

# Complexity and Robustness Trade-Off for Traditional and Deep Models

Lead Guest Editor: Shahzad Sarfraz

Guest Editors: Adil Mehmood Khan, Manuel Mazzara, Salvatore Distefano, and Muhammad Ahmad





---

# **Complexity and Robustness Trade-Off for Traditional and Deep Models**

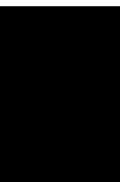
Complexity

---

## **Complexity and Robustness Trade-Off for Traditional and Deep Models**

Lead Guest Editor: Shahzad Sarfraz

Guest Editors: Adil Mehmood Khan, Manuel  
Mazzara, Salvatore Distefano, and Muhammad  
Ahmad




---

Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in "Complexity." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

Hiroki Sayama , USA

## Associate Editors

Albert Diaz-Guilera , Spain  
Carlos Gershenson , Mexico  
Sergio Gómez , Spain  
Sing Kiong Nguang , New Zealand  
Yongping Pan , Singapore  
Dimitrios Stamovlasis , Greece  
Christos Volos , Greece  
Yong Xu , China  
Xinggang Yan , United Kingdom





## Academic Editors

Andrew Adamatzky, United Kingdom  
Marcus Aguiar , Brazil  
Tarek Ahmed-Ali, France  
Maia Angelova , Australia  
David Arroyo, Spain  
Tomaso Aste , United Kingdom  
Shonak Bansal , India  
George Bassel, United Kingdom  
Mohamed Boutayeb, France  
Dirk Brockmann, Germany  
Seth Bullock, United Kingdom  
Diyi Chen , China  
Alan Dorin , Australia  
Guilherme Ferraz de Arruda , Italy  
Harish Garg , India  
Sarangapani Jagannathan , USA  
Mahdi Jalili, Australia  
Jeffrey H. Johnson, United Kingdom  
Jurgen Kurths, Germany  
C. H. Lai , Singapore  
Fredrik Liljeros, Sweden  
Naoki Masuda, USA  
Jose F. Mendes , Portugal  
Christopher P. Monterola, Philippines  
Marcin Mrugalski , Poland  
Vincenzo Nicosia, United Kingdom  
Nicola Perra , United Kingdom  
Andrea Rapisarda, Italy  
Céline Rozenblat, Switzerland  
M. San Miguel, Spain  
Enzo Pasquale Scilingo , Italy  
Ana Teixeira de Melo, Portugal

Shahadat Uddin , Australia  
Jose C. Valverde , Spain  
Massimiliano Zanin , Spain









## Contents

### **Feature Extraction of Plant Leaf Using Deep Learning**

Muhammad Umair Ahmad , Sidra Ashiq, Gran Badshah , Ali Haider Khan , and Muzammil Hussain 




Research Article (8 pages), Article ID 6976112, Volume 2022 (2022)

### **An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning**

S. M. Taslim Uddin Raju , Amlan Sarker , Apurba Das , Md. Milon Islam , Mabrook S. Al-Rakhami , Atif M. Al-Amri , Tasniah Mohiuddin , and Fahad R. Albogamy 





Research Article (19 pages), Article ID 9928836, Volume 2022 (2022)

### **Caricature Face Photo Facial Attribute Similarity Generator**

Muhammad Irfan Khan , Muhammad Kashif Hanif , and Ramzan Talib 







Research Article (14 pages), Article ID 6709707, Volume 2022 (2022)

### **Affinity Propagation-Based Hybrid Personalized Recommender System**

Iqbal Qasim , Mujtaba Awan, Sikandar Ali , Shumaila Khan , Mogeab A. A. Mosleh , Ahmed Alsanad , Hizbullah Khattak , and Mahmood Alam 






Research Article (12 pages), Article ID 6958596, Volume 2022 (2022)

### **A Deep Neural Network Model for the Detection and Classification of Emotions from Textual Content**

Muhammad Zubair Asghar , Adidah Lajis , Muhammad Mansoor Alam , Mohd Khairil Rahmat , Haidawati Mohamad Nasir , Hussain Ahmad, Mabrook S. Al-Rakhami , Atif Al-Amri, and Fahad R. Albogamy








Research Article (12 pages), Article ID 8221121, Volume 2022 (2022)

### **Employing Machine Learning-Based Predictive Analytical Approaches to Classify Autism Spectrum Disorder Types**

Muhammad Kashif Hanif , Naba Ashraf , Muhammad Umer Sarwar , Deleli Mesay Adinew , and Reehan Yaqoob 





Research Article (10 pages), Article ID 8134018, Volume 2022 (2022)

### **Monitoring Population Phenology of Asian Citrus Psyllid Using Deep Learning**

Maria Bibi , Muhammad Kashif Hanif , Muhammad Umer Sarwar , Muhammad Irfan Khan , Shouket Zaman Khan , Casper Shikali Shivachi , and Asad Anees 



Research Article (10 pages), Article ID 4644213, Volume 2021 (2021)

### **Price Risk Measurement of China's Soybean Futures Market Based on the VAR-GJR-GARCH Model**

Chuan-hui Wang , Li-ping Wang, Wei-feng Gong , Hai-xia Zhang , and Xia Liu 


Research Article (12 pages), Article ID 8912024, Volume 2021 (2021)

### **Using Proximity Graph Cut for Fast and Robust Instance-Based Classification in Large Datasets**

Stanislav Protasov  and Adil Mehmood Khan 


Research Article (9 pages), Article ID 2011738, Volume 2021 (2021)

**Equipment Maintenance Support Effectiveness Evaluation Based on Improved Generative Adversarial Network and Radial Basis Function Network**

Zhen Li , Jianping Hao, and Cuijuan Gao




Research Article (11 pages), Article ID 1332242, Volume 2021 (2021)

**Development of ANPR Framework for Pakistani Vehicle Number Plates Using Object Detection and OCR**

Salma, Maham Saeed, Rauf ur Rahim, Muhammad Gufran Khan , Adil Zulfiqar, and Muhammad Tahir Bhatti




Research Article (14 pages), Article ID 5597337, Volume 2021 (2021)

**Sublemma-Based Neural Machine Translation**

Thien Nguyen , Huu Nguyen , and Phuoc Tran 





Research Article (9 pages), Article ID 5935958, Volume 2021 (2021)

**COVID-19 Detection Using Deep Convolutional Neural Networks and Binary Differential Algorithm-Based Feature Selection from X-Ray Images**

Mohammad Saber Iraj , Mohammad-Reza Feizi-Derakhshi , and Jafar Tanha 

Research Article (10 pages), Article ID 9973277, Volume 2021 (2021)


**A New Classification Method in Ultrasound Images of Benign and Malignant Thyroid Nodules Based on Transfer Learning and Deep Convolutional Neural Network**

Weibin Chen , Zhiyang Gu , Zhimin Liu, Yaoyao Fu, Zhipeng Ye, Xin Zhang , and Lei Xiao 

Research Article (9 pages), Article ID 6296811, Volume 2021 (2021)




**Diagnosis of COVID-19 Using a Deep Learning Model in Various Radiology Domains**




Yousef Alhwaiti , Muhammad Hameed Siddiqi , Madallah Alruwaili , Ibrahim Alrashdi, Saad

Alanazi , and Muhammad Hasan Jamal

Research Article (10 pages), Article ID 1296755, Volume 2021 (2021)

**Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques**

Muhammad Yaseen Khan , Abdul Qayoom , Muhammad Suffian Nizami , Muhammad Shoaib


Siddiqi , Shaukat Wasi , and Syed Muhammad Khaliq-ur-Rahman Raazi 

Research Article (18 pages), Article ID 2553199, Volume 2021 (2021)

**Urdu Handwritten Characters Data Visualization and Recognition Using Distributed Stochastic Neighborhood Embedding and Deep Network**

Mujtaba Husnain , Malik Muhammad Saad Missen , Shahzad Mumtaz , Dost Muhammad Khan ,

Mickael Coustaty , Muhammad Muzzamil Luqman , Jean-Marc Ogier , Hizbullah Khattak , Sikandar

Ali , and Ali Samad 

Research Article (15 pages), Article ID 4383037, Volume 2021 (2021)

# Contents

## **Research on the Capability Maturity Evaluation of Intelligent Manufacturing Based on Firefly Algorithm, Sparrow Search Algorithm, and BP Neural Network**

Li Shi , Xuehong Ding , Min Li, and Yuan Liu



Research Article (26 pages), Article ID 5554215, Volume 2021 (2021)

## **Selection and Ranking of Fog Computing-Based IoT for Monitoring of Health Using the Analytic Network Approach**

Dong Xue, Shah Nazir , Zhiqiang Peng , and Hizbullah Khattak 




Research Article (11 pages), Article ID 9964303, Volume 2021 (2021)

## **An Overview of Computational Models for Industrial Internet of Things to Enhance Usability**

Zhen Ying , Iftikhar Ahmad, Saima Mateen, Asad Zia, Ambreen, Shah Nazir , and Neelam Mukhtar

Research Article (11 pages), Article ID 5554685, Volume 2021 (2021)

## **Ontology-Based Smart System to Automate Higher Education Activities**

Nada Abdullah Alrehaili , Muhammad Ahtisham Aslam , Dimah Hussein Alahmadi, Dina Abdullah Alrehaili , Muhammad Asif, and Muhammad Sheraz Arshad Malik





Research Article (20 pages), Article ID 5588381, Volume 2021 (2021)

## **Accelerating Spark-Based Applications with MPI and OpenACC**

Saeed Alshahrani , Waleed Al Shehri , Jameel Almalki , Ahmed M. Alghamdi , and Abdullah M. Alammari 





Research Article (17 pages), Article ID 9943289, Volume 2021 (2021)

## **Solution of the Systems of Delay Integral Equations in Heterogeneous Data Communication through Haar Wavelet Collocation Approach**

Hualing Wu , Rohul Amin , Asmatullah Khan, Shah Nazir , and Sultan Ahmad 


Research Article (11 pages), Article ID 5805433, Volume 2021 (2021)

## **An Efficient Decision Support System for the Selection of Appropriate Crowd in Crowdsourcing**

Yongjun Huang, Shah Nazir , Jiyu Wu , Fida Hussain Khoso, Farhad Ali , and Habib Ullah Khan 






Research Article (11 pages), Article ID 5518878, Volume 2021 (2021)

## **Vibration Reliability Analysis of Drum Brake Using the Artificial Neural Network and Important Sampling Method**

Zhou Yang , Unsong Pak, and Cholu Kwon

Research Article (14 pages), Article ID 5517634, Volume 2021 (2021)



## **An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification**

Aijaz Ahmad Reshi , Furqan Rustam , Arif Mehmood , Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad , Hessa Alsuwailem, and Gyu Sang Choi 

Research Article (12 pages), Article ID 6621607, Volume 2021 (2021)



**Optimal Economic Modelling of Hybrid Combined Cooling, Heating, and Energy Storage System Based on Gravitational Search Algorithm-Random Forest Regression**

Muhammad Shahzad Nazir , Sami ud Din , Wahab Ali Shah, Majid Ali, Ali Yousaf Kharal, Ahmad N. Abdalla, and Padmanaban Sanjeevikumar






Research Article (13 pages), Article ID 5539284, Volume 2021 (2021)

**Improving Transformer-Based Neural Machine Translation with Prior Alignments**

Thien Nguyen , Lam Nguyen , Phuoc Tran , and Huu Nguyen 

Research Article (10 pages), Article ID 5515407, Volume 2021 (2021)

**Intelligent Computation Offloading for IoT Applications in Scalable Edge Computing Using Artificial Bee Colony Optimization**

Mohammad Babar , Muhammad Sohail Khan, Ahmad Din , Farman Ali , Usman Habib , and Kyung Sup Kwak 





Research Article (12 pages), Article ID 5563531, Volume 2021 (2021)

**Deep CNN and Deep GAN in Computational Visual Perception-Driven Image Analysis**

R. Nandhini Abirami , P. M. Durai Raj Vincent , Kathiravan Srinivasan , Usman Tariq , and Chuan-Yu Chang 


Review Article (30 pages), Article ID 5541134, Volume 2021 (2021)

**Design and Implementation of Novel LMI-Based Iterative Learning Robust Nonlinear Controller**

Saleem Riaz , Hui Lin , Farkhanda Afzal , and Ayesha Maqbool 





Research Article (13 pages), Article ID 5577241, Volume 2021 (2021)

**Prediction of Seepage Pressure Based on Memory Cells and Significance Analysis of Influencing Factors**

Mengdie Zhao, Haifeng Jiang, Shoukai Chen , and Yajing Bie




Research Article (10 pages), Article ID 5576148, Volume 2021 (2021)

**A Computationally Efficient User Model for Effective Content Adaptation Based on Domain-Wise Learning Style Preferences: A Web-Based Approach**

Dong Pan , Anwar Hussain , Shah Nazir , and Sulaiman Khan 

Research Article (15 pages), Article ID 6634328, Volume 2021 (2021)

**Isolated Handwritten Pashto Character Recognition Using a K-NN Classification Tool based on Zoning and HOG Feature Extraction Techniques**

Juanjuan Huang , Ihtisham Ul Haq, Chaolan Dai, Sulaiman Khan , Shah Nazir , and Muhammad Imtiaz

Research Article (8 pages), Article ID 5558373, Volume 2021 (2021)

**Features to Text: A Comprehensive Survey of Deep Learning on Semantic Segmentation and Image Captioning**

Ariyo Oluwasammi , Muhammad Umar Aftab , Zhiguang Qin , Son Tung Ngo , Thang Van Doan , Son Ba Nguyen , Son Hoang Nguyen , and Giang Hoang Nguyen 

Research Article (19 pages), Article ID 5538927, Volume 2021 (2021)

## Contents

---

### **Transfer Learning and Semisupervised Adversarial Detection and Classification of COVID-19 in CT Images**

Ariyo Oluwasanmi , Muhammad Umar Aftab , Zhiguang Qin , Son Tung Ngo , Thang Van Doan , Son Ba Nguyen , and Son Hoang Nguyen 







Research Article (11 pages), Article ID 6680455, Volume 2021 (2021)

### **Multicriteria-Based Crowd Selection Using Ant Colony Optimization**

Guan Wang, Farhad Ali, Jonghoon Yang, Shah Nazir , Ting Yang , Abdullah Khan, and Muhammad Imtiaz



Research Article (11 pages), Article ID 6622231, Volume 2021 (2021)

### **Probabilistic Model-Based Malaria Disease Recognition System**

Rahila Parveen , Wei Song , Baozhi Qiu , Mairaj Nabi Bhatti , Tallal Hassan , and Ziyi Liu 

Research Article (11 pages), Article ID 6633806, Volume 2021 (2021)

### **Machine Learning Techniques for Quantification of Knee Segmentation from MRI**

Sujeet More , Jimmy Singla , Ahed Abugabah, and Ahmad Ali AlZubi

Review Article (13 pages), Article ID 6613191, Volume 2020 (2020)

## Research Article

# Feature Extraction of Plant Leaf Using Deep Learning

Muhammad Umair Ahmad <sup>1</sup>, Sidra Ashiq,<sup>2</sup> Gran Badshah <sup>3</sup>, Ali Haider Khan <sup>1</sup>  
and Muzammil Hussain <sup>1</sup>

<sup>1</sup>Department of Computer Science, School of Systems & Technology, University of Management and Technology, Lahore, Pakistan

<sup>2</sup>Department of Computer Science, National College of Business Administration & Economics, Multan, Pakistan

<sup>3</sup>Department of Computer Science, King Khalid University, Abha 61421, P.O. Box 960, Saudi Arabia

Correspondence should be addressed to Gran Badshah; [gdoostan@kku.edu.sa](mailto:gdoostan@kku.edu.sa) and Ali Haider Khan; [s2018288001@umt.edu.pk](mailto:s2018288001@umt.edu.pk)

Received 13 April 2021; Revised 21 April 2022; Accepted 27 April 2022; Published 28 May 2022

Academic Editor: Shahzad Sarfraz

Copyright © 2022 Muhammad Umair Ahmad et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Half a million species of plants could be existing in the world. Classification of plants based on leaf features is a critical job as feature extraction (includes shape, margin, and texture) from binary images of leaves may result in duplicate identification. However, leaves are an effective means of differentiating plant species because of their unique characteristics like area, diameter, perimeter, circularity, aspect ratio, solidity, eccentricity, and narrow factor. This paper presents the extraction of plant leaf gas alongside other features from the camera images or a dataset of images by applying a convolutional neural network (CNN). The extraction of leaf gas enables identification of the actual level of chlorophyll (Ch) and nitrogen (N) which may help to interpret future predictions. Our contribution includes the study of texture and geometric features, analyzing ratio of Ch and N in both healthy and dead leaves, and the study of color-based methods via CNN. Several steps are included to obtain the results: image preprocessing, testing, training, enhancement, segmentation, feature extraction, and aggregation of results. A vital contrast of the results can be seen by considering the kind of image, whether a healthy or dead leaf.

## 1. Introduction

Almost millions of species of plants can be found on the Earth surface as a result of botanical research. The literature proposed a wide range of techniques to recognize the plant type, most likely leaf recognition and visual classification of plants through image processing and computer vision [1]. It is a challenging problem and requires tactical dealing with intraclass variable textures and asymmetrical shapes. Commonly, a plant is recognized by recognizing its specific organs such as leaves, flowers, fruits, or bark or their combination. Belhumeur et al. introduced the usage of such a system for quick classification and recognition of plant species from an entire collection; i.e., the process of hours can be accomplished within seconds [2]. Similarly, SIFT descriptors integrated with a pack of Word models were applied for leaf recognition in [3]. The biologists have discovered numerous types of leaves using machine learning classifiers [4–7] and computer vision techniques [1], but still,

there have been some kinds of leaves left to be identified which need to be demarcated.

An open-source plant recognition problem was given as challenge to research community in the 2016 edition of LifeCLEF which targets identification of the unknown and never-seen categories on the basis of plant characteristics like leaf shape, leaf veins, flower, fruit, stem, and branch of the entire plant [8]. With the advancement of artificial intelligence and neural networks, the research community can make their solutions more optimal in several domains. Automated plant recognition via the neural network using image processing is a critical area which allows recognition of leaf images with an accuracy of 80% to 97% [9, 10].

Neural network works in the same fashion as the human brain, founded on mathematical formulas/models. The functional principles of neural networks target understanding and recognizing patterns among different components. The fundamental unit of neural network is a neuron which is trained by repetitive tasks and gets experienced just

like human brain through acquired knowledge in the training. The focus of training and acquiring knowledge is to establish a connection between input and output. After training, the system can make predictions about what it has been trained. Convolutional neural network (CNN) is a class of deep neural networks first proposed by LeCun [11]. Common applications of CNN can be found in computer vision, natural language processing, and speech recognition [12–15]. CNN functions are based on the human vision processing system. It distinguishes a feature using local receptive field and shared weight and associates this feature to the feature map, saving computational load. Furthermore, a process of subsampling is performed to achieve the invariance among features with respect to geometric distortion. CNNs are considered better than classic neural networks on images because layers in CNN inherent the properties of input images whereas feedforward neural networks cannot make sense of order in their inputs.

This study provides a method to analyze leaf gas and the leaf features like area, diameter, perimeter, circularity, aspect ratio, solidity, eccentricity, and narrow factor of healthy and dead leaf dataset. The leaf attributes like leaf area, diameter, leaf chlorophyll, and leaf nitrogen are calculated and analyzed through CNN implemented in MATLAB. The result proves this method to be an efficient attempt. The paper is organized in five sections. Related work of the research community in the current field is presented in Section 2 with close comparison. Section 3 illustrates the methodology with the proposed cluster. Simulated results are discussed in Section 4. The paper is concluded and discourses the future directions of our work in Section 5.

## 2. Related Work

In the past few decades, the research community has focused on the field of artificial intelligence by working in digital image processing, computer vision technique, and machine learning to provide a platform between human and machine theory [2, 16–22]. This work is widely used in several companies and medical fields for classification and identification of plants which play a vital role in the Earth's ecosystem. Many plant species are at the edge of extinction in recent days. In order to save the Earth's biosphere, the flora diversity catalogue and study of plant databases is an important step. Techniques used for leaf recognition using shape, descriptor, size, and texture have been focused for many years. Wu et al. [23] used the probabilistic neural network (PNN) to automatically classify the leaf features of 32 plants. The accuracy rate is above 90% as the algorithm is fast in execution. Automatic plant identification via leaf characteristics is a challenging task and constrained by many complications which include geometric deformations, illumination variations, and interspecies and intraspecies levels. To overcome these constraints, Yahiaoui et al. [24] proposed a boundary-based approach using Otsu algorithm with a better classification rate by segmenting a scanned or scanned-like image into foreground and background pixel sets,

which helps to obtain the binary image and subsequently extract the boundary for the description stage. In the past, researchers used the shape of leaves as one of the classification feature of plants [4, 17, 25, 26] because plants can be identified through distinct shape attributes of the leaf even by nonexperts. Figure 1 shows the comparison of related studies.

The enhanced neural networks like PNN, ANN, and CNN have significantly improved the resulting ratio and accuracy rate at a minimum cost of iterations. The concept of a pre-trained CNN model for plant recognition was also proposed by Lee et al. [27], which achieved a performance of up to 99.6% verified through DN visualization. Lee et al. deduced that shape attributes of a leaf should be avoided as a choice of plant classification; however, venation structure is an important feature to distinguish plant species. Nitrogen (N), being an integral part of chlorophyll (Ch), plays a vital role in plant growth as Ch absorbs light energy for photosynthesis [28]. Plants with sufficient N contents in Ch are green and healthy; otherwise, plants are pale-green or yellow. Therefore, the status of Ch and N can be determined by exploiting leaf color property using image processing. Ali et al. [29] used the Dark Green Color Index (DGCI) model to find out the level of Ch and N contents from color images of leaves. The DGCI covers dark green color on a scale of 0 and 1. The measurements of Ch and N were recorded at three different stages of plant development through laboratory-based methods and using a SPAD-502 device (a hand-held absorbance meter used to measure relative greenness and Ch and N contents).

## 3. Methodology with the Proposed Cluster

Extraction of leaf features using the principles of visual image processing helps in plant classification, and a training dataset is used to train the CNN. The input image needs to be preprocessed and is recognized after passing through a variety of steps. A color image is composed of color pixels where each pixel can have red, green, and blue color planes. So, the input image can be assumed as a three-dimensional matrix corresponding to three color planes having color values of the pixel as matrix entries. Figure 2 represents the procedural flow of the proposed model as a block diagram. Images can exist in several color spaces, e.g., grayscale, RGB, HSV, and CMYK. The computational intensity is directly proportional to the number of dimensions of the image. CNN plays a role in reducing images to a form that is easier to process and does not lack features necessary for making a good prediction. This makes CNN design a system not only good at learning features but also scalable to massive datasets. Otsu algorithm as proposed by Yahiaoui et al. [24] is used for segmentation of the input image foreground and background and then three color planes to input in the HSV model.

*3.1. Preparing the Training Dataset.* The proposed model has used the Caltech dataset for leaf gas analysis by extracting individual leaf features which can be statistical or geometric.

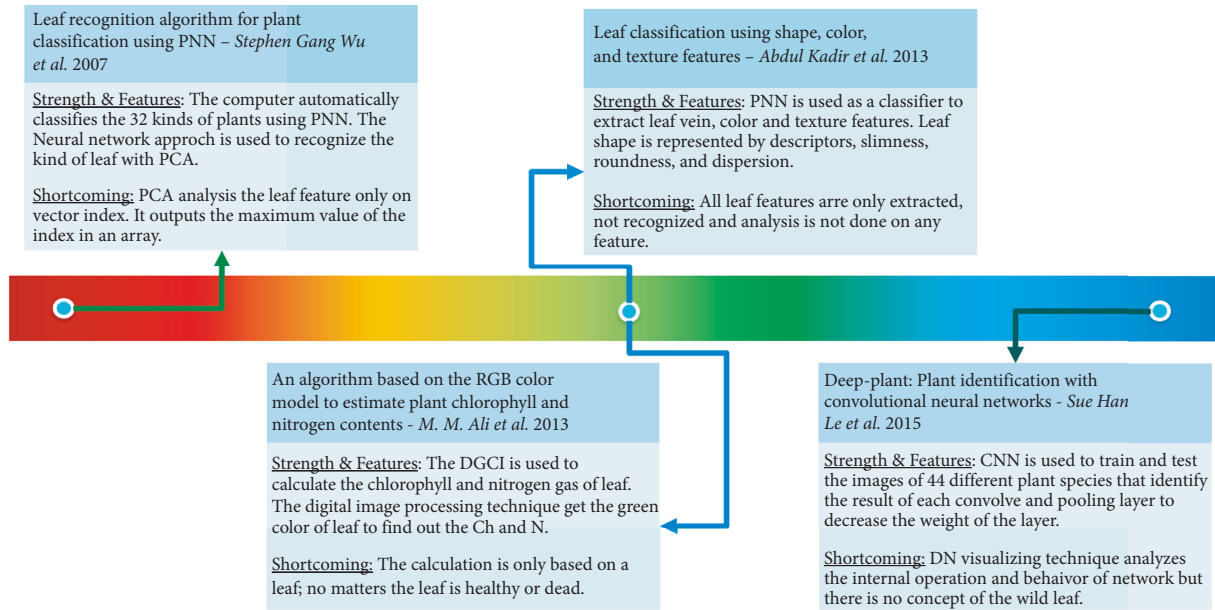


FIGURE 1: A close comparison of related work.

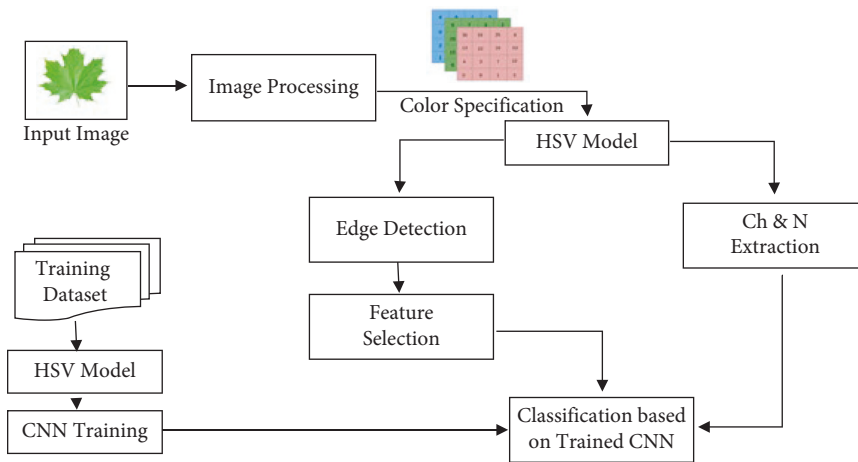


FIGURE 2: Block diagram of the proposed model.

The MNIST database, containing 60,000 image examples, is used to train the convolutional neural network with 10,000 image examples. As a result of training, network generates two datasets of images and labels necessary for classification of input images as shown in Table 1.

After downloading the MNIST data files, we unzip the files in the MNIST directory. The database contains both healthy and dead leaf images. Figure 3 shows the sample of healthy and dead leaves which have been trained by the CNN.

**3.2. Image Processing.** In computer science, image processing is used to obtain an improved version of digital images or extract some useful information from it after applying different operations. The objectives of image processing may include visualization, image sharpening and

TABLE 1: Generated datasets.

Datasets	Images	Labels
Train dataset	Train-images-idx3-ubyte	Train-labels-idx1-ubyte
Test dataset	t10k-images-idx3-ubyte	t10k-labels-idx1-ubyte

repair, image restoration, pattern measurement, and recognition. A digital image can be thought as a 2D matrix of pixel values. In CNN, an image should be thought as a 3D matrix where depth represent color channels of either red, green, or blue, unless not talking about a grayscale image which needs to be converted to the 3D matrix. Most commonly, the pixel size is 8 bits or 1 byte; therefore, a single pixel can represent a value between 0 and 255 as the intensity of color for color images, where 0 corresponds to black and 255 corresponds to white for grayscale images. The proposed model uses RGB, HSV, and HSB color conversions.

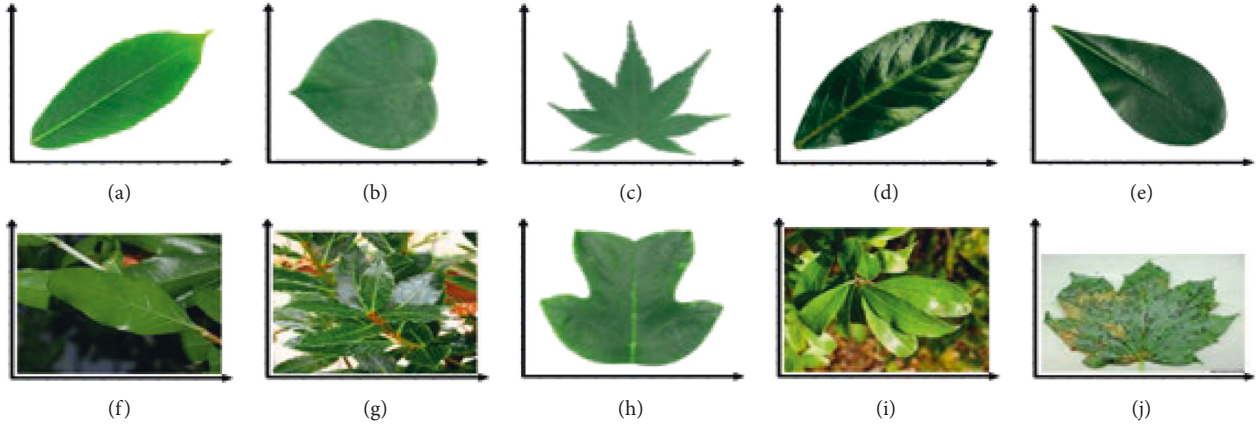


FIGURE 3: Sample of the leaf database.

Chlorophyll is estimated from the RGB image, the hue saturation value (HSV) is used to identify leaf color and histogram, and hue saturation and intensity is used for measuring leaf nitrogen. Figure 4 shows the flow chart of the HSV model.

Computing the number of colour bands/channels of a health or dead leaf image is a preprocessing step followed by physical rotation in order to direct the leaf apex on the right side, which is considered the initial point of the process. Then, image resizing is done and pixels are separated into three RGB color matrices of size  $400 \times 300 \times 3$ . The corresponding grayscale value of the RGB color image can be found by averaging or weighted averaging. Both methods are the same but some weighing factor is given to each of the color intensities in weighted averaging which can be defined as

$$\text{Grav} = 0.3 * I_R + 0.78 * I_G + 0.14 * I_B. \quad (1)$$

Here,  $I_R$ ,  $I_G$ , and  $I_B$  are the color intensities of red, green, and blue, respectively, of a pixel and multiplied by a predefined value. The resultant of (1) will be a grayscale image.

HSV is a cylindrical-coordinate representation of RGB obtained through Gray world algorithm or the White patch algorithm. This color space is used for illumination value which indicates the light source. Figure 5 shows the HSV color representation of healthy and dead leaves. The conversion of the RGB digital image into HSV color space includes color detection, mask recognition, and finding the number of blobs in the image. The parameters like area, mean, max-mean, and min-mean are determined where the blobs in the image are identified unless blob size is more than 10 pixels. For more than 10-pixel blob size, parameters would have some values. HSV low and high thresholding are also used followed by histogram analysis applied to the processing image. At last, mean values of H, S, and V are computed.

**3.3. CNN Architecture.** Typical machine learning models used for classification are support vector machine [30] and AdaBoost [31] whose performance is based on extracted feature points. However, these models cannot extract the optimal feature points because learning and classification

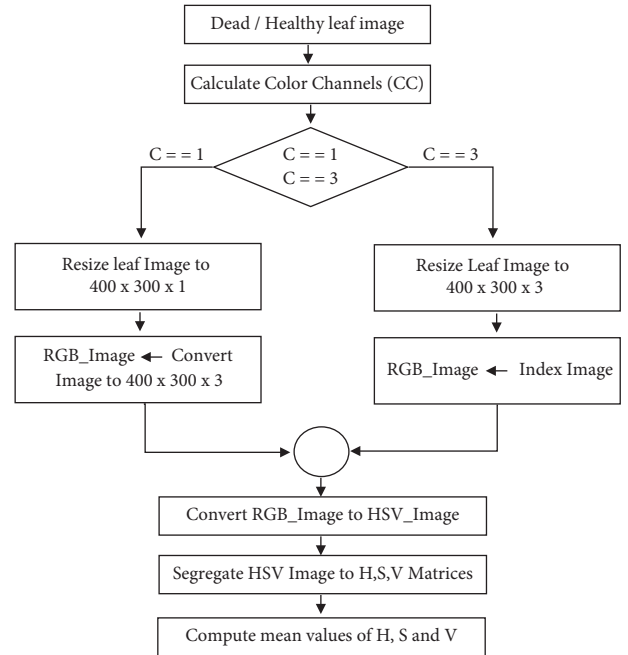


FIGURE 4: Flow chart of the HSV model.

proceed independently. The CNN is the neural network model mirroring the human visual system [32]. To understand the architecture of CNN, all layers of CNN can be categorized as (i) convolution layer: works the same as lateral geniculate nucleus (LGN) detecting the boundary edge of objects, (ii) pooling layer: corresponds to visual cortex (V3) used to identify the color of the whole object, and (iii) fully connected layer: acts as the lateral occipital cortex (LOC) to detect the color and shape of objects.

The proposed CNN is based on a 9-layer structure where each layer filters the distinct features of the processing image. The input image is transformed as an array of pixel values of volume  $28 \times 28 \times 3$ . The front layer is always the convolution layer extracting the maximum or accumulated leaf features, which are passed to the pooling layer. The output of the model can be a single class or a group of classes that evidently describe the leaf image. There are totally 3 nodes with

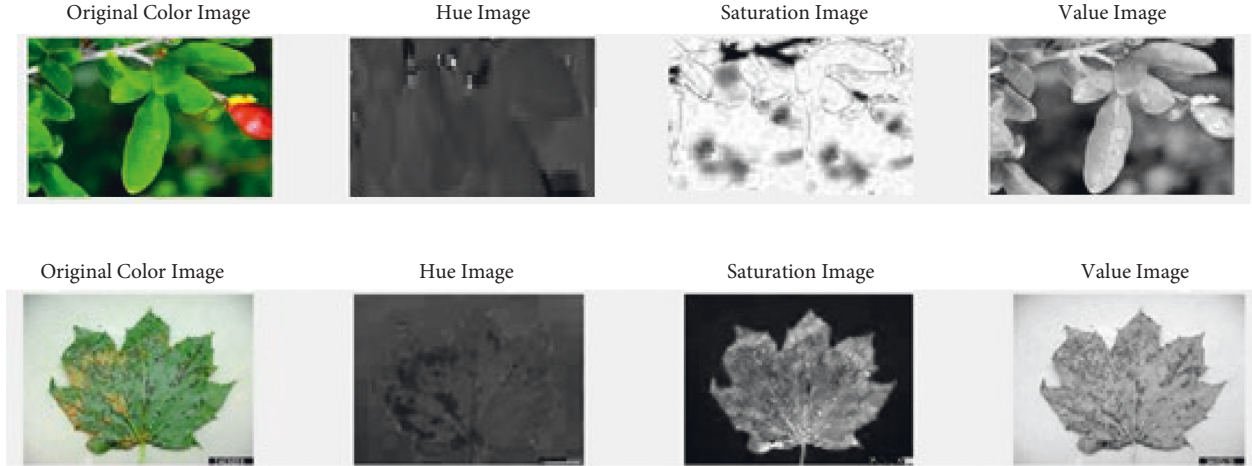


FIGURE 5: HSV image representation of healthy and dead leaf images.

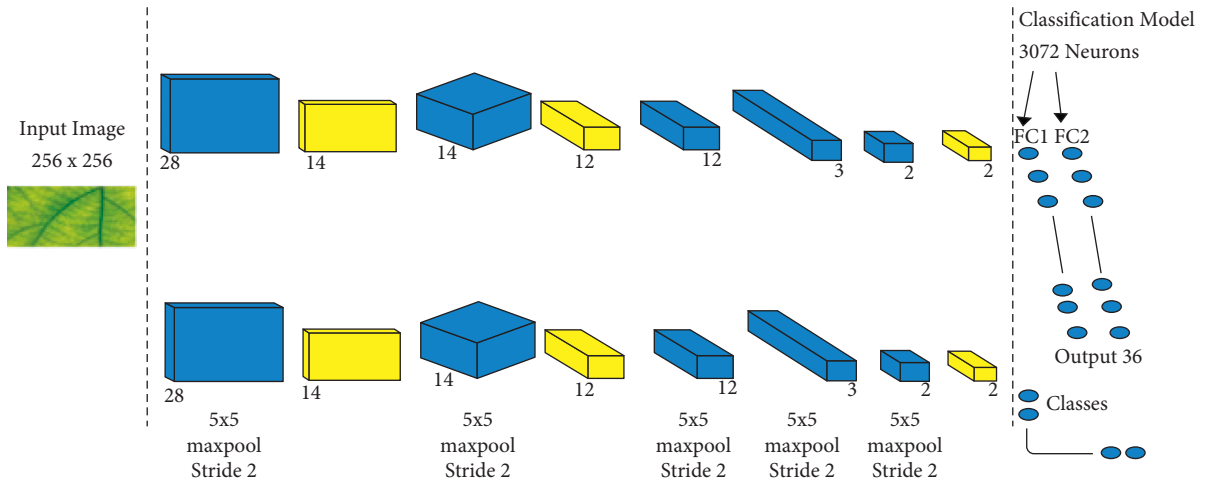


FIGURE 6: Proposed CNN architecture.

9 vector forms. Figure 6 depicts the proposed CNN architecture.

**3.4. Edge and Boundary Detection.** Edge detection is used in many applications of image processing, particularly object recognition and classification system. Edge detection is the process of identifying points in a digital image at which the pixel value (brightness) changes sharply or has discontinuities. The combination of these points can be organized as a set of curved line segments known as edges. Edge detection techniques are gaining popularity because of being robust to the conditions where illumination changes abruptly. In this proposed model, edge detection of leaf images is based on the region-based approach which engages Prewitt and Roberts filter methods that enable us to sharply identify the discontinuities between two regions in a greyscale image. It uses the Prewitt operator which is a discrete differentiation operator and approximates the gradient of the image intensity function. The Prewitt operator returns the corresponding gradient vector or the norm of the vector at every pixel in the processing image. The Prewitt operator is relatively inexpensive in terms of computations because this

activates on image convolution with a small, separable, and integer-valued filter with 90-degree rotation. To emphasize the edges, Sobel and Canny filters are used in the edge detection algorithm. The mean square error (MSE) can be computed as follows:

$$\text{MSE} = \sqrt{\frac{1}{MN} \sum_{k=1}^{MN} (m_{ik} - \bar{m}_{ik})^2}. \quad (2)$$

Here,  $\bar{m}_{ik}$  is the mean of the original image and  $m_{ik}$  is the mean of the filtered image of the  $i$ th color plane at the  $k$ th pixel.

$$\text{Leaf EDT} = e * \log_{10}\left(\frac{v^2}{\text{MSE}}\right), \quad (3)$$

where  $e = 10$  and  $v$  lies between  $MN - 1$  and  $255$ . The use of the Canny filter emphasizes the whole leaf edge, whereas Prewitt and Roberts filters focus only the upper leaf part.

**3.5. Feature Extraction.** Separation of the leaf object (foreground) from its background is known as segmentation.

TABLE 2: Dead vs. healthy leaf gas analysis.

Leaf state	Leaf class	Ch ( $\mu\text{g/ml}$ )	N (%)	Area	Diameter
Dead leaf	<i>Alternaria</i>	19.759	0.544704	17330	1.58666
	Bacteria	0.688	0.75	210	1.87
	<i>Cercosporin</i>	1.32937	0.523	7930	3.52565
Healthy leaf	1	34.58	0.4557	2671	1.9551
	2	16.53	0.64	15071	1.67

TABLE 3: Recognition rate.

Leaf state	Leaf class	Ch ( $\mu\text{g/ml}$ )	N (%)	Recognition rate (%)
Dead leaf	<i>Alternaria</i>	19.759	0.544704	97.2
	Bacteria	0.688	0.75	94.5
	<i>Cercosporin</i>	1.32937	0.523	92.6
Healthy leaf	1	34.58	0.4557	97.7
	2	16.53	0.64	96.3

This process adopts adaptive threshold K-Means method. After segmentation, geometric features are extracted from the segmented image. For example, aspect ratio and roundness ( $R$ ) of the leaf can be computed from the following:

$$\text{aspect ratio} = \frac{W}{L}, \quad (4)$$

$$R = \frac{4\pi A}{p^2}. \quad (5)$$

The leaf color falls under morphological features. Several statistical parameters like mean, skewness, and kurtosis can be computed in the color space to represent color features of the leaf. This method has low computational complexity and is applicable to real-time processing. As the processing image comprises three color planes (red, green, and blue), it helps in the estimation of Ch and N using the mean of all three color planes after leaf contour as follows:

$$\text{ChN} = G - \left( \frac{R+B}{2} \right). \quad (6)$$

In (6), values of red and blue are included for normalization purpose and giving the estimate of ChN of healthy and dead leaves. Nitrogen can be computed using (7) where HE, St, and Bg are hue value, saturation intensity, and brightness intensity of the colored image, respectively.

$$N = \frac{1}{3} \times \left( \frac{HE - 60}{60} + (1 - St) + (1 - Bg) \right). \quad (7)$$

The maximum distance between two points on the boundary of the leaf object in the processing image is known as the effective diameter. The effective diameter can be calculated using another morphological feature of a leaf called area as formulated in

$$\text{effective diameter} = 2 \times \sqrt{\frac{\text{area}}{\pi}}. \quad (8)$$

Skewness ( $S_i$ ) and kurtosis ( $K_i$ ) are the imperative color moments used to represent color distribution in processing

and retrieving images. Equations (9) and (10) formulate both parameters for the  $i$ th color plane, respectively.

$$S_i = \sqrt[3]{\frac{1}{MN} \sum_{k=1}^{MN} (m_{ik} - \bar{m}_{ik})^3}, \quad (9)$$

$$K_i = \sqrt[4]{\frac{1}{MN} \sum_{k=1}^{MN} (m_{ik} - \bar{m}_{ik})^4}. \quad (10)$$

#### 4. Numerical Results and Discussion

This work implements leaf recognition and Ch and N analysis using CNN and contributes to the domain of leaf classification no matter whether the leaf is healthy or dead. Two databases, namely, Caltech and Flavia, are used for training and experimentation purpose. The Caltech database was prepared by Caltech University containing 102 categories of leaves consisting of 8000 images. This database contains at least 40 images of each category which makes it attractive for large-scale systems developed to recognize and analyze the leaf images. The second database Flavia contains healthy leaf images collected from different areas. The dataset of diseased leaf images is also included to analyze dead leaves in order to achieve the diversity of experimentation. The final dataset contains 467 healthy leaf images and 60 dead images which corresponds 70% of Caltech and 30% of other datasets. Table 2 and Table 3 show that all results are obtained within the error limit of  $5 \times 10^{-7}$  using the backpropagation process.

Table 2 characterizes the numerical results of dead and healthy leaves against Ch, N, area, and diameter. First, leaves are categorized according to their state, i.e., dead and health leaves. Healthy leaves are further categorized into two classes, 1 and 2. The first class contains normal images of leaves, whereas the second class comprises noisy images.

Table 3 represents the recognition rate of dead and healthy leaves against Ch and N. Our proposed model



TABLE 4: Performance comparison.

Sr.	Classifier	Goal	Result	Work
1	CNN	Identification of plant leaves using image features and ANN	94.4%	33
2	PNN	Plant leaf classification using the centroid distance and axis of the least inertia method	80.10–82.05%	34
3	KNN	Recognition of whole and deformed plant leaves using statistical shape features and neuro-fuzzy classifier	79.7%	35
4	SVM	Comparative analysis of leaf classification and recognition by different SVM classifiers	90.9%	36
5	CNN	Plant leaf gas analysis	97.7%	<b>Proposed</b>

reduces 1.4% recognition rate when assigned to recognize the noisy version of normal healthy class images. To test the trained CNN efficiency, dead leaves of three classes were examined. Several leaf features, as discussed above, are extracted to test the proposed model. It is observed that the ratio of Ch of the healthy leaf will be greater than that of the dead leaf and vice versa for nitrogen ratios, because the number of bacterial blobs is identified as more than that on the healthy leaf.

Comparing the computed results of plant leaf recognition system performance with existing research studies is essential. This study achieved outstanding results compared to existing research studies [4, 33–35]. However, the mentioned work has a different dataset and different types of classes, so it is not justified to directly compare their results. Table 4 shows the comparative results related to the study.

## 5. Conclusion and Future Directions

Leaf recognition plays a significant role in the science of plant classification. Convolutional neural networks (CNNs) have proven their capability in various applications of image processing and computer vision. This paper critically discusses the related work and then aims feature extraction of plant leaves with gas analysis, specifically, under two leaf states, i.e., healthy and dead leaves, using CNN. The proposed model separates the integrated complications like geometric deformations, varying illumination in the sample images, and interspecies and intraspecies levels to be handled with lower complexity. The work shows how feature extraction of leaf and gas analysis can be accomplished using the HSV model while the trained CNN classifies the leaves using color specifications without mathematical or statistical study. Two leaf databases, Caltech and Flavia, are used to train and test CNN's efficiency along with a deceased set of leaves. The accuracy of recognition ratio for the proposed model is almost 98%.

This work can be extended by training larger datasets, particularly dead leaves, validating the recognition ratio using CNNs. The extraction of advanced features of digital images with image acquisition, adaptive image enhancement, and various boundary detection algorithms with new developing tools is required. The work can be evolved using different machine learning and deep learning techniques, e.g., autoencoders. The way of learning of domain adaptation is another big task for researchers in the future.

## Data Availability

The processed data are available upon request from the corresponding author.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] J. Wäldchen and P. Mäder, "Plant species identification using computer vision techniques: a systematic literature review," *Archives of Computational Methods in Engineering*, vol. 25, no. 2, pp. 507–543, April 01 2018.
- [2] P. N. Belhumeur, D. Chen, S. Feiner et al., "Searching the world's herbaria: a system for visual identification of plant species," in *Proceedings of the European Conference on Computer Vision*, pp. 116–129, Springer, Marseille, France, October 2008.
- [3] M. Sulc and J. Matas, "Texture-based leaf identification," in *Proceedings of the European Conference on Computer Vision*, pp. 185–200, Springer, Zurich, Switzerland, September 2014.
- [4] J. Chaki, R. Parekh, and S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers," *Pattern Recognition Letters*, vol. 58, pp. 61–68, 2015.
- [5] T. Munisami, M. Ramsurn, S. Kishnah, and S. C. Pudaruth, "Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers," *Procedia Computer Science*, vol. 58, pp. 740–747, 2015.
- [6] C. Kalyoncu and Ö. Toygar, "GTCLC: leaf classification method using multiple descriptors," *IET Computer Vision*, vol. 10, no. 7, pp. 700–708.
- [7] F. M. Kheirkhah and H. Asghari, "Plant leaf classification using GIST texture features," *IET Computer Vision*, vol. 13, no. 4, pp. 369–375.
- [8] H. Goëau, P. Bonnet, and A. Joly, "Plant identification in an open-world (lifeclef 2016)," in *Proceedings of the CLEF 2016-Conference and Labs of the Evaluation Forum*, pp. 428–439, Évora, Portugal, September 2016.
- [9] S. J. Kho, S. Manickam, S. Malek, M. Moseleh, and S. K. Dhillon, "Automated plant identification using artificial neural network and support vector machine," *Frontiers in Life Science*, vol. 10, no. 1, pp. 98–107, 2017.
- [10] B. Şekeroglu and Y. J. P. C. S. İnan, "Leaves recognition system using a neural network," *Procedia Computer Science*, vol. 102, pp. 578–582, 2016.
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [12] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of*

- the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4277–4280, Kyoto, Japan, March 2012.
- [13] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, June 2015.
- [14] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. Hossein Anisi, “CNN-RNN based method for license plate recognition,” *CAAI Transactions on Intelligence Technology*, vol. 3, no. 3, pp. 169–175, 2018.
- [15] E. Wengrowski, M. Purri, K. Dana, and A. Huston, “Deep CNNs as a method to classify rotating objects based on monostatic RCS,” *IET Radar, Sonar & Navigation*, vol. 13, no. 7, pp. 1092–1100.
- [16] S. Fiel and R. Sablatnig, “Automated Identification of Tree Species from Images of the Bark,” *leaves or needles. na*, vol. 13, 2010.
- [17] N. Kumar, P. N. Belhumeur, A. Biswas et al., “Leafsnap: a computer vision system for automatic plant species identification,” in *Proceedings of the European Conference on Computer Vision*, pp. 502–516, Florence, Italy, October 2012.
- [18] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, “Experiments of Zernike moments for leaf identification,” *Journal of Theoretical and Applied Information Technology*, vol. 41, no. 1, pp. 82–93, 2012.
- [19] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, “Performance improvement of leaf identification,” *system using principal component analysis*, vol. 44, pp. 113–124, 2012.
- [20] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, “Rotation-invariant image and video description with local binary pattern features,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.
- [21] B. Li and Z. Zhao, “Research on leaf recognition and simulation,” in *Proceedings of the IET Conference Proceedings*, pp. 43–52, Shenzhen, China, December 2012.
- [22] K.-B. Lee, K.-S. J. I. J. o. B.-S. Hong, and Bio-Technology, “An implementation of leaf recognition system using leaf vein and shape,” vol. 5, no. 2, pp. 57–66, 2013.
- [23] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, “A leaf recognition algorithm for plant classification using probabilistic neural network,” in *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology*, pp. 11–16, IEEE, Madrid, Spain, December 2007.
- [24] I. Yahiaoui, O. Mzoughi, and N. Boujemaa, “Leaf Shape Descriptor for Tree Species Identification,” in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, pp. 254–259, Melbourne, Australia, July 2012.
- [25] Z.-Q. Zhao, L.-H. Ma, Y.-m. Cheung, X. Wu, Y. Tang, and C. L. P. Chen, “ApLeaf: an efficient android-based plant leaf identification system,” *Neurocomputing*, vol. 151, pp. 1112–1119, 2015.
- [26] N. Jamil, N. A. C. Hussin, S. Nordin, and K. Awang, “Automatic plant identification: is shape the key feature?” *Procedia Computer Science*, vol. 76, pp. 436–442, 2015.
- [27] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, “Deep-plant: plant identification with convolutional neural networks,” in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, pp. 452–456, Quebec City, Canada, September 2015.
- [28] W. Lee, S. W. Searcy, and T. Kataoka, “Assessing nitrogen stress in corn varieties of varying color,” *ASAE Annual International Meeting*, pp. 1–24, 1999, <http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.33.790>.
- [29] M. Ali, A. Al-Ani, D. Eamus, and D. Tan, “An algorithm based on the RGB colour model to estimate plant chlorophyll and nitrogen contents,” in *Proceedings of the International Conference on Sustainable Environment and Agriculture*, Switzerland, September, September 2014.
- [30] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [31] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” *Icml*, vol. 96, pp. 148–156, 1996.
- [32] W. S. Jeon and S. Y. Rhee, “Plant Leaf Recognition Using a Convolution,” *Neural Network*, vol. 17, no. 1, pp. 26–34, 2017.
- [33] R. Janani and A. Gopal, “Identification of selected medicinal plant leaves using image features and ANN,” in *Proceedings of the 2013 International Conference on Advanced Electronic Systems (ICAES)*, IEEE, Pilani, India, September 2013.
- [34] K. Mahdikhanelou and H. Ebrahimnezhad, “Plant leaf classification using centroid distance and axis of least inertia method,” in *Proceedings of the In 2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, pp. 1690–1694, IEEE, Tehran, Iran, May 2014.
- [35] V. Srivastava and A. Khunteta, “Comparative analysis of leaf classification and recognition by different SVM classifiers,” in *Proceedings of the In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 626–631, IEEE, Coimbatore, India, July 2018.

## Research Article

# An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning

**S. M. Taslim Uddin Raju** <sup>1</sup>, **Amlan Sarker** <sup>2</sup>, **Apurba Das** <sup>3</sup>, **Md. Milon Islam** <sup>1</sup>,  
**Mabrook S. Al-Rakhami** <sup>4</sup>, **Atif M. Al-Amri** <sup>4,5</sup>, **Tasniah Mohiuddin** <sup>6</sup>,  
**and Fahad R. Albogamy** <sup>7</sup>

<sup>1</sup>Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

<sup>2</sup>Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh

<sup>3</sup>Department of Industrial Engineering and Management, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

<sup>4</sup>Research Chair of Pervasive and Mobile Computing, Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

<sup>5</sup>Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

<sup>6</sup>Department of Computer Science and Engineering, Military Institute of Science and Technology, Dhaka 1216, Bangladesh

<sup>7</sup>Computer Sciences Program, Turabah University College, Taif University, Taif 21944, Saudi Arabia

Correspondence should be addressed to Mabrook S. Al-Rakhami; malrakhami@ksu.edu.sa

Received 4 April 2021; Revised 11 November 2021; Accepted 8 December 2021; Published 25 February 2022

Academic Editor: Shahzad Sarfraz

Copyright © 2022 S. M. Taslim Uddin Raju et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims to introduce a robust framework for forecasting demand, including data preprocessing, data transformation and standardization, feature selection, cross-validation, and regression ensemble framework. Bagging (random forest regression (RFR)), boosting (gradient boosting regression (GBR) and extreme gradient boosting regression (XGBR)), and stacking (STACK) are employed as ensemble models. Different machine learning (ML) approaches, including support vector regression (SVR), extreme learning machine (ELM), and multilayer perceptron neural network (MLP), are adopted as reference models. In order to maximize the determination coefficient ( $R^2$ ) value and reduce the root mean square error (RMSE), hyperparameters are set using the grid search method. Using a steel industry dataset, all tests are carried out under identical experimental conditions. In this context, STACK<sub>1</sub> (ELM + GBR + XGBR-SVR) and STACK<sub>2</sub> (ELM + GBR + XGBR-LASSO) models provided better performance than other models. The highest accuracies of  $R_2$  of 0.97 and 0.97 are obtained using STACK<sub>1</sub> and STACK<sub>2</sub>, respectively. Moreover, the rank according to performances is STACK<sub>1</sub>, STACK<sub>2</sub>, XGBR, GBR, RFR, MLP, ELM, and SVR. As it improves the performance of models and reduces the risk of decision-making, the ensemble method can be used to forecast the demand in a steel industry one month ahead.

## 1. Introduction

Demand forecasting indicates the prediction of the future needs of a product or service [1]. It is necessary to follow a procedure to attain a crystalline graph of the demand for identifying the pulse of the customer's need to hold their position in the market. From the last era, the steel industry in

Bangladesh is a fast-growing industry in the local market. The industries managed to manufacture a large amount of steel to fulfill both local and international markets, but producing a large amount of steel without proper forecasting causes various problems. Demand prediction is used to support many fundamental business assumptions, including turnover, total revenues, income, capital consumption,

chance evaluation and moderation plans, scope quantification, transportation and distribution plans, and more. Any type of misdeed assessment could cost decaying or scarcity of raw materials. It can also lead to overproduction or underproduction. All these cases erode the entire supply chain and total income, resulting in opportunity cost. Again, the entire industry setup depends on this demand, such as the amount of raw material, labor, and space. For these whole arrangements, time is also a crucial issue, as some processes have predefined deadlines that must be perfectly synchronized. For smart business strategy, the most important thing is to forecast the demand precisely but the industries do not have any intelligent method to measure the need perfectly. They follow the time series of their sales data and often skip factors, such as raw material supply, availability, and the number of workers at the factories, significantly influencing steel production.

Forecasting methods can be classified into three categories: (1) statistical methods, (2) artificial intelligence-based methods such as single machine learning (ML) methods, and (3) ensemble/hybrid methods. Most steel industries in Bangladesh use traditional statistical approaches. Statistical approaches, such as exponential smoothing [2], moving average [3], autoregressive moving average [4], and autoregressive integrated moving average [5], are most frequently used for time series prediction. The major drawbacks of these techniques are that the parameter values are fixed using statistical calculations. The error of estimation increases when the fluctuations in the entered data are high and do not yield convincing results for complicated time series patterns [6]. Thus, the companies need an intelligent decision support system that considers several factors.

Several researchers reveal that in the investigation of most cases, ML approaches have drawn much attention and could provide more accuracy than could traditional approaches [7]. Single artificial intelligence-based models, such as support vector machine (SVM), extreme machine learning, heuristic techniques, and multilayer perceptron (MLP), are widely used in various industrial aspects to predict demands because they demonstrate promising results in the areas of control, prediction, and pattern recognition [8–10]. Support vector regression (SVR) is popular for predicting future demand because of its outstanding generalization capability and no dependency over input space dimensionality [11]. It produces higher accuracy in agribusiness prediction [8] and supply chain demand forecasting [12]. Recently, MLP is used for monthly water demand prediction [10], wind speed prediction [13], and water demand prediction [14]. For improving MLP's prediction accuracy, different MLP architectures were used, and an optimization algorithm was used to tune its parameters [15]. The extreme learning machine (ELM) is another advanced model, which is a single hidden layer feed-forward neural network (SLFN) model with incremental learning speed and fast convergence, making it efficient and fast in learning [16]. It is widely used in applications, such as sales forecasting demand of fashion retailing [17] and sales prediction for the retail industry [18].

Since demand forecasting in steel industries is considerably challenging, it is impossible to solve this problem accurately using single ML models. No single model is ideally suited for various ML applications. Each method and application domain has some prerequisites, advantages, assumptions, and characteristics [19]. Generally, the performance of combined forecasting models is better than that of a single forecasting model [20]. The literature has described several strategies to enhance the predictive performance of regression models, and one of these is the regression ensemble [8]. The regression ensemble theory is built on ML, whose roots are related to the concept of divide-to-conquer, solving the constraints of ML models working in isolation [21]. An ensemble model is one in which numerous base models are constructed to address the same problem, with each model learning the dataset's feature attributes and making a prediction. As a result, the separate model's forecasts are integrated to generate the final projection. By combining the mean or weighted average, ensemble approaches for regression problems can be developed. The simple method of grouping regression ensembles by mean and weighted average is to use mean and weighted average. The regression ensemble models construct a collection of models in order to improve the predictive power of the selected models and the numerical goal variables [22, 23]. Ensemble methods are used in several studies, such as forecasting for energy consumption [24], agribusiness prediction [8], and wind power forecasting [25]. Although numerous frameworks have been established, there is always a need for improved forecasting accuracy and robustness, particularly in the steel industry.

This study proposes a new pipeline for demand forecasting in steel industries. From this aspect, this study explores the capacity of predictive regression ensemble models by comparing the ensembles among themselves and considering the single reference models to forecast the demand. The proposed pipeline includes data preprocessing, feature selection, hyperparameter tuning, cross-validation, and regression ensemble approaches to outperform the state-of-the-art results. Instead of using the median value of the attribute, the mean value of the attribute is utilized to fill in the empty area since it has a more central tendency to the mean of the attribute distribution than the median. The appropriate features are selected using feature selection algorithms (correlation-based, principal component analysis (PCA), and independent component analysis (ICA)) to avoid redundancy and model overfitting problems. Different single ML techniques, such as SVR, MLP, and ELM, are adopted as reference models. The ensemble bagging (RFR), boosting (GBR and XGBR), and stacking (STACK) models are used in our proposed framework to enhance demand forecasting robustness and efficiency. The grid search technique with cross-validation is used to select the optimal hyperparameters for each ML model. Comprehensive experiments are conducted on different data preprocessing and a combination of ML techniques to minimize the RMSE and maximize  $R^2$  of demand forecasting models. All experiments are carried out under the same experimental settings and with the same data set as the previous experiment. Finally,

we investigate the performance of regression ensemble approaches and verify that ensemble approaches outperform single reference models. The contributions of this paper are summarized as follows:

- (i) Collect the dataset from a well-known steel industry in Bangladesh.
- (ii) Present a modification of the theory underlying regression ensembles based on bagging (RFR), boosting (GBR and XGBR), and stacking (STACK) as well as single models (SVR, MLP, and ELM) (details in Supplementary Material Appendix A).
- (iii) Find the best preprocessing pipeline using filling missing values, data transformation, standardization, and feature selection algorithms where the number of selected features is also varied.
- (iv) Implement different ML regression models with its optimal hyperparameters, obtained using grid search algorithms with cross-validation. Investigate and analyze the performance of bagging, boosting, and stacking ensemble approaches and compare them with each other on the same dataset and preprocessing under the same experimental condition.
- (v) Verify the superiority of the proposed ensemble approaches using Friedman test and Wilcoxon signed rank test.

The remainder of the paper is arranged in the following manner: Section 2 describes a collection of related studies for the purpose of forecasting. Section 3 illustrates the suggested approach, dataset, feature selection methods, and assessment measures. Various experimental findings are documented in Section 4 based on the interpretation of the data. Section 5 provides a conclusion as well as a scope for further development.

## 2. Related Works

Forecasting demand for industrial products is an urgent matter since a massive portion of a company's planning process is based on the amount of product to be produced. To meet the increasing demand, precise demand forecasting is required. In this section, we will discuss the work that has been done to anticipate demand in a variety of disciplines and will describe numerous exemplary studies.

Ribeiro and dos Santos Coelho [8] proposed a system for agribusiness prediction using ensemble methods. Bagging, boosting, and stacking ensembles along with single reference models named SVR, MLP, and KNN were used for their purposes. In this experiment, it was shown that ensemble methods performed better than single models. They obtained MAPE of 0.9787 and 0.7394 for both cases for best ensemble models. They did not apply any metaheuristics algorithm for optimizing hyperparameters. Yu et al. [9] developed an ensembling and decomposition algorithm with EEML for crude oil price forecasting. In Ref. [12], they introduced a system by ensembling regression algorithms and time series algorithms to forecast the supply chain

demand. The system showed superior outcome because of the reality of invalidating the over-gauging and under-determining. Cankurt [26] employed a variety of regression models, including M5P and M5-Rule model trees, bagging, boosting, randomization, stacking, and voting, to anticipate tourism demand. In this case, they obtained  $R$  of 0.986 and a RAE of 14.96. The bagging and boosting methods have great significance for the improvement of performances in regression tree models.

Yang et al. [27] developed a system for forecasting agriculture commodities using the bagging and combining approaches with the Heterogeneous Autoregression (HAR) model. HAR model along with bagging and the principal component combination shows outstanding performance for agriculture commodities forecasting. In Ref. [28], they introduced a system by ensembling empirical mode decomposition (EEMD) to analyze global food price volatility. Tao et al. [29] proposed a method using a combination of ensemble empirical mode decomposition (EEMD), extreme learning machine (ELM), and ARIMA for forecasting hog price. They obtained the best-estimated accuracy of  $R = 0.848$ . Ribeiro et al. [30] designed nonlinear prediction models based on ensemble aggregation in order to improve the prediction accuracy of electricity load forecasting. In the proposed system, they used hourly load values from Italy in 2015 and Global Energy Forecasting Competition in 2012 to validate their proposed framework. Compared to the multilayer perceptron neural network (MPNN) and regression tree approach, their proposed forecasting framework based on wavelet ensemble provided a better performance.

da Silva et al. [31] introduced a decomposition-ensemble learning strategy for multi-step forward extremely short-term forecasting, which involved aggregating many regression models. They employed a range of preprocessing strategies to account for the system's high degree of input correlation. Across all time horizons, the proposed models outperform the CEEMD, STACK, and single models. In Ref. [32], they presented an excellent rolling decomposition-ensemble model for gasoline forecasting, which was both accurate and efficient. The researchers' experimental results demonstrate that the rolling decomposition-ensemble model is both accurate and resilient when it comes to projecting gasoline consumption levels and trends. A unique wind speed ensemble forecasting system (WSEFS) was developed by Liu et al. [33] in order to enhance point forecasting (PF) and interval forecasting (IF). They obtained MAPE of 1.9322%, 2.1579%, and 2.2808% for the 1<sup>st</sup> step, 2<sup>nd</sup> step, and 3<sup>rd</sup> step, respectively. The experimental results showed that the MOMA ensemble forecasting system is better than MOGWO and MODA. In order to estimate the sediment movement in open channels, Ebtehaj and Bonakdari [35] developed the ELM algorithm [35]. In all training and testing modes, the FFNN-ELM outperformed the FFNN-BP and GP methods, which were previously used. For the testing mode, they found RMSE = 0.121 and MARE = 0.023, respectively.

Considering the existing literature in Table 1, it is observed that ensemble models contribute significantly to determine predictions, more than traditional models in each

TABLE 1: Summary of most recent works for demand forecasting in various fields with their input factors and performances.

Publication	Objectives	Domain	Performance	Finding
Ribeiro and dos Santos Coelho [8]	Ensembling bagging (RFR), boosting (GBR and XGBR), and stacking (STACK), as well as adopting reference model SVR, MLP, and KNN	Agribusiness prediction	MAPE = 0.0093–1.6354, RMSE = 0.0013–0.0680	The ensemble approach outperforms the single models, especially the STACK model
Yu et al. [9]	Developing an ensembling and decomposition algorithm with ensemble empirical mode decomposition (EEMD) and extended extreme learning machine (EELM)	Forecasting crude oil price	MAPE = 0.0003, RMSE = 0.1431	This ensembling method shows better results than some existing popular model as well as single model in terms of accuracy, speed, and resilience
Adhikari et al. [12]	Ensembling time series methods and regression techniques in order to reduce forecast error from the actual value	Supply chain demand forecasting	TS FACC = 62% – 69% Reg FACC = 62% – 65%, En FACC = 65% – 71%	Showed superior outcome because of the reality of invalidating the overgauging and underdetermining and bringing the conjecture esteems closer to the genuine in the vast majority of the cases
Cankurt [26]	Developing M5P and M5-Rule model trees, randomization, boosting, bagging, voting, and stacking in order to anticipate the demand for tourism in Turkey	Tourism demand forecasting	$R = 0.9866$ , $R^2 = 0.973$ , RAE = 14.96, and RRSE = 16.77	The bagging and boosting methods have great significance for the improvement of performances in regression tree models
Yang et al. [27]	Developing the bagging and combining approaches with heterogeneous autoregression (HAR) model for the prediction of agriculture commodities' future	Forecasting agriculture commodities	$R^2 = 0.6263 - 0.3080$	HAR model with bagging shows outstanding performance comparing with AR benchmark
Wang et al. [28]	Ensembling empirical mode decomposition to analyze global food price volatility	Forecasting food price volatility	MSE = 74.29, MAE = 6.969, MAPE = 3.799	This model can successfully analyze the fluctuation of 3 types of agricultural commodities
Tao et al. [29]	Developing a combination of EEMD, ELM, and ARIMA	Forecasting hog price	$R = 0.281 - 0.848$	This model outperforms for the selected parts and claimed itself as an alternative for short-term forecasting for hog price
Ribeiro et al. [30]	Design of nonlinear prediction models for the ensemble aggregation of waveNet ensemble	Electricity load time series	–	All preprocessing stages and aggregation techniques contribute to overall performance, although perhaps not all to the same extent as a ceiling analysis would indicate
da Silva et al. [31]	For multi-step forward extremely short-term forecasting, decomposition-ensemble learning approaches are used. These methods include K-Nearest neighbors (KNN), partial least squares regression (PLSR), Ridge regression (RR), support vector regression (SVR), and Cubist regression (CR).	Wind energy forecasting	MAE = 101.32, MAPE = 8.63, RMSE = 138.97	CEEMD–BC–STACK a stacking-ensemble learning technique that significantly improved the accuracy of weak models CEEMD by merging and forecasting with a strong model.

TABLE 1: Continued.

Publication	Objectives	Domain	Performance	Finding
Yu et al. [32]	Proposing decomposition-ensemble learning model (ARIMA, SVR, ANN, RVFL, KRR, and ELM)	Gasoline forecasting	MAPE = 0.02 – 0.04	Decomposition-ensemble is better for prediction. Ensemble model or instantaneous frequency analysis is applicable for complex and irregular characteristics.
Liu et al. [33]	Developing a revolutionary wind speed ensemble forecasting system (WSEFS) to enhance point forecasting (PF) and interval forecasting (IF)	Wind speed forecasting	MAPE – 1 <sup>st</sup> step, 2 <sup>nd</sup> step, and 3 <sup>rd</sup> step are 1.9322%, 2.1579%, and 2.2808%, respectively.	VMD technology is better than mayfly algorithm (MA) and ICEEMDAN. MOMA ensemble forecasting system is better than MOGWO and MODA.
Cook and Weisberg [34]	Developing imperialist competitive algorithms (ICA) and particle swarm optimization (PSO) algorithms were compared with the results of the MLP neural network trained with the back propagation algorithm	Nondeposition sediment transport prediction	MAPE = 2.7% – 6.52% and RMSE = 0.009 – 0.042	In comparison to the PSO and MLP algorithms, the ICA method is more accurate for computing the densimetric Froude number in pipe channels
Ebtehaj and Bonakdari [35]	Developing an extreme learning machine (ELM) and comparing with back propagation (BP), genetic programming (GP), and existing sediment transport equation	Sediment transport estimation	RMSE = .309 and MARE = .059	FFNN-ELM performs well and is also an alternative method in predicting the Fr

\* MAPE = mean absolute percentage error, RMSE = root mean square error, MSE = mean square error, MAE = mean absolute error,  $R$  = Pearson correlation,  $R^2$  = coefficient of determination, FACC = forecast accuracy check, TS FACC = time series FACC, Reg FACC = regression FACC, En FACC = ensemble FACC, RAE = relative absolute error, RRSE = root relative square error, CEEMD = complete ensemble empirical mode decomposition, VMD = variational mode decomposition, MOMA = multiobjective Mayfly algorithm, ICEEMDAN = improved complete ensemble empirical mode decomposition with adaptive noise, MOGWO = multiobjective grey wolf optimizer, MODA = multiobjective dragonfly algorithm, FFNN = feed-forward neural network, Fr = densimetric Froude number, MARE = mean absolute relative error.

case. Although several frameworks have already been developed, there is still a need for improvement in the accuracy and robustness of demand forecasting, especially in the steel industry. To sum up, there is up to now no proper pipeline for data preprocessing, features selection, hyperparameter tuning, and finally developed a regression ensemble method. This study uses bagging, boosting, and two-level stacking ensemble methods by analyzing the time series of historical data from the steel industry to achieve more propriety of forecasting results for demand. The steel industry follows the traditional time series trend to predict the demand, which fluctuates at a high quantity. To avoid this problem, this study combines multiple approaches instead of using a traditional single method to determine the precise result for the industry.

### 3. Materials and Methods

This section contains a concise description of the materials and method used. The suggested framework is depicted in Figure 1. The following are the primary phases in our suggested framework: (i) collection of industrial environmental data as the primary inputs of the framework; (ii) preprocessing the data including filling the missing values, Yeo–Johnson transformation, and standardization; (iii)

discarding the irrelevant and redundant features to avoid overfitting of the models; (iv) applying the grid search algorithm with cross validation for hyperparameter tuning for each machine learning model; (v) development of two-level stacking ensemble method, where machine learning models with optimal hyperparameters are used as the baseline model; and (vi) evaluation metrics used to evaluate the proposed framework. These blocks are explained in the following sections.

**3.1. Data Collection.** The data were collected from a well-known prominent steel company named Bangladesh Steel Re-Rolling Mills Ltd., in Chittagong, Bangladesh. During the industrial attachment, some raw data were procured from sources, such as workers, production leaders, and human resources. Later, the data were closely knitted to build the dataset. The dataset comprises 132 cases and six input features from January 2009 to December 2019 (11 years). The key responsibility is to identify the demand of every month based on other factors. The dataset holds the amount of raw material used in a month, availability, the number of workers, working days, and other attributes. The data were gathered from their monthly and annual industrial reports from their official website, such as financial reports,

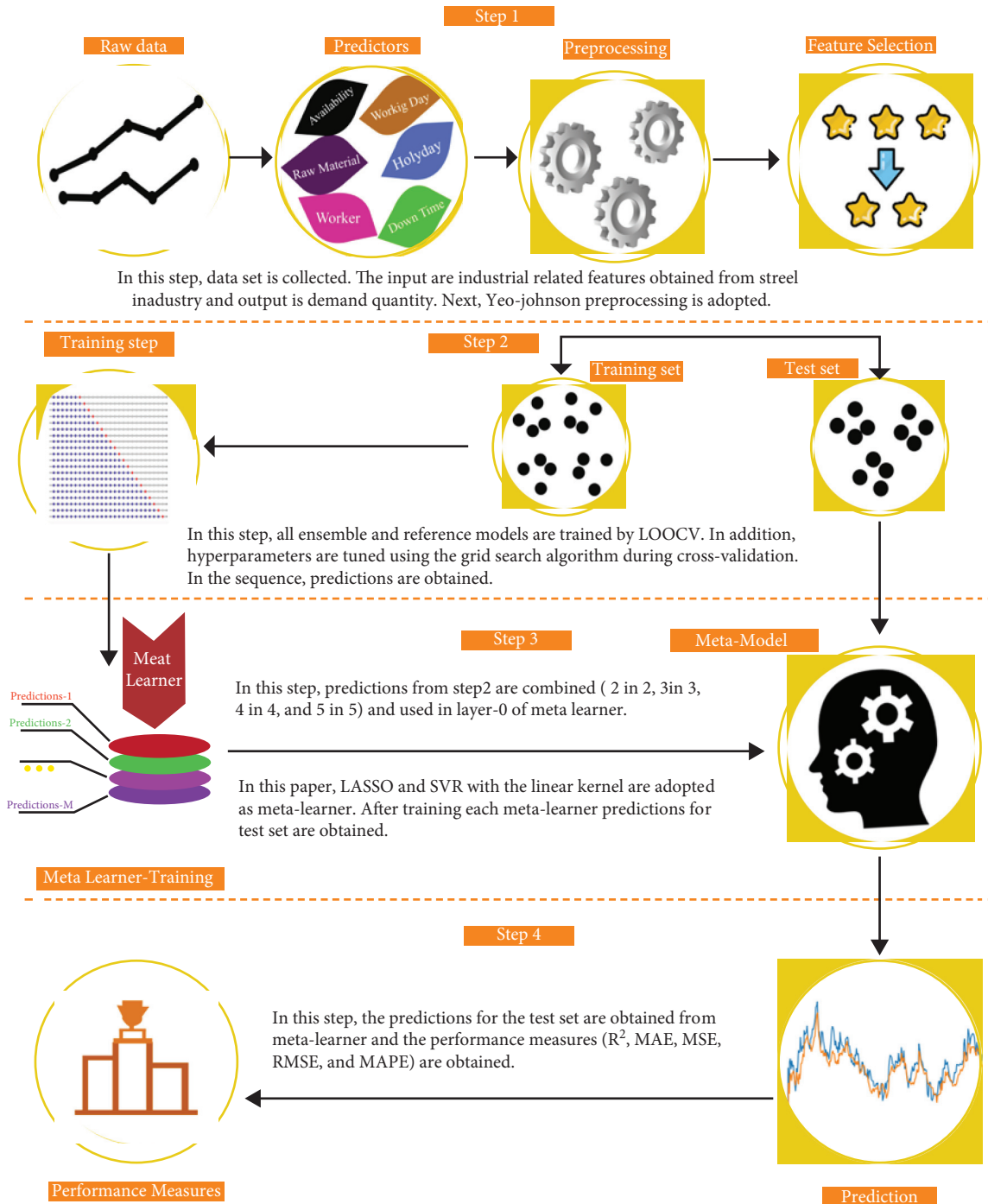


FIGURE 1: Proposed architecture of automatic demand forecasting.

production reports, and some other necessary factors directly affecting their production achievements. Table 2 describes each feature and shows a statistical summary.

**3.2. Data Preprocessing.** The data preprocessing stage comprises missing value imputation and power transformation of data. Raw data inherit some missing attributes from various features that must be filled before applying any ML technique. Several imputation techniques can fill

missing values. In our proposed method, the mean-based imputation technique is used, where the missing value is filled with the mean of the attributes of that specific feature.

After the imputation of missing or null values, the data power transformation is performed. In regression analysis, transformations are crucial [36]. Parametric, monotonic transformations are power transformations used to make data more Gaussian-like. This technique is useful in heteroscedasticity problems or other circumstances where data normality is required. Among the two most popular power



TABLE 2: Statistical description of influence features for forecasting the demand.

Indices	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	Outcome
Mean	8231.97	52444.08	167.42	28.93	1.46	60.09	51205.62
S.D.	2269.08	12385.54	43.10	1.28	0.90	9.68	10848.99
$Q_{25}$	6985.25	43731.75	135.00	28.00	1.00	52.00	45167.18
$Q_{50}$	7883.50	52941.50	150.00	29.00	1.00	62.00	52418.97
$Q_{75}$	9640.50	59861.00	200.00	30.00	2.00	67.25	61496.95
Range	(4010, 10178)	(30275, 60000)	(100, 200)	(26, 30)	(0, 4)	(38, 76)	(26072, 62275)
Skewness	0.297	0.123	0.454	0.322	1.403	-0.309	-0.624
Kurtosis	-0.520	-0.801	-0.740	2.569	1.180	-0.829	-0.553
Data type	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric

\* S.D. = standard deviation,  $F_1$  = availability,  $F_2$  = raw material,  $F_3$  = worker,  $F_4$  = working day,  $F_5$  = holiday,  $F_6$  = down time, outcome = demand level,  $Q_{25}$  = first quartile,  $Q_{50}$  = second quartile,  $Q_{75}$  = third quartile.

transformations methods are the Box-Cox and Yeo-Johnson transformations. Here, the Yeo-Johnson transformation is used because the Box-Cox transformation demands that input data are strictly positive, whereas both positive and negative data are endorsed by the Yeo-Johnson transformation [37]. The description of the Yeo-Johnson transformation can be given using

$$y^* = \begin{cases} \frac{((y+1)^\lambda - 1)}{\lambda}, & \text{if } \lambda \neq 0, y \geq 0, \\ \log(y+1), & \text{if } \lambda = 0, y \geq 0, \\ -\frac{[(-y+1)^\lambda \{2-\lambda\} - 1]}{(2-\lambda)}, & \text{if } \lambda \neq 2, y < 0, \\ \log(-y+1), & \text{if } \lambda = 2, y < 0, \end{cases} \quad (1)$$

where  $y^*$  is the transformed value,  $y$  is a list of  $n$  strictly positive numbers, and  $\lambda$  is a hyperparameter used to control the transformation. Here, Scikit-learn implementation of PowerTransformer (method = "Yeo-Johnson", \*, standardize = True) is used, performing the Yeo-Johnson power transformation operation with implicit data standardization with zero mean and unit variance to the transformed output.

**3.3. Feature Selection.** Feature selection or reduction reduces irrelevant, redundant, or partially important features that might mislead the model prediction, as the accuracy of an ML model depends on the features on which it has been trained. Feature reduction reduces the chances of overfitting because of the reduction of the redundant feature and lessens the model's complexity. Several feature selection or reduction techniques exist. In our proposed method, PCA, ICA [36], and correlation-based feature selection algorithms were used to discard irrelevant features.

PCA is frequently employed in this capacity due to its adaptability and ease of implementation. PCA works on the premise of dividing data into an orthogonal space so that the eigenvectors corresponding to the greatest eigenvalues preserve the maximum data variance. PCA is a technique that focuses on the covariance matrix and second-order statistics. ICA decomposes observable data linearly into

statistically independent components. For the correlation-based method, it classifies characteristics using a heuristic evaluation function that takes into account the correlation between the target outcome and their features. The design structure of both PCA and ICA follows the default implementation of Scikit-learn except the  $n\_components$  parameter, resembling the number of features to be chosen by the respective algorithm, as the value of the parameter is driven from hyperparameter tuning. The design of PCA can be illustrated, respectively, such as (n\_components, copy, whiten, svd\_solver, tol, iterated\_power) = ({4, 5, 6}, True, False, auto, 0.0, auto). Algorithms 1-3 summarize the procedures of PCA, ICA, and correlation-based feature selection algorithms, respectively.

**3.4. Hyperparameters Determination.** Hyperparameters define those values directly controlling the learning process of ML techniques and can be arbitrarily set by the user before starting the training phase. The correct combination of values is significant in achieving the best and quality model. Choosing the correct values for the optimal model is known as hyperparameter optimization or hyperparameter tuning [38]. Grid search and random search are both well-known techniques when tuning the hyperparameters of an estimator. This study used the grid search method based on cross-validation, resulting in the most precise predictions [39]. This algorithm splits the range of parameter values to be upgraded into the grid and across all points to obtain the optimal parameters. Different parameter combinations were evaluated for each model, which were divided into training and test sets using the cross-validation method [39]. Table 3 provides an overview of hyperparameters tuned using ML techniques and their range of tuning.

**3.5. Cross-Validation in Time Series.** Cross-validation is a widely used validation approach for tuning hyperparameters and assessing the effectiveness of machine learning techniques [40]. Different parameters must be stated for each case depending on the dataset. A grid search technique combined with cross-validation is effective at identifying the optimal hyperparameter combination for each model. As a consequence, forecasting errors associated with test samples may be decreased, allowing for the determination of the ideal

**Input:**  $m$ - dimensional input data matrix  $X \in \mathbb{R}^m$  with number of samples  $N$ , and variance threshold  $T_{\text{var}}$   
**Output:** reduced  $L$ - dimensional data matrix  $Y \in \mathbb{R}^L, L < m$ ,  
 Load  $X \in \mathbb{R}^m$ , and calculate mean for each feature,  $\mu_j = 1/N \sum_{i=1}^N X_{ij}$  for  $j = 1, 2, \dots, m$ ; subtract the mean from each corresponding dimension,  $X_{ij} = X'_{ij} - \mu_j$  for  $j = 1, 2, \dots, m$  and  $i = 1, 2, \dots, N$ ;  
 /\* Make each signal uncorrelated to each other \*/  
 Calculate covariance matrix of  $X'$ ,  $\sum_{m \times m} 1/N - 1 [X']^T = X'$ ;  
 Solve the  $\sum_{m \times m}$  as  $\sum_{m \times m} = V^{-1}DV$ , where  $V \in \mathbb{R}^m$  is the matrix of eigenvector and  $D_{m \times m}$  is the diagonal matrix containing eigenvalues on both sides of the diagonal matrix ;  
 Sort the eigenvector matrix  $V$  in the descending order to the first  $L$ - eigenvector that have variance  $\geq T_{\text{val}}$  and form a projection matrix  $P_{m \times L}$ ;  
 Finally, project on the PCA space,  $Y = P^T X$ ;

ALGORITHM 1: Steps for the implementation of principal component analysis (PCA).

**Input:**  $m$ - dimensional input data matrix  $X \in \mathbb{R}^m$  with number of samples  $N$ , and variance threshold  $T_{\text{var}}$   
**Output:** reduced  $L$ - dimensional data matrix  $Y \in \mathbb{R}^L, L < m$ ,  
 Select a nonquadratic nonlinear function  $g$ ;  
 Initialize  $W$  as  $X = WH$ , where  $W \leftarrow$  ratio of source during mixing,  $H \leftarrow$  matrix contains different components, and  $X \leftarrow$  mixed output;  
 Perform PCA on  $X$ , as  $X \leftarrow \text{PCA}(X)$  as in Algorithm 1;  
**while**  $W$  changes **do**  
   Update  $X \leftarrow E\{Xg(W^T X)\} - E\{g'(W^T X)\}$ ;  
   Normalize  $X \leftarrow W/\|W\|$ ;  
 Derive the new dataset by taking  $Y = W^T X$ , where  $Y \in \mathbb{R}^L$ ;

ALGORITHM 2: Steps for the implementation of independent component analysis (ICA).

**Input:**  $m$ - dimensional input data matrix  $X \in \mathbb{R}^m$  with number of samples  $N$ , and expected outcome,  $Y_O \in \mathbb{R}$   
**Output:** reduced  $L$ - dimensional data matrix  $Y \in \mathbb{R}^L, L < m$ ,  
**for**  $p \leftarrow 1$  **to**  $p \leq m$  **do**  
 $r_{pO} = \sum (X_p - \bar{X}_p)(Y_O - \bar{Y}_O) / \sqrt{\sum (X_p - \bar{X}_p)^2} \sqrt{\sum (Y_O - \bar{Y}_O)^2}$   
 Sort the correlation  $r_{pO}$  in descending order to choose first  $L$  features for  $Y \in \mathbb{R}^L$ ;

ALGORITHM 3: Steps for the implementation of correlation-based feature selection (Corr).

collection of hyperparameters that enhance predictive performance while minimizing model overfitting [41]. The leave-one-out cross-validation procedure is acceptable in this scenario when dealing with time series data [42]. Alternatively, this method can be considered a sequential block cross-validation procedure and a subset of K-fold cross-validation.

Thus, the training set is iteratively constructed, with the training and validation sets being utilized concurrently, a process known as rolling cross-validation. This procedure is performed several times, with each iteration increasing the amount of observations in the training set and decreasing them in the validation set. The associated training set comprises only observations that happened before the observation in the test set. The dataset is partitioned into training and test sets, with 70% of the data used for training and verifying the models. The time series split notion is to divide the training set in half at each iteration, assuming that the validation set is still ahead of

the training split. It is initially trained on a limited subset of data in order to forecast the next data point. Following that, the forecasted data points are incorporated into the succeeding training dataset, and subsequent data points are forecasted. This process is repeated until the complete training set has been utilized. Calculate the training outcome by estimating iteration performance assessments.

**3.6. Structure of Stacked Ensemble Modeling.** STACK modeling was conducted by considering two stages, level 0 and level 1, and the predictions of the base learner (level 0) are combined with the meta-learner (level 1). From the previous studies, it is shown that the support vector regression (SVR) and selection operator (LASSO) regression are used as the meta-learner [8, 25]. The key advantages of adopting SVR, and especially layer-1 in the STACK technique, are its ability to identify predictor nonlinearities and subsequently exploit

TABLE 3: Different machine learning techniques with hyperparameters to be tuned by the grid search algorithm during cross-validation.

Algorithms	Hyperparameters	Explanation	Grid
SVR	C	Regularization parameter of the error term	$1 - 1 \times 10^{-3}$
	Kernel	Kernel types applied in the algorithm	Linear, polynomial, RBF
	Epsilon	Border of tolerance	0.1 - 1
	Gamma	Kernel coefficient for rbf	$1 \times 10^{-3} - 0.1$
RFR	n_estimators	Number of trees in a forest	5 - 15
	Criterion	Measurement of the quality of a split	mae or mse
	max_depth	Highest depth of the tree	2 - 10
	min_samples_leaf	Least number of instances needed to split an internal node	2 - 10
	min_samples_split	Least number of instances needed to be at a leaf node	2 - 10
MLP	initial_learning_rate	Learning rate value at the starting point of training	$1 \times 10^{-3} - 0.1$
	Solver	Used for weight optimization	lbfgs, sgd, Adam
	learning_rate_adjustment	Learning the rate adjustment depending on the cost function's current value	Constant, adaptive
	hidden_layer_sizes	Layer: Number of layers between input and output layers Neurons: Number of hidden layer neurons	1 ~ 7 (4, 8, 12)
	activation_functions	Output of each neuron	Logistic, tanh, relu
	Alpha ( $L_2$ penalty)	Reduces the influence of input parameters	$1 \times 10^{-3} - 0.1$
ELM	n_neurons	Number of hidden layer neurons	8
	activation_functions	Transformation function of hidden layer neurons	relu
	Alpha	Regularization strength	0.001
GBR	n_estimators	Number of boosting stages to carry out	200 - 500
	max_features	Number of features while considering the best split	Sqrt
	min_samples_leaf	Least number of instances needed to be at a leaf node	2 - 10
	max_depth	Utmost depth of individual regression estimators	2 - 10
	learning_rate	Shrinks the contribution of each tree	0.1 - 1
	Loss	Loss function based on order information of input variables	ls, lad, huber, quantile
XGBR	cosample_bytree	Subsample ratio of columns while building each tree	0.1 - 1
	Subsample	Subsample ratio of training samples	0.1 - 1
	reg_lambda	$L_2$ regularization On weight	0.1 - 1
	reg_alpha	$L_1$ regularization On weigh	0.1 - 1
	min_child_weight	Least sum of sample weight required in a child	1 - 10
	learning_rate	Step size reduction to prevent overfitting	$1 \times 10^{-3} - 0.1$
LASSO	Alpha ( $L_1$ penalty)	A constant value that multiplies $L_1$	$1 \times 10^{-4} - 1 \times 10^2$

them to improve demand forecasts [8]. The SVR with linear kernel and selection operator (LASSO) regression model was utilized as a meta-learner in our experiment (level 1).

The following steps were adopted in this work.

- (1) After doing the training session of the SVR/LASSO, RFR, MLP, ELM, GBR, and XBR models, the predicted results are combined (2 in 2, 3 in 3, 4 in 4, and 5 in 5) to build a STACK (SVR/LASSO) layer 0. Stack layer 0 does not use the model used in layer 1.
- (2) For each STACK model, 56 models are analyzed, and best one is chosen for the study based on the test set results.
- (3) The findings in Tables S1 and S2 indicate that models numbered 1-15 indicate a model combination of 2 in 2, models numbered 16-35 indicate a model combination of 3 in 3, models numbered 36-50 indicate a model combination of 4 in 4, and models numbered

51-56 indicate a model combination of 5 in 5 in the order specified in step 1.

- (4) The performance evaluation measurements are achieved for the training and test sets after training each STACK model.

The working procedure of the stacking ensemble in this paper is described in Algorithm 4.

**3.7. Performance Measures.** Estimating the model's accuracy is crucial in designing ML models to define how well the model is predicting. It is used to determine the goodness of fit among models and data to compare various models for model selection. If  $y_1, y_2, \dots, y_t$  are  $T$  actual values and  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t$  are corresponding predicted values, then the formulas are for evaluating the accuracy of the regression models as follows:

```

Input: Input dataset  $D = \{X_i, y_i\}_{i=1}^m$ , where  $(X \in \mathbb{R}, y \in Y)$ ,  $\Theta_{\text{set}}$  is the set of optimal hyperparameter for each based regression model,  $M$  is number of based model,  $T$ .
Output: final forecast demand level  $Y_f$  and performance indices.
Step 1: learn first-level base regression models;
/* Loop for train and evaluate the first-level individual /regressor *
for  $t \leftarrow 1$  to  $T$  do
  Divide the dataset  $D$  into  $D^{\text{train}}$  and  $D^{\text{test}}$ ;
  /* 70% data for training and validation, 30% for test set */
  /* Leave-One-Out Cross-Validation */
  for  $i \leftarrow 1$  to  $K$  ( $K \leftarrow \text{size of } D^{\text{train}}$ ) do
     $D_i^{\text{val}} = D^{\text{train}}(i, :)$   $\Rightarrow D_i^{\text{train}} = D^{\text{train}}/D_i^{\text{val}}$ ;
    Train  $M_t$  with optimal hyperparameter set  $\Theta_{\text{set}}$  on  $D_i^{\text{train}}$ ;
    Predict the demand level for  $M_t$  with  $D_i^{\text{val}}$ :  $h_t \leftarrow M_t(D_i^{\text{val}})$ ;
Step 2: create a new dataset from  $D$ ;
for  $t \leftarrow 1$  to  $T$  do
  Create a new dataset  $D_{m \times l}' = \{X_l', y_i\}$  for meta-regressor,
  Where  $X_l' = \{h_1, h_2, \dots, h_l\}$ ,  $h_l \leftarrow$  output of  $i^{\text{th}}$  model,  $l \leftarrow$  number of based model;
Step 3: learn second-level regressor model;
/* Loop for train and evaluate the final-level meta-regressor model
* for  $j \leftarrow 1$  to  $K$  ( $K \leftarrow \text{size of } D^{\text{train}}$ ) do
   $D_j^{\text{val}} = D^{\text{train}}(j, :)$   $\Rightarrow D_j^{\text{train}} = D^{\text{train}}/D_j^{\text{val}}$ ;
  Train the meta-model  $H_{\text{meta}}$  with  $D_j^{\text{train}}$  using  $\Theta_{\text{set}}$ ;
  Predict the demand level for  $H_{\text{meta}}$  with  $D_j^{\text{val}}$ ;
Test set  $D^{\text{test}}$  are used for the prediction and performance measure ( $P_{H_{\text{meta}}}$ ) using  $H_{\text{meta}}$ 
return  $P_{H_{\text{meta}}}$ ;

```

ALGORITHM 4: Demand forecasting using Stacking Ensemble techniques using cross-validation.

$$\begin{aligned}
 R^2 &= 1 - \frac{\sum_{t=1}^T (y_t - \hat{y}_t)^2}{\sum_{t=1}^T (y_t - \bar{y})^2}, \\
 \text{MAE} &= \frac{1}{T} \sum_{t=1}^T |y_t - \hat{y}_t|, \\
 \text{RMSE} &= \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2}, \\
 \text{MAPE} &= \frac{1}{T} \sum_{t=1}^T \left| \frac{y_t - \hat{y}_t}{y_t} \right|
 \end{aligned} \tag{2}$$

where  $\bar{y} = 1/T \sum_{t=1}^T y_t$  and in this paper, training set  $t = 1, \dots, 90$  and test set  $t = 91, \dots, 132$  are adopted.

Along with the performance evaluation matrix mentioned above, several statistical tests [43, 44] are performed in this study to ensure the superiority of the proposed approach. The Friedman test is used to examine if the absolute percentage errors (APE) of the two models differ statistically significantly. Once statistical significance has been established, post hoc tests (nonparametric tests), such as the Wilcoxon signed-rank test, can be employed to assess if the APEs of the models change when compared to one another (lower tail) [44, 45]. Wilcoxon's null hypothesis indicates that there is no difference in APE between models 1 and 2, but the alternative hypothesis states that model 1 has a lower APE than model 2.

## 4. Experimental Results and Discussion

In this section, the preparatory analysis of steel industrial data used in this study is demonstrated in Section 4.2. The performance of the adopted models and statistical tests for test set errors are described in Section 4.3. Tables S1 and S2 represent the performance measurement indices of the 56 generated models.

*4.1. Experimental Setup.* A single computer (Asus X556U with an Intel® Core (TM) i5-7200U, central processor unit running at 2.50 GHz, 8.0 GB of random access memory, and an Nvidia GeForce 940MX graphics card) running the Windows 10 operating system was used to create the findings provided in Section 4. In order to implement the machine learning approaches and ensemble methods, we used the Python 3.6 programming language in conjunction with the Spyder computing environment, which is included in Anaconda.

*4.2. Exploratory Analysis.* Correlation analysis is a statistical approach used to determine the connection between two numerical variables. From an ML viewpoint, it indicates how the features correspond to the outcome. However, it is challenging to identify how features are interconnected. Data visualization can help determine how individual features might correlate with the outcome. Pearson's correlation coefficient is used to identify the relationship between two variables in a statistical analysis. In the range of +1 to -1, it means that there is no correlation at all, +1 indicates that there is a perfect positive correlation, and -1

indicates that there is a perfect negative correlation, according to the definition. After the Yeo–Johnson transformation has been performed to the training data set, the correlation matrix for the exploratory variables is shown in Figure 2. Figure 2 depicts the color scale of its association, which is represented on the right-hand side of the illustration. The light color indicates a close relation of 0, whereas the intense color indicates a close relation of +1 or -1. The indicators ( $F_1$ ,  $F_2$ , and  $F_3$ ) and the response variable (Demand) are highly positively correlated. Thus, the increment or decrement in the value of one tends to increment or decrement those that are highly correlated. However, indicator ( $F_5$ ) is negatively correlated to the outcome (Demand), indicating that if the number of holidays in a month increases, the number of demands decreases and vice versa.

**4.3. Evaluation of Proposed Models.** In this study, the proposed models are trained using a set of optimal hyperparameters achieving the maximum predictive performance of each model achieved by grid search. The steel production data from January 2009 to December 2019, covering 132 months, are taken as the training and testing sets.

Table 3 presents an overview of hyperparameters tuned for each ML model, their explanation, and turning ranges. Table 4 represents the quantified results for selecting the best performing preprocessing and the number of selected features and ML models, where  $R^2$  with standard deviation is stated for comparison. Table 5 summarizes each model's capacity to obtain the highest  $R^2$  using the suggested pipeline, along with the optimal preprocessing and feature selection algorithms and the number of selected features. In addition, Table 5 illustrates the best-tuned hyperparameters using the grid search. The analysis of Table 4 reveals that when suitable preprocessing is used, various models produce superior outcomes. The different architectures of the MLP model are shown in Table 6. Table 7 summarizes the performance metrics used to evaluate each model, which include  $R^2$ , MAE, RMSE, and MAPE. When either correlation-based or PCA-based feature selection is applied, each model achieves the best results for filling missing values, Yeo–Johnson transformation, and data normalization (Tables 4 and 5). For SVR, the estimated accuracy of  $R^2 = 0.931$  is obtained from preprocessed data and correlation-based feature selection.

The comprehensive experiments were performed on the same dataset to get the best architecture for the MLP model. Eight separate MLP models (Table 6) were implemented and evaluated, with 1–7 hidden layers, where the number of neurons served as a hyperparameter for selecting the best numbers. The experimental results in Figure 3 indicate that the optimal architecture is the MLP layout with  $M = 4$  hidden layers ( $H_1$ ,  $H_2$ ,  $H_3$ , and  $H_4$ ) and  $N_1 = 12$ ,  $N_2 = 12$ ,  $N_3 = 12$ , and  $N_4 = 8$  neurons. In addition, the presence of additional hidden layers with fewer samples, like in the steel dataset, limits the MLP model's capability (Figure 3). Because of the limited data, such as in the steel dataset, the wide depth of the MLP model could be overfitted and cause gradient fading problems. Table 3 lists the optimal hyperparameters of the best MLP model. The models have used the ReLU activation function and Adam solver. It was trained on 200 epochs with a constant learning

rate, batch size, and a regularization parameter of 0.01, 32, and 0.1, respectively. To reduce overfitting, the dropout layer was used, randomly dropping 60% of neurons. The highest accuracy  $R^2$  from the MLP model is 0.961 when we perform data preprocessing and PCA-based feature selection. Similarly, the ELM model with eight neurons in the hidden layer obtained the best result. Table 3 lists the optimal hyperparameters of the best ELM model. The model used the ReLU as the transformation function of hidden layer neurons, and the optimal regularization parameter was 0.001. The best-estimated accuracy ( $R^2$ ) of the ELM model with preprocessed data and correlation-based feature selection is 0.942.

Feature selection methods are used to improve the overall performance of each model (correlation-based, PCA, and ICA). It is possible to reduce the dimensions of a higher-dimensional space to a lower-dimensional space using PCA by selecting the orthogonal projections with the highest variance. The ICA theory implies that data are only partly independent if their variances across characteristics are larger than their covariance. The number of computers being used has a significant impact on PCA performance. Because the ICA-based feature selection technique is used to find newly specified mutually independent components, it is possible that correlation with the desired output will be lost when the procedure is used to discover new predefined mutually independent components. Due to the fact that both PCA and ICA create new components in an unsupervised manner, it is not possible to guarantee greater performance on the steel dataset. Correlation-based feature selection, on the other hand, takes into consideration the relationship between quality and outcomes in order to discover the most closely related features. As shown in Table 4, the majority of models perform better when four features,  $F_1$ ,  $F_2$ ,  $F_3$ , and  $F_6$ , are used. These four features were chosen using a correlation-based feature selection technique.

Further improvement of demand forecasting was obtained using regression ensemble models. Bagging (RFR), Boosting (GBR and XGBR), and stacking (STACK) regression ensembles were adopted to improve the performance of demand forecasting. Table 5 presents the performance evaluation of the adopted models. Furthermore, the results are sorted regarding  $R^2$  in the ascending order for the test set results. Finally, the best models present the lower RMSE and higher  $R^2$  in the test set. RFR is the ensemble learner built-in unpruned decision tree, and it reduced the effects of overfitting by combining multiple trees. Table 5 shows the optimal hyperparameters for the RFR model. The best-estimated accuracy ( $R^2$ ) of the RFR model is 0.966 obtained from preprocessed data and PCA-based feature selection. The RFR performance of the models is better for SVR, MLP, and ELM in terms of the RMSE, that is, it has lower RMSE values. GBR and XGBR are also used to increase the accuracy of forecasts. Extreme gradient boosting is a specific variant of the gradient boosting strategy that discovers the ideal tree model by employing a more exact approximation than the conventional gradient boosting method. The best-estimated accuracy ( $R^2$ ) of the GBR model is 0.969, obtained from preprocessed data and correlation-based feature selection. The XGBR can reduce the loss by showing an extreme gradient capability. The highest accuracy ( $R^2$ ) of XGBR is 0.974, and the lowest RMSE is 0.151. The RMSE of XGBR is significantly lower

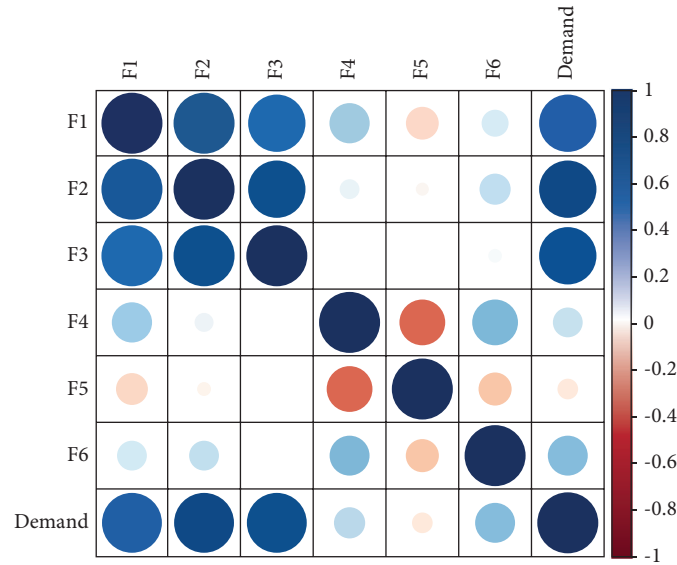


FIGURE 2: Correlation matrix for all influence features corresponding to the demand.

TABLE 4: Summary of all extensive experiments to select the best performing preprocessing feature selection methods with the number of features and regression models.

ML method (s)	Preprocessing	Algorithm	n_features	Performance ( $R^2$ )
SVR	Raw data	N/A	6	0.898
	Processed data	N/A	4	0.919
		<b>Corr</b>	<b>4</b>	<b>0.931</b>
		PCA	4	0.923
		ICA	4	0.923
RFR	Raw data	N/A	6	0.918
	Processed data	N/A	4	0.929
		Corr	4	0.966
		<b>PCA</b>	<b>4</b>	<b>0.967</b>
		ICA	4	0.952
MLP	Raw data	N/A	6	0.845
	Processed data	N/A	4	0.911
		Corr	4	0.957
		<b>PCA</b>	<b>4</b>	<b>0.961</b>
		ICA	4	0.950
ELM	Raw data	N/A	6	0.841
	Processed data	N/A	4	0.849
		<b>Corr</b>	<b>4</b>	<b>0.942</b>
		PCA	4	0.887
		ICA	4	0.909
GBR	Raw data	N/A	6	0.934
	Processed data	N/A	4	0.944
		<b>Corr</b>	<b>4</b>	<b>0.969</b>
		PCA	4	0.949
		ICA	4	0.934
XGBR	Raw data	N/A	6	0.949
	Processed data	N/A	4	0.953
		<b>Corr</b>	<b>4</b>	<b>0.974</b>
		PCA	4	0.956
		ICA	4	0.952

\* N/A = none. Note: the best approaches were shown in bold type.

than the reference models and RFR and GBR. The best result of XGBR is obtained when a child's minimum amount of weight is less than 4, and a subsample ratio to construct a tree is 0.7.

Finally, the stacking ensemble method is used for integrating multiple-base models in order to reduce prediction errors to the smallest possible amount. According to the results from the test set, level 0 of the STACK<sub>1</sub> method is

TABLE 5: Best--performing ML model and preprocessing and tuned hyperparameters with the highest possible accuracy ( $R^2$ ).

ML techniques	Best preprocessing	Best hyperparameters	Performance
SVR	Processed data Corr (n_attributes: 4)	C: 100 Kernel: RBF Epsilon: 0.1 Gamma: 0.001	$R^2$ : 0.931
	Processed data PCA (n_attributes: 4)	n_estimators: 10 Criterion: mse max_depth: 8 min_samples_leaf: 2 min_samples_split: 2	$R^2$ : 0.966
MLP	Processed data PCA (n_attributes: 4)	initial_learning_rate: 0.01 Solver: Adam learning_rate_adjustment: Constant hidden_layer_sizes: (12, 12, 12, 8) activation_functions: relu Alpha ( $L_2$ penalty): 0.01	$R^2$ : 0.961
ELM	Processed data Corr (n_attributes: 4)	n_neurons: (8) activation_functions: relu Alpha: 100	$R^2$ : 0.942
GBR	Processed data Corr (n_attributes: 4)	n_estimators: 250 max_features: Sqrt min_samples_leaf: 2 max_depth: 2 learning_rate: 0.2 Loss: lad	$R^2$ : 0.969
	Processed data Corr (n_attributes: 4)	cosample_bytree Subsample reg_lambda reg_alpha min_child_weight learning_rate	$R^2$ : 0.974
XGBR	Processed data Corr (n_attributes: 4)		
STACK (SVR) <sub>1</sub>	Processed data Corr (n_attributes: 4)	C: 100 kernel: RBF	$R^2$ : 0.977
STACK (LASSO) <sub>2</sub>	Processed data Corr (n_attributes: 4)	Alpha ( $L_1$ penalty): 0.001	$R^2$ : 0.977

TABLE 6: The different architectures of MLP with the corresponding number of hidden layers and the number of neurons in each layer.

Numerous architectures	The number of hidden layers and the number of neurons in each layer
Architecture <sub>1</sub>	$H_1 \in \mathbb{R}^8$
Architecture <sub>2</sub>	$H_1 \in \mathbb{R}^8, H_2 \in \mathbb{R}^{12}$
Architecture <sub>3</sub>	$H_1 \in \mathbb{R}^4, H_2 \in \mathbb{R}^8, H_3 \in \mathbb{R}^{12}$
Architecture <sub>4</sub>	$H_1 \in \mathbb{R}^{12}, H_2 \in \mathbb{R}^{12}, H_3 \in \mathbb{R}^{12}, H_4 \in \mathbb{R}^8$
Architecture <sub>5</sub>	$H_1 \in \mathbb{R}^{12}, H_2 \in \mathbb{R}^{12}, H_3 \in \mathbb{R}^8, H_4 \in \mathbb{R}^{12}, H_5 \in \mathbb{R}^8$
Architecture <sub>6</sub>	$H_1 \in \mathbb{R}^{12}, H_2 \in \mathbb{R}^{12}, H_3 \in \mathbb{R}^8, H_4 \in \mathbb{R}^{12}, H_5 \in \mathbb{R}^4, H_6 \in \mathbb{R}^{12}$
Architecture <sub>7</sub>	$H_1 \in \mathbb{R}^8, H_2 \in \mathbb{R}^8, H_3 \in \mathbb{R}^{12}, H_4 \in \mathbb{R}^{12}, H_5 \in \mathbb{R}^{12}, H_6 \in \mathbb{R}^{12}, H_7 \in \mathbb{R}^{12}$

formed of the models ELM, GBR, and XGBR, with SVR as the first model in level 1. For STACK<sub>2</sub>, the levels 0 and 1 are made of ELM, GBR, and XGBR, with LASSO as the level 1 component. All of the models in Table A1 have the same performance ( $R^2$ ) as the models numbered 14, 24, 33, 35, 45, 50, and 55. Model 35, on the other hand, is selected for the STACK<sub>1</sub> technique because its complexity is smaller than that of other configurations, and it has the lowest MAPE. In a similar process, the models numbered 33, 35, 50, and 56 in Table A2 exhibit the same level of performance ( $R^2$ ). For the STACK<sub>2</sub> technique, model 35 is also picked because its complexity is lower than that of other configurations, and it has the lowest MAPE of any of the models tested. The best-

estimated accuracy of STACK<sub>1</sub> is 0.977, whereas the best-estimated accuracy of MAPE is 0.445. In a similar vein, the best-estimated accuracy of STACK<sub>2</sub> is 0.977, and the best-estimated accuracy of MAPE is 0.463. According to Table 7, based on the findings of the test phase, the approaches based on ensemble learning produced results that were compatible with the objective of minimizing error.

Figure 4 illustrates the violin graph for the APE distribution of each model that was utilized to produce predictions for the test set, as shown by the APE distribution of each model. The mean APE is shown by the white dot in the center of the chart. Ensemble-based techniques, as compared to other models, significantly lower the APE to the absolute bare

TABLE 7: Comparing stacking ensemble model with the best performing ML models.

Models	$R^2$	MAE	RMSE	MAPE
SVR	0.931	0.202	0.246	0.902
ELM	0.942	0.183	0.226	0.880
MLP	0.961	0.149	0.186	0.569
RFR	0.966	0.133	0.172	0.517
GBR	0.969	0.125	0.164	0.401
XGBR	0.974	0.120	0.151	0.619
<b>STACK<sub>2</sub></b>	<b>0.977</b>	<b>0.112</b>	<b>0.143</b>	<b>0.463</b>
<b>STACK<sub>1</sub></b>	<b>0.977</b>	<b>0.112</b>	<b>0.144</b>	<b>0.445</b>

The best approaches were shown in bold type.

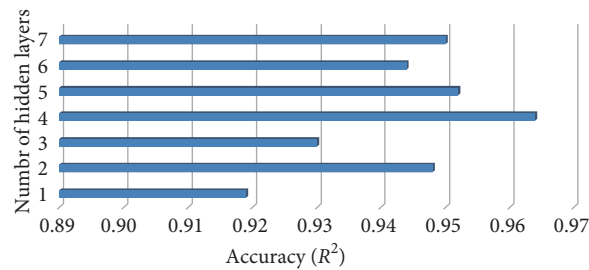


FIGURE 3: Performance of several MLP architectures with the purpose of picking the optimal one with the maximum accuracy ( $(R)^2$ ), where the best corresponding models are presented in Table 6.

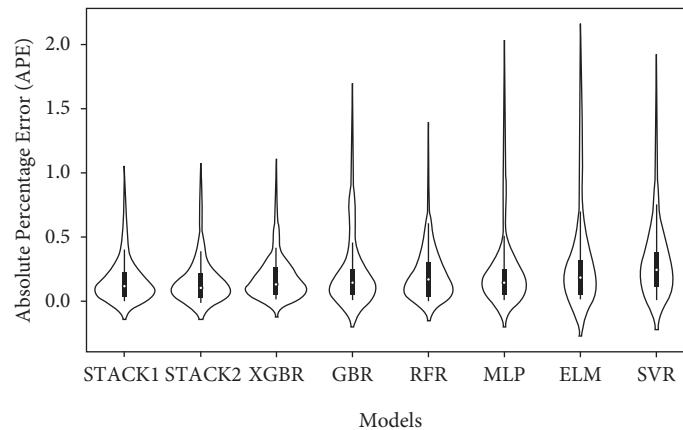


FIGURE 4: Violin plot to represent the APE of the models.

minimum. In this way, we can show that a model (for the test set) with lower metric values in Table 7 has a more stable APE and less volatility than a model with higher metric values. The Friedman test established that the APEs for the accepted models varied in the test set ( $\chi^2_7 = 72.1875$ ,  $p$ -value  $< 0.05$ ). This implies that there exist models with observed APE values that are equal to or less than those of the others. In addition, Table 8 depicts the results of the Wilcoxon signed rank test (lower tail) for measuring the APE reduction of the assessed models in the test set, in the presence of a statistically significant difference as revealed by the Friedman test ( $\chi^2_7 = 72.1875$ ,  $p$ -value  $< 0.05$ ).

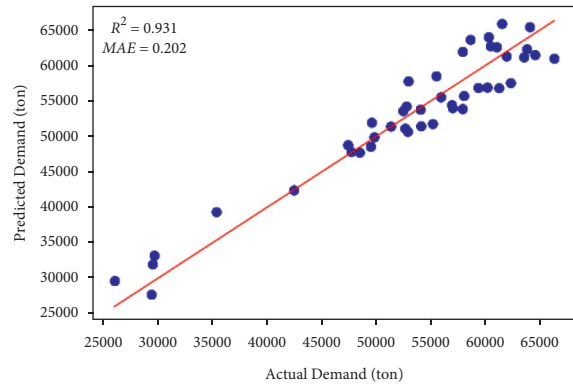
At the 5% level of significance, the APE of the STACK<sub>1</sub> model is fewer than the APEs of the RFR, MLP, ELM, and SVR models, as shown in Table 8. It is statistically equivalent when the STACK<sub>1</sub> model is compared to other models with error rates at the 5% threshold of statistical significance. In

addition, when the 5% threshold of significance is utilized to compare the models, Table 8 reveals that the APE of the STACK<sub>2</sub> model is lower than the APEs of the RFR, MLP, ELM, and SVR models. Using the % level of statistical significance, the STACK<sub>2</sub> model is compared to other models, and the errors are statistically equivalent. This highlights the advantages of the stacking ensemble models that we provide. Ensemble-based models, on average, have a lower APE than ELM and SVR. As a result, the ability of this approach to learn the data could be described using smaller estimation errors and variance between the ensemble methods than with the others, confirming the validity of this methodology. At the 5% level of significance, the APE of the STACK<sub>1</sub> model is fewer than the APEs of the RFR, MLP, ELM, and SVR models, as shown in Table 8. When the STACK<sub>1</sub> model is compared to other models, the errors are statistically equal at the 5% level. Similarly, Table 8 reveals that the APE of the STACK<sub>2</sub> model

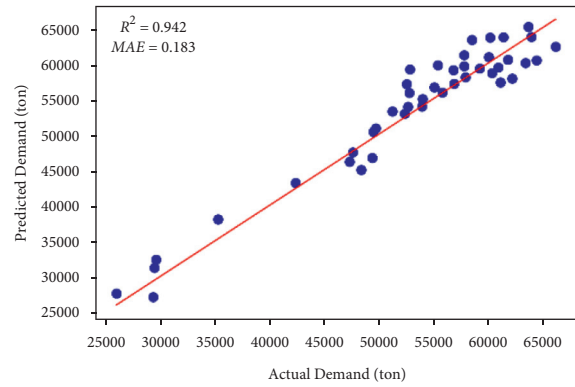


TABLE 8: Wilcoxon signed rank test statistic ( $W$ ) (lower tail) and  $p$  value for APE comparisons.

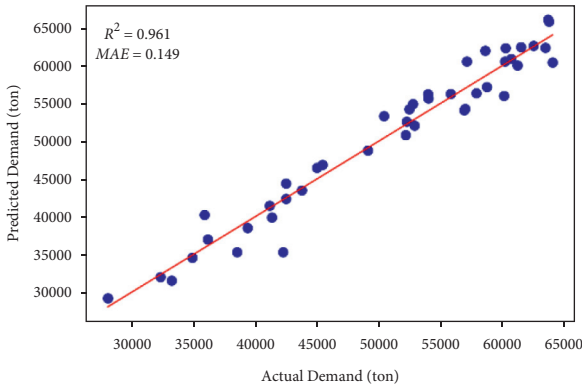
Model <sub>1</sub> vs Model <sub>2</sub>	$W$	$p$ - value	Model <sub>1</sub> vs. Model <sub>2</sub>	$W$	$p$ value	Model <sub>1</sub> vs. Model <sub>2</sub>	$W$	$p$ value
STACK <sub>1</sub> vs STACK <sub>2</sub>	3794	>0.05	STACK <sub>2</sub> vs RFR	2947	<0.05	GBR vs RFR	3740	>0.05
STACK <sub>1</sub> vs XGBR	4085	>0.05	STACK <sub>2</sub> vs MLP	2882	<0.05	GBR vs MLP	3374	>0.05
STACK <sub>1</sub> vs GBR	3383	>0.05	STACK <sub>2</sub> vs ELM	2270	<0.05	GBR vs ELM	2897	<0.05
STACK <sub>1</sub> vs RFR	3057	<0.05	STACK <sub>2</sub> vs SVR	1901	<0.05	GBR vs SVR	2392	<0.05
STACK <sub>1</sub> vs MLP	2856	<0.05	XGBR vs GBR	3814	>0.05	RFR vs MLP	3857	>0.05
STACK <sub>1</sub> vs ELM	2213	<0.05	XGBR vs RFR	3462	>0.05	RFR vs ELM	3048	<0.05
STACK <sub>1</sub> vs SVR	1843	<0.05	XGBR vs MLP	3144	<0.05	RFR vs SVR	2372	<0.05
STACK <sub>2</sub> vs XGBR	4023	>0.05	XGBR vs ELM	2720	<0.05	MLP vs ELM	3222	<0.05
STACK <sub>2</sub> vs GBR	3333	>0.05	XGBR vs SVR	1957	<0.05	MLP vs SVR	2593	<0.05
						ELM vs SVR	3657	>0.05



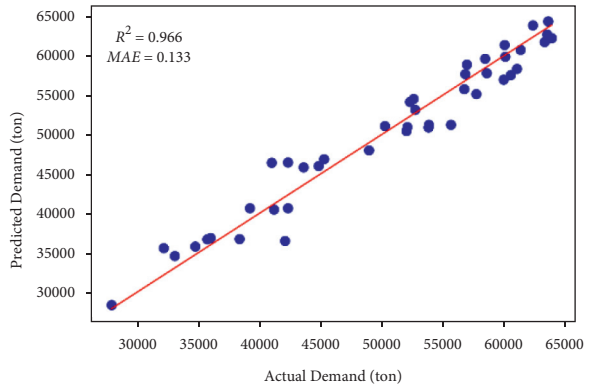
(a)



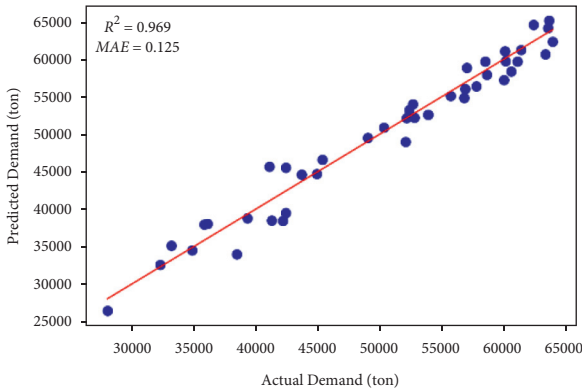
(b)



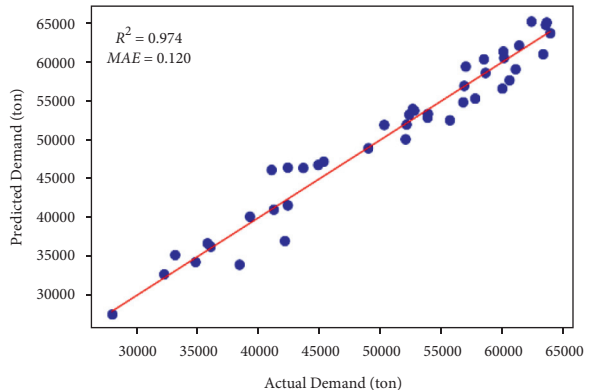
(c)



(d)



(e)



(f)

FIGURE 5: Continued.

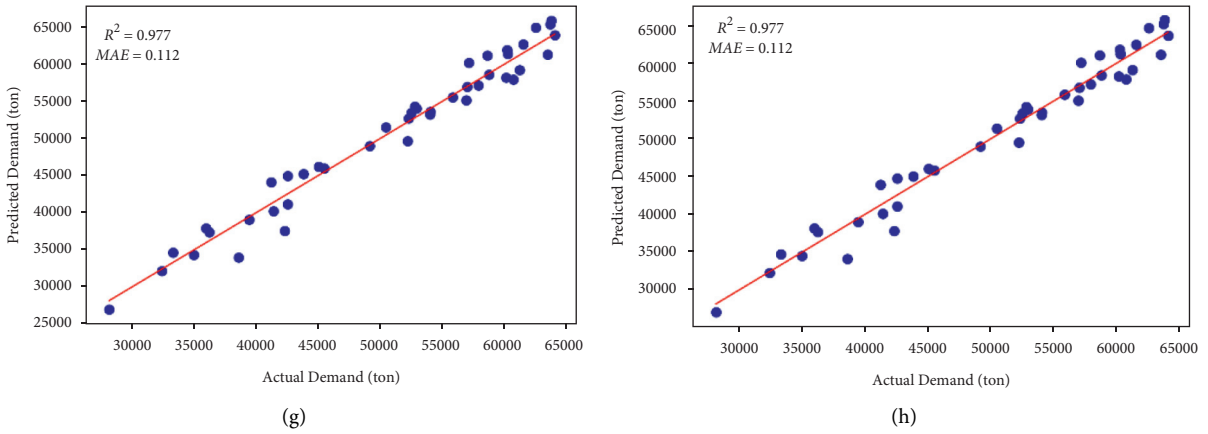


FIGURE 5: Correlation-based comparison between predicted demand and actual demand during the testing phase (a) SVR, (b) ELM, (c) MLP, (d) RFR, (e) GBR, (f) XGBR, (g)  $STACK_2$ , and (h)  $STACK_1$ .

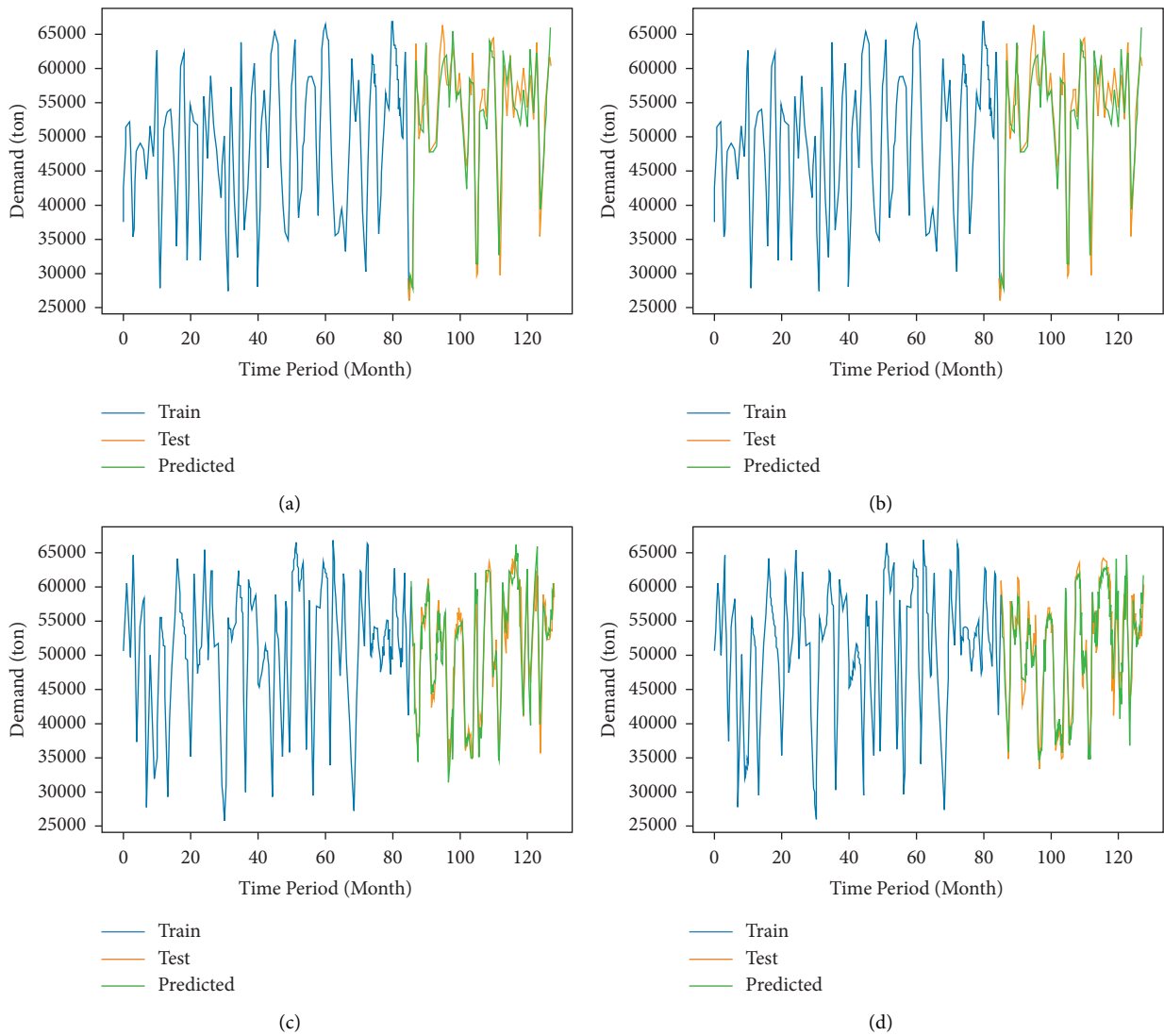


FIGURE 6: Continued.

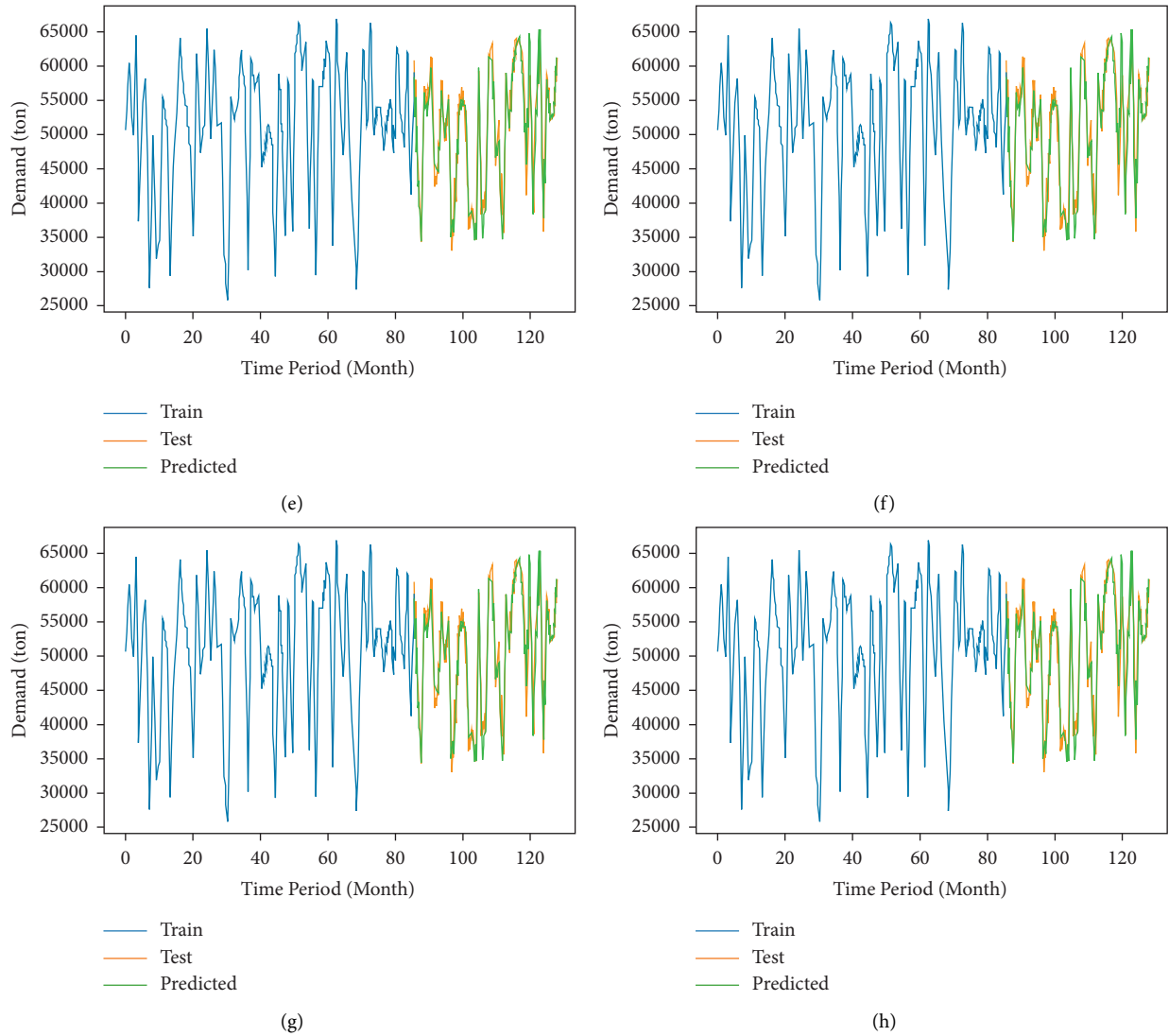


FIGURE 6: Graphical representation of actual vs predicted demand obtained by (a) SVR, (b) ELM, (c) MLP, (d) RFR, (e) GBR, (f) XGBR, (g)  $STACK_2$ , and (h)  $STACK_1$  models, respectively.

is less than that of the RFR, MLP, ELM, and SVR models at the 5% level of significance. When the  $STACK_2$  model is compared to other models, the errors are statistically equal at the 5% level. This demonstrates the advantages of the stacking ensemble models we proposed. Ensemble-based models, on average, have a lower APE than ELM and SVR. Thus, the capacity of this method to learn the data could be expressed using lower estimation errors and variance between the ensemble methods than with the others, demonstrating the correctness of this approach.

Furthermore, a relationship between the actual and predicted demand was established. Figures 5 and 6 show these techniques for better understanding. Figure 5 shows the correlation-based comparison between actual and predicted demand levels for both reference models and regression ensemble models. Figure 6 provides a pictorial view of actual vs predicted demand. Figure 6 shows that the demand pattern arbitrarily fluctuates because of the impact of the variables affecting it.

As shown in Figures 5 and 6, models that are capable of providing predictions that are consistent with the observed values are able to learn from data behavior. The improved performance attained during the training phases is maintained during the test phases, suggesting that the regression ensemble methodology is reliable in terms of achieving established predictions. This is supported by the capability of machine learning models to manage nonlinearities and model the complicated interaction between response variables and input features.

## 5. Conclusions

Precise demand forecasting significantly influences improving the performance and durability of the steel industry. This study compares the predictive performance of  $STACK$ , GBR, XGBR, and RFR regression ensembles and the MLP, ELM, and SVR reference models. In order to improve the prediction performance of regression ensemble models, data preparation

and feature selection procedures are critical. The proposed preprocessing scheme improves the raw dataset quality, where filling the missing values and data standardization are the main concerns. The Yeo–Johnson transformation is used to influence the features and response variables. While PCA and ICA solely focus on interfeature redundancy, correlation-based feature selection might improve interfeature correlation. Hyperparameters are tuned to find the optimal hyperparameter set for each ML technique using a grid search algorithm. The best-performing models are combined in STACK<sub>1</sub> to form level 0. SVR with linear kernels and LASSO regressions are adopted as meta-learners in level 1. The Friedman and Wilcoxon signed-rank tests (lower tail) are used to validate the models' APE differences. Regarding the findings, two models may be used to forecast one month as follows: STACK<sub>1</sub> (ELM + GBR + XGBR-SVR) and STACK<sub>2</sub> (ELM + GBR + XGBR-LASSO). The test set results demonstrate that ensemble approaches outperform single models, notably the STACK model, in forecasting demand in the steel industry.

Future research will (i) develop other ensemble techniques and integrate other ML regression techniques into the ensemble; (ii) include other influence variables such as occasion and political factors; (iii) collect more information, in this case only 132 months of production data are used; and (iv) extend to other industrial fields to evaluate their generality and flexibility to predict several types of demand. [36, 46, 47].

## Data Availability

The data used in this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University for funding through Vice Deanship of Scientific Research Chairs. The authors also thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support. The authors are also grateful for the support from Taif University, Taif, Saudi Arabia. The APC is funded by Taif University Researchers Supporting Project Number (TURSP-2020/331), Taif University, Taif, Saudi Arabia.

## Supplementary Materials

Figure S1: general architecture of stacking ensemble; Figure S2: architecture of MLP, with  $M$  hidden layer ( $\mathbf{H}$ ) and  $N_M$  neurons in  $\mathbf{H}_M$  layer, for forecasting the demand in the proposed framework; Figure S3: Single hidden layer architecture of ELM is used in this study; Table S1: performance metrics for evaluating STACK models that are used to anticipate steel demand one month ahead when the

metalearner is SVR with a linear kernel; Table S2: performance metrics for evaluating STACK models that are used to anticipate steel demand one month ahead when the metalearner is LASSO. (*Supplementary Materials*)

## References

- [1] W. J. Stevenson, M. Hojati, and J. Cao, *Operations Management*, McGraw-Hill Irwin, New York, NY, USA, 2007.
- [2] V. Bianco, O. Manca, S. Nardini, and A. A. Minea, "Analysis and forecasting of nonresidential electricity consumption in Romania," *Applied Energy*, vol. 87, no. 11, pp. 3584–3590, 2010.
- [3] A. Azadeh, S. F. Ghaderi, and S. Sohrabkhani, "Forecasting electrical consumption by integration of neural network, time series and anova," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1753–1761, 2007.
- [4] M. Manera and A. Marzullo, "Modelling the load curve of aggregate electricity consumption using principal components," *Environmental Modelling & Software*, vol. 20, no. 11, pp. 1389–1400, 2005.
- [5] V. Ş. Ediger and S. Akar, "Arima forecasting of primary energy demand by fuel in Turkey," *Energy Policy*, vol. 35, no. 3, pp. 1701–1708, 2007.
- [6] W.-Y. Chang, "A literature review of wind forecasting methods," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 161–168, 2014.
- [7] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," *European Journal of Operational Research*, vol. 184, no. 3, pp. 1140–1154, 2008.
- [8] M. H. D. M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing*, vol. 86, Article ID 105837, 2020.
- [9] L. Yu, W. Dai, and L. Tang, "A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting," *Engineering Applications of Artificial Intelligence*, vol. 47, pp. 110–121, 2016.
- [10] A. Altunkaynak and T. A. Nigussie, "Monthly water demand prediction using wavelet transform, first-order differencing and linear detrending techniques based on multilayer perceptron models," *Urban Water Journal*, vol. 15, no. 2, pp. 177–181, 2018.
- [11] M. Awad and R. Khanna, "Support Vector Regression," in *Efficient Learning Machines*, pp. 67–80, Apress, Berkeley, CA, USA, 2015.
- [12] N. C. D. Adhikari, R. Garg, S. Datt, L. Das, S. Deshpande, and A. Misra, "Ensemble Methodology for Demand Forecasting," in *Proceedings of the 2017 International Conference On Intelligent Sustainable Systems (ICISS)*, pp. 846–851, Palladam, India, December 2017.
- [13] R. C. Deo, M. A. Ghorbani, S. Samadianfard, T. Maraseni, M. Bilgili, and M. Biazar, "Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data," *Renewable Energy*, vol. 116, pp. 309–323, 2018.
- [14] I. S. Msiza, F. V. Nelwamondo, and T. Marwala, "Water demand forecasting using multi-layer perceptron and radial basis functions," in *Proceedings of the 2007 International Joint Conference On Neural Networks*, pp. 13–18, Orlando, FL, USA, August 2007.

- [15] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, Article ID 76516, 2020.
- [16] C. Chen, K. Li, M. Duan, and K. Li, "Extreme Learning Machine and its Applications in Big Data Processing," in *Big Data Analytics For Sensor Network Collected Intelligence*, pp. 117–150, Elsevier, 2017.
- [17] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419, 2008.
- [18] F. L. Chen and T. Y. Ou, "Sales forecasting system based on gray extreme learning machine with taguchi method in retail industry," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1336–1345, 2011.
- [19] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
- [20] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [21] F. Divina, A. Gilson, F. Gómez-Vela, M. García Torres, and J. Torres, "Stacking ensemble learning for short-term electricity consumption forecasting," *Energies*, vol. 11, no. 4, p. 949, 2018.
- [22] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–40, 2012.
- [23] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018.
- [24] M. A. Khairalla, X. Ning, N. T. Al-Jallad, and M. O. El-Faroug, "Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model," *Energies*, vol. 11, no. 6, p. 1605, 2018.
- [25] M. Tahir, R. El-Shatshat, and M. Salama, "Improved stacked ensemble based model for very short-term wind power forecasting," in *Proceedings of the 2018 Fifty Third International Universities Power Engineering Conference (UPEC)*, pp. 1–6, Glasgow, UK, September 2018.
- [26] S. Cankurt, "Tourism Demand Forecasting Using Ensembles of Regression Trees," in *Proceedings of the 2016 IEEE Eighth International Conference on Intelligent Systems (IS)*, pp. 702–708, Sofia, Bulgaria, September 2016.
- [27] K. Yang, F. Tian, L. Chen, and S. Li, "Realized volatility forecast of agricultural futures using the har models with bagging and combination approaches," *International Review of Economics & Finance*, vol. 49, pp. 276–291, 2017.
- [28] L. Wang, W. Duan, D. Qu, and S. Wang, "What matters for global food price volatility?" *Empirical Economics*, vol. 54, no. 4, pp. 1549–1572, 2018.
- [29] X. Tao, L. Chongguang, and B. Yukun, "An improved eemd-based hybrid approach for the short-term forecasting of hog price in China," *Agricultural Economics*, vol. 63, no. 3, pp. 136–148, 2017.
- [30] G. T. Ribeiro, V. C. Mariani, and L. D. S. Coelho, "Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 272–281, 2019.
- [31] R. G. da Silva, M. H. D. M. Ribeiro, S. R. Moreno, V. C. Mariani, and L. d. S. Coelho, "A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting," *Energy*, vol. 216, Article ID 119174, 2021.
- [32] L. Yu, Y. Ma, and M. Ma, "An effective rolling decomposition-ensemble model for gasoline consumption forecasting," *Energy*, vol. 222, Article ID 119869, 2021.
- [33] Z. Liu, P. Jiang, J. Wang, and L. Zhang, "Ensemble forecasting system for short-term wind speed forecasting based on optimal sub-model selection and multi-objective version of mayfly optimization algorithm," *Expert Systems with Applications*, vol. 177, Article ID 114974, 2021.
- [34] R. D. Cook and S. Weisberg, "Graphs in statistical analysis: is the medium the message?" *The American Statistician*, vol. 53, no. 1, pp. 29–37, 1999.
- [35] I. Ebtehaj and H. Bonakdari, "Assessment of evolutionary algorithms in predicting non-deposition sediment transport," *Urban Water Journal*, vol. 13, no. 5, pp. 499–510, 2016.
- [36] I. Ebtehaj, H. Bonakdari, and S. Shamshirband, "Extreme learning machine assessment for estimating sediment transport in open channels," *Engineering with Computers*, vol. 32, no. 4, pp. 691–704, 2016.
- [37] I.-K. Yeo and R. A. Johnson, "A new family of power transformations to improve I. Ebtehaj and H. Bonakdari, "Assessment of evolutionary algorithms in predicting non-deposition sediment transport," *Urban Water Journal*, vol. 13, no. 5, pp. 499–510, 2016.
- [38] I.-K. Yeo and R. A. Johnson, "A New Family of Power Transformations to Improve Normality or symmetry," *Biometrika*, vol. 87, no. 4, pp. 954–959, 2000.
- [39] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [40] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.
- [41] H. Yasin, R. E. Caraka, A. Hoyyi, and Tarno, "Prediction of crude oil prices using support vector regression (svr) with grid search-cross validation algorithm," *Global Journal of Pure and Applied Mathematics*, vol. 12, no. 4, pp. 3009–3020, 2016.
- [42] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.
- [43] P. Shine, M. D. Murphy, J. Upton, and T. Scully, "Machine-learning algorithms for predicting on-farm direct water and electricity consumption on pasture based dairy farms," *Computers and Electronics in Agriculture*, vol. 150, pp. 74–87, 2018.
- [44] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques," *Expert Systems with Applications*, vol. 112, pp. 353–371, 2018.
- [45] M.-W. Li, Y.-T. Wang, J. Geng, and W.-C. Hong, "Chaos cloud quantum bat hybrid optimization algorithm," *Nonlinear Dynamics*, vol. 103, no. 1, pp. 1167–1193, 2021.
- [46] B. E. Flores, "The utilization of the wilcoxon test to compare forecasting methods: a note," *International Journal of Forecasting*, vol. 5, no. 4, pp. 529–535, 1989.
- [47] G.-F. Fan, S. Qing, H. Wang, W.-C. Hong, and H.-J. Li, "Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting," *Energies*, vol. 6, no. 4, pp. 1887–1901, 2013.

## Research Article

# Caricature Face Photo Facial Attribute Similarity Generator

Muhammad Irfan Khan , Muhammad Kashif Hanif , and Ramzan Talib 

Department of Computer Science, Government College University, Faisalabad, Pakistan

Correspondence should be addressed to Muhammad Kashif Hanif; [mkashifhanif@gcuf.edu.pk](mailto:mkashifhanif@gcuf.edu.pk)

Received 27 May 2021; Revised 10 December 2021; Accepted 28 January 2022; Published 22 February 2022

Academic Editor: Muhammad Ahmad

Copyright © 2022 Muhammad Irfan Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Caricatures can help to understand the perception of a face. The prominent facial feature of a subject can be exaggerated, so the subject can be easily identified by humans. Recently, significant progress has been made to face detection and recognition from images. However, the matching of caricature with photographs is a difficult task. This is due to exaggerated features, representation of modalities, and different styles adopted by artists. This study proposed a cross-domain qualitative feature-based approach to match caricature with a mugshot. The proposed approach uses Haar-like features for the detection of the face and other facial attributes. A point distribution measure is used to locate the exaggerated features. Furthermore, the ratio between different facial features was computed using different vertical and horizontal distances. These ratios were used to calculate the difference vector which is used as input to different machine and deep learning models. In order to attain better performance, stratified  $k$ -fold cross-validation with hyperparameter tuning is used. Convolution neural network-based implementation outperformed the machine learning-based models.

## 1. Introduction

A picture is worth a thousand words—a truth that is derived from experience [1]. It refers that an image can convey complex and even multiple ideas in a more effective way as compared to verbal description. Cartoons and caricatures are designed by artists to represent a short story or theme within an image. A cartoon is a nonrealistic or sometimes semirealistic style to interpret or visually explain some concept. According to Kleeman, cartoon is enjoyable for someone if he understands the cartoonist's viewpoint [2]. James Gillray has been known as the founder of political cartoons [3]. Apparently, caricatures and cartoons are entirely similar in nature. Caricatures and cartoons can be sketched either by some humans or by a computer. The element of verisimilitude makes a caricature different from the cartoon. Caricatures are always of real persons, while the cartoon can be fabricated for unrealistic persons. Caricature is an image of a person (politician or someone famous in the public) made by an excessive depiction of some characteristics of the person. Caricatures, in nature, capture the physical traits of some person which are exaggerated for

humorous effects. For example, if the ears of a person are much prominent than an average one, then ears will be portrayed much larger than common.

The human brain can identify the person depicted in a caricature more quickly as the exaggerated features act as an eye-catcher and thus help to identify the person in a short time (Figure 1). According to [5], well-designed caricatures are more easy to recognize than perfect portraits.

Researchers have designed different applications and algorithms for face detection and recognition [6]. Most of these applications work for heterogeneous face recognition. However, these techniques are not appropriate for recognizing a person from caricatures.

This work tackles the problem of matching a caricature to a mugshot using different machine and deep learning algorithms. In order to match caricature to a photograph, qualitative facial attributes are defined. The qualitative facial features such as forehead height and width and nose height and width were used to encode the physical appearance of the face. These qualitative features help to determine whether a face is a mean face or digressed from a mean face. We proposed statistical learning techniques to measure the



FIGURE 1: Examples: photos and corresponding caricatures. For each subject, the first column contains a photo, and the next 3 columns contain caricatures by different caricaturists [4].

weights of these features. These features were used in machine and deep learning algorithms to detect and recognize a face.

In this work, different approaches were discussed to match caricature with a mugshot. The contributions of this study are described as follows:

- (i) Some features are more explanatory for caricatures when compared with photographs. For this reason, analytical representation of facial components of caricature and photographs was used.
- (ii) We employed Haar-like features to tackle the challenge of facial feature extraction from exaggerated artistic work in caricature.
- (iii) In order to detect facial landmarks, different horizontal and vertical distances were computed using the Euclidean distance. Moreover, we devised different thresholds for different facial attributes to detect face shape and to incorporate maximum features for better performance.
- (iv) Attribute-based proportionality technique was used to minimize the cross-domain differences.
- (v) A difference vector was computed based on qualitative features. Then different machine and deep learning-based algorithms were employed to characterize the performance.

The rest of the paper is organized into different sections. Section 2 describes the heterogeneous face recognition and role of facial attributes. Section 3 discusses related work. Section 4 describes the proposed technique used in this work. Results and evaluations of the proposed method are elaborated in Section 5. Section 6 gives the conclusion of this study.

## 2. Heterogeneous Face Recognition

Heterogeneous face recognition (HFR) is a known paradigm for face recognition and matching of different modalities. One of the major applications of HFR is sketch-based face recognition (SBFR) which deals with matching facial sketches to photographs. SBFR can be classified on the basis of how the sketches are produced. There exist four widely used categories of sketches (Figure 2):

- (i) Viewed sketches: mugshots are referred to as artists (Figure 2(a))
- (ii) Forensic sketches: hand-drawn sketches on the basis of witnesses (Figure 2(b))
- (iii) Composite sketches: made by experts by using specific software (Figure 2(c))
- (iv) Caricature sketches: sketches with exaggerated facial features (Figure 2(d))

The eigen-transformation algorithm which was proposed by Tang and Wang [11] is the foundation of synthesis-based HFR techniques. In the HFR system, the first step is to characterize face images in different modalities. The most common representations of the face are holistic, patch-based, component-based, and analytics (Figure 3).

Each face part (e.g., mouth, nose, eyes, and lips) can be represented independently using component-based representations [14] (Figure 3(a)). This can help to separately measure the features of each component in the matching process. Match score among two face images can be produced by some component-fusion scheme. The capabilities of both linear and nonlinear misalignment across modalities

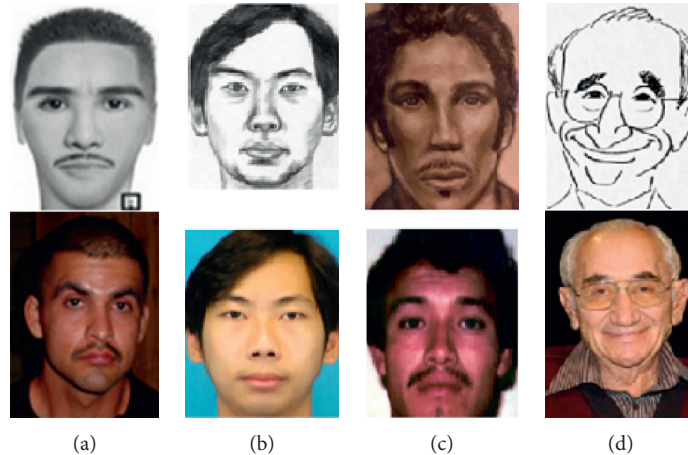


FIGURE 2: The first row is of sketches and the second row is of mugshot photos. (a) A composite sketch created using the software FACES [7, 8]. (b) Viewed sketch [9]. (c) Forensic sketch [9]. (d) Caricature [10].

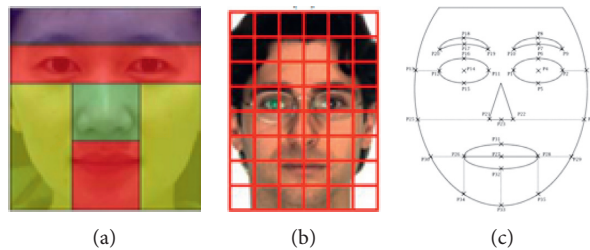


FIGURE 3: (a) Five facial components corresponding to the eye, eyebrow, cheeks, nose, and mouth. (b) Patch-based holistic representation [12]. (c) Analytical representation with fiducial points [13].

can be improved when components are detected correctly and matched [14].

Face image can also be represented by a single vector using global holistic representations [15]. The appearance of each image is encoded in patches with a feature vector for each patch in patch-based holistic representations (Figure 3(b)). There exist a variety of approaches to use these patches, for example, learning a classifier per patch [16] or concatenation as a large feature vector [11]. This approach can encode all information of available appearance. However, a high-dimensional feature vector can have sensitivity for expression and alignment variations which can result in overfitting [17].

In analytical representations, the face is modeled geometrically by detecting facial components and fiducial points on the face [18] (Figure 3(c)). This representation is relatively invariant to modality when fitting a model to a face in different modalities. However, it can require human involvement to avoid ineffectiveness in face-based model fitting. Moreover, it does not support facial expressions [18].

### 3. Related Work

This section presents the previous work related to feature-based heterogeneous face recognition for caricature and sketch. In HFR, feature-based approaches concentrate on developing a feature descriptor for the images. These feature

descriptors are unvarying to the modality but varying to the identity of the person. The popular image feature descriptors are Gabor transform, scale-invariant feature transform (SIFT) [11], local binary pattern (LBP) [19], and histogram of averaged oriented gradients (HAOG) [20]. Sketch and photographs are matched directly after they are encoded using one of these descriptors. Table 1 summarizes the recognition approaches used in feature-based HFR.

SIFT features offer a compact vector representation of the image. Klare et al. [11] proposed an invariant SIFT feature-based method to match sketches and photographs. They modeled SIFT feature vectors from the mugshot images and concatenated them jointly for sketches and mugshot images. Then, Euclidean distances are estimated between concatenated SIFT feature vectors of both sketch and mugshot images for nearest neighbor (NN) matching.

Bhatt et al. [23] proposed a method based on extended uniform circular LBP descriptors. They also employed a weight optimization technique based on genetic algorithm (GA) [26] to seek optimum weights for each facial patch. Lastly, NN matching is executed by using the chi-square distance measure.

A self-similarity descriptor was proposed by Khan et al. [21]. Features were extracted individually from local regions of photographs and sketches. A small image patch is correlated within its larger neighborhood to obtain the self-similarity features. Self-similarity reduces the modality gap



TABLE 1: Feature-based HFR matching methods.

Features	Recognition approach	Publication
SIFT	NN	[11]
Self-similarity	NN	[21]
Gabor shape	NN, chi-square	[22]
HAOG	NN, chi-square	[20]
EUCLBP	Weighted chi-square	[23]
LRBP	NN, PMK, chi-square	[24]
CITE	PCA + LDA	[16]
Geometric features	K-NN	[25]

since it remains comparatively invariant to the sketch-photograph modality variation.

Galoogahi et al. [24] proposed Local Radon Binary Pattern (LRBP) as a new face descriptor. This descriptor was helpful to directly match sketches and mugshots. In this framework, the mugshots are transfigured into Radon space. Afterward, these transformed images are encoded by LBP. As a final step, LRBP is computed by concatenating the histogram of local LBPs. A distance measurement based on pyramid match kernel (PMK) [27] is used to perform matching.

Another face descriptor was introduced by Zhang et al. [16] on the basis of coupled information-theoretic encoding. This descriptor uniquely captures discriminative local facial structures. Thus, a coupled encoding was obtained through an information-theoretic projection tree. Klare et al. [11] combined their SIFT descriptor with a common representation space strategy which was projection-based. This improvement was made on the assumption that even though direct comparison of sketches and mugshots is not possible, the distribution of interface similarities will be comparable within the mugshot and sketch domain. Consequently, for each sketch and mugshot, re-encoding is performed to obtain a vector of their Euclidean distances for the training set of sketches and mugshots accordingly. This common representation acts as the invariant to modality.

The caricature recognition task is similar to forensic sketch recognition. Besides the modality shift challenge, they hold either incomplete or imprecise information due to the judgment based on witness' personal feelings and opinions, and shortcoming of memory. A system for automatically matching sketches to photographs was proposed by Uhl et al. [28]. They geometrically standardized the sketch and mugshot to assist comparison after extracting the facial features from sketches and mugshots. As the final step, eigen analysis was performed for matching. Although their method was outmoded as compared to modern methods, they attracted researchers towards forensic sketch and caricature-based face recognition problems. A study was carried out by Klare et al. [8] in which they introduced an approach that utilized projection-based and feature-based contributions. They presented a framework named local feature-based discriminant analysis (LFDA). In LFDA, they independently represent both photos and sketches using SIFT and multiscale local binary patterns (MLBPs).

The algorithm proposed by Bhatt et al. [19] combines the projection- and feature-based contributions to enhance

recognition achievement. They encoded structural information in local facial regions by using multiscale circular Webber's local descriptor.

Generally, it is considered that the caricature recognition problem can be solved by sketch recognition methods. During earlier research, a semantic face graph was proposed to match facial mugshots that were converted into caricatures [29]. A photograph is transformed into a sketch (or vice versa) so that cross-modal differences could be eliminated. Different face recognition methods can be employed in this transformation (photograph to sketch or sketch to photograph) [30]. The cross-model differences are comparatively more than that of view-based sketches (sketches drawn from a photograph). For this purpose, the photographs and sketches are encoded into a common space MLBP [31] and SIFT [32] descriptors. Weights are allocated to facial regions by using a multiscale circular Weber's descriptor [19]. To minimize cross-domain gaps, 68 facial attributes were proposed [33]. However, automatic extraction of these facial features was left unsolved. A midlevel attribute representation was used to define a method for cross-modality matching [34].

For caricature and photograph matching, there are some commonly used datasets for benchmarking (Table 2). Each dataset consists of pairs of photographs and caricatures/sketches. However, these datasets differ based on viewed sketches, drawn by an artist, and the sketches drawn by using the software.

The existing work seems to be tedious due to the extraction of facial attribute features. The attribute features used are labeled by humans [33]. Based on these features, support vector machines (SVMs), multiple kernel learning (MKL), and logistic regression (LR) were applied to estimate the similarity level of a photograph and a caricature. A method to extract the facial attribute features from a photograph was proposed by Klare et al. [9]. However, manual work was done to identify the attribute features of caricatures. They used a genetic algorithm to find the weights of these attributes.

In most of the existing work, facial attributes were labeled manually for caricatures. In this study, qualitative feature extraction is performed automatically. We tried to minimize the cross-domain differences by using attribute-based proportionality. Furthermore, exaggerated features are difficult to handle in Haar-like features. Therefore, we employed qualitative feature matching using machine and deep learning techniques.

## 4. Methodology

This section presents the methodology adopted to match caricature with a mugshot. Face detection is the most fundamental and essential part of the face recognition process. Caricature face detection is distinct from that of face detection from mugshots. The reason is that caricatures do not preserve facial features to a massive scope. Moreover, the variations may occur in caricatures regarding the artistic style. Therefore, facial feature extraction and face detection

TABLE 2: Existing datasets for caricatures.

	Dataset information	
	Subjects	Images
Klare et al. [33]	196	392
Abaci and Akgul [9]	200	400
Mishra [4]	100	Caricature: 8,928, face: 1,000

in caricature are much complicated as that of mugshots or other photos.

Different techniques can be employed due to variations in the properties of caricatures and photographs. These variations are differences in resolution, color, and skin textures. Figure 4 shows the proposed methodology. The most basic step is to identify the qualitative features of photos and caricatures. These qualitative features are evaluated to check the matches. Higher values signify the correct matching of photographs and caricatures. Numerical encoding of features is comparatively different due to exaggerated features in caricature. For example, a wider nose is exaggerated by the artist to make it prominent. Thus, caricatures can grab and exaggerate the prominent features of a person/face while the unimportant features are discarded. The artists outline the caricature by drawing face outlines such as lips, eyes, and nose in physically exact locations with few exaggerations to make it funny but identifiable. Furthermore, some other basic attributes such as hairstyle (or some special cap/turban), beard, eye-glasses, and mole are also drawn to distinguish the caricature. For this reason, we used some qualitative facial features.

Caricature and photographs can have noise or blur images. Preprocessing is performed in order to match caricatures with photographs. This step removes noise, converts photos to grayscale, and normalizes the size with respect to eye and mouth coordinates. Then, the Haar cascade classifier was employed to detect faces from caricatures and photographs using the Haar features [35]. The reason to use this algorithm is the high detection rate and fast processing. This technique integrates different classifiers which can eliminate nonface regions within an image. Moreover, the AdaBoost algorithm is used in this approach to take important features.

First, fewer explanatory features in face appearance [36] are observed. These less explanatory features are useful for caricature recognition. These features include head shapes (i.e., oval, rectangle, circle, square, heart, or triangle) (Figure 5). In addition to head shape features, some facial attributes such as nose (Figure 6) and eye shapes (Figure 7) are used. We focused on extracting eyes, nose, lips, and chin from the image (Figure 8). Moreover, we calculated the sum of pixels in a feature window by integral image concept to avoid the summing up individually. More training data are required to extract the facial features from caricatures (Figure 9).

Facial landmarks play an important role to determine facial features. The distances between these landmarks are the key information that must be used effectively. The horizontal and vertical distances (Figures 10 and 11)

between different facial attributes are important in caricature recognition. The reason is these features are also exaggerated by an artist to create a caricature.

To determine the shape of the face, we marked horizontal distances between some facial landmarks (Figure 12) which are  $H_1, H_2, H_3, H_4, H_5$  and vertical distances from the respective horizontal line to the Y-coordinate of the deepest landmark present at the chin as  $V_1, V_2, V_3, V_4, V_5$ .

Horizontal distances  $H_1, H_2, H_3, H_4, H_5$  are computed using Euclidean distance as follows:

$$H_i = \sqrt{(x_{12-i})^2 - (x_i)^2}, \quad 1 \leq i \leq 5. \quad (1)$$

Vertical distances  $V_1, V_2, V_3, V_4, V_5$  are calculated using the following equation:

$$V_j = \sqrt{(y_6)^2 - (x_j)^2}, \quad 2 \leq j \leq 5. \quad (2)$$

Index of minimum and maximum horizontal distance  $N_{\min}$  and  $N_{\max}$  from  $H_2, H_3, H_4, H_5$  is calculated using equation (3) and equation (4), respectively.

$$N_{\min} = \arg_n \min((H_2|H_3|H_4|H_5) \leq 0.95H_1), \quad (3)$$

$$N_{\max} = \arg_n \max((H_2|H_3|H_4|H_5) \leq 0.95H_1). \quad (4)$$

$N_{\max}$  is used to find maximum horizontal distance and relevant vertical distance using equation (5) and equation (6):

$$H_{\max} = H_{N_{\max}}, \quad (5)$$

$$V_{H_{\max}} = V_{N_{\max}}. \quad (6)$$

Then two ratios  $r$  and  $t$  are computed the following equations:

$$r = \frac{V_{H_{\max}}}{H_{\max}}, \quad (7)$$

$$t = \frac{V_5}{V_2}. \quad (8)$$

Next, a threshold value of 0.35 is used to determine the category of the face. If  $t$  is less than equal to 0.5, then the face shape is square; otherwise, it is rectangle.

```

if  $r \leq 0.35$  then
  if  $t \leq 0.5$  then
    Face is Square
  else
    Face is Rectangle
  end if
end if

```

If the ratio  $r \geq 0.35$ , then  $\theta_1$  and  $\theta_2$  are computed using the following equation:

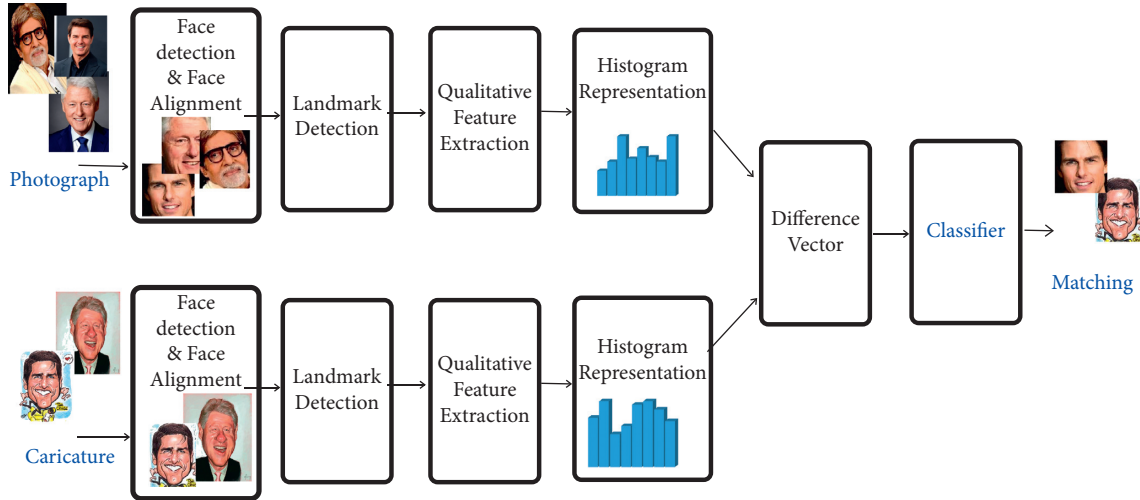


FIGURE 4: Overview of the proposed model for caricature identification.

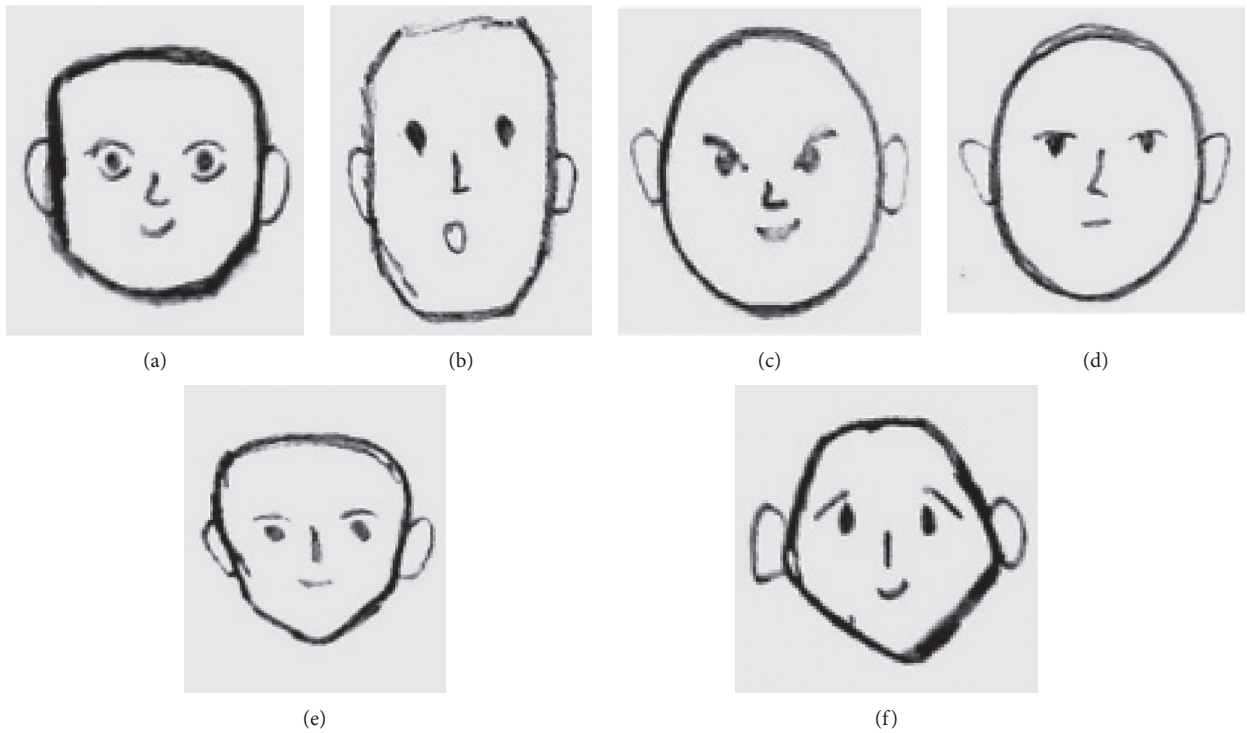


FIGURE 5: Features: generally less explanatory for photos but much explanatory for caricatures. (a) Square face, (b) rectangle face, (c) circle face, (d) oval face, (e) heart face, and (f) triangle face.

$$\theta_1 = \arctan\left(\frac{y_3 - y_4}{x_3 - x_4}\right),$$

$$\theta_2 = \arctan\left(\frac{y_4 - y_5}{x_4 - x_5}\right).$$
(9)

Then  $r$  is recalculated as

$$r = \frac{|\theta_1 - \theta_2|}{65},$$
(10)

and face shape is recognized by

if  $r \leq 0.35$  then  
 if  $t \leq 0.5$  then  
 Face is Circle  
 Else  
 Face is Oval  
 end if  
 else

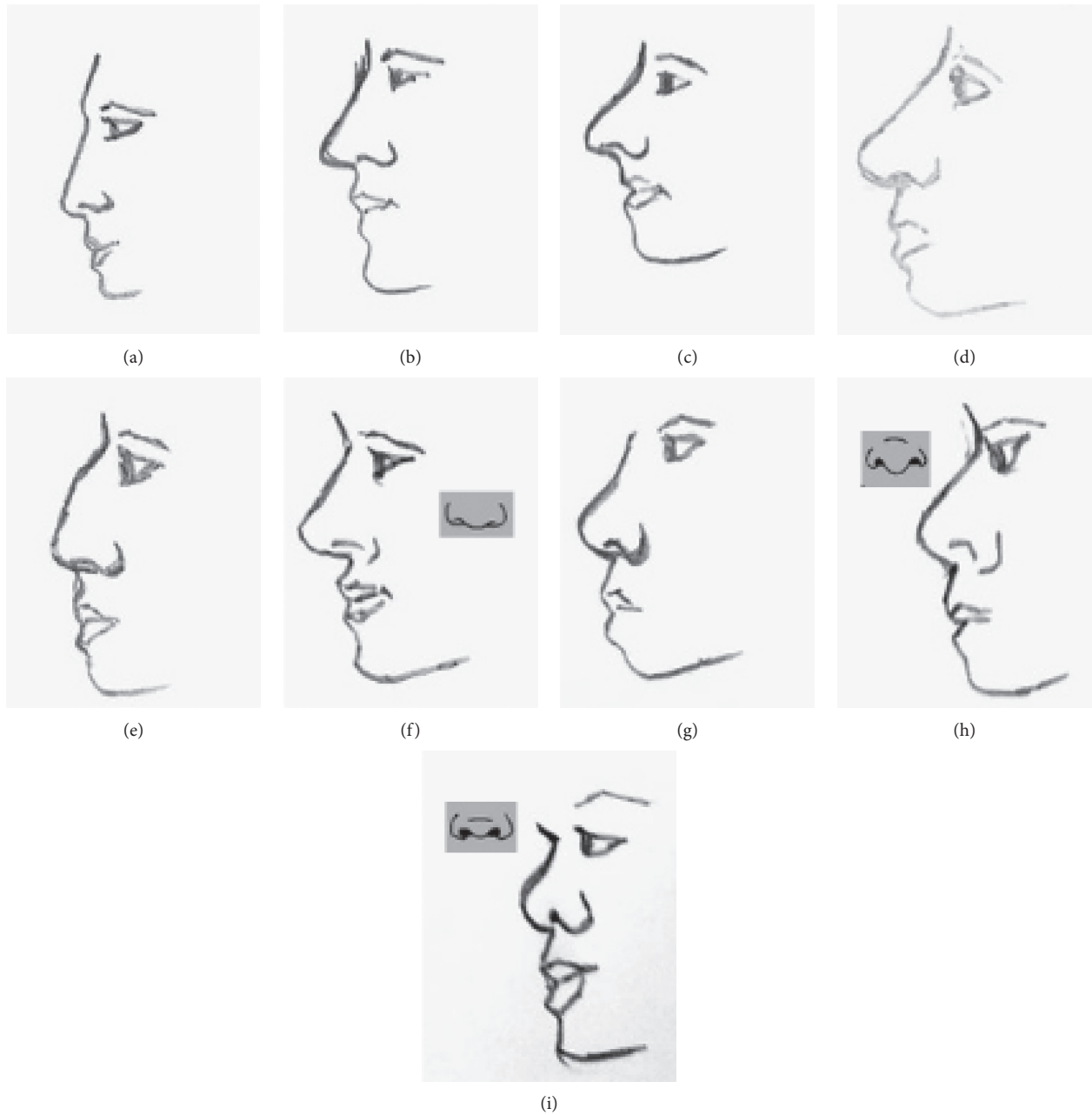


FIGURE 6: Nose shapes: (a) Grecian nose (drops straight down from the forehead), (b) Roman nose (slightly aquiline), (c) Aquiline nose (eagle-like convex), (d) droopy nose (tip very low, an effect of aging), (e) hooked nose (broken profile), (f) button nose (rounded and small, tip turns up so slightly that nostrils are not visible), (g) upturned nose (concave), (h) snub nose (aka blunt, short, and upturned, mostly found in Asians), and (i) funnel nose (African nose, nostrils pass over to the bridge).

```

if  $t \leq 0.5$  then
  Face is Heart
else
  Face is Triangle
end if
end if

```

Furthermore, a difference vector is calculated for each possible pair of caricatures and photographs in the training set. The difference vector is marked as +1 and -1 for true and

false match of caricature and photograph, respectively. Suppose  $\{(Z_i, t_i), Z_i \in \mathbb{R}, t_i \in \{-1, +1\}, i = 1, 2, \dots, n\}$  are the  $n$ -pair of difference vectors.  $t_i$  is +1 if  $Z_i$  is a difference vector for a true match of caricature and photo.  $t_i$  is set to -1 for false match. Thus, for  $n$ -subjects in the training set, positive samples for true matches are  $n$ , whereas negative samples for false matches are  $n^2 - n$ .

For improved qualitative feature matching mechanisms, machine learning approaches for the selection and weighting of feature subsets are employed. In this study, we compared the performance of LR, SVM, MKL, and convolutional

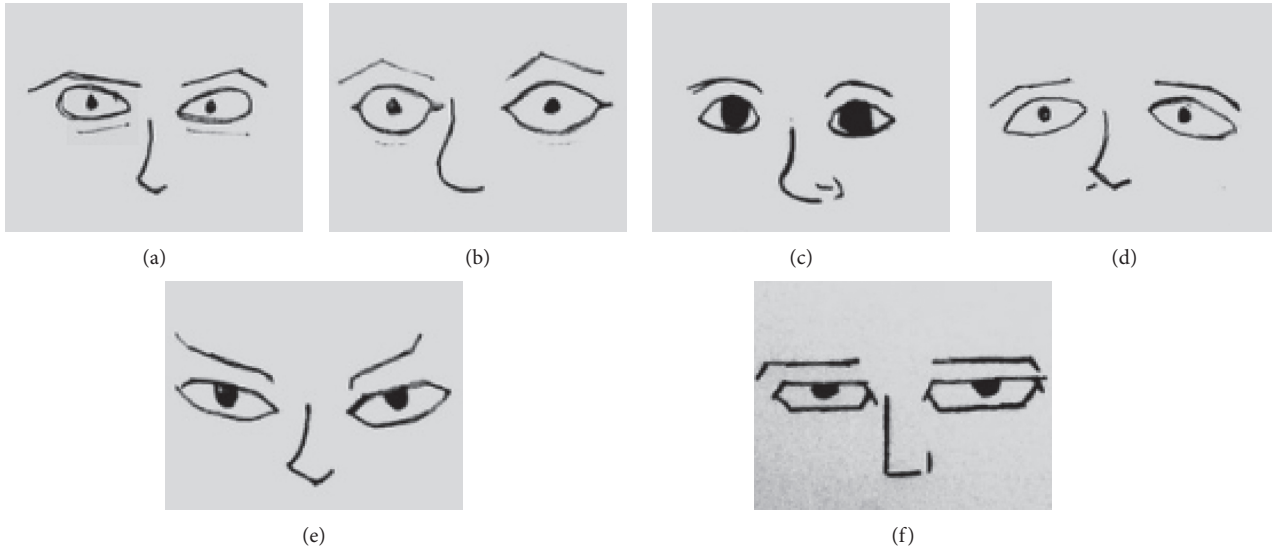


FIGURE 7: Eye shapes: (a) almond eye, (b) round-shaped eye, (c) small eye, (d) downturned eye, (e) upturned eye, and (f) deep set eye.

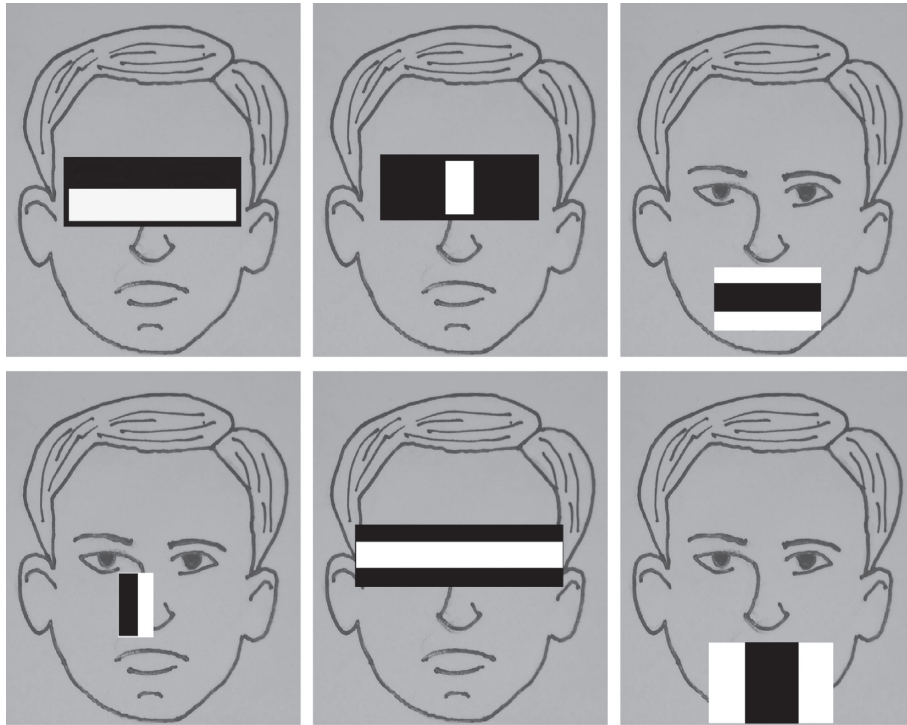


FIGURE 8: Haar features for facial attributes.

neural network (CNN) for matching. LR searches for a function that outlines the difference vectors to numerical labeling (here these are +1 or -1). A similarity score is obtained as an output. Similarity value is computed as

$$f(Z) = X\alpha - \log(\exp(Z\alpha) + 1), \quad (11)$$

where  $Z$  is the difference vector between a caricature and a photograph. The drawback of the LR method is it works only for the linear dependency of features. For this reason, MKL and SVM are employed to work with nonlinear dependencies [37].

For  $n$  training images, a set of base kernels is  $\{F_i \in \mathbb{R}^{n \times n}, i = 1, 2, \dots, 25\}$ . To combine these base kernels, the coefficient  $q = (q_1, q_2, \dots, q_r)^T \in \mathbb{R}_+^r$  is used. The kernel matrix is  $F(q) = \sum_{i=1}^r q_i F_i$ . Convex-concave optimization of the MKL dual formulation is used to get the coefficient vector  $q$  [38]. MKL is used with the nearest neighbour. This helped to obtain the weighted differences for each feature vector. SVM algorithm is used with a single kernel utilizing all feature components at the same time [39].

Moreover, this study also employs CNN for caricature and photograph matching. Input images are resized and

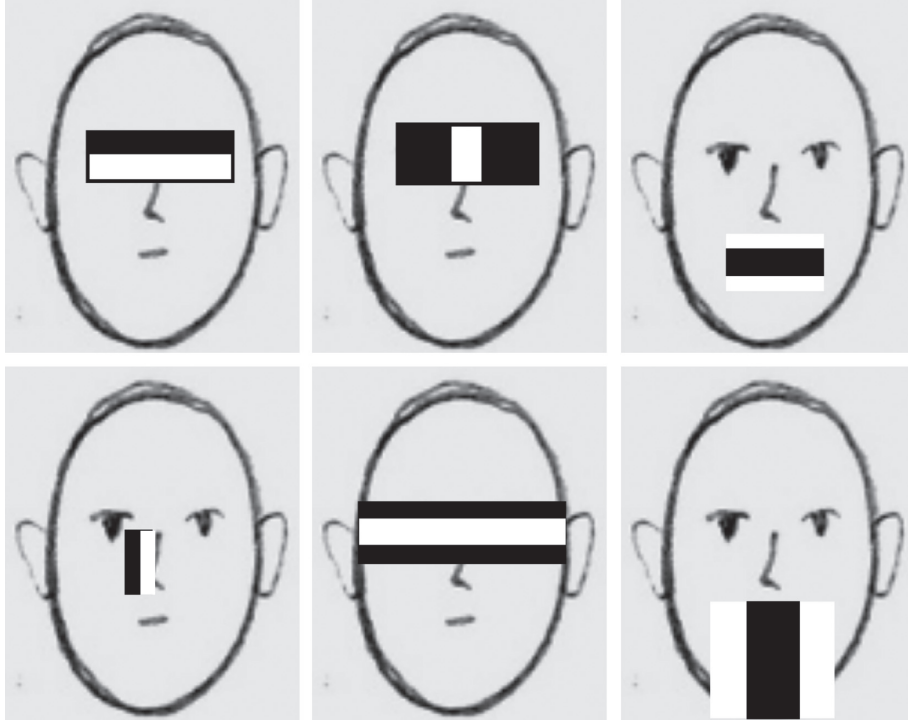


FIGURE 9: Haar features for facial attributes in caricatures.

padded to size  $224 \times 224$ . Then these images are converted to grayscale. These transformed images are used as input to CNN. The proposed CNN architecture consists of ten convolutional layers, five pooling layers, and one fully connected layer. Max operator is applied in four pooling layers and the last max-pooling layer uses average. The approximation of a large filter is obtained by applying several small filters. Also, the number of parameters is turned down by eliminating the redundancy of fully connected layers.

## 5. Results and Discussion

This section discusses the results of techniques used to match caricature with a photograph. Google Colab is used for the implementation of machine and deep learning models. Google Colab is a free online cloud-based Jupyter notebook environment having high computational power. We used the IIIT-CFW database [4]. There are 8929 cartoon faces and 1000 genuine faces of 100 renowned personalities. For training, 80% of the dataset is used and the rest of the dataset is used for testing.

We employed stratified 10-fold cross-validation and hyperparameter tuning to find the best parameters for each approach. Stratified  $k$ -fold cross-validation helps to measure the performance by splitting the dataset into  $k$  subsets. One of these subsets is taken as a testing subset and others as training subsets. This procedure is iterated  $k$  times for different subsets. Stratified  $k$  fold cross-validation divides the data such that the proportions between classes are the same in each fold [40]. The machine and deep learning models

were retrained for the best parameters obtained using hyperparameter tuning.

Figures 13 and 14 show the result of the proposed model using precision, recall, F1-scores, and normalized confusion matrix of five celebrities selected from the dataset. The proposed model has a high value of precision and recall (Figure 13). This means the proposed model has a low false-positive rate and a low false-negative rate. Diagonal values in the normalized confusion matrix also validate the performance of the proposed model (Figure 14).

Moreover, receiver operating characteristic (ROC) and cumulative match characteristic (CMC) analysis are used to evaluate the performance. The ROC analysis is made by true-positive rate (TPR) versus false-positive rate (FPR). TPR and FPR calculations are given in equations (12) and (13). The accuracy is computed using equation (14).

$$\text{TPR} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \times 100, \quad (12)$$

$$\text{FPR} = \frac{\text{true negative}}{\text{true positive} + \text{false negative}} \times 100, \quad (13)$$

$$\text{accuracy} = \frac{\text{TPR} + \text{FPR}}{2} \times 100. \quad (14)$$

The results of TPR at fixed FPR of 1% and 10% are listed in Table 3.

The CMC curve shows the caricature recognition accuracy (Figure 15). CMC evaluates the frequency that a caricature is

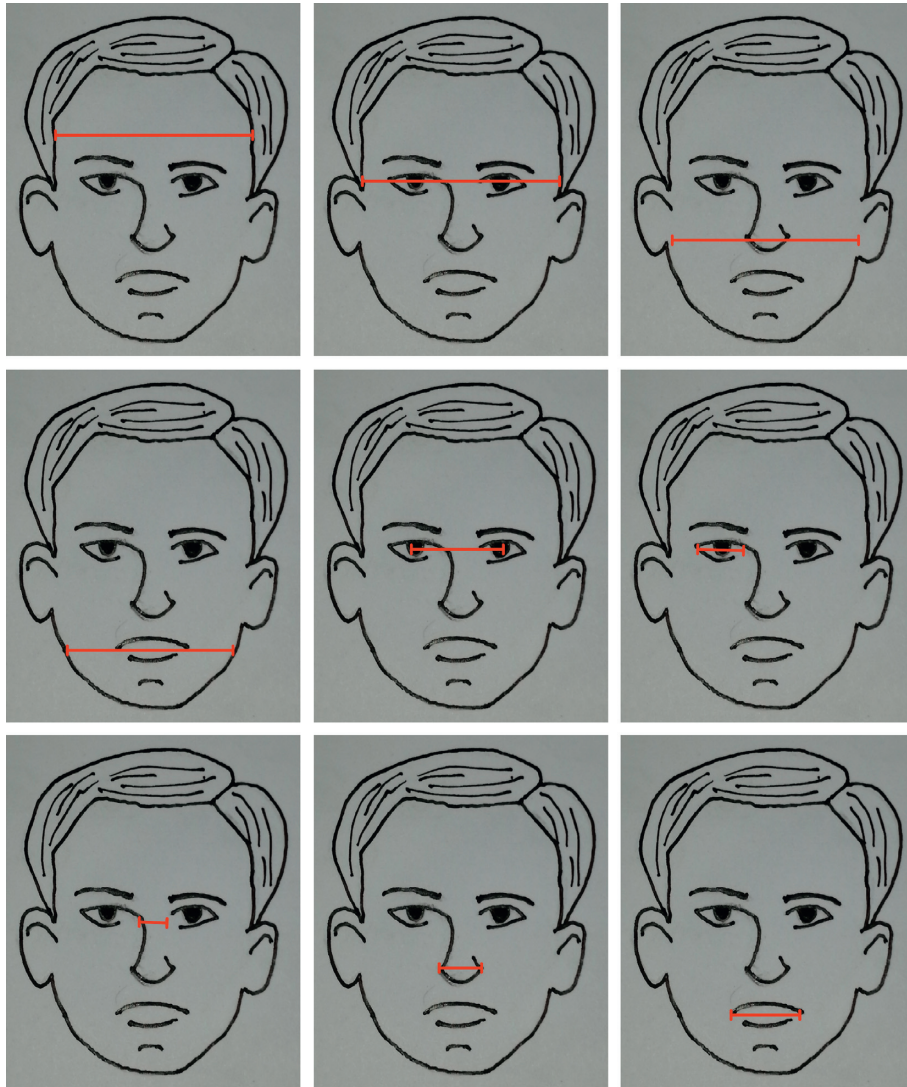


FIGURE 10: Horizontal distances between landmarks for basic facial attributes.

matched with the same identity when it is searched in a dataset of photos. The ranks are plotted at the  $X$ -axis and the frequency percentage is plotted at the  $Y$ -axis. The percentage of times that at least one out of the top  $n$  matches in the dataset is the same caricature is plotted as the frequency at Rank  $n$ . The Rank 1 and Rank 10 scores are listed in Table 4. Our proposed approach attained approximately 78% accuracy using CNN. The existing study attained 74% accuracy [33]. The reason for improved performance is using proportionality for facial features.

However, the existing study used different weights for facial features.

Testing is performed by training the algorithm only on photographs (without caricatures). It is observed that when knowledge is transferred from the mugshot domain to the caricature domain, the results are improved a certain percentage. We can infer the importance of different qualitative features with the help of vector estimation. We can find the important features using the assigned weights to these qualitative features.

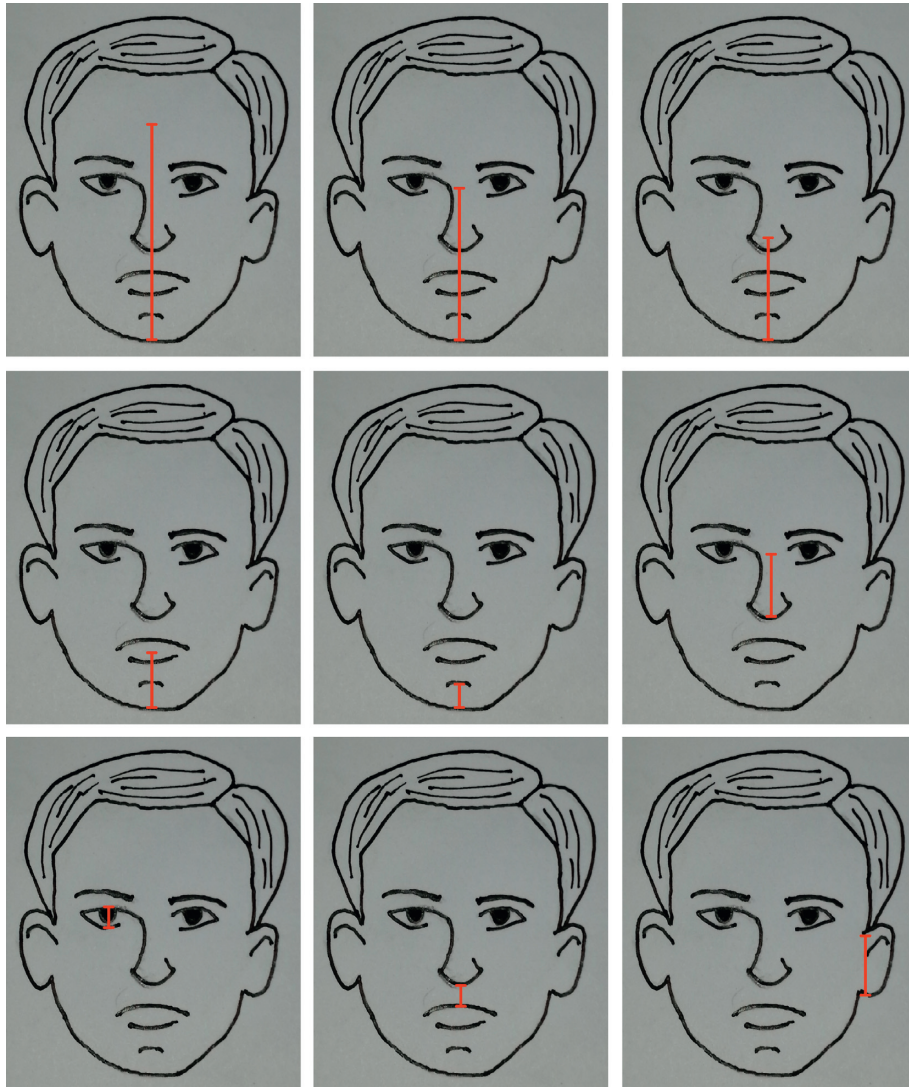


FIGURE 11: Vertical distances between landmarks for basic facial attributes.

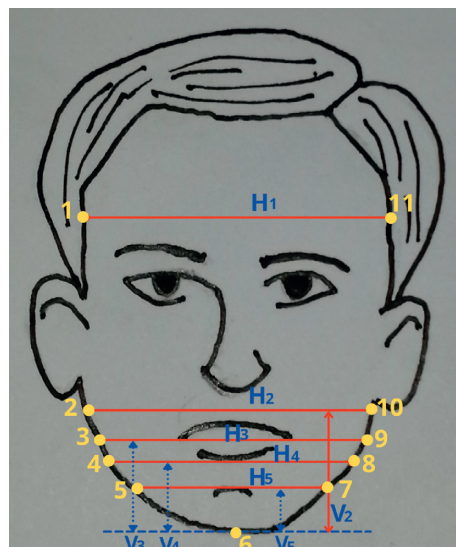


FIGURE 12: Basic distances between facial landmarks to determine the face shape.



	precision	recall	f1-score
Obama	0.91	0.95	0.93
Hillary	0.93	0.87	0.90
Amitab	0.89	0.94	0.91
Clinton	0.95	0.95	0.95
Tom Cruise	1.00	0.93	0.96
accuracy			0.93
macro avg	0.94	0.93	0.93
weighted avg	0.93	0.93	0.93

FIGURE 13: Confusion matrix of caricature recognition results.

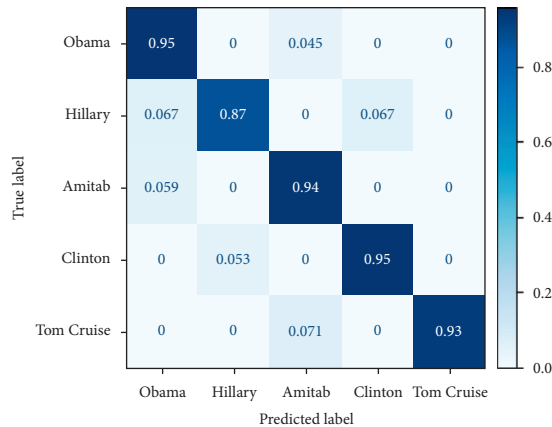


FIGURE 14: Normalized confusion matrix.

TABLE 3: Performance analysis of different classifiers.

Method	TPR at FPR = 1%	TPR at FPR = 10%
LR	13.1 ± 2.1	54.7 ± 3.3
SVM	13.9 ± 1.9	58.6 ± 4.3
MKL	9.4 ± 3.8	45.4 ± 3.9
CNN	24.6 ± 2.6	73.4 ± 4.7

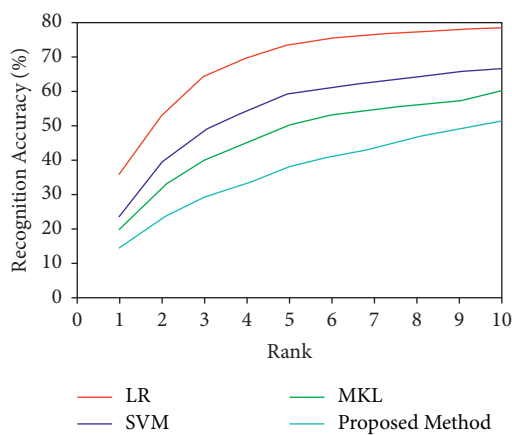


FIGURE 15: Successful caricature to photo-matching rates (Rank 1 to Rank 10).

TABLE 4: Average scores of caricature recognition accuracy (%) for different classifiers.

Method	Rank 1	Rank 10
LR	19.7 ± 2.1	59.8 ± 3.3
SVM	23.3 ± 4.6	66.4 ± 2.9
MKL	14.1 ± 2.9	51.1 ± 3.7
CNN	35.8 ± 2.7	78.3 ± 3.3

## 6. Conclusion

Caricature face detection and feature extraction required extensive effort because of misalignment problems due to exaggeration of facial features. This study proposes a cross-domain facial qualitative feature matching of caricature with photographs. Haar was employed to detect and extract the facial features from caricatures and mugshots. Euclidean distance was used to compute vertical and horizontal distances to calculate the ratio between different facial attributes. A difference vector based on qualitative features was designed which is used as input for different machine and deep learning algorithms. The proposed approach using CNN performed better when compared with other techniques and attained approximately 78% accuracy. In the future, we are interested to identify and match pose variant caricatures using different machine and deep learning techniques.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## References





- [1] W. Gardner, *Speakers give sound advice*, Syracuse Post Standard, New York, NY, USA, 1911.
- [2] G. Kleeman, "Not just for fun: using cartoons to investigate geographical issues," *New Zealand Geographer*, vol. 62, no. 2, pp. 144–151, 2006.

- [3] M. Rowson, "Satire, sewers and statesmen: why james gillray was king of the cartoon," *The Guardian*, vol. 16, 2015.
- [4] A. Mishra, S. N. Rai, A. Mishra, and C. V. Jawahar, "Iiit-cfw: a benchmark database of cartoon faces in the wild," in *Proceedings of the European Conference on Computer Vision*, pp. 35–47, Springer, Amsterdam, Netherlands, October 2016.
- [5] R. Mauro and M. Kubovy, "Caricature and face recognition," *Memory & Cognition*, vol. 20, no. 4, pp. 433–440, 1992.
- [6] P. J. Grother, P. J. Grother, P. J. Phillips, and G. W. Quinn, *Report on the evaluation of 2D still-image face recognition algorithms*, Citeseer, 2011.
- [7] S. Klum, H. Han, A. K. Jain, and B. Klare, "Sketch based face recognition: forensic vs. composite sketches," in *Proceedings of the 2013 international conference on biometrics (ICB)*, pp. 1–8, IEEE, Madrid, Spain, June 2013.
- [8] H. Han, B. F. Klare, K. Bonnen, and A. K. Jain, "Matching composite sketches to face photos: a component-based approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 191–204, 2012.
- [9] B. Klare, Z. Li, and A. K. Jain, "Matching forensic sketches to mug shot photos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 639–646, 2010.
- [10] B. Abaci and T. Akgul, "Matching caricatures to photographs," *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 295–303, 2015.
- [11] X. Tang and X. Wang, "Face sketch synthesis and recognition," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 687–694, IEEE, Nice, France, October 2003.
- [12] B. Klare and A. K. Jain, "Sketch-to-photo matching: a feature-based approach," *Biometric technology for human identification VII. International Society for Optics and Photonics*, Article ID 766702, 2010.
- [13] P. C. Yuen and C. H. Man, "Human face image searching system using sketches," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 4, pp. 493–504, 2007.
- [14] S. Liu, D. Yi, Z. Lei, and S. Z. Li, "Heterogeneous face image matching using multi-scale features," in *Proceedings of the 2012 5th IAPR International Conference on Biometrics (ICB)*, pp. 79–84, IEEE, New Delhi, India, April 2012.
- [15] D. Yi, R. Liu, R. Chu, Z. Lei, and S. Z. Li, "Face matching between near infrared and visible light images," in *Proceedings of the International Conference on Biometrics*, pp. 523–530, Springer, 2007.
- [16] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *Proceedings of the CVPR 2011*, pp. 513–520, IEEE, Colorado Springs, CO, USA, June 2011.
- [17] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: a survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.
- [18] S. Pramanik and D. Bhattacharjee, "Geometric feature based face-sketch recognition," in *Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, pp. 409–415, IEEE, Salem, India, March 2012.
- [19] H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa, "Memetically optimized mcwld for matching sketches with digital face images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1522–1535, 2012.
- [20] H. K. Galoogahi and T. Sim, "Inter-modality face sketch recognition," in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, pp. 224–229, IEEE, Melbourne, Australia, July 2012.
- [21] Z. Khan, Y. Hu, and A. Mian, "Facial self similarity for sketch to photo matching," in *Proceedings of the 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 1–7, IEEE, Fremantle, Australia, December 2012.
- [22] H. Kiani Galoogahi and T. Sim, "Face photo retrieval by sketch example," in *Proceedings of the 20th ACM international conference on multimedia*, pp. 949–952, Nara, Japan, November 2012.
- [23] H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa, "On matching sketches with digital face images," in *Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–7, IEEE, Washington, DC, USA, September 2010.
- [24] H. K. Galoogahi and T. Sim, "Face sketch recognition by local radon binary pattern: LRBP," in *Proceedings of the 2012 19th IEEE International Conference on Image Processing*, pp. 1837–1840, IEEE, Orlando, FL, USA, October 2012.
- [25] S. Pramanik and D. Bhattacharjee, "Geometric feature based face-sketch recognition," 2013, <https://arxiv.org/abs/1312.1462>.
- [26] D. E. Goldberg, *Genetic algorithms*, Pearson Education India, Delhi, India, 2006.
- [27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," vol. 2, pp. 2169–2178, in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2169–2178, IEEE, New York, NY, USA, June 2006.
- [28] R. G. Uhl and N. da Vitoria Lobo, "A framework for recognizing a facial image from a police sketch," in *Proceedings of the CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–593, IEEE, San Francisco, CA, USA, June 1996.
- [29] R.-L. Hsu and A. K. Jain, "Semantic face matching," vol. 2, pp. 145–148, in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 145–148, IEEE, Lausanne, Switzerland, August 2002.
- [30] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 1955–1967, 2008.
- [31] D. Maturana, D. Mery, and A. Soto, "Face recognition with local binary patterns, spatial pyramid histograms and naive bayes nearest neighbor classification," in *Proceedings of the 2009 International Conference of the Chilean Computer Science Society*, pp. 125–132, IEEE, Santiago, Chile, November 2009.
- [32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [33] B. F. Klare, S. S. Bucak, A. K. Jain, and T. Akgul, "Towards automated caricature recognition," in *Proceedings of the 2012 5th IAPR International Conference on Biometrics (ICB)*, pp. 139–146, IEEE, New Delhi, India, April 2012.
- [34] S. Ouyang, T. Hospedales, Y.-Z. Song, and X. Li, "Cross-modal face matching: beyond viewed sketches," in *Proceedings of the Asian Conference on Computer Vision*, pp. 210–225, Springer, Singapore, November 2014.
- [35] Y. Yanwei Pang, X. Xuelong Li, Y. Yuan Yuan, D. Dacheng Tao, and J. Jing Pan, "Fast haar transform based feature extraction for face representation and recognition," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 441–450, 2009.

- [36] B. Klare and A. K. Jain, "On a taxonomy of facial features," in *Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–8, IEEE, Washington, DC, USA, September 2010.
- [37] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, no. 6, 2008.
- [38] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.
- [39] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.
- [40] A. C. Müller and S. Guido, *Introduction to machine learning with Python: A guide for data scientists*, O'Reilly Media, Inc., Sebastopol, CA, USA, 2016.

## Research Article

# Affinity Propagation-Based Hybrid Personalized Recommender System

Iqbal Qasim <sup>1</sup>, Mujtaba Awan,<sup>2</sup> Sikandar Ali <sup>3,4</sup>, Shumaila Khan <sup>1</sup>,  
Mogeeb A. A. Mosleh <sup>5</sup>, Ahmed Alsanad <sup>6</sup>, Hizbullah Khattak <sup>7</sup>,  
and Mahmood Alam <sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Science and Technology, Bannu, Pakistan

<sup>2</sup>Department of Software Engineering, Riphah International University, Islamabad, Pakistan

<sup>3</sup>Department of Information Technology, The University of Haripur, Haripur 22621, Khyber Pakhtunkhwa, Pakistan

<sup>4</sup>Beijing Key Lab of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China

<sup>5</sup>Faculty of Engineering and Information Technology, Taiz University, Taiz 6803, Yemen

<sup>6</sup>STC's Artificial Intelligence Chair, Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

<sup>7</sup>Department of Information Technology, Hazara University Mansehra, Mansehra, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Sikandar Ali; sikandar@cup.edu.cn, Mogeeb A. A. Mosleh; mogeebmosleh@taiz.edu.ye, and Ahmed Alsanad; aasanad@ksu.edu.sa

Received 15 June 2021; Revised 21 September 2021; Accepted 7 January 2022; Published 27 January 2022

Academic Editor: Muhammad Ahmad

Copyright © 2022 Iqbal Qasim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A personalized recommender system is broadly accepted as a helpful tool to handle the information overload issue while recommending a related piece of information. This work proposes a hybrid personalized recommender system based on affinity propagation (AP), namely, APHPRS. Affinity propagation is a semisupervised machine learning algorithm used to cluster items based on similarities among them. In our approach, we first calculate the cluster quality and density and then combine their outputs to generate a new ranking score among clusters for the personalized recommendation. In the first phase, user preferences are collected and normalized as items rating matrix. This generated matrix is then clustered offline using affinity propagation and kept in a database for future recommendations. In the second phase, online recommendations are generated by applying the offline model. Negative Euclidian similarity and the quality of clusters are used together to select the best clusters for recommendations. The proposed APHPRS system alleviates problems such as sparsity and cold-start problems. The use of affinity propagation and the hybrid recommendation technique used in the proposed approach helps in improving results against sparsity. Experiments reveal that the proposed APHPRS performs better than most of the existing recommender systems.

## 1. Introduction

Recommender systems (RSs) play a vital role in the adaptive web utilizing sophisticated algorithms to reduce the ever-growing information load. Ricci et al. [1] have defined recommender systems, software tools, and techniques, which provide choices to the user for a product selection. During the past decade, the recommender system has been used in several research domains, for example, information retrieval, cognitive science, e-commerce applications, knowledge

management systems, and approximation theory [2, 3]. With the popularity of social media, such product recommender systems are getting intelligent, as they often incorporate user's comments about a particular product as input and update their underlying algorithms accordingly. However, due to the speedy growth of Internet technologies, web data is growing enormously; therefore, fetching relevant data from heterogeneous resources has become a difficult task [4]. Recommender system is an effective technology in helping users to tackle this problem by automatically recommending the

related information based on their personalized preferences, which are usually stored in a user profile.

Personalized recommendations are commonly presented as the items being rated. While carrying out this rating, the RS attempts to predict the most appropriate items on the basis of the user's preferences along with other recommendation parameters that vary from portal to portal. To predict an appropriate list of choices, RSs fetch user rating information and translate that to the user-item matrix.

The job of a current day recommender system is to predict item ratings and then make a recommendation close enough to the user preference. The ever-first system of recommendation named Tapestry [5] was developed in the mid 1990s. The most common research papers are focused on movie recommendation studies [6]. However, a great volume of the literature about the RS can be found to recommend music [7–9], best television show [9], best book to read [10], related document fetching in e-learning [11], knowledge system management [12], e-commerce [13], applications in markets [14], and web search [15].

RSs can be divided into three main categories, including collaborative filtering (CF) and content-based and hybrid methods [16]. Content-based filtering (CB) recommendations are based on the past decisions of users. The content-based filtering is efficient in finding text and items related to a certain interest by using techniques such as Boolean queries, though it has some limitations too. The content-based techniques often have the problem of lack of diversity in items recommendation [17]. In literature, content-based approaches are widely used in information retrieval [3] and information filtering [18] research. Some approaches used in content-based systems are News Weeder [19], Info Finder [20], and News Dude [21].

Collaborative filtering (CF) gathers ratings of users as opinions about certain items. The recommendation depends on the opinion of the user similarities and dissimilarities to the active user (neighbor) [22]. They aim that the collected information can be very effective for new recommendations. The collaborative filtering techniques are found to be much effective; however, it has some limitations too. One of its limitations is that the user rating analysis may be minimum; thus, the quality to recommend something would be poor. Also, collaborative filtering has common issues of sparsity, cold-start problem, and new user problem [23, 24]. Collaborative filtering algorithms are further categorized into model-based and memory-based algorithms [25]. Memory-based techniques like user-based KNN [26] use the whole user-item rating to locate  $k$  nearest neighbors for the active user and then use the ratings of these neighbors to generate recommendations. However, memory-based collaborative filtering (CF) algorithms are not scalable as the size of users grows. In contrast, model-based techniques like clustering [27] and Bayesian model [28] build a model first from the rating matrix and then use it for generating recommendations. The downside of the model-based algorithm is that the model must be rebuilt after adding a new to the rating matrix. To combine the strengths of the two stated approaches, a hybrid of model-based and memory-based techniques like Region KNN [25] and clustering-based KNN [28] can be used.

Hybrid recommender systems are the systems that combine the features of two or more recommendation techniques; for example, content-based and collaborative filtering [29] are combined for the aim of using the powers of these techniques and overcoming the weaknesses of any individual technique [30, 31]. The hybrid recommender system can be much efficient for increasing prediction accuracy.

In this paper, a hybrid approach has been proposed that combines collaborative filtering and content-based filtering for improving prediction accuracy and valid recommendations. Moreover, our proposed system uses both the model-based and memory-based features of collaborative filtering for clustering the user similarity and user-item rating matrix, respectively. The algorithm we have proposed for our system is a clustering algorithm named affinity propagation [32]. Affinity propagation algorithm produces clusters in a more efficient and accurate manner and hence provides better recommendations as compared to the baseline clustering algorithms. The main contributions of this work are as follows:

- (i) The use of affinity propagation algorithm that is effective in case of sparse datasets.
- (ii) A hybrid recommendation technique to improve the prediction and recommendation accuracy.
- (iii) While utilizing the features of CF techniques, unlike classical CF recommenders, we combine the density of a cluster with the similarity measure to select a range of clusters for the generation of recommendations. This way of picking clusters gives the active user a decent set of auxiliary recommendations.

The remainder of this paper is organized as follows. In Section 2, some of the related works are presented. Section 3 briefly describes the affinity propagation technique. Section 4 is reserved for the proposed system. In Section 5, we have explained the experimental setup and findings of our proposed APHPRS. Finally, in Section 6, the paper is concluded.

## 2. Related Work

Recommender systems focus on presenting the most relevant information to the users. In literature, different recommender systems are available that are developed for various application domains. These recommender systems are based on different filtering techniques and utilize various recommendation methods. Bilal and Hamad in [33] proposed a recommendation system for mobile application development. The system was designed to help the mobile application developers by recommending attractive designs and artifacts. They use the collaborative filtering technique by proposing a unique measurement that takes into account both similarity and trust information to demonstrate prediction accuracy. In another work, an ontology-based recommender system was proposed for advertisement on social sites [34]. They applied shared ontology for representing advertisement as well as the interest of users. The use of ontology in recommender systems is one of the new trends in recommender system research. A system for improving the recommendation accuracy of any recommender system

was proposed in [35]. They had presented an approach named Dual Training Error based Correction (DTEC) for the improvement of recommendation performance.

Clustering is one of the techniques that can be utilized for accurate prediction in a recommender system. Clustering can be defined as the group of a certain set of objects based on their characteristics, aggregated according to their similarities. Clustering algorithms have been shown to surpass similarity metrics in terms of locating users that are similar to the target user. Clustering techniques also contribute to the resolving of data sparsity and high dimensionality issues [36]. Pham et al. [37] used the hierarchical clustering algorithm [38] for clustering users based on the social information of users and then used traditional collaborative filtering for rating predictions. A movie recommender system using the performance comparison of seven different clustering algorithms was proposed in [39]. They had optimized the  $k$  value of different clustering techniques for the movie recommender system. Moreover, they had applied social network analysis for verification of the recommendation quality of their proposed system. Shindi [40] proposed the centering-bunching based clustering (CBBC) algorithm for recommender systems, that is, “hybrid personalized recommender system using centering-bunching based clustering algorithm (CBBCHPRS).” CBBC algorithm is used to cluster the rating matrix. This algorithm is better than traditional  $k$ -mean and  $k$ -medoid in which centroids are initially calculated appropriately, resulting in the suitable formation of clusters; however, the technique still needs the number of clusters to be prespecified as input. Similarity measures are then used to choose the most suitable cluster/s for the generation of rating predictions. Bedi et al. [41] used an ant colony-based clustering algorithm to cluster the user-item rating matrix, and then the most similar cluster/s to the active user is/are selected for the generation of recommendations. Shindi [31] proposed a fast  $k$ -medoid algorithm to cluster the rating matrix and select the cluster that is most related to the active user for the generation of rating predictions. In this work, we propose a machine learning algorithm known as affinity propagation [32] for our proposed hybrid (of model-based and memory-based) personalized recommender system, which can automatically find the appropriate clusters in a given data set without the need for clusters number to be prespecified. The details about affinity propagation are presented in section 3.

### 3. Affinity Propagation

Affinity propagation is a novel semisupervised machine learning algorithm that is used for clustering and is referred to as “clustering by passing messages between data points” [32]. It has been observed that affinity propagation can locate relevant information clusters with much fewer errors and in a short time compared with the other available techniques. The central point of a cluster, which is itself a data point, is called an exemplar. A common approach for clustering is learning a group of exemplars so that the total squared differences among data points with their closest exemplars can be minimized. Affinity propagation considers every data

point to be equally probable for being selected as an exemplar and take it as a node in the network of message passing system. This approach takes as input similarities between data points. Based on the input similarities, the transfer of messages between data points took place, and eventually, an adequate number of clusters with their corresponding exemplars are generated. In contrast to other techniques, affinity propagation automatically generates the number of clusters and never requires the number of clusters to be specified in advance. The similarity between two points  $s(i, k)$  indicates how similar two points are to each other by calculating a negative Euclidean distance between them:

$$s(i, k) = - \| U_i - U_k \|^2, \quad (1)$$

where  $i \neq k$ .

Affinity propagation takes a numbered value  $s(k, k)$  as a similarity between each data point  $k$ , and the points that have a higher value of  $s(k, k)$  are more chances to become an exemplar. This numbered value is known as “preferences.” The number of clusters emerges according to the input preferences but also arises during the message-exchange process. Since affinity propagation assumes every data point as an exemplar initially, the preferences are kept as a common value. This preference may be set to the minimum of all input similarities that will produce clusters less in number, or it can be set to the median of all input similarities that will produce a moderate number of clusters.

Moreover, the messages that are to be sent between the data points are categorized into two types: responsibility and availability [32]. The responsibility messages  $r(i, k)$  are those messages that are sent from data point  $i$  to the data point  $k$  representing how well it is appropriate for data point  $k$  to be the exemplar for data point  $i$  in consideration of other probable exemplars for a point  $I$  (Figure 1(a)), while the availability messages  $a(i, k)$  are the messages that are sent from data point  $k$  to the data point  $i$  representing how well it is suitable for point  $i$  to pick point  $k$  as it is exemplary considering the support from other points that point  $k$  should be an exemplar (Figure 1(b)). At any point in time, the two messages can be consolidated for determining the exemplars. Affinity adds a value  $\lambda \in [0, 1]$  called damping factor in message passing to periodic variations of in certain circumstances. The process of message passing between data points is ended when the exemplar decision does not change for a certain number of iterations, usually 10.

Figure 2 shows the message-exchange process of affinity propagation. (A) Sending responsibilities  $r(i, k)$  are the messages sent from data points to candidate exemplars to specify the favoring value of each data point for candidate exemplar compared to other exemplars. (B) Sending availabilities  $a(i, k)$  are those messages that are sent from candidate exemplars to data points indicating the degree by which candidate exemplar can be selected as a cluster center for the data point.

In the initial iteration of affinity propagation, the availabilities are initialized as zero, that is,  $a(i, k) = 0$ . The responsibilities in this step are then calculated by the rule (equation (2)). In succeeding iterations, while few data

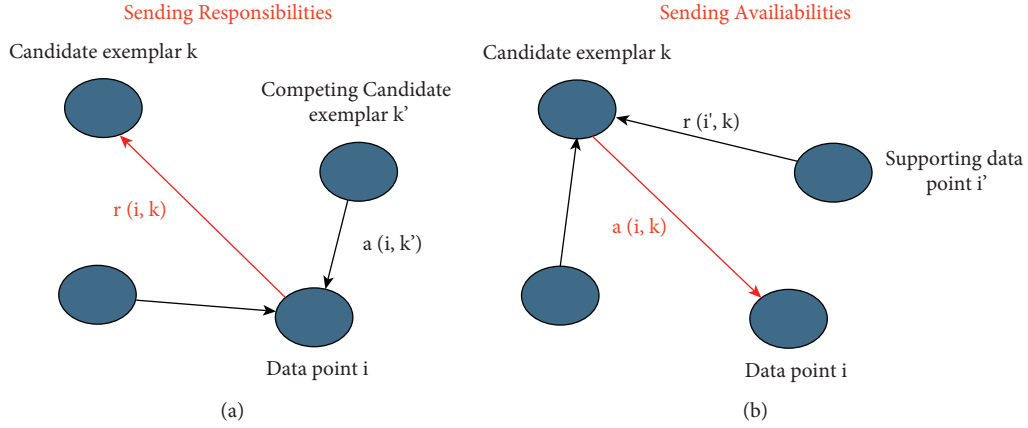


FIGURE 1: Message-exchange process of affinity propagation [32].

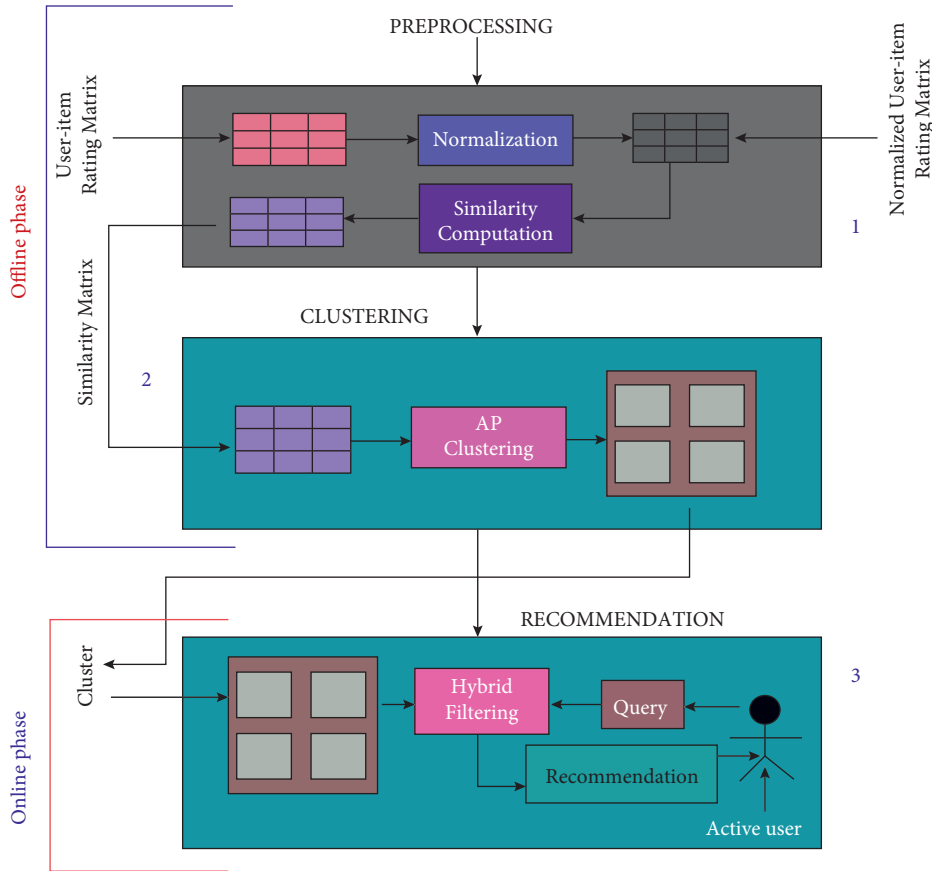


FIGURE 2: Affinity Propagation-Based Hybrid Personalized Recommender System.

points are efficiently associated with other exemplars, their availabilities decrease lower than zero as approved using the availability update rule (equation (3)). The found negative availabilities brought much decrease in the similarity value provided as input  $s(i, k')$  in responsibility update rule and detached the already supportive candidate exemplars. When the value is  $k=i$ , the value for responsibility  $r(k, k)$  is initialized as input preference that point  $k$  is selected as an exemplar,  $s(k, k)$ , minus the largest of the similarities between point  $i$  and all other candidate

exemplars. The found “self-responsibility” proves point  $k$  as an exemplar and is based on the input preference mitigated by how poorly it is to be allocated to a different exemplar [32].

$$r(i, k) = s(i, k) - \{a(i, k') + s(i, k')\}, \quad (2)$$

where  $r(i, k)$  represents the update rule for responsibility,  $s(i, k)$  represents the input similarity of data point  $i$  to its exemplar  $k$ , and  $\max_{k', s.t. k' \neq k} a(i, k') + s(i, k')$  is the maximum similarity of point  $i$  with other exemplars.

$$a(i, k) = \{0, r(k, k) + \}. \quad (3)$$

The above rule of availability is used to collect information of data points as if the candidate exemplar might be helpful in making a good exemplar. The availability  $a(i, k)$  is set to the self-responsibility  $r(k, k)$  plus the sum of the positive responsibilities candidate exemplar  $k$  receives from other points. The proposed affinity propagation algorithm has a drawback of high computational cost. Its computational complexity is of the order  $O(n^2t)$ , where “ $n$ ” represents the total number of the data points and “ $t$ ” is the number of all iterations until final clusters are made. Moreover, the proposed algorithm may fail to produce effective results when the similarity matrix is not generated well.

#### 4. The Proposed Affinity Propagation-Based Hybrid Personalized Recommender System

The proposed Affinity Propagation-Based Hybrid Personalized Recommender System (APHPRS) combines the features of content-based methods and collaborative filtering methods for handling the issues like sparsity and cold-start problems. The proposed method works in two phases (Figure 2). In the offline phase, it performs the preprocessing on data. In this phase, a rating matrix is generated and normalized, and then a pairwise similarity matrix for users is generated of the normalized matrix of user-item ratings. The similarity matrix is then loaded into an affinity propagation algorithm for clustering. Once the clusters are formed, they are kept in some database file so that the clusters can be used for generating recommendations in future.

The second phase (phase 2) is about generating recommendations for the active user. In this phase, the similarity metric and the number of user preferences are combined in a specific cluster for finding suitable clusters for the generation of recommendations. Moreover, the quality of item ratings in each cluster is also recorded. Based on this measure, suitable clusters get selected out of the list of chosen clusters for the rating predictions. After this step, the recommendations are generated using the weighted average of item ratings in the chosen clusters. Unlike classical collaborative recommenders, we combine the density of a cluster with the similarity measure to select a range of clusters for the generation of recommendations. This way of picking clusters gives the active user a decent set of auxiliary recommendations. The results are then further refined by choosing cluster/s having maximum quality ratings. The detailed procedure and working of the proposed APHPRS are described in the following by taking the example of the Jester dataset.

Figure 2 represents the proposed APHPRS system. (1) The preprocessing step: user-item rating matrix is taken from the Jester dataset and normalized. After normalization, a pairwise similarity matrix between users is calculated. (2) The clustering step: it is the step where the found similarity matrix is provided to the AP algorithm for grouping those users who have similar ratings. (3) The recommendation step: in this step, the recommendations take place for active

users. Here, the similarity between active users and clusters is computed to find the best clusters for generating recommendations. The rating quality of each item not rated by an active user is calculated in the selected clusters. To generate the recommendations, clusters are further selected based on the rating quality of an item. Note that step 1 and step 2 are the parts of the offline phase, while step 3 is the online phase of our proposed system.

*4.1. Offline Phase.* The proposed APHPRS starts with an offline phase that consists of two different steps: preprocessing and clustering. The steps in this phase are done in offline mode for faster execution of recommendations and to reduce running time. As the proposed algorithm has a little high computational cost (i.e.,  $O(n^2t)$ ), the calculation in the offline phase prevents the system from slow recommendations. In this step, the data in the form of user-item ratings is collected and clustered using the proposed algorithm. Moreover, a similarity matrix is also obtained for the user ratings, which is then used as an input for the proposed affinity propagation algorithm. Here, in this phase, we do not need to have online processing as all of its steps are done without connection with an active user.

*4.1.1. Preprocessing.* In this step, we took Jester dataset as input and normalized it for future processing of our system. We then compute the similarity between different entries available in the Jester dataset and fetch the similarity to the next step. The details of preprocessing are presented in the succeeding steps:

(1) *Jester Dataset.* The Jester dataset is an online dataset for the Web-based Joke Recommender System. In the Jester dataset, each row represents a different user, while each column (except the first column) shows the rated score given by a particular user. The first column gives a specific number of jokes rated by a particular user. The remaining columns give the ratings for different jokes. The user-item rating matrix collected from Jester data consisted of item ratings on a scale of  $-10$  to  $10$ , and in any cell, the value  $99$  represents null or no rating of the item (joke). Table 1 shows a sample of the Jester dataset that we have taken only 10 entries to keep the table readable to the reader.

(2) *Normalization.* In this step, we have taken the Jester dataset (Table 1) as input and removed the first column. Table 1 represents a sample of the Jester dataset, where the first column shows the number of jokes rated by a particular user while the rest of the columns show the rated values of each joke. Note that we have picked only 10 jokes in the below table. We then normalized the ratings to the scale from  $0$  to  $1$ , where  $0$  represents null or no rating for a particular joke (Table 2). It is noteworthy that we have kept the positive and null ratings only as the recommendations will be made on the basis of these two factors. Unlike Jester dataset, we have removed the negative ratings in the normalized rating matrix. The normalized values are calculated using



TABLE 1: A sample taken from Jester dataset before normalization.

Number of jokes rated	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
74	-7.82	8.79	-9.66	-8.16	-7.52	-8.5	-9.85	4.17	-8.98	-4.76
100	4.08	-0.29	6.36	4.37	-2.38	-9.66	-0.73	-5.34	8.88	9.22
49	99	99	99	99	9.03	9.27	9.03	9.27	99	99
48	99	8.35	99	99	1.8	8.16	-2.82	6.21	99	1.84
91	8.5	4.61	-4.17	-5.39	1.36	1.6	7.04	4.61	-0.44	5.73
100	-6.17	-3.54	0.44	-8.5	-7.09	-4.32	-8.69	-0.87	-6.65	-1.8
47	99	99	99	99	8.59	-9.85	7.72	8.79	99	99
100	6.84	3.16	9.17	-6.21	-8.16	-1.7	9.27	1.41	-5.19	-4.42
100	-3.79	-3.54	-9.42	-6.89	-8.74	-0.29	-5.29	-8.93	-7.86	-1.6
72	3.01	5.15	5.15	3.01	6.41	5.15	8.93	2.52	3.01	8.16

TABLE 2: A sample of normalized rating matrix in the scale of 0 to 1 taken from Jester dataset.

Users	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
U1	0.109	0.9395	0.017	0.092	0.124	0.075	0.0075	0.7085	0.051	0.262
U2	0.704	0.4855	0.818	0.7185	0.381	0.017	0.4635	0.233	0.944	0.961
U3	0	0	0	0	0.9515	0.9635	0.9515	0.9635	0	0
U4	0	0.9175	0	0	0.59	0.908	0.359	0.8105	0	0.592
U5	0.925	0.7305	0.2915	0.2305	0.568	0.58	0.852	0.7305	0.478	0.7865
U6	0.1915	0.323	0.522	0.075	0.1455	0.284	0.0655	0.4565	0.1675	0.41
U7	0	0	0	0	0.9295	0.0075	0.886	0.9395	0	0
U8	0.842	0.658	0.9585	0.1895	0.092	0.415	0.9635	0.5705	0.2405	0.279
U9	0.3105	0.323	0.029	0.1555	0.063	0.4855	0.2355	0.0535	0.107	0.42
U10	0.6505	0.7575	0.7575	0.6505	0.8205	0.7575	0.9465	0.626	0.6505	0.908
Active user	0.864	0.7695	0	0	0.289	0	0.9465	0.675	0	0.806

0 indicates item being not rated

$$n_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \quad (4)$$

where  $n_i$  is the value being normalized,  $x_i$  is the value before normalization,  $\min(x)$  is the lowest value, and  $\max(x)$  is the highest-rated value. The calculated values for each rating are given in Table 2. Table 2 represents the normalized form of Jester data, where the first column has been removed, and the ratings are converted on a scale between 0 and 1. Note that the negative ratings have been skipped, as the recommendation will be made on positive ratings.

(3) *Similarity Matrix.* Affinity propagation needs a pairwise matrix of similarity values to be fed as input. We have obtained the similarity among users according to their rating for items (Table 3). We have used the negative square error or Euclidean distance for calculating the similarity. The similarity matrix can be found using

$$s(i, k) = -\|U_i - U_k\|^2, \quad (5)$$

where  $i$  and  $k$  can be any two users and  $i \neq k$ .

The similarity matrix constructed for the normalized rating matrix has 10 users; hence, the similarity matrix will have  $n(n-1) = 10(9)$  entries that will lead us to create a table having 90 entries. For the sake of brevity, a short version of that matrix is reproduced here (Table 3). Here, columns 1 and 2 represent indices of the two users, while the column represents the negative Euclidean distance between the corresponding pair of users.

TABLE 3: A sample similarity matrix.

$U_i$	$U_k$	$s(i, k)$
1	2	-3.3837435
1	3	-3.4048888
1	4	-1.1776203
2	1	-3.3837435
2	3	-5.7244563
2	4	-4.0768918
3	1	-3.4048888
3	2	-5.7244563
3	4	-1.700498
4	1	-1.1776203
4	2	-4.0768918
4	3	-1.700498

4.1.2. *Clustering.* We used affinity propagation as a clustering algorithm for grouping items based on their similarity. The default damping factor of  $\lambda$  is initialized as 0.5; see details about the damping factor in Section 2. The used clustering algorithms have generated three clusters of the given similarity matrix.

We have chosen affinity propagation as in this clustering algorithm, and we do not need the number of clusters to be prespecified. Moreover, the proposed algorithm is effective in the case of sparse data, and it takes advantages of data sparsity when similarities are well-conducted [42]. The details of the found clusters are shown in Table 4.

Table 4 represents the clusters found using affinity propagation. The first column shows three different clusters,

TABLE 4: Users in each cluster with exemplar.

Cluster	User	Exemplar
1	3, 7	3
2	1, 4, 6, 9	6
3	2, 5, 8, 10	10

while the second column shows the number of users in each cluster. In the third column, the central point (exemplar) of each cluster is recorded.

**4.2. Online Phase.** In this phase, predictions and recommendations are made for the active user. The steps of this phase are done online, as the active user is considered while processing each step. A detailed description of the online phase is given in the next sections.

**4.2.1. Selecting the Best Cluster(s).** Selecting the best cluster(s) for the generation of recommendations is dependent on two aspects: (1) the number of users in a cluster and (2) the similarity of that cluster with the active user. The match score of a certain cluster is obtained by

$$\text{Match}_{\text{score}(i)} = \frac{\rho(i) \cdot sm(i)}{\sum_{i=1}^n \rho(i) \cdot sm(i)}, \quad (6)$$

where  $sm(i)$  is the similarity between the exemplar of  $i_{\text{th}}$  cluster and the active user,  $\rho(i)$  is the density of the  $i_{\text{th}}$  cluster, and  $n$  represents the complete set of clusters formed.

A similarity measure is used to find the cluster of users having preferences that match the most with the profile of the active user. There are numeral different measures used for calculating similarities like Pearson correlation, Euclidean distance measure, and vector similarity measure. We have used Euclidean distance measure to find the similarity between the profile of an active user and cluster. The Euclidean distance can be calculated using equation (7):

$$D(i) = \left\{ \sum_{j=1}^d |\text{exp}_{i,j} - u_j|^2 \right\}^{1/2}, \quad (7)$$

where  $d$  is the total number of items or attributes of a user,  $\text{exp}_{i,j}$   $i_{\text{th}}$  attribute of the exemplar of cluster  $j$ , and  $u_j$  is the  $j_{\text{th}}$  user.

Hence, the similarity of the  $i_{\text{th}}$  cluster with the active user is calculated as follows:

$$sm(i) = \frac{1}{D(i)}. \quad (8)$$

The density of the cluster can be calculated using the following equation:

$$\rho(i) = \frac{\text{population of users in cluster } i}{\text{total population of users}}. \quad (9)$$

The density of a cluster increases with the increase of users in a cluster and vice versa. Those clusters whose match score lies within the range (highest score  $-\alpha \leq \text{match\_score} \leq$  highest score) are selected for

recommendations generation. In this paper,  $\alpha = 0.2$ . By using this range, unlike collaborative filtering, not only the clusters having the highest score but also the other clusters having their scores slightly less than the maximum one are selected. The match score, density  $\rho$ , and similarity values calculated are shown in Table 5. Table 5 represents the calculated similarity  $sm(j)$  with an active user, the density  $\rho(i)$  of each cluster, and the matching score between clusters, where the first column shows the functions for different clusters, while the rest of the column shows the corresponding function measure values for each cluster.

The clusters selected are 2 and 3 because their match score lies within the range  $0.4311 - 0.2 \leq \text{match score} \leq 0.4311$ .

**4.2.2. Calculation of Rating Quality.** Rating quality any item explains how similar the ratings of different users are in any specific cluster. The rating quality can be found using

$$\text{Qty} = \frac{(\text{ub}_{\text{rating}} + \text{mean}_{\text{rating}})}{2 * \text{ub}_{\text{rating}}}, \quad (10)$$

where  $\text{ub}_{\text{rating}}$  is the maximum rate score of an item and  $\text{mean}_{\text{rating}}$  represents the mean value of item ratings in any cluster. If  $\text{ub}_{\text{rating}}$  and  $\text{mean}_{\text{rating}}$  values are equal, we get quality rating Qty as 1 that represents good quality. Hence, a greater value of Qty will lead to higher rating quality and vice versa. The calculated rating quality of different unrated jokes is shown in Table 6. Table 6 shows the rating quality of unrated jokes in different clusters, where the first column represents the active user unrated jokes, while the second and third columns show the corresponding quality ratings, that is, clusters 2 and 3, respectively.

**4.2.3. Prediction of New Item Ratings.** After finding the quality of each unrated item on the basis of rating quality, the clusters are then selected from the initially chosen clusters (Table 7). Instead of selecting the cluster with the quality for an unrated item, those clusters where the quality ‘Qty’ of each unrated item is in the range [17] are additionally retrieved from the cluster set that was initially chosen for the generation of rating predictions. In this paper,  $\alpha = 0.1$ . Table 7 shows the ratings predicted for unrated jokes. If only one cluster is selected, then the mean rating of the item in the selected cluster is calculated; otherwise, the rating is calculated as the weighted average. For jokes 4, 6, and 9, the rating is calculated as a weighted average of ratings in clusters 2 and 3, while for joke 3, the average rating is calculated from cluster 3.

Item’s rating is then predicted using

$$\text{Rating} = \frac{\sum_{i=1}^k (\text{Qty}_i * \text{mean}_{\text{rating}})}{\sum_{i=1}^k \text{qty}_i}, \quad (11)$$

where  $\text{Qty}_i$  represents the item’s quality in the chosen cluster,  $\text{mean}_{\text{rating}}$  is the rating of the item in any selected cluster, and  $k$  represents the number of clusters being selected.

TABLE 5: Cluster selecting procedure.

Function	Cluster 1	Cluster 2	Cluster 3
$sm(i)$	0.5393	0.7002	0.6544
$\rho(i)$	0.2	0.4	0.4
Match score( $i$ )	0.166	0.4311	0.4029

TABLE 6: Rating quality of unrated jokes in selected clusters.

Unrated jokes by an active user	Quality rating in cluster 2	Quality rating in cluster 3
J3	0.636	0.8685
J4	0.7592	0.8112
J6	0.7413	0.792
J9	0.7429	0.8063

TABLE 7: Ratings predicted for the active user.

Jokes	Clusters selected	Predicted ratings
J3	3	0.7064
J4	2, 3	0.27
J6	2, 3	0.4403
J9	2,3	0.34

4.2.4. *Generation of Top-N Recommendations.* Once the item ratings are predicted, the next step is to provide the active user with top-N suggestions. For example, if  $N=1$ , joke 3 will be suggested. For  $N=2$ , jokes 3 and 6 will be suggested, and so on.

## 5. Experimental Setup

5.1. *Clustering Performance Evaluation.* The performance of the proposed clustering algorithm was evaluated using the IRIS dataset that is available in the UCI repository (<https://archive.ics.uci.edu/ml/datasets/Iris>). The IRIS dataset contains a total of 150 objects divided into three main classes, where each class has 50 objects. The classes are labelled as “Setosa,” “Versicolor,” and “Virginica.” The performance of clusters was evaluated in terms of the accuracy of grouping each and every object according to the true classes. In the experiments, we compared the proposed algorithm with many baseline algorithms used for clustering. The proposed affinity propagation algorithm outperforms the existing algorithm as it clusters the object more accurately (see Table 8). The accuracy has been improved due to the message sending process of the proposed algorithm. Moreover, the proposed algorithm does not need the cluster number to be prespecified, and it automatically generates the cluster centroids (exemplars).

Table 8 shows the evaluation of the proposed clustering algorithm on the IRIS dataset. In this table, the first column represents the various classes of the IRIS dataset and the second column holds the different algorithms and their clustering results. This table clearly reveals that the affinity propagation algorithm generates the clusters with more accuracy.

TABLE 8: Clustering performance on IRIS dataset by the proposed algorithm and baseline algorithms.

IRIS classes	Clustering algorithms			
	$k$ -means	$k$ -medioids	CBBC	Affinity propagation
Setosa	50	50	50	50
Versicolor	34	41	44	47
Virginica	66	59	56	53

5.2. *Performance Evaluation of the Proposed Recommender System.* APHPRS has been implemented in MATLAB version 7.8. The experiment was conducted on the Jester dataset. We also implemented another collaborative filtering-based recommender system [43] manually for performance comparison, using  $R$ -precision and MAE metrics. To evaluate the performance, we have used two different sizes of datasets. Dataset 1 is a small subset of Jester data that consists of a user-item rating matrix having 10 users and 10 jokes (Table 2). Dataset 2 is a large subset of the Jester dataset consisting of 70,000 users and 100 jokes. The proposed APHPRS was evaluated by checking its prediction quality and recommendation quality.

Prediction quality is tested by relating the predicted user rating to the test set of actual user ratings. For the evaluation of predictive accuracy, Mean Absolute Error (MAE) is used as a universal metric [44]. Hence, we used the MAE metric for evaluating the prediction quality of the proposed APHPRS. If we represent the set of rating prediction for items as  $p_1, p_2, p_3, \dots, p_N$ , and represent the set of corresponding actual user ratings as  $q_1, q_2, q_3, \dots, q_N$ , then MAE can be computed using

$$\text{MAE} = \frac{\sum_{i=1}^N |p_i - q_i|}{N}. \quad (12)$$

The low MAE value will lead to greater quality of a recommender system, that is, smaller values of MAE, indicating that rating predictions are much similar to real ratings. For the experiment on dataset 1, we have randomly selected 7 different users from the Jester dataset and gradually increased the size of the test set from 1, 2, 3, 4, 5, 6, and 7 (Figure 3). We have compared our proposed APHPRS

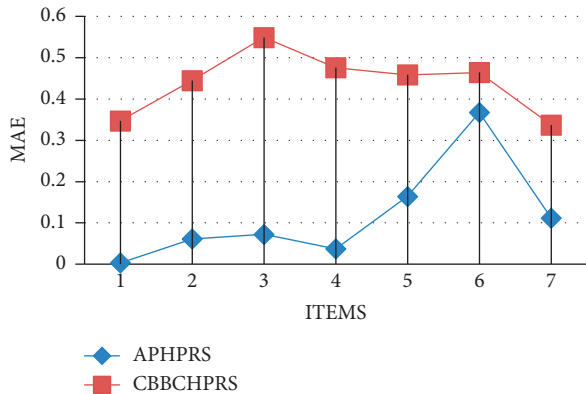


FIGURE 3: MAE for the two recommender systems for dataset 1.

system with the existing hybrid personalized recommender system using a centering-bunching-based clustering algorithm (CBBCHPRS) [40]. The proposed algorithm and the CBBCHPRS both use the same number of users of the Jester dataset.

Figure 3 represents the MAE calculated for our proposed APHPRS in comparison with CBBCHPRS. The result shows that our proposed system has a smaller MAE that will generate better recommendations as compared to other recommender systems. In the figure, the horizontal numbers show seven different items of the Jester dataset, while the vertical numbers show the corresponding MAE score of both systems for the selected items.

The MAE curves of our algorithm in Figure 3 are superior as compared to the other algorithm, which shows that our algorithm generates quality predictions for item ratings. For recommendation quality, the most widely used metrics are precision and recall. Precision is the recommended items that are relevant out of the total number of recommended items. A recall is the number of relevant recommended items out of the total number of relevant items. These two metrics are problematic in a case when at a given cutoff point, for example, precision@N, the number of relevant recommendations is less than Said et al. [43] have proposed a metric for recommendation quality, called  $R$ -precision. It has been shown empirically that  $R$ -precision reflects more accurately the quality of recommendation compared to the traditionally used metrics. Hence, we used  $R$ -precision for the evaluation of the recommendation quality of our recommender system.  $R$ -precision is computed as

$$R\text{-precision} = \frac{r}{R}, \quad (13)$$

where  $R$  shows the total number of relevant recommendations, while  $r$  is the number of retrieved recommendations that are relevant. Here,  $R$  is used as a cutoff point and varies from query to query. For the experiment on dataset 1, we randomly selected 6 different users from the Jester dataset and generated top 1, 2, 3, 4, 5, and 6 recommendations. The results obtained are shown in Figure 4.

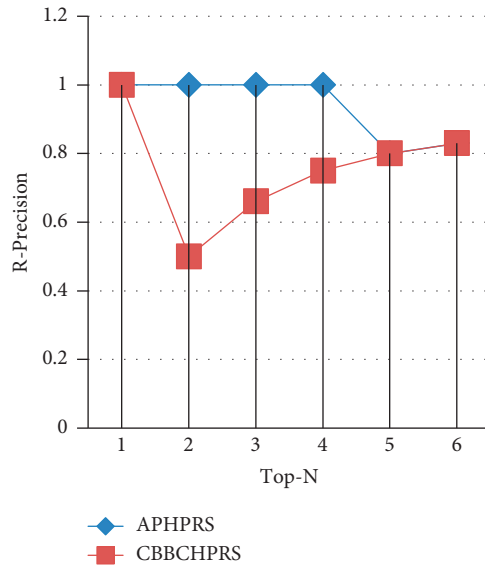


FIGURE 4:  $R$ -precision for the two recommender systems for dataset 1.

Figure 4 shows  $R$ -precision generated for the top- $N$  recommendations. The results were compared with CBBCHPRS, which shows that our proposed APHPRS has better performance in comparison with other systems.

Another experiment was performed on dataset 2, where we randomly selected 10 different users from the Jester dataset and gradually increased the size of the test set from 3, 6, 9, 12, 15, 18, 21, 24, 25, and 27. The results obtained are shown in Figure 4.

A second experiment has been conducted using a different dataset (dataset 2) taken from Jester data. For this experiment, we randomly selected 10 active users from the Jester dataset and generated the top 5, 10, and 15 recommendations. The results of the second experiment were compared with Bat Algorithm (BA) used in [45]. The results obtained show that our proposed recommender system produces high  $R$ -precision as compared to the existing BA.

Figure 5 represents the MAE calculated for our proposed APHPRS in comparison with the BA algorithm using dataset 2. The results show that our proposed system has a smaller MAE in most cases as compared to other systems. In this figure, the MAE values of the ten users were summed up for each top- $N$  recommendation.

A second experiment has been conducted using dataset 2 taken from Jester data. For this experiment, we randomly selected 10 different users from the Jester dataset and generated the top 5, 10, and 15 recommendations. The experiment was conducted to find the  $R$ -precision values. The results obtained show that our proposed recommender system produces high  $R$ -precision as compared to the existing BA (Figure 6).

It can be seen in Figure 6 that the  $R$ -precision values of our system are higher than those of the other system, which indicates the quality of our recommendations.

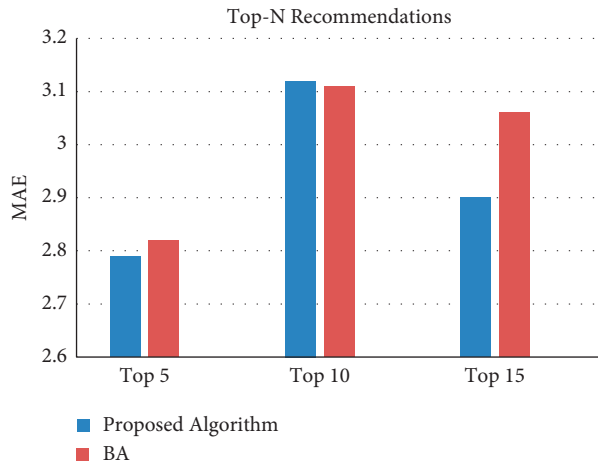


FIGURE 5: MAE for the two recommender systems for dataset 2.

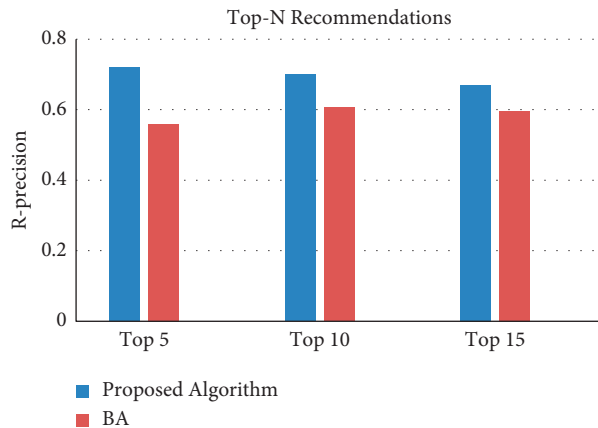


FIGURE 6: R-precision for the two recommender systems for dataset 2.

## 6. Conclusion

Recommender systems are the tools and techniques used for handling a load of information by filtering a large amount of information and then suggesting a piece of information relevant to the user. This paper proposed APHPRS, an Affinity Propagation-Based Hybrid Personalized Recommender System collecting user preferences of the matrix of items rated by different users, clustered it, and generated recommendations for the user. The clustering method used in this work outperforms other baseline methods as tested on IRIS data; however, this method fails to produce effective results when the similarity matrix is not calculated well. We have combined the features of content-based approaches and collaborative filtering approaches for making the system hybrid. The previous work done in this domain suggests that only similarity score was considered by the existing recommender systems, whereas we incorporated the cluster's quality and density and combined both with the similarity score for selecting a range of quality clusters for recommendations. This helps in discovering the additional clusters which have similarities in accordance with user preference.

Evaluation of our proposed APHPRS was made using the Jester dataset. Furthermore, the APHPRS is compared with another hybrid system CBBCHPRS. The performance of our system was evaluated in terms of prediction quality and recommendation quality. The prediction quality was calculated using the MAE score, while recommendation quality was measured in terms of  $R$ -precision. The slight decrease in the MAE score is evidence that sparsity has less impact on our proposed system. Similarly, the  $R$ -precision of our proposed system is much higher as compared to the existing systems. The increase in  $R$ -precision shows that our proposed approach has better results in recommending items.

In future work, we will use the Seed Affinity Propagation (SAP) algorithm that is an advanced version of affinity propagation for finding clusters. We will also consider ontology as a knowledge base for improving prediction accuracy in the future.

## Data Availability

The data supporting this research are from previously reported studies and datasets, which have been cited. The processed data are available at (<https://goldberg.berkeley.edu/jester-data/>).

## Conflicts of Interest

The authors declare no potential conflicts of interest.

## Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University, for funding through the Vice Deanship of Scientific Research Chairs.

## References







- [1] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender Systems Handbook*, pp. 1–34, Springer, Berlin, Germany, 2015.
- [2] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.
- [3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [4] A. S. Lampropoulos, G. A. Tsihrintzis, Tsihrintzis, and A. George, "A survey of approaches to designing recommender systems," in *Multimedia Services in Intelligent Environments*, pp. 7–30, Springer, Berlin, Germany, 2013.
- [5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [6] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, F. García-Sánchez, and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10990–11000, 2012.
- [7] H.-C. Chen and A. L. P. Chen, "A music recommendation system based on music data grouping and user interests," in *Proceedings of the Tenth International Conference on*

- Information and Knowledge Management*, pp. 231–238, Atlanta, GA, USA, October 2001.
- [8] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, “Musicbox: personalized music recommendation based on cubic analysis of social tags,” *IEEE Transactions on Audio Speech and Language Processing*, vol. 18, pp. 407–412, 2009.
  - [9] Z. Yu, X. Zhou, Y. Hao, and J. Gu, “TV program recommendation for multiple viewers based on user profile merging,” *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82, 2006.
  - [10] R. G. Crespo, O. S. Martínez, J. M. C. Lovelle, B. C. P. García-Bustelo, J. E. L. Gayo, and P. O. D. Pablos, “Recommendation system based on user interaction data applied to intelligent electronic books,” *Computers in Human Behavior*, vol. 27, no. 4, pp. 1445–1449, 2011.
  - [11] C. Porcel and E. Herrera-Viedma, “Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries,” *Knowledge-Based Systems*, vol. 23, no. 1, pp. 32–39, 2010.
  - [12] J. Bobadilla, F. Serradilla, and A. Hernando, “Collaborative filtering adapted to recommender systems of e-learning,” *Knowledge-Based Systems*, vol. 22, no. 4, pp. 261–265, 2009.
  - [13] R. Katarya and O. P. Verma, “Recommender system with grey wolf optimizer and FCM,” *Neural Computing & Applications*, vol. 30, no. 5, pp. 1679–1687, 2018.
  - [14] E. Costa-Montenegro, A. B. Barragáns-Martínez, and M. Rey-López, “Which App? A recommender system of applications in markets: implementation of the service for monitoring users’ interaction,” *Expert Systems with Applications*, vol. 39, no. 10, pp. 9367–9375, 2012.
  - [15] K. McNally, M. P. O’Mahony, M. Coyle, P. Briggs, and B. Smyth, “A case study of collaboration and reputation in social web search,” *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 1, pp. 1–29, 2011.
  - [16] U. Liji, Y. Chai, and J. Chen, “Improved personalized recommendation based on user attributes clustering and score matrix filling,” *Computer Standards & Interfaces*, vol. 57, pp. 59–67, 2018.
  - [17] R. Forsati and M. R. Meybodi, “Effective page recommendation algorithms based on distributed learning automata and weighted association rules,” *Expert Systems with Applications*, vol. 37, no. 2, pp. 1316–1330, 2010.
  - [18] N. J. Belkin and W. B. Croft, “Information filtering and information retrieval,” *Communications of the ACM*, vol. 35, no. 12, pp. 29–38, 1992.
  - [19] K. Lang, “Newsweeder: learning to filter netnews,” in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, Elsevier, Tahoe, CA, USA, July 1995.
  - [20] B. Krulwich and C. Burkey, “Learning user information interests through extraction of semantically significant phrases,” in *Proceedings of the AAAI spring Symposium on Machine Learning in Information Access*, p. 110, Palo Alto, CA, USA, March 1996.
  - [21] D. Billsus and M. J. Pazzani, “A hybrid user model for news story classification,” in *Proceedings of the CISM International Centre for Mechanical Sciences Um99 User Modeling*, pp. 99–108, Banff, Canada, June 1999.
  - [22] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix,” *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.
  - [23] L. H. Son, “Dealing with the new user cold-start problem in recommender systems: a comparative review,” *Information Systems*, vol. 58, pp. 87–104, 2016.
  - [24] F. Zhang, J. Cheng, and Z. Ma, “A survey on fuzzy ontologies for the Semantic Web,” *The Knowledge Engineering Review*, vol. 31, no. 3, pp. 278–321, 2016.
  - [25] Xi Chen, X. Liu, Z. Huang, and H. Sun, “Regionknn: a scalable hybrid collaborative filtering algorithm for personalized web service recommendation,” in *Proceedings of the 2010 IEEE International Conference on Web Services*, Miami, FL, USA, July 2010.
  - [26] M. R. McLaughlin, Herlocker, and L. Jonathan, “A collaborative filtering algorithm and evaluation metric that accurately model the user experience,” in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 329–336, Sheffield, UK, July 2004.
  - [27] L. H. Ungar, Foster, and P. Dean, “Clustering methods for collaborative filtering,” *AAAI Workshop on Recommendation Systems*, pp. 114–129, 1998.
  - [28] A. M. Rashid, S. K. Lam, G. Karypis, and J. Riedl, “ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm,” in *Proceedings of the webKDD*, Philadelphia, PA, USA, August 2006.
  - [29] X. Yang, Y. Guo, Y. Liu, and H. Steck, “A survey of collaborative filtering based social recommender systems,” *Computer Communications*, vol. 41, pp. 1–10, 2014.
  - [30] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, “A literature review and classification of recommender systems research,” *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, 2012.
  - [31] S. K. Shinde and U. V. Kulkarni, “Hybrid personalized recommender system using fast K-medoids clustering algorithm,” *Journal of Advances in Information Technology*, vol. 2, pp. 152–158, 2011.
  - [32] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
  - [33] B. Abu-Salih, H. Alsawalqah, B. Elshqairat, T. Issa, P. Wongthongtham, and K. K. Premi, “Toward a knowledge-based personalised recommender system for mobile app development,” *JUCS - Journal of Universal Computer Science*, vol. 27, no. 2, pp. 208–229, 2021.
  - [34] F. García-Sánchez, R. Colomo-Palacios, and R. Valencia-García, “A social-semantic recommender system for advertisements,” *Information Processing & Management*, vol. 57, no. 2, Article ID 102153, 2020.
  - [35] C. Panagiotakis, H. Papadakis, A. Papagrigoriou, and P. Fragopoulou, “Improving recommender systems via a dual training error based correction approach,” *Expert Systems with Applications*, vol. 183, Article ID 115386, 2021.
  - [36] M. Ramezani, P. Moradi, and F. Akhlaghian, “A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains,” *Physica A: Statistical Mechanics and Its Applications*, vol. 408, pp. 72–84, 2014.
  - [37] M. C. Pham, Y. Cao, R. Klamma, and M. Jarke, “A clustering approach for collaborative filtering recommendation using social network analysis,” *Journal of Universal Computer Science*, vol. 17, pp. 583–604, 2011.
  - [38] A. Clauset, M. E. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, Article ID 066111, 2004.
  - [39] L. Debby, S. Jenq, and P. Seda, “Design of an unsupervised machine learning-based movie recommender system,” *Symmetry*, vol. 12, no. 2, p. 185, 2020.

- [40] S. K. Shinde and U. Kulkarni, "Hybrid personalized recommender system using centering-bunching based clustering algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1381–1387, 2012.
- [41] P. Bedi, R. Sharma, and H. Kaur, "Recommender system based on collaborative behavior of ants," *Journal of Artificial Intelligence*, vol. 2, no. 2, pp. 40–55, 2009.
- [42] D. Dueck, *Affinity Propagation: Clustering Data by Passing Messages*, Citeseer, Princeton, New Jersey, USA, 2009.
- [43] M. K. Najafabadi, M. N. R. Mahrin, S. Chuprat, and H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Computers in Human Behavior*, vol. 67, pp. 113–128, 2017.
- [44] A. H. Nabizadeh, A. Jorge, and J. P. Leal, "Long term goal oriented recommender systems," in *Proceedings of the WEBIST*, Lisbon, Portugal, May 2015.
- [45] S. Yadav, Vikesh, Shreyam, and S. Nagpal, "An improved collaborative filtering based recommender system using bat algorithm," *Procedia Computer Science*, vol. 132, pp. 1795–1803, 2018.

## Research Article

# A Deep Neural Network Model for the Detection and Classification of Emotions from Textual Content

Muhammad Zubair Asghar <sup>1,2</sup>, Adidah Lajis <sup>1</sup>, Muhammad Mansoor Alam <sup>1,3</sup>,  
Mohd Khairil Rahmat <sup>1</sup>, Haidawati Mohamad Nasir <sup>1</sup>, Hussain Ahmad,<sup>2</sup>  
Mabrook S. Al-Rakhami <sup>4</sup>, Atif Al-Amri,<sup>4,5</sup> and Fahad R. Albogamy<sup>6</sup>

<sup>1</sup>Center for Research & Innovation, CoRI, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia

<sup>2</sup>Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan

<sup>3</sup>Faculty of Computing, Riphah International University, Islamabad, Pakistan

<sup>4</sup>Research Chair of Pervasive and Mobile Computing, Information Systems Department,  
College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

<sup>5</sup>Software Engineering Department, College of Computer and Information Sciences, King Saud University,  
Riyadh 11543, Saudi Arabia

<sup>6</sup>Computer Sciences Program, Turabah University College, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

Correspondence should be addressed to Mabrook S. Al-Rakhami; malrakhami@ksu.edu.sa

Received 28 May 2021; Accepted 27 December 2021; Published 15 January 2022

Academic Editor: Muhammad Ahmad

Copyright © 2022 Muhammad Zubair Asghar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Emotion-based sentimental analysis has recently received a lot of interest, with an emphasis on automated identification of user behavior, such as emotional expressions, based on online social media texts. However, the majority of the prior attempts are based on traditional procedures that are insufficient to provide promising outcomes. In this study, we categorize emotional sentiments by recognizing them in the text. For that purpose, we present a deep learning model, bidirectional long-term short-term memory (BiLSMT), for emotion recognition that takes into account five main emotions (Joy, Sadness, Fear, Shame, Guilt). We use our experimental assessments on the emotion dataset to accomplish the emotion categorization job. The datasets were evaluated and the findings revealed that, when compared to state-of-the-art methodologies, the proposed model can successfully categorize user emotions into several classifications. Finally, we assess the efficacy of our strategy using statistical analysis. This research's findings help firms to apply best practices in the selection, management, and optimization of policies, services, and product information.

## 1. Introduction

For a long time, computer-based emotion detection and classification has been a hot topic of research. Emotion detection can be done using a variety of mediums, including text, photographs, video, and audio [1, 2]. In recent years, social media sites such as Twitter, Facebook, and Instagram have undergone an unexpected global expansion. Twitter, for example, had over 200 million monthly active users by the fourth quarter of 2021 [3]. As long as analytical researchers working on social networking material can address the particular obstacles presented by such content, advances

in computational linguistics and text analytics allow researchers to extract and evaluate textual emotions provided by users on social media via big data sources.

Computational linguistic experts have performed many studies to detect and identify emotions at various levels, including words, expressions, sentences, and analysis [4–7]. Many studies, on the other hand, focus on emotion-related bearing terms, with little attention paid to textual clues to emotions, which, if included, may improve the output of cognitive-based sentiment classification for social media data. Hence, the research and development of text-based emotion classification systems are motivated by studies of emotions expressed in text.



The proposed emotion-based sentiment analysis system for social media is based on previous work on emotion classification in the social media paradigm [5, 8]. Previous studies lacked the capability to represent emotion signals using advanced feature representations and do not exploit the capability of individual deep learning models for emotion classes. However, we present a social media-based cognitive-based sentiment analysis framework that focuses on detecting and classifying textual emotions using an advanced feature representation scheme and a fusion of deep learning models.

*1.1. Research Motivation.* Over the last few years, emotion-based SA applications have become increasingly popular on the Internet for gauging the opinions, emotions, and sentiments of individuals on different issues and policies. However, it is often difficult to analyze text with existing emotion detection methods to detect emotions from social media content. Therefore, it is essential to extract and analyze social media content to automatically classify emotions.

The difference between social media and traditional blogs is that the former incorporates complex textual material. When opposed to text information alone, such material contains text and emotion cues, which are more suitable for expressing and conveying people's subtle thoughts, emotions, and personal characteristics [1]. However, emotion-based sentiment analysis, which is based on the identification of emotional cues, is still in its early stages.

There has been extensive work carried out in the area of text-based analysis of feeling [8], construction of lexicons [4], cognition and analysis of aspects of feeling [6], and analysis of visual feelings [1, 9]. However, further research is required in the area of cognitive-based social media analysis, with a focus on extracting and categorizing emotions from social media content.

*1.2. Problem Statement.* In this work, we address the problem of emotion classification of English text using a deep learning technique. Considering a set of reviews =  $\{R_1, R_2, R_3, \dots, R_n\}$  given as input, the aim is to develop a classifier for assigning an emotion label  $E_i \in \{J-S, F-G, F-S\}$  for the review  $r_i$ , where J-S, F-G, and F-S show different emotion labels. It is our goal in this study to develop a powerful deep learning-based emotion recognition system that is capable of accurately classifying provided text reviews into the required emotional category.

*1.3. Research Questions.* RQ1: How can textual content be identified and classified into different emotion groups using BiLSTM, a deep learning technique?

RQ2: What is the proposed technique's efficiency in comparison with other machine learning and deep learning techniques?

RQ3: How can the efficacy of the proposed technique for emotion classification be estimated in comparison to baseline studies?

*1.4. Proposed Contributions.* The following are the main contributions of the research work:

- (1) To efficiently capture the semantics of words, an advanced function representation scheme called word embedding is used.
- (2) For the emotion classification challenge, we present a deep neural network called Bi-LSTM (Bi-LSTM), which learns contextual information in both directions, forward and backward, using forward and backward LSTM.
- (3) We tested different ML classifiers as well as deep learning models.
- (4) In comparison to state-of-the-art techniques, our proposed technique yielded productive performance.

The goal is to improve the scientific literature by detecting emotional indicators in textual material on social networking sites. The proposed system will help companies identify and analyze their consumers' attitudes and emotions toward a program, policy, or product, which can help them make better decisions.

The following is how the rest of the article is organized: (i) Section 2 discusses the current methods and frameworks related to emotion-based sentiment analysis, (ii) Section 3 presents the proposed framework for emotion-based sentiment analysis, (iii) experimental setup is introduced in this Section 4, and (iv) the proposed system is concluded with its drawbacks and potential future guidance in Sections 5 and 6.

## 2. Related Work

Due to the ambiguity of emotions and a large number of emotional terms, detecting and classifying emotions in sentiment analysis is a challenging activity. There have been several techniques and methods proposed for detecting and classifying emotions from social media textual content. This segment gives a summary of the current state of emotion detection and classification science.

The efficacy of different machine learning models is evaluated by [8]. Variant machine learning models such as SVM, NB, DT, LR, XGBoost, KNN, and backpropagation neural (BPN) classifier are tested using the state-of-the-art ISEAR emotion dataset to achieve this goal. According to the results, BPN had the highest accuracy of all the classifiers, with a score of 71.27 percent. The work has a few limitations: (i) it is limited to five emotion classes, (ii) it uses only the ISEAR dataset, (iii) conventional features are used, and (iv) classical machine learning models are used. Future directions may include utilizing various emotion combinations, (ii) utilizing other benchmark datasets, (iii) conducting additional deep learning experiments, and (iv) utilizing a word embedding feature representation approach. The authors of [10] are tasked with categorizing emotions in relation to Indonesian texts. The study used a variety of machine learning algorithms, including NB, SVM, KNN, and minimal optimization, to achieve this goal. Various text cleaning tasks (tokenization, stop word deletion, and case

conversion) are also introduced. During the tests, 10-fold cross-validation was used, and the findings show that the minimal optimization technology outperformed other techniques. For the extraction of variant tweets, we use emotion-word hashtags as well as the “Hashtag Emotion Corpus” data collection [11]. Furthermore, the study used an emotion-labeled tweet dataset to create an emotion dictionary of rich terms. Experiments show that the output of the SVM model is most closely linked to the primary emotion classes. The inclusion of emotion words with variant synonyms, on the other hand, can aid in improving the system’s effectiveness. The identification of emotions in the human speech was performed by [12] in their research. Different speech recognition models are used, which are then followed by various speech features (peak to peak distance). In comparison to alternative methods, the dataset used in the experiments contains 30 distinct subjects, and the highest precision is achieved. An et al. [13] tackled the issue of music emotion classification based on lyrics in their research. They accomplish this by crawling the music lyrics with their tags on Baidu, a well-known network platform. Following that, a naive Bayes classifier that has been trained on four different datasets is used. According to the findings, the final classification accuracy on D-4 was 68 percent (Dataset). Using other algorithms, on the other hand, could improve the proposed system’s effectiveness. Dini and Bittar [14] conducted research to classify emotions on Twitter. Two corpora have been created for this purpose: ETCC for emotion classification of tweets and ETCR for emotion relevance of tweets. This dataset is used to train and evaluate machine learning models, as well as to test a rule-based model. The results show that the symbolic technique outperformed the ML algorithm in determining tweet significance, but the ML algorithm is the best option for tweet emotion classification. The goal in the future is to create a hybrid model that combines both approaches, and marked data quality testing will be discussed as well. Kaewyong et al. [15] suggested a lexicon-based approach in their work on automated feedback inquiry for student feedback. To begin, data was gathered from over 1100 student responses about teaching faculty. Following that, preprocessing techniques are used, followed by using a sentiment lexicon to assign sentiment scores to opinion terms. In comparison with the other approaches, the findings show that the proposed solution has the best efficiency. Sen et al. [16] set out to determine a novel method for multitask learning of input word embedding by using supervised prompts for emotions and sentiment tasks. After that, they looked at using jointly qualified embedding to improve emotion detection. With a test accuracy of 57.46 percent, the findings show that KMeans (ES-SWE) outperformed the other approaches. In the future, it is expected to perform a configuration experiment in which word embedding is mutually trained, sensitive to emotion-topic, and used for emotion or topic downline recognition. Furthermore, their proposed approach can be used to solve other classification problems besides emotion and sentiment identification. The work can also be applied to n-gram embedding. Kollias et al. [17] investigated deep convolutional neural networks for facial

expression and emotion detection (DCNNs). In comparison with the competing models, the experimental findings show that the proposed model outperforms them. However, accurate device estimation can be accomplished by developing a real-world application for human-computer communication. Poria et al. [18] use the convolutional learning approach to tackle the problem of extracting emotions from the media content, specifically text, audio, and video. In comparison to the baseline work, the inner layer of the network used an activation feature and achieved an accuracy of 96.55 percent for the MOUD dataset and 76.85 percent for the IEMOCAP dataset. To conduct microblog sentiment analysis, Severyn and Moschitti [19] proposed a deep learning-based Convolutional Neural Network (CNN). The input to his proposed model is seed words that have been correctly trained using a deep learning model. The key advantage of the framework is that it does not need support attributes to train the model on Twitter data records. The proposed model achieved the best results at the sentence and phrase stages, according to the results of the experiments. Gupta et al. [20] present a new profound learning approach to emotion identification in textual conversations, namely, sentiment and semanticized LSTM (SS-LSTM). Two datasets, ISEAR and SemEval2007 Affective Text, were used in the experiments. The experimental results show that the proposed method outperformed state-of-the-art machine learning as well as other deep learning approaches, with an average F1-score of 71.34 percent. However, the technique can be improved by using a context-sensitive framework to train a model. Cambria [21] uses effective computer methods to investigate the problem of the identification of feelings and emotional predictions. Text classification is carried out in the positive and negative sense module, and emotional clues are identified in the emotional prediction system. First, feelings are found in the proposed approach, and then a certain type of emotion is assigned to the unwanted feeling. To define and classify emotions from textual material, a number of research works are categorized as emotions, as seen in the literature review above.

*2.1. Research Gap.* To resolve the limitations of the aforementioned recent emotion-based sentiment classification techniques, it is necessary to detect and identify emotions from textual material. To close this void, we propose a comprehensive emotion-based sentiment system for emotion classification expressed in online social media. As a result, for emotion classification, we used automated feature engineering techniques followed by a deep neural network called Bi-LSTM.

### 3. Proposed Methodology

The current section presents the proposed method for emotion classification’s comprehensive architecture.

*3.1. Acquiring Data.* We have obtained a benchmark data set called “ISEAR” [8] containing 5474 records. Each input review is tagged with a separate class of three emotion class

sets such as J-S, F-S, and F-G. This work employs the Python language [1], and the library used is Keras, which is based on the TensorFlow deep learning framework [1]. Figure 1 provides a description of the dataset.

**3.1.1. Training Set.** The model has been trained using the training dataset, with 80 percent of the data used for model training [1]. Figure 2 shows an exemplary review of training examples.

**3.1.2. Validation Set.** In the training phase, the model is typically accurate, but in the testing phase, the model’s performance declines. As a result, in order to overcome the model’s performance error in terms of underfitting and overfitting, the validation set must be used [22]. Keras has two methods for determining the optimum model parameters: manual data validation and automatic data validation [3]. We are currently employing manual data validation in our current project.

**3.1.3. Testing Set.** Based on new/unseen cases, the testing set is used to evaluate the performance of the model. It is used once the model has been trained properly using both the train and validation sets. The test set is responsible for the model’s ultimate prediction [23]. Table 1 contains test set samples for emotions classification.

**3.2. Main Modules of the Proposed System.** The proposed method consists of three main modules: (i) Embedding Layer-based Word Representation, (ii) Bi-LSTM-based Forward and Backward Context Information Saving, and (iii) Sigmoid Layer-based Classification. The first module’s goal is to obtain a numeric representation of the terms, which will be fed into the second module, which will produce an encoded representation of features. Bi-LSTM is used to create this encoded representation, which keeps track of both the forward and backward contextual details of the word within a series. Finally, a sigmoid activation mechanism is used to perform classification in the final module (see Figure 3). The following is a breakdown of each module:

**3.2.1. Words Representation Exploiting Embedding Layer.** The emotion dataset is represented as a set of more than one user review and an individual review  $E$  “*I felt very happy when I won the soccer pool*” involving a series of  $r$  words, i.e.,  $w_1, w_2, w_3,$  and  $w_r$ . A single term  $w_i$  “felt” represents an embedding vector  $w_i-R_n$  that includes real values [0.6, 0.9, 0.2]. An embedding matrix is made up of the embedding vectors of each term. During this project, the kera embedding layer was used. The embedding matrix is a two-dimensional matrix denoted by  $DR(rn)$ , where  $r$  denotes the length of the input review and  $n$  denotes the embedding dimension. After that, the next layer receives an embedding matrix  $D$ , also known as a sentence/input matrix.

Task	Dataset with title	Number of Review Text	Labels
Emotion Classification	ISEAR	5474	Joy (1094), Fear (1095), Sadness (1096), Shame (1096), Guilt (1093)

FIGURE 1: The dataset’s description.

Review Text ID	Review Text	Label
1.	When I was involved in a traffic accident.	F-G
2.	When I lost the person who meant the most to me.	J-S
3.	When I did not speak the truth.	F-S
4.	When I caused problems for somebody because he could not keep the appointed time and this led to various consequences.	F-G
6.	When I got a letter offering me the Summer job that I had applied for.	J-S
7.	When I was going home alone one night in Paris and a man came up behind me and asked me if I was not afraid to be out alone so late at night.	F-G
9.	When my friends did not ask me to go to a New Year’s party with them.	J-S
10.	When my uncle and my neighbour came home under police escort.	F-G

FIGURE 2: Train set examples for emotion classification.

**3.2.2. Bi-LSTM Layer.** The Bi-LSTM layer is responsible for learning long dependencies. It aids in the saving of the two preceding and succeeding contexts in the form of an encoded user analysis. A unidirectional LSTM, on the other hand, saves only the information from the previous context, leaving out the information from the subsequent context. As a result, Bi-LSTM gathers even more information for encoded review processing. To learn past and future meaning knowledge of tokens (words), Bi-LSTM employs forward and backward LSTM [2]. *Forward LSTM.* During forward LSTM, the processing of the sequence is done from the left toward right through the concatenation of binary inputs. The first one is present (current) input “ $x_1$ ,” while the second one is prior input/hidden\_state “ $h_{t-1}$ .” Forward LSTM generates a certain outcome “ $h$ ” regarding a provided sequence of input:  $x_1, x_2, x_3, \dots, x_{y-1}$ . *Backward LSTM.* For backward LSTM, the processing of the sequence is done right toward the left through the concatenation of two inputs, the first one is the current input “ $x_1$ ,” and the second one is the next input/future hidden state “ $h_{t+1}$ .” Forward LSTM generates a certain outcome “ $h$ ” regarding a provided sequence of input:  $x_{z+1}, \dots, x_3, x_2, x_1$ .

The integration of forward  $h$  and backward  $\bar{h}$  context representations is done to generate different sentence matrices  $H = [h_1, h_2, h_3, \dots, h_y]$ , while  $H \in \mathbb{R}^{y \times m}$ . The integration of forward and backward output is conducted through the computation of an element-wise sum (see (1)).

TABLE 1: Test set examples for emotion classification.

Review text ID	Review text	Label
5.	When I realise that I value material possessions more than the well-being of my family, I feel a pang of guilt wash over me. I'm feeling quite self-obsessed at the moment.	F-G
8.	When my child was born.	J-S

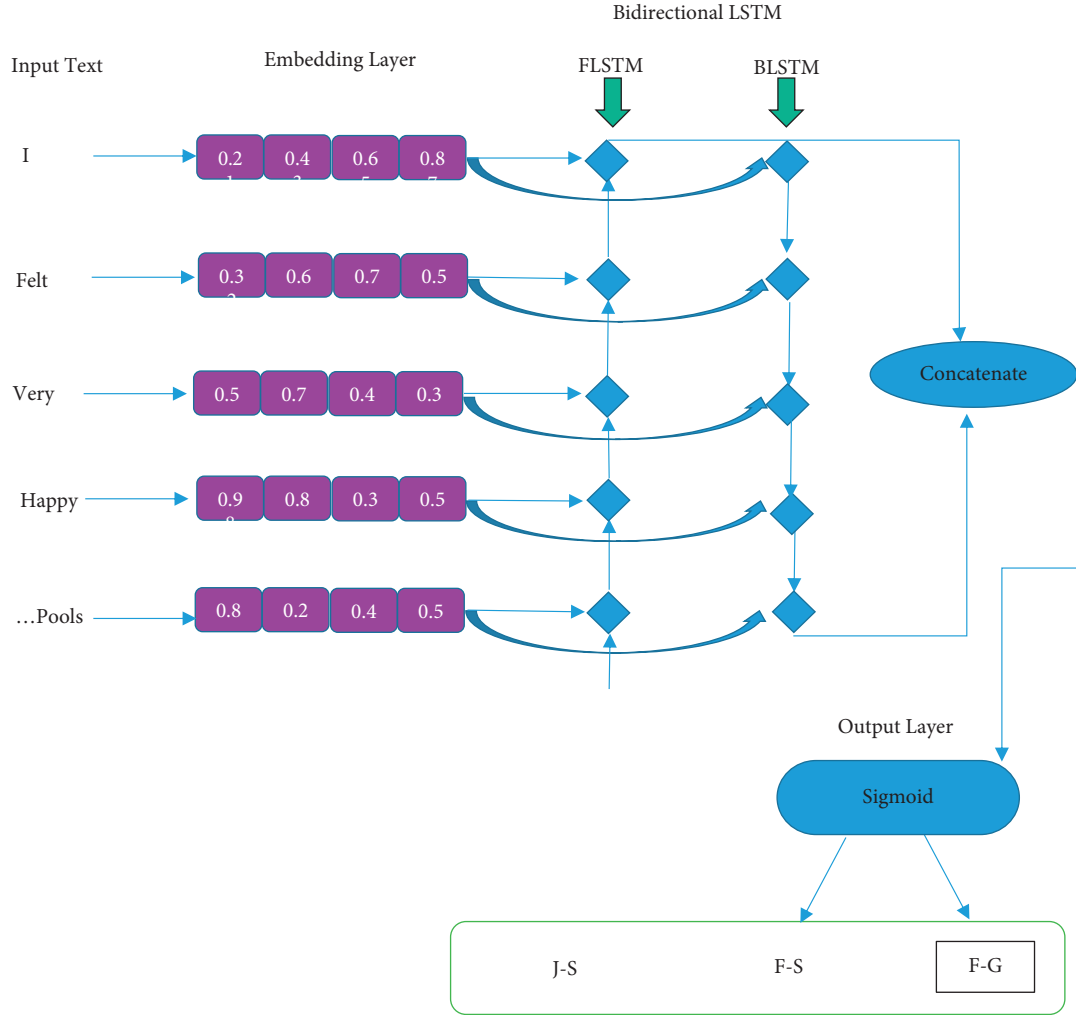


FIGURE 3: The Bi-LSTM model architecture concerning emotion classification.

$$h = \vec{h} \oplus \overleftarrow{h}, \quad (1)$$

Lastly, the new representation of a sentence matrix (1) is fed into a classification layer regarding final classification work.

The equations ((2)–(7)) used for forward LSTM [24] are given as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (2)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (4)$$

$$c \sim_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (5)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c \sim_t, \quad (6)$$

$$h_t = o_t \odot \tanh(c_t). \quad (7)$$

The equations ((8)–(13)) used for backward LSTM [24] are given as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t+1} + b_f), \quad (8)$$

$$i_t = \sigma(W_i x_t + U_i h_{t+1} + b_i), \quad (9)$$

$$o_t = \sigma(W_o x_t + U_o h_{t+1} + b_o), \quad (10)$$

$$c \sim_t = \tanh(W_c x_t + U_c h_{t+1} + b_c), \quad (11)$$

$$c_t = f_t \odot c_{t+1} + i_t \odot c \sim_t, \quad (12)$$

$$h_t = o_t \odot \tanh(c_t), \quad (13)$$

where  $n$  represents the size of the input,  $m$  represents the size of the cell state,  $x_t$  depicts the vector of input with  $n \times 1$  size,  $f_t$  depicts the vector of forget gate with  $m \times 1$  size,  $i_t$  depicts the vector of input gate with  $m \times 1$  size,  $o_t$  depicts the vector of output gate with  $m \times 1$  size,  $h_t$  depicts the vector of output with  $m \times 1$  size, and  $c_t$  depicts a vector of cell state with  $m \times 1$  size.

The output gate weight matrices are represented as  $W_o$ ,  $W_i$ , and  $W_f$  with size  $m \times n$ . Weight matrices of the output gate are represented as  $U_c$ ,  $U_o$ ,  $U_i$ , and  $U_f$  with size  $m \times m$ . The bias vector is represented by  $b_c$ ,  $b_o$ ,  $b_i$ , and  $b_f$  with size  $m \times 1$ . The tangent function is denoted by  $\tanh$ , and the logistics sigmoid function is denoted by  $\sigma$ .

Every gate in Bi-LSTM performs its own function. The function of the forget gate  $f_t$  is to delete useless information. The input gate  $i_t$  takes a decision about storing which information. Lastly, the output gate  $o_t$  calculates the final output  $h_t$ .

The notations used during forward and backward LSTM are listed in Table 2.

**3.2.3. Feature Classification Using Sigmoid Layer.** This layer performs the classification of input features (final representation) obtained from the previous module. We add a dense layer with two neurons that have a sigmoid function for this purpose. The sigmoid activation function performed a nonlinear operation, and its task was to calculate the probability of various emotion classes. It converts the weighted sum into a number between 0 and 1. Therefore, after the output layer passes the review text "I felt very happy when I won the football pool," it is tagged with one of the three binary classes "J-S," "F-S," or "F-G."

The probability of each of the emotion classes is calculated using a softmax activation function. The net input for classifying the final emotion representation (equation (14)) can be approximated as follows:

$$uj = \sum_i^1 wix_i + b, \quad (14)$$

where "w" denotes a weight vector, "x" denotes a vector of inputs, and "b" denotes a bias factor.

The phases of the Bi-LSTM system for emotion categorization are depicted in Algorithm 1.

## 4. Results and Discussion

The experiments in this work are implemented in Python using the Keras library (All things Keras, n.d.) which is based on the TensorFlow deep learning framework [25]. The

machines utilized for testing are Intel Core i7 with a 64 bit OS and 8 GB internal storage. The original dataset is divided into three sets: training, testing, and validation.

**4.1. Answer to RQ1.** To answer RQ1, "How can textual content be identified and classified into different emotion groups using BILSTM, a deep learning technique?" Different experiments are carried out with respect to different emotion cues such as Joy (J), Sadness (S), Fear (F), Shame (S), and Guilt (S) for the output assessment of the proposed deep learning model (G). We created three binary classifiers for this experiment: Joy-Sadness, Fear-Shame, and Fear-Guilt.

**4.1.1. Experiment #1: Joy-Sadness.** Experiment #1 was performed to determine the efficiency of the proposed BILSTM model with respect to Joy-Sadness emotion signals, as shown in Table 3. According to the experimental findings, the BILSTM model performed better for the emotion clue "joy" with an F1-score of 0.89 and recall of 0.91, while both emotion clues "joy" and "sadness" performed best in terms of precision (0.88). The overall precision is 0.88 percent.

**4.1.2. Experiment #2: Fear-Shame.** Experiment #2 was performed to determine the efficacy of the proposed BILSTM model regarding Fear-Shame emotion clues, as shown in Table 4. The BILSTM model had the best performance for the "Shame" emotion hint, with precision (0.89), recall (0.91), F1-score (0.89), and overall accuracy of 0.86 percent, according to the results.

**4.1.3. Experiment #3: Fear-Guilt.** Experiment #3 was performed to determine the efficacy of the proposed BILSTM model regarding Fear-Guilt emotion signals, and the findings are summarized in Table 5. According to the findings, the BILSTM model performed well for both "Fear" and "Shame" emotion clues, with accuracy (0.89), recall (0.89), and F1-score (0.89). The overall precision is 0.89 percent.

**4.2. Answer to RQ2.** To find the answer to RQ2, "What is the efficiency of the proposed technique in relation to other machine learning and deep learning techniques?", we conducted experiments to compare the efficiency of a word embedding scheme trained using the BILSTM method to machine learning classifiers that use traditional feature representation schemes such as the TF-IDF and Count-Vectorizer. The assessment results are presented in Tables 6 and 7.

**4.2.1. BILSTM and Machine Learning Approaches Attempting to Exploit Traditional Features in Comparative Study.** The proposed BILSTM model's output is compared with that of various machine learning approaches using classical feature representation schemes, as shown by the following experiments:

TABLE 2: Notations used during forward and backward LSTM.

Notation	Depiction
$W_c, W_o, W_i, W_f$	Weight matrices of the input gate
$U_c, U_o, U_i, U_f$	Weight matrices of the output gate
$b_c, b_o, b_i, b_f$	Biases
$h_{t-1}$	Prior cell output
$h_t$	Hidden state
$h$	The output obtained from element-wise sum regarding prior cell output and subsequent cell output
$h_{t+1}$	subsequent(future) cell output
$c_t$	Current state cell memory
$i_t$	Gate of input
$f_t$	Gate of forget
$o_t$	Gate of output
$x_t$	Current input
$\sigma$	Sigmoid operation
$\tanh$	Hyperbolic tangent function
$H$	Sentence matrix obtained through Bi-LSTM

```

Data: Emotion dataset “M”, Train Set “TAS”, Test Set “TSS”
Result: Review Text Label: “J-S”, “F-S”, “F-G”
Start
// Review Text Encoding toward machine understandable word vectors (real valued)
while each review text  $R \in M$  do
  while each word  $T \in M$  do
    (1) Word(token) indices allocation
  End while
End while
Initializing Hyperparameter
(2) embed_dim = 100, 128,300, max_features = 2000, epochs = 7, batch_size = 32, train set = 90%, test size = 10%
//Deep Learning model training
while each review text  $R \in M_{TAS}$  do
  (3) Generate all word embedding vectors in  $R = [r_1, r_2, r_3, \dots, r_n]$ 
  (4) Implement Bi-LSTM operation exploiting equations (1)–(13)
End while
// Allocating a label to Review Text final depiction
while each Review Text  $R \in M_{TSS}$  do
  (5) Trained(learned) model is built
  (6) Employ a softmax classifier using Eq. 14, for the classification of output obtained from the Bi-LSTM into “J-S”, “F-S”, “F-G”
End while
End

```

ALGORITHM 1: Emotion classification through Bi-LSTM neural network.

TABLE 3: Proposed BILSTM model performance evaluation results.

Emotion cues	F1-score	Recall	Precision	Accuracy (%)
Joy	0.89	0.91	0.88	0.88
Sadness	0.86	0.83	0.88	

TABLE 4: Proposed BILSTM model performance evaluation results.

Emotion cues	F1-score	Recall	Precision	Accuracy (%)
Fear	0.84	0.85	0.83	0.86
Shame	0.88	0.86	0.89	

TABLE 5: Proposed BILSTM model performance evaluation results.

Emotion cues	F1-score	Recall	Precision	Accuracy (%)
Fear	0.89	0.89	0.89	0.89
Guilt	0.89	0.89	0.89	

TABLE 6: Comparative results of the proposed method with the baseline studies for emotion classification.

Models	Emotion classes	Metrics results				
		Avg. acc (%)	Avg. pre (%)	Avg. rec (%)	Avg. F1-score (%)	
Baseline Study: Neural Network Technique	Back propagation neural Classifier (BPN) [8]	Joy (J), Sadness (S), Fear (F), Shame (S), Guilt (G)	71.27	0.67	0.66	0.67
Proposed Study: Deep Learning Technique	BISLTM	Joy (J), Sadness (S), Fear (F), Shame (S), Guilt (G)	87.66	87.66	87.66	87.66

TABLE 7: J-S emotion class significance.

	Proposed BILSTM correct classification	Proposed BILSTM misclassification	Total
KNN correct classification	165	11	176
KNN misclassification	26	17	43
Total	191	28	219

When we ran a McNemar-type of the test, we found that the chi-squared statistic was 5.5.3, and a 1 represents the degree of freedom, which means that the 2-tailed  $p$  value is 0.021. Consequently, the null hypothesis is deemed to be false, and an alternative hypothesis is adopted.

*ML Driven.* We estimate the output of different classical feature representation strategies for machine learning classifiers, such as Countvectorizer and TF-IDF, in these experiments. The method used in Countvectorizer is called count-of-words, and the text is converted into a feature vector using the TF-IDF scheme. Many classifiers are employed by Countvectorizer, TF, and TF  $\times$  IDF feature encoding techniques, with KNN achieving the best accuracy score of 80.82 percent and XGBoost achieving the worst accuracy score of 71.23 percent.

*Proposed BILSTM Model.* We used the BISLTM model over a word embedding feature representation scheme to perform an emotion classification task in this experiment. The main advantage of using a word embedding over the traditional BOW system is that the BOW model’s output degraded as the vocabulary grew larger, while deep neural network models yielded more successful results.

Table 8 shows that the BILSTM (proposed method) exploits a word embedding scheme to generate more efficient findings than traditional feature representation methods such as Countvectorizer and TF-IDF.

*4.2.2. Comparison of Proposed BILSTM with Variants of Deep Learning Models.* Different studies are carried out to assess the efficacy of the proposed BILSTM model in comparison to various DL models.

(1) *DL Driven.* Individual CNN, Individual LSTM, Individual BILSTM, and Individual RNN with an advanced feature representation scheme were used to test various deep learning models (word embedding). The Individual GRU model achieves the highest accuracy of 82 percent among all DL classifiers, while the Individual RNN model achieves the lowest accuracy of 66 percent.

This segment uses two datasets to compare our approach with different deep learning approaches such as Individual BILSTM, Individual LSTM, Individual CNN, and Individual

RNN. The results of the experiments in Tables 8 and 9 are summarized as follows.

(2) *Comparing Proposed BILSTM with Individual CNN.* In the initial experiments for the emotion classification task, the proposed BILSTM model is compared to the individual CNN. In terms of accuracy, precision, recall, and F1-score, the proposed model outperformed the individual CNN model (accuracy = 88 percent, precision = 88 percent, recall = 88 percent, and F1-score = 88 percent). The decrease in CNN performance is due to the fact that the text classification task requires the preservation of sequential data, which a single CNN cannot help with. In addition, a large dataset must be given to the CNN model in order for it to enhance its accuracy.

(3) *Comparing Proposed BILSTM with Individual LSTM.* In the second experiment, an Individual LSTM is compared to the proposed model in an emotion classification task. The downside of the unidirectional LSTM is that it only retains previous information and not subsequent information. Maintaining information on both sides of words (previous and subsequent) helps in a greater understanding of sentence context. As a result, when compared to the proposed deep neural network model, an Individual LSTM layer will reduce the performance by up to 10% for J-S, 2% for F-G, and 1% for F-S, as shown in Table 8.

(4) *Comparing Proposed BILSTM with Individual GRU.* For the classification of emotions, a performance comparison between the proposed BILSTM model and an Individual GRU is performed in the third experiment. The main disadvantage of GRU is that it does not perform feature extraction and instead focuses on preserving contextual details. In comparison with the proposed BILSTM model, GRU’s output dropped by up to 6% for J-S, 3% for F-G, and 4% for F-S, as shown in Table 8.

(5) *Comparing Proposed BILSTM with Individual RNN.* A comparison of the proposed model and the RNN model is carried out in the final experiment. The RNN’s output is

TABLE 8: Evaluation results of variant deep learning and the proposed technique.

Classifier/model	Accuracy (%)		Precision (%)		Recall (%)		F1-score (%)	
	J-S F-S	F-G	J-S F-S	F-G	J-S F-S	F-G	J-S F-S	F-G
Individual CNN [26]	74 76	75	74 76	75	74 76	75	74 76	75
Individual LSTM [20]	78 85	87	78 85	87	78 85	87	78 85	87
Individual GRU	82 82	86	82 82	86	82 82	86	82 82	86
Individual RNN	66 80	78	74 83	83	73 83	83	72 83	83
Proposed (BILSTM)	88 86	89	88 86	89	88 86	89	88 86	89

TABLE 9: Evaluation results of machine learning and the proposed technique.

Classifier/model	Accuracy (%)		Precision (%)		Recall (%)		F1-score (%)	
	J-S F-S	F-G	J-S F-S	F-G	J-S F-S	F-G	J-S F-S	F-G
SVM [8]	79 67.73	81.74	79 75	82	79 68	82	79 82	67
KNN	80.82 79.55	79	81 79	79	81 79	79	81 79	79
LR	78.08 81.36	82.19	78 81	82	78 82	82	78 81	82
RF	73.52 72.27	67.12	73 72	67	73 72	67	73 72	67
MNB [27]	74.89 84.09	85.84	77 84	86	76 84	86	75 84	86
DT [27]	76.71 76.82	73.52	77 76	74	77 76	73	77 76	73
XGBoost	71.23 73.18	72.15	80 73	72	71 73	72	67 73	72
Proposed (BILSTM)	88 86	89	88 86	89	88 86	89	88 86	89

degraded because it is unable to manipulate additional broad sequences, demonstrating the RNN’s inability to monitor long-range relationships. It is necessary to retain the details for a long time in order to maintain the context, which an Individual RNN cannot do because it just keeps track of short-term memory sequences.

In comparison to the BILSTM, RNN resulted in a performance decline of up to 22% for J-S, 11% for F-G, and 6% for F-S, as shown in Table 8.

**4.2.3. Why the Proposed BILSTM Model Is Better?** As we employed “Bidirectional” LSTM (BILSTM) on the supplied datasets, the findings outperform other deep learning approaches. The purpose of the BILSTM model is to retain background data from both sides of an expression, i.e., the left and right sides, within a phrase. After receiving data from the embedding layer, the BILSTM generates an enhanced encoding of the data that takes into consideration both the current and earlier input information. As a consequence, it is obvious that the BILSTM

deep learning model can successfully collect current and past background information through time and produce predictions. The suggested deep learning model performed well on the dataset in categorizing the input text into various emotions such as Pleasure, Sadness, Fear, Guilt, and Shame.

**4.3. Answer to RQ.3.** To answer RQ3, “How to estimate the efficiency of the proposed technique regarding emotion classification with regard to baseline studies?”, we ran an experiment to see how well the proposed BILSTM model performed in comparison to the baseline sample. The experimental results are mentioned in Table 6.

**4.3.1. Proposed (BILSTM) Compared with Baseline (BPN).** In this experiment, we compared the output of the proposed BILSTM approach with that of [8], who used a back-propagation neural classifier to classify emotions. With an accuracy of 87.66 percent, the proposed method outperformed the current state-of-the-art analysis, as shown in



TABLE 10: F-G emotion class significance.

	Proposed BILSTM correct classification	Proposed BILSTM misclassification	Total
MNB correct classification	150	10	160
MNB misclassification	19	40	59
Total	169	50	219

With one degree of freedom, we observed that the 2-tailed  $p$  value was 0.137 with a chi-square statistic of 2.22. As a result, the null hypothesis is discarded in favor of an alternative hypothesis.

TABLE 11: F-S emotion class significance.

	Proposed BILSTM correct classification	Proposed BILSTM misclassification	Total
MNB correct classification	145	20	165
MNB misclassification	29	25	54
Total	174	45	219

A chi-square statistic  $a = 1.3$  gives us a  $p$  value of 0.253 for the two-tailed McNemar’s test, and a degree of freedom of 1. An alternate hypothesis is therefore accepted, and the null hypothesis is rejected.

Table 6. The following are the reasons for our model’s improved performance: The reason for our improved model output is that we used a “bidirectional” LSTM model that is effective at maintaining both the left and right contextual details of the series. Furthermore, BILSTM is good at storing knowledge over a long period of time. As a result, keeping information for a long time is very useful for text classification and prediction tasks. BPN, on the other hand, is not suitable for classification issues [1]. In addition, the BPN classifier has issues with delay convergence, neural network weights declining over local optima, and network insensitivity [28].

4.3.2. *Significance Test.* We also ran experiments to see whether the proposed BILSTM classifier, which uses the word embedding function, is statistically different from KNN for J-S and MNB for F-G and F-S, to perform a significance test for the “emotion classification task.” The findings suggest that our proposed BILSTM model’s efficacy in the emotion classification task for J-S, F-G, and F-S was greatly enhanced by innovative features (word encoding), as seen in Tables 7, 10, and 11.

## 5. Conclusion

We further analyzed why and how users react in different emotional states. To carry out the research task, we proposed applying a Bi-LSTM technique. The study will include the following modules: (i) take the data, (ii) prepare the data, and (iii) apply the deep learning algorithm. For the emotion classifier task, we used a deep learning model, namely, Bi-LSTM. Bi-LSTM performs two tasks at once; i.e., it can remember both the forward and backward sequence of the previous time sequence [8]. After using numerous other programs to encode the text, the decoded text is manually classified as J, F, and G. Experiments to apply different machine learning and deep learning algorithms to emotion datasets were also conducted. The results show that the proposed Bi-LSTM model produced improved results in terms of improved accuracy (87.66%), precision (87.66%),

recall (87.66%), and F1-score (87.66%) with respect to the compared studies.

The following are some potential constraints on the proposed work:

- (1) We perform the emotional classification of text content
- (2) The research is limited to random word embedding, with no use of word representation models such as Glove, FastText, or word2vec
- (3) Emotions are not classified using other configurations of deep learning techniques
- (4) The current study’s content is solely focused on the ISEAR and Twitter emotion datasets
- (5) In the work, limited emotion clues are exploited
- (6) The current research does not mention emotion intensities such as strong negative, strong positive, weak negative, and weak positive, which must be discussed to make the system more effective
- (7) The research is limited to texts written in English
- (8) AUC, density, and error rate can be used to better estimate the classifier’s performance; however, the proposed study topic is confined to performance metrics like accuracy, preciseness, recall, and F1-score
- (9) A modest number of machine learning models are currently being used for testing, but that can be extended

## 6. Future Directions

The following is a list of probable future options for the research work:

- (1) Photographs and videos may be used to expand the work
- (2) Other previously trained schemes such as Glove, word2vec, and FastText can be used for word embedding layer in future work

- (3) For emotional classification, we will investigate the various combinations of deep neural networks
- (4) Exploring additional emotional classification tasks in different data sets
- (5) One potential plan in the future is to increase the performance of the research project by extending the different emotion clues
- (6) The research will be expanded to include other languages to test the efficacy of the proposed model in other languages
- (7) The problem of long computation times can be addressed using GPU, which allows for large-scale dataset experiments
- (8) Using a combination of deep neural network systems to solve the problem of emotion classification would be more efficient. As a result, we will use other neural networks in the future
- (9) In the future, we will put more emphasis on using ensemble approaches to improve device efficiency
- (10) We will focus on adding more base models and looking for other criteria that might further improve the overall accuracy of the proposed work
- (11) In future studies, we will look at combining our proposed approach with other NLP strategies such as part of speech and marking to achieve better results for NLP problems

## Data Availability

Two publically available datasets (ISEAR and SemEval2007) were used to support the findings of this study. The two datasets were cited and included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University for funding through the Vice Deanship of Scientific Research Chairs. The APC was funded by Taif University Researchers Supporting Project Number (TURSP-2020/331), Taif University, Taif, Saudi Arabia. The authors would like to acknowledge the support.

## References

- [1] A. Khattak, M. Z. Asghar, M. Ali, and U. Batool, "An efficient deep learning technique for facial emotion recognition," *Multimedia Tools and Applications*, pp. 1–35, 2021.
- [2] M. Z. Asghar, F. Subhan, H. Ahmad et al., "Senti-eSystem: a sentiment-based eSystem -using hybridized fuzzy and deep neural network for measuring customer satisfaction," *Software: Practice and Experience*, vol. 51, no. 3, pp. 571–594, 2021.
- [3] A. Khattak, A. Khan, H. Ullah et al., "An efficient supervised machine learning technique for forecasting stock market trends," in *Information and Knowledge in Internet of Things*, pp. 143–162, Springer, Cham, New York, NY, USA, 2022.
- [4] Y. Wang, F. Subhan, S. Shamshirband, M. Z. Asghar, I. Ullah, and A. Habib, "Fuzzy-based sentiment analysis system for analyzing student feedback and satisfaction," *Computers, Materials & Continua*, vol. 62, no. 2, pp. 631–655, 2020.
- [5] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.
- [6] D. Xue, L. Wu, Z. Hong, S. Guo, and L. Gao, "Deep learning-based personality recognition from text posts of online social networks," *Applied Intelligence*, vol. 48, no. 11, pp. 4232–4246, 2018.
- [7] U. Gupta, A. Chatterjee, R. Srikanth, and P. Agrawal, "A sentiment-and-semantics-based approach for emotion detection in textual conversations," 2017, <https://arxiv.org/abs/1707.06996>.
- [8] M. Z. Asghar, F. Subhan, M. Imran, F. M. Kundi, A. Khan, S. Shamshirband, A. Mosavi, P. Csiba, and A. R. Varkonyi-Koczy, "Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1093–1118, 2020.
- [9] H. Ahmad, M. U. Asghar, M. Z. Asghar, A. Khan, and A. Mosavi, "A Hybrid Deep Learning Technique for Personality Trait Classification from Text," *IEEE Access*, vol. 9, 2021.
- [10] N. A. S. Winarsih and C. Supriyanto, "Evaluation of classification methods for Indonesian text emotion detection," in *Proceedings of the 2016 International Seminar on Application for Technology of Information and Communication (ISEmantic)*, pp. 130–133, IEEE, Indonesia, Asia, August 2016.
- [11] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets," *Computational Intelligence*, vol. 31, no. 2, pp. 301–326, 2015.
- [12] A. Davletcharova, S. Sugathan, B. Abraham, and A. P. James, "Detection and analysis of emotion from speech signals," *Procedia Computer Science*, vol. 58, pp. 91–96, 2015.
- [13] Y. An, S. Sun, and S. Wang, "Naive Bayes classifiers for music emotion classification based on lyrics," in *Proceedings of the 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 635–638, IEEE, Wuhan, China, May 2017.
- [14] L. Dini and A. Bittar, "Emotion analysis on twitter: the hidden challenge," in *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pp. 3953–3958, LREC, Portorož, Slovenia, May 2016.
- [15] P. Kaewyong, A. Sukprasert, N. Salim, and A. Phang, "The possibility of students' comments automatic interpret using lexicon based sentiment analysis to teacher evaluation," in *Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Science*, vol. AICS2015, pp. 179–189, George Malaysia, September 2015.
- [16] A. Sen, M. Sinha, S. Mannarswamy, and S. Roy, "Multi-task representation learning for enhanced emotion categorization in short text," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 324–336, Springer, Jeju, Republic of Korea, April 2017.
- [17] D. Kollias, A. Tagaris, and A. Stafylopatis, "On line emotion detection using retrainable deep neural networks," in *Proceedings of the Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pp. 1–8, IEEE, Athens Greece, December 2016.

- [18] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *Proceedings of the Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 439–448, IEEE, Barcelona, Spain, December 2016.
- [19] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval 2015*, pp. 959–962, ACM, New York, NY, USA, August 2015.
- [20] S. Gupta, R. Singh, and V. Singla, "Emoticon and text sarcasm detection in sentiment analysis," in *Proceedings of the First International Conference on Sustainable Technologies for Computational Intelligence*, pp. 1–10, Springer, Jaipur, Rajasthan, India, March 2019.
- [21] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.
- [22] H. M. Khan, F. M. Khan, A. Khan, M. Z. Asghar, and D. M. Alghazzawi, "Anomalous behavior detection framework using HTM-based semantic folding technique," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 5585238, 2021.
- [23] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Applied Sciences*, vol. 11, no. 24, Article ID 11634, 2021.
- [24] J. Asghar, S. Akbar, M. Z. Asghar, B. Ahmad, M. S. Al-Rakhami, and A. Gumaei, "2021, Detection and classification of psychopathic personality trait from social media text using deep learning model," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 5512241, 10 pages, 2021.
- [25] M. K. Abadi, R. Subramanian, S. M. Kia, P. Avesani, I. Patras, and N. Sebe, "2015, DECAF: MEG-based multimodal database for decoding affective physiological responses," *IEEE Transactions on Affective Computing*, vol. 6, no. 3, pp. 209–222.
- [26] M. Baali and N. Ghneim, "Emotion analysis of Arabic tweets using deep learning approach," *Journal of Big Data*, vol. 6, no. 1, pp. 1–12, 2019.
- [27] M. Osama and S. R. El-Beltagy, "A transfer learning approach for emotion intensity prediction in microblog text," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, pp. 512–522, Springer, Cairo, Egypt, October 2019.
- [28] N. Prasad, R. Singh, and S. P. Lal, "Comparison of back propagation and resilient propagation algorithm for spam classification," in *Proceedings of the 2013 5th International Conference on Computational Intelligence, Modelling and Simulation*, pp. 29–34, IEEE, Seoul, Republic of Korea, September 2013.

## Research Article

# Employing Machine Learning-Based Predictive Analytical Approaches to Classify Autism Spectrum Disorder Types

Muhammad Kashif Hanif <sup>1</sup>, Naba Ashraf <sup>1</sup>, Muhammad Umer Sarwar <sup>1</sup>,  
Deleli Mesay Adinew <sup>2</sup> and Reehan Yaqoob <sup>1</sup>

<sup>1</sup>Department of Computer Science, Government College University, Faisalabad, Pakistan

<sup>2</sup>Department of Computer Science, Dilla University, Dilla, Ethiopia

Correspondence should be addressed to Deleli Mesay Adinew; mesaya@du.edu.et

Received 28 May 2021; Revised 28 November 2021; Accepted 10 December 2021; Published 10 January 2022

Academic Editor: Sheng Du

Copyright © 2022 Muhammad Kashif Hanif et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Autism spectrum disorder is an inherited long-living and neurological disorder that starts in the early age of childhood with complicated causes. Autism spectrum disorder can lead to mental disorders such as anxiety, miscommunication, and limited repetitive interest. If the autism spectrum disorder is detected in the early childhood, it will be very beneficial for children to enhance their mental health level. In this study, different machine and deep learning algorithms were applied to classify the severity of autism spectrum disorder. Moreover, different optimization techniques were employed to enhance the performance. The deep neural network performed better when compared with other approaches.

## 1. Introduction

Autism spectrum disorder (ASD) is an inherited long-living and neurological disorder that starts in the early age of childhood with complicated causes [1]. The person with ASD can have mental disorders such as anxiety, miscommunication, and limited repetitive interest. ASD can affect the person ability to perform the function properly in the different stages of life. Therefore, initial diagnosis and treatment are tremendously important [2]. One of the most important symptoms of ASD is the behaviour of the affected person with others [3]. Normally, children with autism speak very little and stay quiet. They can adopt specific behaviour from movies and cartoons. For this reason, they can show a risky unexpected behaviour [4].

According to the World Health Organization, ASD affects about 1% population of worldwide [5] and the ratio around the world is increasing very rapidly [1, 6]. According to the Centers for Disease Control and Prevention, the prevalence of ASD has risen to approximately 1 in 68. The frequency of occurrence of ASD in males is approximately

four times higher than that in females [7]. ASD can be present in ethnic, racial, and economic groups. In the United States, most children are not diagnosed with ASD until they reach four years [8]. ASD affects about 1.4% population in the region of South Asia [9].

There exists no biological test to diagnose ASD. Current practices to diagnose ASD rely on behavioural patterns [10]. Autism detection in the earlier stage can prevent the patient situation from more deteriorating and also help to decrease the costs that are linked with overdue diagnosis [1]. According to the Diagnostic and Statistical Manual of Mental Disorders, 5th Edition (DSM-5), the severity of ASD can be determined on a spectrum, which has three levels of severity ranging from mild to severe symptoms: Level 1: needs support, Level 2: needs substantial support, and Level 3: needs very substantial support [11, 12].

Machine learning is a growing field, which uses mathematical learning, statistical estimation, and information theories to find useful patterns in the large amount of data [13–16]. Recently, deep learning is the most trending area of research, which is the subset of machine learning and uses

the neural network architectures to model the high-level abstraction in data [17]. These structures contain several layers with processing units that apply linear or nonlinear transformations to the input data [17]. In recent years, different types of deep learning architectures have been already applied to the supervised and unsupervised datasets in the medical field [18–20].

Different studies have employed machine learning and deep learning techniques to predict, diagnose, and classify ASD [21]. The objective of this study was to apply machine learning and deep learning techniques to classify the severity of ASD. The proposed techniques employ cross-validation with hyperparameter tuning. Furthermore, this study performs statistical analysis to compare the models. The experimental results depict that DNN outperforms other methods.

The next section presents the related work. Then, the proposed methodology is discussed. Afterwards, the results are presented. The last section concludes the outcomes.

## 2. Review of Literature

Researchers have applied different techniques to diagnose the ASD. Dvornek et al. [22] applied the recurrent neural networks with long short-term memory for the classification of individuals with ASD and typical controls for the Autism Brain Imaging Data Exchange (ABIDE) dataset and attained 68.5% accuracy. Van den Bekerom [23] used the NSCH dataset and applied the four machine learning classification algorithms to classify the severity of ASD. The result to classify the severity of ASD attained the accuracy of 0.50% to 0.54% and also used the one-way method, which improved the prediction accuracy of the severity of ASD of 54.1% to 90.2%.

Heinsfeld et al. [24] used the machine learning techniques such as SVM, RF, and DNN in the ABIDE dataset, and the result showed the accuracy of 0.65% to 0.70%. Bi et al. [25] used the multiple SVM to classify the patients and normal controls. Altay and Ulas [26] applied the linear discriminant analysis (LDA) and K-nearest neighbor (KNN) to diagnose the ASD and attained 90.8% and 88.5% accuracy, respectively. Mohammadian Rad and Furlanello [27] applied deep learning to the automatic detection of stereotypical motor movements. They used the convolutional neural network (CNN) to learn a discriminative feature space from raw data. They also combined the long short-term memory with CNN to model the temporal patterns in a sequence of multi-axis.

Kong et al. [28] used the DNN for ASD classification. They extracted the features for each subject and ranked them, and the top 3000 features were used as input to DNN for classification. The proposed method attained 90.39% accuracy. Eslami and Saeed [29] proposed the Auto-ASD-Network model and classified the subjects of ASD from healthy subjects. They used the deep learning to find the useful patterns from the dataset. They applied the auto-tune model to optimize the hyperparameters of SVM and achieved 70% accuracy for the fMRI dataset. Wilson and Rajan [30] applied the deep learning algorithm to detect the

ASD from the brain imaging dataset and attained 70% accuracy in the detection of ASD. Pream et al. [31] used the supervised machine learning techniques to identify the syndromic ASD. They attained 98% and 94% accuracy using SVM and decision tree, respectively. Nasser et al. [32] build the artificial neural network model to diagnose the ASD. The data were gathered from the ASD screening application that contains ASD test outcomes based on queries from users. Table 1 provides the summary of the related work.

## 3. Methodology

The classification of severity of ASD is difficult due to dependence on different features. This study proposes the machine learning models to classify the severity of ASD (Figure 1). In this work, the survey-based national survey of children’s health dataset is used, which was collected from 2011 to 2012. The survey was conducted in the United States [23]. The participants of survey were children of age 2 to 17 years in the United States. The dataset consists of 95 677 records.

Next, the dataset is preprocessed to clean the data and remove the irrelevant parts of dataset. First, the columns that have a single value or few unique values were identified. VarianceThreshold class was used to remove these columns. Next, the duplicate rows were identified and removed. The rows having null values in most of the columns were removed. Moreover, the imputation technique with mean value was employed to handle the remaining missing values. Label encoding was used to encode the target label into numeric values. The dataset has imbalance classes (Figure 2). The classes in dataset were encoded to 0, 1, 2, and 3, which corresponds to no ASD, Level 1 ASD, Level 2 ASD, and Level 3 ASD, respectively.

Most of the machine learning algorithms are sensitive to the data scaling. For this reason, it is good practice to adjust the data representation [33]. In this study, different scaling techniques were used and StandardScaler technique showed better performance. Therefore, this data transformation technique was used for machine learning pipeline.

Furthermore, different dimensionality reduction techniques were considered [34, 35]. We employed principal component analysis (PCA) for this study. PCA rotates the dataset in a way such that features are statistically uncorrelated. Then, the subset of the rotated features are used based on their importance [33]. For the sake of model development, we used 70% of total dataset as training and the remaining 30% dataset as testing.

In this study, we considered random forest (RF), support vector machine (SVM), naive Bayes (NB), K-nearest neighbor (KNN), and deep neural network algorithms to predict and classify the severity of ASD. We also applied hyperparameter tuning with stratified k-fold cross-validation for each machine and deep learning model to obtain the best parameters for each model. Stratified k-fold cross-validation was used to attain more reliable estimate of generalization of performance [33].

Cross-validation was used to estimate the performance for each parameter combination. For this purpose, the

TABLE 1: Summary of related work.

Sr. No.	Authors	Objectives	Method	Dataset	Accuracy
1	Dvornek et al [22]	Used the long short-term memory network to identify the autism from resting-state fMRI.	Recurrent neural networks (long short-term memory)	ABIDE	68.5%
2	Van den Bekerom [23]	To use the machine learning to predict the ASD.	Machine learning algorithms (SVM, RF, and NB) and 1-way method	National Survey of Children’s Health (NSCH) data	0.49% to 0.54% in 4 classes using machine learning and 54.1% to 90.2% using the 1-way method
3	Heinsfeld et al [24]	To apply the deep learning algorithm to identify the ASD.	SVM, RF, DNN	ABIDE	65%, 63%, 70%
4	Bi et al [25]	To apply the random SVM to classify the ASD.	Random support vector cluster	ABIDE	96.15%
5	Altay and Ulas (2018)	To use the classification method to identify the children who have ASD or not.	Discriminant analysis, KNN	292 samples	91%, 89%
6	Kong et al [28]	To identify the ASD using DNN.	DNN	ABIDE	90.39%
7	Eslami and saeed [29]	To diagnose the ASD using the Auto-ASD-Network based on the deep learning and SVM.	DNN, SVM, state-of-the-art classifier	4 fMRI datasets	80%
8	Wilson and rajan [30]	To use the brain imaging dataset to identify the ASD.	DL algorithms	ABIDE	70%
9	Sudha and Vijaya [31]	To identify the ASD by supervised algorithms.	Decision tree, SVM, MLP		98%, 96%, 95%
10	Nasser et al [32]	To identify the ASD by artificial neural network.	ANN	Dataset collected from the ASD screening app	100%

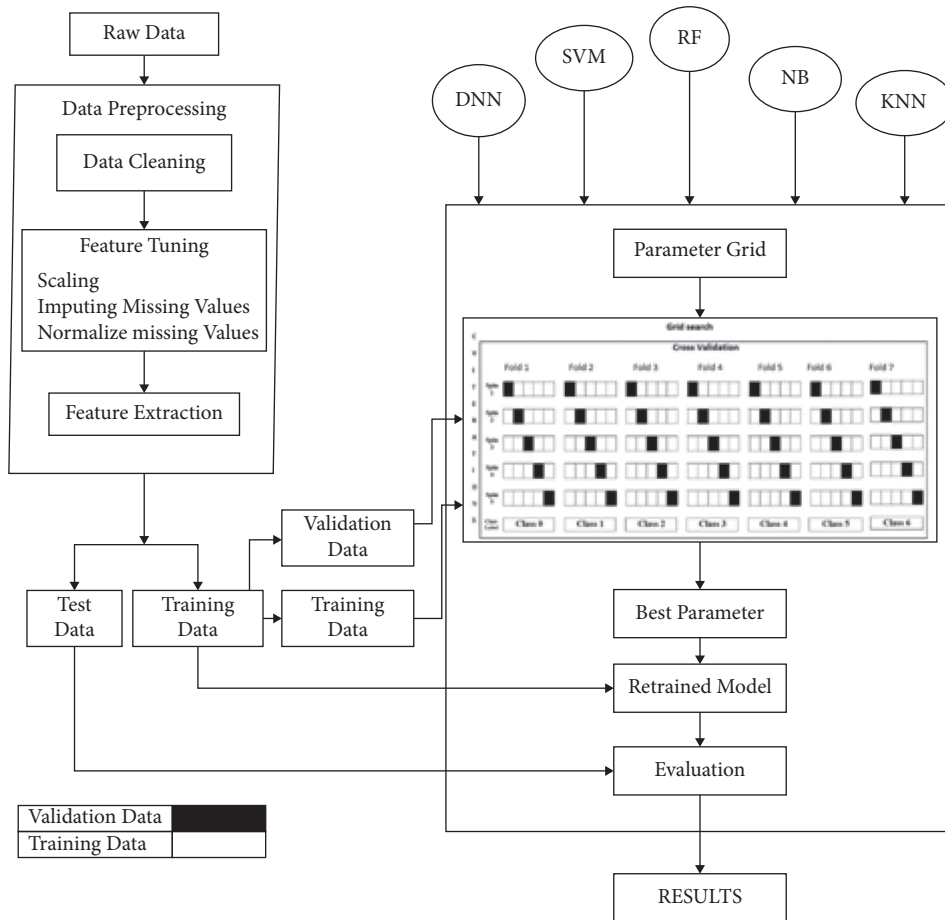


FIGURE 1: Proposed methodology.

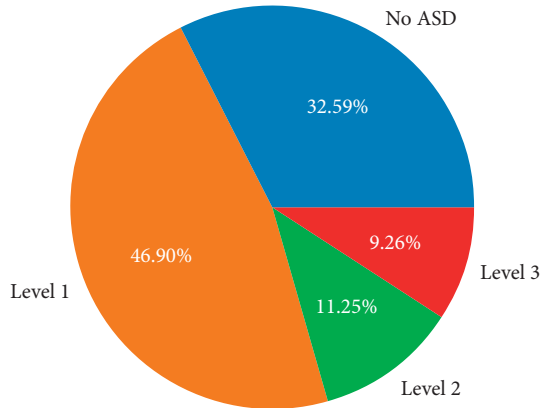


FIGURE 2: Class distribution.

training and validation data are split for each parameter setting. The accuracy values were computed for each parameter setting for each split in the cross-validation. The mean validation accuracy is calculated for each parameter setting [33]. The best parameters for each model were obtained using the best cross-validation performance [33]. The model was retrained using these best parameters to attain the best performance. Then, the model was evaluated using test data.

For each classifier, we used the pipeline function of Sklearn library and employed standard scalar, PCA, and classifier in this order. For each classifier, PCA parameter  $n\_components$  having range (2, 10, 5) is used. RF is the combination of many trees and one of the most successful ensemble learning methods [36]. The different values of  $n\_estimators$ ,  $min\_samples\_split$ ,  $min\_samples\_leaf$ , and  $max\_features$  parameters for RF classifier were considered. To obtain the best parameters for RF using grid search, parameter values of  $n\_estimators$ ,  $min\_samples\_split$ ,  $min\_samples\_leaf$ , and  $max\_features$  [50, 150, 200], [2, 5, 10], [1, 2, 4], and ["auto," "rbf"] were considered, respectively. After that, we applied the fivefold cross-validation with grid search for the training and validation datasets. After applying the fivefold cross-validation with grid search, the best parameter value obtained for  $n\_estimators$  is 200,  $min\_samples\_split$  is 5,  $min\_samples\_leaf$  is 4,  $max\_features$  is auto, and  $n\_components$  of PCA is 7. Then, the RF classifier was retrained using these best parameters.

SVM is one of the best classification methods that can handle multiple variables. Kernel, gamma, and C parameters were considered for SVM classifier. The different parameter values of kernel, gamma, and C used were ["linear," "rbf"], [0.001, 0.01, 0.1], and [0.001, 0.01, 0.1], respectively. After applying cross-validation with grid search, the best parameters obtained for kernel, gamma, and C were "linear," 0.001, and 0.001, respectively. For PCA,  $n\_components$  value is 2. Then, the SVM is retrained with these best parameters. For NB,  $var\_smoothing$  parameter containing the  $np.log\_space(0, -9, num = 20)$  is used and value 1.0 is obtained as the best value using cross-validation and hyperparameter tuning. The best parameter of  $n\_components$  for PCA is 2. KNN is a nonparametric method and is based on the proximity of the sample feature and the training set [37]. The

$n\_neighbors$  and  $p$  parameters of KNN with values [1–7] and [1, 2, 5] were used, respectively. After that, we applied the fivefold cross-validation in combination with grid search for the training dataset and validation dataset. The best parameter value for  $n\_neighbors$  is 7,  $p$  is 1, and  $n\_components$  of PCA is 7.

Moreover, cross-validation and grid search were applied to DNN to find the best parameters. The parameters considered for DNN were activation, batch\_size, epochs, optimizer, learn\_rate, momentum, init\_mode, dropout\_rate, and weight\_constraint. The parameter values and best values for each parameter are listed in Table 2.

## 4. Results and Discussion

The objective of this work was to classify the severity of ASD. This section describes the results of different machine and deep learning models considered to classify the severity of ASD. For the model development, we used different functions of Keras and Scikit-learn libraries. Keras is easy to use, fast, and open-source neural network library that runs on top of Theano or TensorFlow. It provides the easy workflow to train and define the neural network in just a few lines of code, but Keras does not handle the low-level computation. To build the DNN model, we adapted the sequential model. Scikit-learn library was used to design SVM, NB, RF, and KNN models. It is the library of machine learning that is written in Python programming language. For visualization of results, matplotlib, seaborn, and PyCaret libraries were used. Moreover, the Google Colab platform was used to create models and obtain the results.

To evaluate the performance of these machine and deep learning models, learning curves and precision-recall curves were used. A learning curve is used to show the performance when there is change in the training set or time. In this study, we used model performance on  $x$ -axis and training set on  $y$ -axis for learning curve. It can detect high variance or high bias in the model. The dataset considered in this work has class imbalance. For this reason, we used the precision-recall curve to check the performance of different models [38]. Machine and deep learning models were trained on dataset using different parameters. Table 2 lists the best parameters obtained for each model using grid search with cross-validation.

Figure 3 summarizes the learning curves of different parameters for DNN model and also shows the loss and accuracy on training sets and test sets. DNN parameters considered were batch\_size, epochs (Figure 3(a)), optimization algorithm (Figure 3(b)), learn\_rate and momentum (Figure 3(c)), weight initialization (Figure 3(d)), activation function (Figure 3(e)), and dropout regularization (Figure 3(f)). For each case considered, the train and test loss is decreasing in second epochs and these losses converge. The accuracy of train and test is almost the same. This means the model is neither underfitting nor overfitting. Moreover, this shows that model does not have high variance or high bias. The reason for better performance of model is the use of stratified cross-validation.

TABLE 2: Parameters considered for different machine and deep learning models.

Model	Parameters	Best Parameters
RF	(i) <code>n_estimator</code> = [50, 150, 200] (ii) <code>min_samples_split</code> = [2, 5, 10] (iii) <code>min_samples_leaf</code> = [1, 2, 4] (iv) <code>max_features</code> = [auto, rbf]	(i) <code>n_estimator</code> = 200 (ii) <code>min_samples_split</code> = 5 (iii) <code>min_samples_leaf</code> = 4 (iv) <code>max_features</code> = auto (v) <code>PCA n_components</code> = 7
SVM	(i) <code>kernel</code> = [linear, rbf] (ii) <code>gamma</code> = [0.001, 0.01, 0.1] (iii) <code>C</code> = [0.001, 0.01, 0.1]	(i) <code>kernel</code> = linear (ii) <code>gamma</code> = 0.001 (iii) <code>C</code> = 0.001 (iv) <code>PCA n_components</code> = 2
NB	(i) <code>var_smoothing</code> = <code>np.logspace</code> (0, 9, num = 20)	(i) <code>var_smoothing</code> = 1.0 (ii) <code>PCA n_components</code> = 2
KNN	(i) <code>n_neighbors</code> = [1-7] (ii) <code>p</code> = [1, 2, 5]	(i) <code>n_neighbors</code> = 7 (ii) <code>p</code> = 1 (iii) <code>PCA n_components</code> = 7
DNN	(i) <code>activation function</code> = [softplus, softmax, softsign, relu, tanh, sigmoid, hard_sigmoid, linear] (ii) <code>batch_size</code> = [10, 50, 100] (iii) <code>epochs</code> = [10, 20, 40, 50, 80, 100] (iv) <code>optimizer</code> = [RMSprop, SGD, Adagrad, Adadelta, Adam, Adamax, Nadam] (v) <code>learn_rate</code> = [0.001, 0.01, 0.1, 0.2, 0.3] (vi) <code>momentum</code> = [0.0, 0.2, 0.4, 0.6, 0.8, 0.9] (vii) <code>init_mode</code> = [uniform, lecun_uniform, normal, zero, glorot_normal, glorot_uniform, he_normal, he_uniform] (viii) <code>dropout_rate</code> = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] (ix) <code>weight_constraint</code> = [1-5]	(i) <code>activation function</code> = relu (ii) <code>batch_size</code> = 10 (iii) <code>epochs</code> = 10 (iv) <code>optimizer</code> = SGD (v) <code>learn_rate</code> = 0.001 (vi) <code>momentum</code> = 0.0 (vii) <code>init_mode</code> = uniform (viii) <code>dropout_rate</code> = 0.0 (ix) <code>weight_constraint</code> = 1

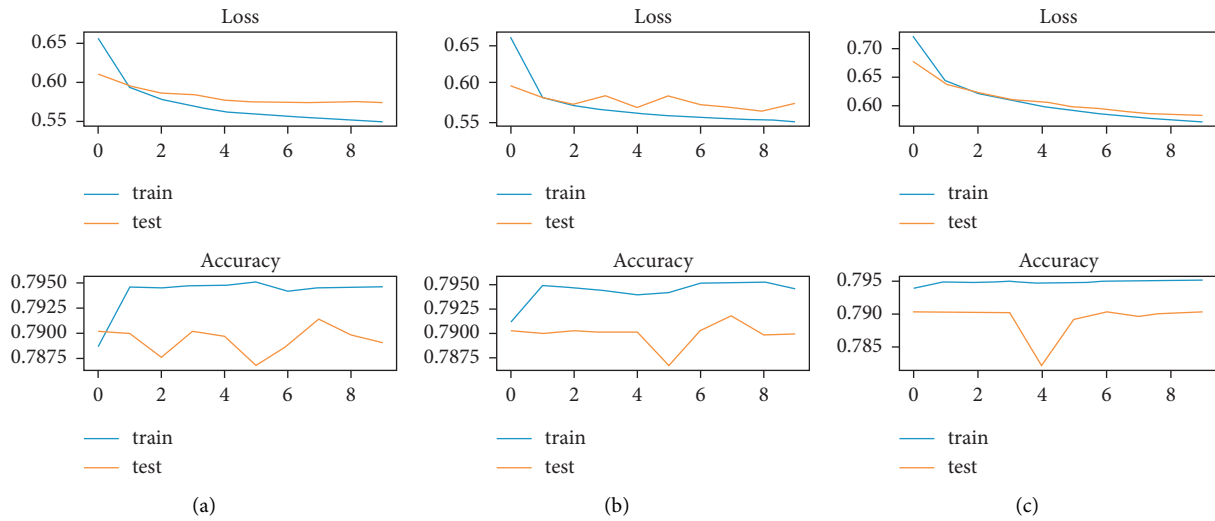


FIGURE 3: Continued.



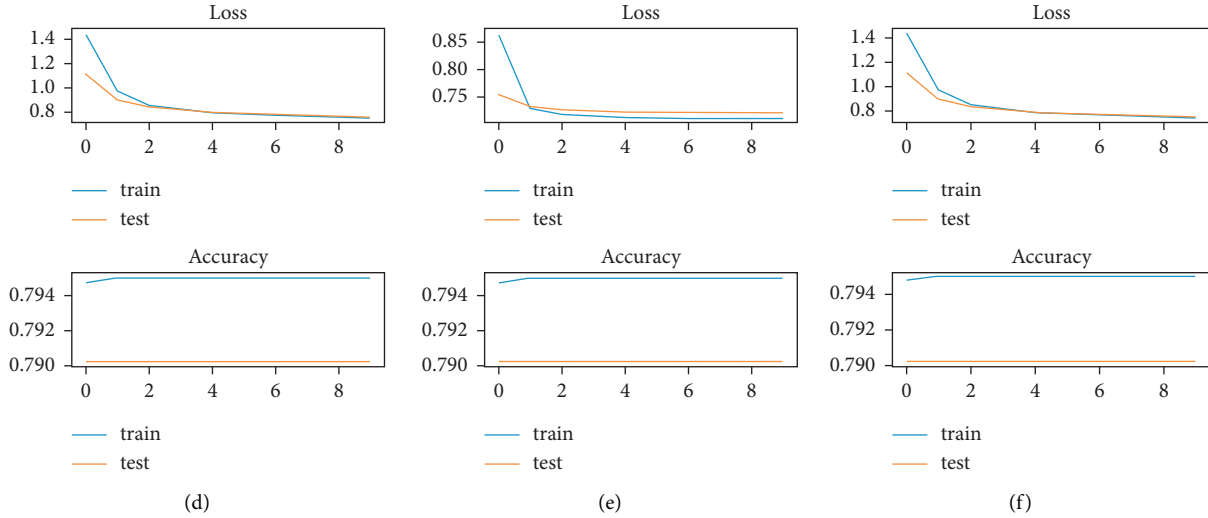


FIGURE 3: Impact of different hyperparameters on the performance of DNN. (a) Epochs and batch\_size. (b) Optimization algorithms. (c) Learning rate and momentum. (d) Weight initialization. (e) Activation function. (f) Dropout regularization.

Next, the learning curves for different machine and deep learning models were computed (Figure 4). The best parameters were obtained for KNN, NB, SVM, RF, and DNN models using cross-validation and hyperparameter tuning. These models were retrained using these best parameters. The learning curves for these models have score on x-axis and training examples on y-axis. Here, higher score means better performance of the model. These learning curves show that training and cross-validation scores are almost the same. The score for training and cross-validation remains almost the same with an increase in training examples. These models attained score in the range of 72% to 80%. The DNN model is good fit as training loss and validation loss gradually decrease and reach a point of stability. Moreover, the gap between training loss and validation loss is small, which means the model has low variance (Figure 4(e)).

Furthermore, confusion matrices were used to visualize the performance for different classes (Figure 5). Most of the classification algorithms correctly classified the data. Class 1 has large number of records in dataset, and most of the algorithms correctly classify data belonging to this class. However, the NB algorithm does not perform well for data belonging to class 3. The reason is distribution of majority class in the training dataset.

Finally, the precision-recall curves were computed (Figure 6). The precision-recall curve has precision on the y-axis and recall on the x-axis. There is trade-off between precision and recall values for different thresholds. For low recall values, the precision value is high. For recall value greater than 0.8, KNN shows a stair step area. This means that a small change in threshold reduces the precision with minor gain in recall value. DNN algorithm attains better average precision compared with other machine learning algorithms. The average precision values are different in the precision-recall curve (Figure 6) and classification algorithm

performance (Table 3). The reason is that the precision-recall curve computes the precision of one class with all other classes in the multi-class dataset.

Table 3 shows the performance of all the machine learning and deep learning algorithms for the NSCH dataset. Accuracy is one of the most used metrics to evaluate classification models. DNN exhibits better performance compared with other approaches and attained the highest accuracy result of 87% in the NSCH dataset. The previous study for this dataset attained 50% to 54% accuracy [23].

These algorithms can be evaluated using a statistical test to check whether they have the same performance. The performance comparison of all the supervised classification models together is difficult. Moreover, the quality and performance of a supervised classification model should be evaluated on independent data [39]. For this reason,  $5 \times 2$  cv paired  $t$ -test was employed to compare the performance of two models [39, 40]. In this work, a level of significance  $\alpha = 0.05$  is used for  $5 \times 2$  cv paired  $t$ -test. For comparison of performance of classification models, this method divides the dataset five times into 50% training data and 50% test data. Two models were fit on the training data and evaluated on the test data for each of the 5 iterations in  $5 \times 2$  cv paired  $t$ -test. Moreover, training data and test data are rotated to compute the performance again [40]. Table 4 shows pairwise performance comparison of classification algorithms. For all cases,  $p \leq \alpha$ . This means that the performance of these algorithms is significantly different for this dataset.

Furthermore, the Kruskal-Wallis and Friedman tests were applied to compare the performance of all models together (Table 5). Table 5 shows the resultant  $p$  value after applying the tests. The null hypothesis for these tests is that the performance of all classifiers is the same. In this case, the null hypothesis is rejected, which means that the performance of classifiers is not the same.

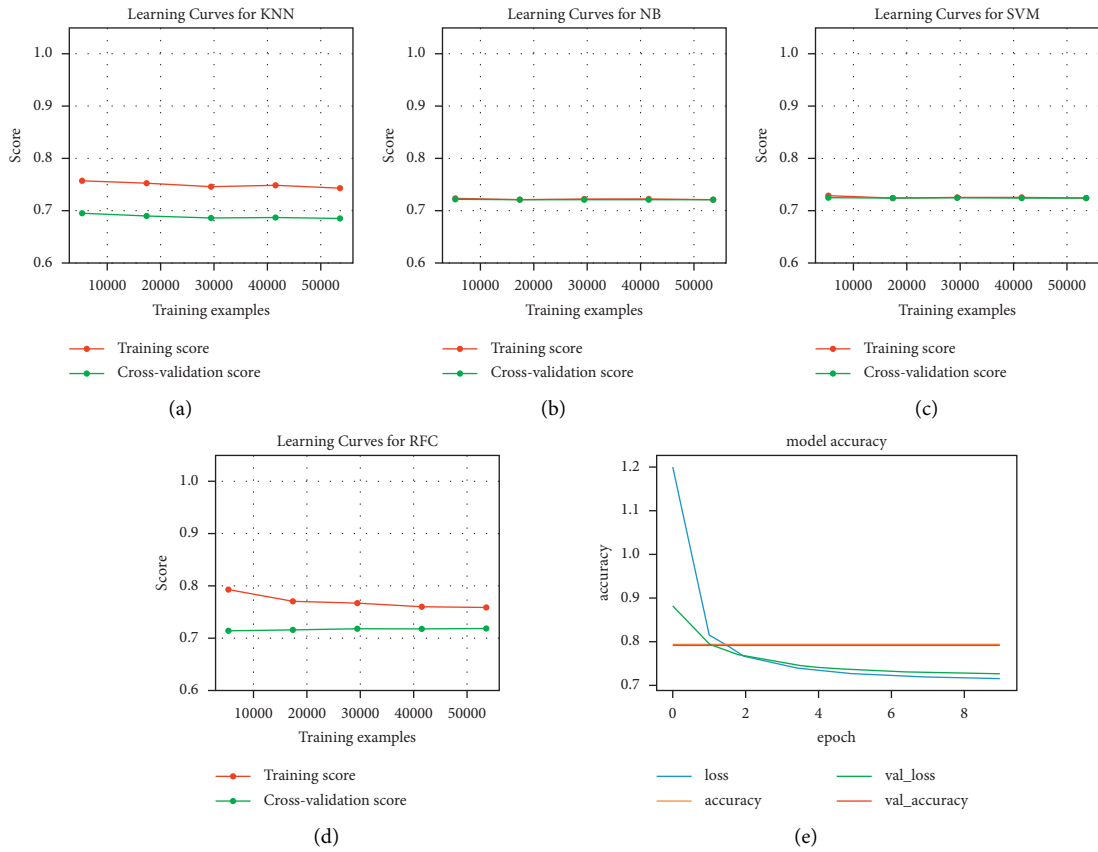


FIGURE 4: Learning curves for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.

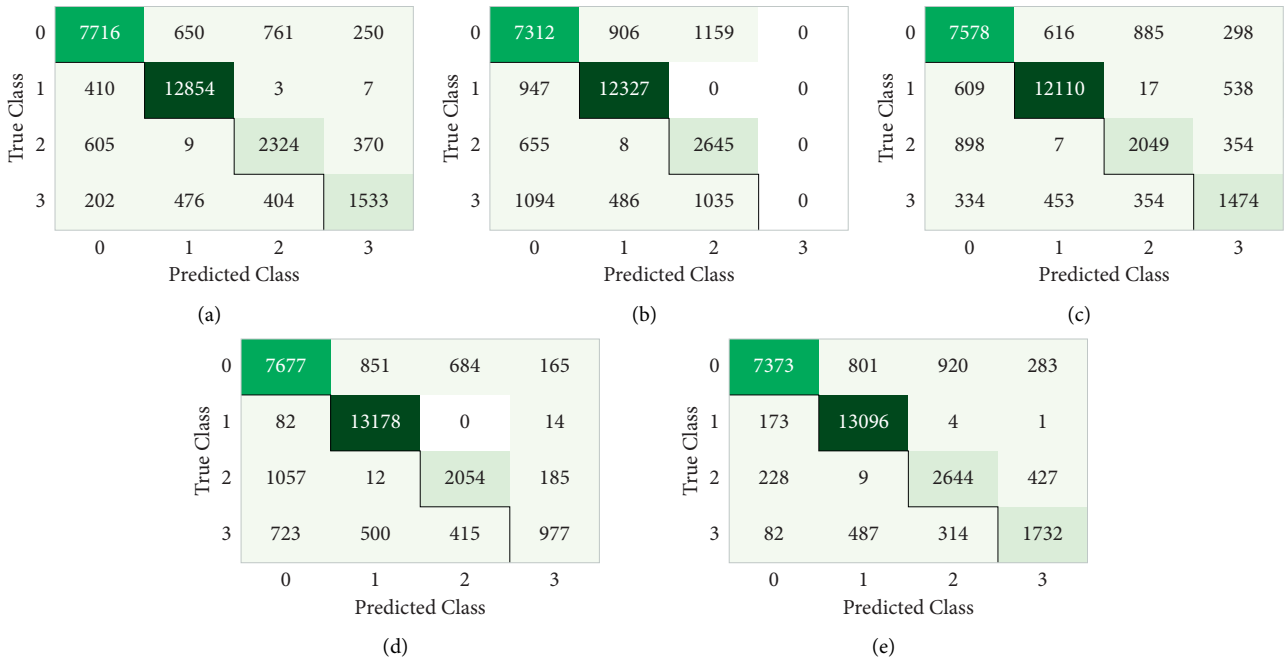


FIGURE 5: Confusion matrix for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.

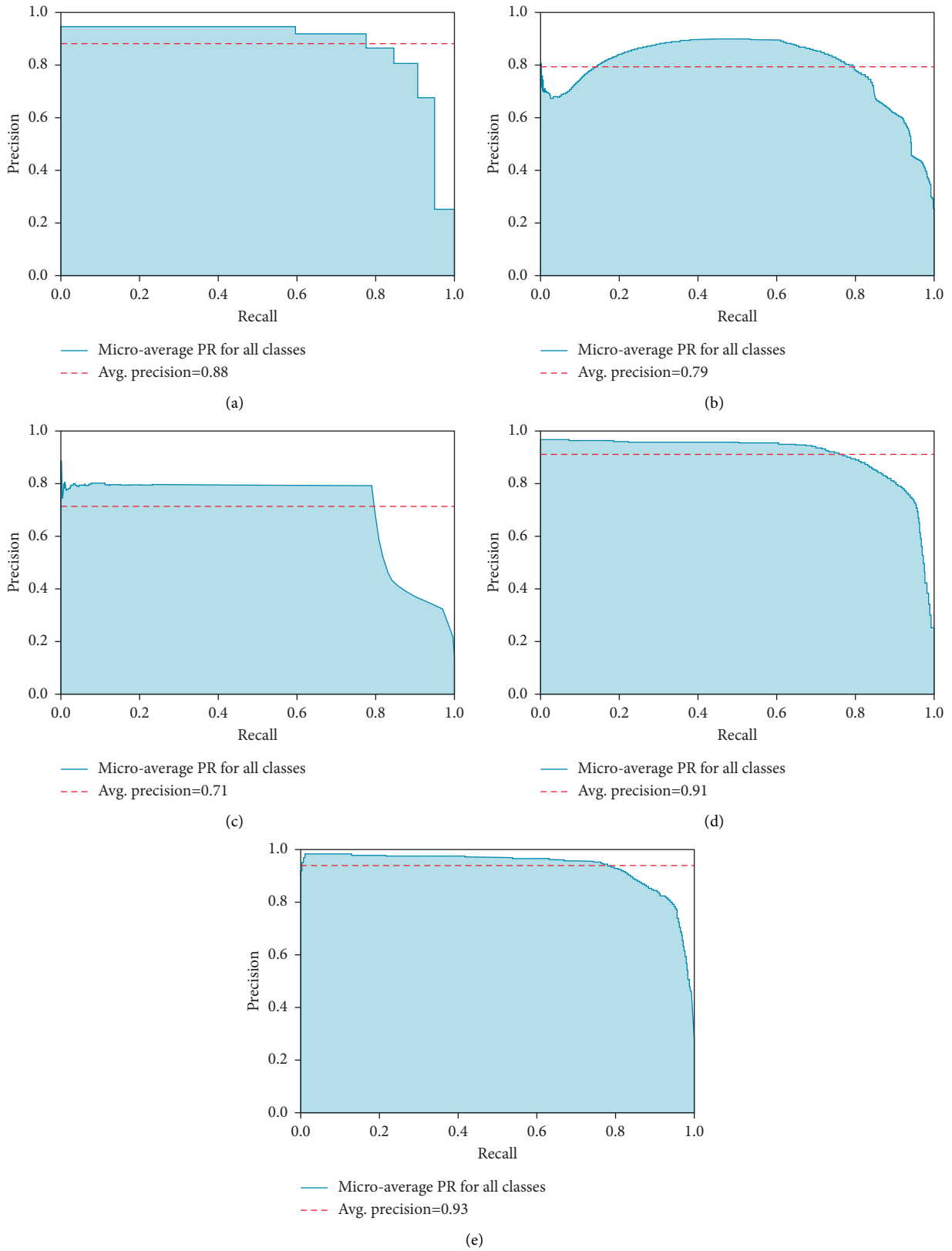


FIGURE 6: Precision-recall curve for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.

TABLE 3: Performance of the classification algorithms.

	DNN	SVM	NB	RF	KNN
Accuracy	0.871 1	0.791 2	0.780 9	0.828 1	0.817 2
Precision	0.876 1	0.815 8	0.722 8	0.830 3	0.856 6
Recall	0.807 5	0.727 8	0.629 6	0.708 2	0.773 8
F1 score	0.870 1	0.814 8	0.749 0	0.828 3	0.856 5

TABLE 4: Performance comparison of the classification algorithms using  $5 \times 2$  cv paired  $t$ -test.

	SVM	NB	RF	KNN
DNN	0.000 1	0.000 1	0.000 1	0.000 1
SVM		0.001 0	0	0
NB			0	0
RF				0.002 0

TABLE 5: Performance comparison of the classification algorithms using the Kruskal–Wallis and Friedman test.

	Kruskal–Wallis test	Friedman test
Accuracy	0.046 0	0.047 1
Precision	0.045 0	0.045 5
Recall	0.045 3	0.451
F1 score	0.046 8	0.047 9

## 5. Conclusion

The early detection of ASD can help to improve the learning capabilities. This study presents different machine and deep learning techniques to classify the severity of ASD. Different experiments were conducted using stratified  $k$ -fold cross-validation and hyperparameter tuning. The objective was to attain the best parameters for each machine and deep learning model and retrain the model using these parameters to obtain better performance. The results depict that DNN has better performance when compared with other models. In the future work, we will apply these techniques to different datasets and modalities.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.








## References

- [1] K. Shahrukh Omar, P. Mondal, N. Shahnaz Khan, R. Karim Rizvi, and N. Islam, "A machine learning approach to predict autism spectrum disorder," in *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–6, IEEE, Cox'sBazar, Bangladesh, Feb 2019.
- [2] D. Liao and Hu Lu, "Classify autism and control based on deep learning and community structure on resting-state fmri," in *Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 289–294, IEEE, Xiamen, China, Mar 2018.
- [3] G. . Murat, "A novel machine learning model to predict autism spectrum disorders risk gene," *Neural Computing & Applications*, vol. 31, no. 10, pp. 6711–6717, 2019.
- [4] S. S. Alwakeel, B. Alhalabi, H. Aggoune, and M. Alwakeel, "A machine learning based wsn system for autism activity recognition," in *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 771–776, IEEE, Miami, FL, USA, Dec 2015.
- [5] S. Selvaraj, P. Palanisamy, S. Parveen, and Monisha, "Autism spectrum disorder prediction using machine learning algorithms," in *Proceedings of the International Conference On Computational Vision and Bio Inspired Computing*, pp. 496–503, Springer, Berlin,Germany, Jan 2019.
- [6] F. Hauck and N. Kliewer, "Machine learning for autism diagnostics: applying support vector classification," in *Proceedings of the Int'l Conf. Heal. Informatics Med. Syst.*, pp. 120–123, IEEE, Yogyakarta, Indonesia, Nov 2017.
- [7] A. Demirhan, "Performance of machine learning methods in determining the autism spectrum disorder cases," *Mugla Journal of Science and Technology*, vol. 4, no. 1, pp. 79–84, 2018.
- [8] D. L. Christensen, J. Baio, D. Bilder et al., "Prevalence and characteristics of autism spectrum disorder among children aged 8 Years - autism and developmental disabilities monitoring network, 11 sites, United States, 2012," *MMWR. Surveillance Summaries*, vol. 65, no. 13, pp. 1–23, 2018.
- [9] A. Minhas, V. Vajaratkar, G. Divan et al., "Parents' perspectives on care of children with autistic spectrum disorder in South Asia - v," *International Review of Psychiatry*, vol. 27, no. 3, pp. 247–256, 2015.
- [10] D. S. Mandell, R. F. Ittenbach, S. E. Levy, and J. A. Pinto-Martin, "Disparities in diagnoses received prior to a diagnosis of autism spectrum disorder," *Journal of Autism and Developmental Disorders*, vol. 37, no. 9, pp. 1795–1802, 2007.
- [11] F. R. Volkmar and B. Reichow, "Autism in dsm-5: progress and challenges," *Molecular Autism*, vol. 4, no. 1, pp. 13–16, 2013.
- [12] A. Kandola, "Levels of autism: everything you need to know," *Med News Today*, vol. 30, no. 2, pp. 48–55, 2019.
- [13] I. El Naqa and M. J. Murphy, "What is machine learning?" in *Machine Learning in Radiation Oncology Machine Learning in Radiation Oncology*, pp. 3–11, Springer, Berlin,Germany, 2015.
- [14] M. B. Usta, K. Karabekiroglu, B. Sahin et al., "Use of machine learning methods in prediction of short-term outcome in autism spectrum disorders," *Psychiatry and Clinical Psychopharmacology*, vol. 29, no. 3, pp. 320–325, 2019.
- [15] F. Fatima, R. Talib, M. K. Hanif, and M. Awais, "A paradigm-shifting from domain-driven data mining frameworks to process-based domain-driven data mining-actionable knowledge discovery framework," *IEEE Access*, vol. 8, pp. 210763–210774, 2020.
- [16] R. Talib, M. Kashif Hanif, F. Fatima, and S. Ayesha, "A multi-agent framework for data extraction, transformation and loading in data warehouse," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.
- [17] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A systematic review of deep learning approaches to educational data mining," *Complexity*, vol. 2019, Article ID 1306039, 22 pages, 2019.

- [18] W. G. Hatcher and W. Yu, "A survey of deep learning: platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.
- [19] D. Nie, R. Trullo, J. Lian et al., "Medical image synthesis with deep convolutional adversarial networks," *IEEE Transactions on Bio-Medical Engineering*, vol. 65, no. 12, pp. 2720–2730, 2018.
- [20] I. Iswanto, K. Purwanto, W. Hastuti, A. Prabowo, and M. Y. Mustar, "Smart smoking area based on fuzzy decision tree algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, pp. 500–504, 2019.
- [21] M. Jiang and Z. Qi, "Learning visual attention to identify people with autism spectrum disorder," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3267–3276, IEEE, Venice, Italy, Oct 2017.
- [22] N. C. Dvornek, P. Ventola, K. A. Pelphrey, and J. S. Duncan, "Identifying autism from resting-state fmri using long short-term memory networks," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging*, pp. 362–370, Springer, Berlin, Germany, Sep 2017.
- [23] B. Van den Bekerom, "Using machine learning for detection of autism spectrum disorder," in *Proceedings of the . 20th Student Conf. IT*, pp. 1–7, 2017.
- [24] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi, "Identification of autism spectrum disorder using deep learning and the abide dataset," *NeuroImage: Clinical*, vol. 17, pp. 16–23, 2018.
- [25] X. A. Bi, Y. Wang, Q. Shu, Q. Sun, and Q. Xu, "Classification of autism spectrum disorder using random support vector machine cluster," *Frontiers in Genetics*, vol. 9, no. 18, p. 18, 2018.
- [26] O. Altay and M. Ulas, "Prediction of the autism spectrum disorder diagnosis with linear discriminant analysis classifier and k-nearest neighbor in children," in *Proceedings of the 2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–4, IEEE, Antalya, Turkey, March 2018.
- [27] N. Mohammadian Rad and C. Furlanello, "Applying deep learning to stereotypical motor movement detection in autism spectrum disorders," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 1235–1242, IEEE, Barcelona, Spain, Dec 2016.
- [28] Y. Kong, J. Gao, Y. Xu, Y. Pan, J. Wang, and J. Liu, "Classification of autism spectrum disorder by combining brain connectivity and deep neural network classifier," *Neuro-computing*, vol. 324, pp. 63–68, 2019.
- [29] T. Eslami and F. Saeed, "Auto-asd-network: a technique based on deep learning and support vector machines for diagnosing autism spectrum disorder using fmri data," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 646–651, ACM, New York, NY, United States, Sep 2019.
- [30] S. M. Wilson and A. S. Rajan, "Analysis of autism spectrum disorder using deep learning and the abide dataset," *International Research Journal of Engineering and Technology*, vol. 6, no. 3, 2019.
- [31] V. Pream, M. S. Sudha, and M. S. Vijaya, "Machine learning-based model for identification of syndromic autism spectrum disorder," in *Integrated Intelligent Computing, Communication and Security Integrated Intelligent Computing, Communication and Security*, pp. 141–148, Springer, Berlin, Germany, 2019.
- [32] I. M. Nasser, M. Al-Shawwa, and S. S. Abu-Naser, "Artificial neural network for diagnose autism spectrum disorder," *International Journal of Academic Information Systems Research*, vol. 3, no. 2, pp. 27–32, 2019.
- [33] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, Inc, California, U.S, 2016.
- [34] M. K. Hanif, S. Ayesha, and R. Talib, "Dimension reduction techniques," in *Big Data, IoT, and Machine Learning Big Data, IoT, and Machine Learning*, pp. 37–50, CRC Press, Boca Raton, Florida, United States, 2020.
- [35] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.
- [36] A. T. Azar, H. I. Elshazly, A. E. Hassanien, and A. M. Elkorany, "A random forest classifier for lymph diseases," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 2, pp. 465–473, 2014.
- [37] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315, IEEE, New Delhi, India, March 2016.
- [38] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, Article ID e0118432, 2015.
- [39] S. Guzman, I. Inza, and J. A. Lozano, "Dealing with the evaluation of supervised classification algorithms," *Artificial Intelligence Review*, vol. 44, no. 4, pp. 467–508, 2015.
- [40] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

## Research Article

# Monitoring Population Phenology of Asian Citrus Psyllid Using Deep Learning

Maria Bibi <sup>1</sup>, Muhammad Kashif Hanif <sup>1</sup>, Muhammad Umer Sarwar <sup>1</sup>,  
Muhammad Irfan Khan <sup>1</sup>, Shouket Zaman Khan <sup>2</sup>, Casper Shikali Shivachi <sup>3</sup>,  
and Asad Anees <sup>4,5</sup>

<sup>1</sup>Department of Computer Science, Government College University, Faisalabad, 38000, Pakistan

<sup>2</sup>Department of Entomology, University of Agriculture Faisalabad Sub-Campus Burewala, Vehari 61010, Pakistan

<sup>3</sup>South Eastern Kenya University, Kitui, Kenya

<sup>4</sup>Cardiovascular Engineering Inc, 1 Edgewater Drive, Norwood, MA 02062, USA

<sup>5</sup>Department of Mathematics, Clausthal University of Technology, Erzstraße 1, D-38678 Clausthal-Zellerfeld, Germany

Correspondence should be addressed to Casper Shikali Shivachi; [cshikali@seku.ac.ke](mailto:cshikali@seku.ac.ke)

Received 28 May 2021; Revised 3 December 2021; Accepted 10 December 2021; Published 31 December 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Maria Bibi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Asian citrus psyllid, *Diaphorina citri* Kuwayama (Liviidae: Hemiptera) is a menacing and notorious pest of citrus plants. It vectors a phloem vessel-dwelling bacterium *Candidatus Liberibacter asiaticus*, which is a causative pathogen of the serious citrus disease known as Huanglongbing. Huanglongbing disease is a major bottleneck in the export of citrus fruits from Pakistan. It is being responsible for huge citrus economic losses globally. In the current study, several prediction models were developed based on regression algorithms of machine learning to monitor different phenological stages of Asian citrus psyllid to predict its population about different abiotic variables (average maximum temperature, average minimum temperature, average weekly temperature, average weekly relative humidity, and average weekly rainfall) and biotic variable (host plant phenological patterns) in citrus-growing regions of Pakistan. The pest prediction models can be used for proper applications of pesticides only when needed for reducing the environmental and cost impacts of pesticides. Pearson's correlation analysis was performed to find the relationship between different predictor (abiotic and biotic) variables and pest infestation rate on citrus plants. Multiple linear regression, random forest regressor, and deep neural network approaches were compared to predict population dynamics of Asian citrus psyllid. In comparison with other regression techniques, a deep neural network-based prediction model resulted in the least root mean squared error values while predicting egg, nymph, and adult populations.

## 1. Introduction

The citrus greening disease, which is also known as Huanglongbing, is a severe affliction to citrus plants that causes significant losses to the citrus economy, caused by a phloem-dwelling bacterium *Candidatus Liberibacter asiaticus*. This incurable and economically damaging disease is transmitted by infection of a sucking pest, Asian citrus psyllid (ACP), which is a transmitting vector agent of *Candidatus Liberibacter asiaticus* phloem-inhabiting bacterium. Effective management of ACP is crucial for preventing the losses caused by Huanglongbing and ACP

complex [1]. There are three types of bacteria, *Candidatus Liberibacter asiaticus*, *Candidatus Liberibacter americanus*, and *Candidatus Liberibacter africanus*, associated with the spreading of Huanglongbing throughout the citrus-growing areas worldwide [2]. Huanglongbing is a vector-borne disease, and its causative agents grow and transmit through ACP [3].

Psyllid population growth rate is directly associated with the flush phenology (biotic factor) of host plants because female adults are only able to lay eggs on young, tender, and succulent plant leaves, and resultantly, the nymphs are more likely to hatch and grow during the season of abundant flush

growth on citrus plants. Availability of flush growth and optimum meteorological conditions leads to large infestations of ACP on citrus plants. Different meteorological conditions such as relative humidity, temperature, and rainfall in the study area are important factors influencing the existence of ACP stages in the field. Citrus host plant phenological characteristics have the tremendous ability to influence psyllid biology, survival, and resultant pest outbreaks under optimum environmental conditions [4–6].

Entomologists carried out various trials previously regarding population change of psyllid over time and these types of studies are significantly useful in future pest prediction and forecasting. Keeping in view, the significant effect of weather factors on insect populations and natural enemies' existence can be correlated with the changing pest-natural enemies' populations, which will better explain the density curves of both psyllid and associated insect enemies [7–9].

The science of ecology in natural sciences studies the mutual relationships among the biotic and abiotic components of the ecosystem to understand ecological processes and make predictions about future trends. Machine learning (ML) techniques have advantages over typical statistical approaches because these techniques are modeling ecological processes in a better way by allowing better decision-making and informed actions in the real world without (or with minimal) human involvement. ML techniques not only provide a flexible framework for the execution of data-driven tasks but also help for the integration of expert knowledge into the system [10].

The abilities of ML algorithms to model high-dimensional and nonlinear data with complex interactions, missing values, and identification of complicated structures from more complex datasets are defeating typical statistical approaches in population modeling [11]. Recent advancements in ML are deep learning (DL) techniques [12]. The DL approaches have the potential of automated feature learning, and the complex structures allow to solve more complex problems faster and accurately and reduce error while modeling regression problems and increasing accuracy in classification problems analysis in the presence of large dataset availability [13].

Machine learning techniques have been used in several studies related to pest's population prediction such as modeling the population dynamics of paddy stem borer (*Scirpophaga incertulas*) [14], the population density of *Scirtothrips dorsalis* Hood [15], risk of Melon thrip (*T. palmi*), diamondback moth (*P. xylostella*) [16], fluctuating trends of *Dendrolimus superans* population [17], population phenology of Black Planthopper (*Nilaparvata lugens*) [18], population occurrence of mosquitoes in correlation with different socioeconomic factors and landscape variables [19], *Prostephanus truncatus* infestation and accompanying damages to maize grain storage in correlation with abiotic factors [20], fluctuating trends of cotton's pest population (*Thrips tabaci* Linde) [21], and the effect of temperature and rainfall monitored by Watts and Worner [22] to the establishment of mealybug (*Planococcus citri*) and aphid (*Myzus persicae*, *Aphis gossypii*, *Eriosoma lanigerum*, and *Brevicoryne brassicae*).

The random forest regressor (RFR) model has been employed by researchers in various fields related to prediction and classification problems; for example, the authors of [23–25] used this ensemble learning approach prediction of dengue, citrus flatid planthopper, and sunn pest's nymphal stage, respectively. For earlier prediction of pest's risk, the multiple linear regression (MLR) model was adopted by numerous researchers. The authors of [26, 27] implemented MLR approach to model potential risk of black planthopper and oriental fruit fly (*Bactrocera dorsalis*) population, respectively.

Deep neural network (DNN) has broader applicability in the following agricultural domains in general. Chlingaryan et al. [10] used DNN for estimation of crop yield prediction. The authors of [28, 29] deployed DNN for the prediction of soil moisture contents, and Scher [30] used DNN for weather conditions' prediction. DNN has been also used for land cover and crop type classification, image identification, and classification of plants and weeds [31–34]. Rammer and Seidl [35] deployed DNN and RFR to predict damages that will occur in the future through bark beetle population outbreak using pest's historical data and concluded that DNN has the tremendous power to model bark beetle outbreaks' dynamics and other ecological prediction problems. This review of previous studies shows a research gap concerning the use of ML and DL models in the prediction of phenological stages of insects-pests. Keeping in view the literatures, the present study was conducted to (a) make comparative analysis of different machine and deep learning techniques to predict phenological stages of ACP and (b) monitor the cumulative effect of different weather factors and host plant phenology on psyllid phenological stages.

In present research, we made a comparative analysis of different regression-based approaches, i.e., DNN, MLR, and RFR models, to predict the population of different ACP-phenological stages using environmental variables and host-plant phenology variables as independent variables. By using the abovementioned regression approaches, we evaluated the combined effect of different independent variables on three ACP-phenological stages, i.e., eggs, nymphs, and adults separately.

## 2. Materials and Methods

**2.1. Study Site and Data Collection.** For data collection to monitor population dynamics of Asian Citrus Psyllid, two study locations, Square No.9 (31° 25'50.4" N; 73° 03'40.2" E; elevation 190 m) and PARS (N31°23'35.20"; E73°01'27.0"; elevation 210 m), were selected from University of Agriculture Faisalabad (UAF), Pakistan. From both study locations, 15 trees of two citrus species, sweet orange (*Citrus sinensis sensu lato*), and kinnow (*Citrus reticulata*) were randomly selected and tagged properly to monitor population fluctuations of ACP on weekly basis from a time course, 26 March 2011 to 20 April 2013. A detailed description of both study sites and ACP-phenological stages' data collection is given in [36]. We used datasets spanning 25 months to reduce experimental errors and to confirm the psyllid response in different weather conditions in different

seasons. If psyllid population increases in spring, then we repeated this for next spring to see if psyllid responds similarly.

Meteorological data during the experimental period regarding daily temperature (maximum, minimum, and average temperature), rainfall, and relative humidity on daily basis were documented from the meteorological observatory of the Crop Physiology (CP) department in the Agricultural faculty of UAF. The effect of meteorological (abiotic) factors was also monitored by calculating the percentage of branches infected with different life stages of ACP, i.e., eggs, nymphs, and adults individually and collectively.

**2.2. Model Development.** In this study, three models, i.e., RFR, DNN, and MLR, were employed to model population dynamics of ACP. The Google Collaboratory was used, and it is an effective cloud computing environment for developing python-based applications.

**2.3. Random Forest Regressor.** Random forest is an ensemble learning approach proposed in [36] and used both for regression and classification problems' analysis [37]. Each random forest is composed of a specified number of decision trees, and each decision tree trains on samples of training data by following a randomized approach called Bagging (Bootstrap aggregating). Random forest regressor returns the output in the form of the mean value calculated from the results of the prediction of all decision trees. It minimizes the effect of model overfitting by introducing randomness in variables and data instances' selection. RFRs have the capability of efficient training and testing. As each prediction is made by random forests (RFs), a built-in mechanism is usually found in RF to calculate test errors, e.g., root mean squared error (RMSE), mean absolute error (MAE), and confidence [38]. Hyperparameter tuning is an important step in the development of models. In order to train the RF, we set the value of `n_estimators` (number of decision trees) as 20 and `random_state=42`, while keeping other hyperparameters with their default values. We used RMSE as a loss function to calculate test errors. The mathematical formulation of RMSE is given as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}, \quad (1)$$

where  $y_i$  and  $\hat{y}_i$  are the actual and predicted values, respectively, and  $n$  is the number of observations. For evaluating the accuracies of the forecasting models, RMSE is a common indicator used in regression problems analysis [39].

**2.4. Deep Neural Network.** Artificial neural networks (ANNs) were developed in the middle of the nineteenth century. The term "deep learning" refers to training of deeper and larger ANNs. Here, deeper and larger are

concerned with more layers and more neurons as compared to ANNs [12]. DNNs are the result of recently developed improved algorithms which are optimizing the weights of the connections [40].

For predicting the population phenology of ACP, we developed a DNN comprised of one input layer consisting of six input neurons/nodes and two hidden layers with six and eight neurons, respectively. The activation function and optimizer used are ReLu and Adam, respectively. DNN architecture also consisted of one output layer with a single neuron to predict each ACP life stage, i.e., eggs, nymphs, and adults separately. We used dense layers to make the model more stable for prediction (Figure 1).

**2.5. Multiple Linear Regression.** To quantify the relationship between different input variables (Average Max Temp, Average Min Temp, Average Weekly Temp, Average Weekly RH, Average Weekly Rainfall, and Branches with Flush) and ACP phenological stages, Pearson correlation analysis was performed. We used Pearson correlation coefficient ( $R$ ) values as criteria to select suitable input variables for developing the MLR model. The MLR model was deployed with a stepwise selection method to monitor the fluctuating trends of ACP population occurrence. Equation (2) for MLR is given below:

$$y = \beta_0 + \beta_{1x_1} + \beta_{2x_2} + \dots + \beta_{kx_k} + \varepsilon, \quad (2)$$

where  $y$  refers to predicted or response variable. The range of predictors or controlled variables starts from  $x_1$  to  $x_k$ .  $\beta_0$  is called the intercept or constant variable and  $\beta_1$  to  $\beta_k$  are the regression coefficients of controlled variables.  $\varepsilon$  is fitted or residual error to indicate the uncertainty in the model [41]. We normalized the dataset before fitting the MLR model on the respective dataset to monitor the population growth of ACP in relation to host plant phenology and different abiotic factors.

**2.6. Feature Importance.** To measure the importance of different input variables for predicting different ACP-phenological stages, we used a feature importance graph using RFR. The feature importance graphs for eggs and nymphs reveal that "branches with flush" is one of the most important variables for ACP egg and nymph growth (Figure 2).

### 3. Results

**3.1. Effect of Abiotic Factors on Population Fluctuations of ACP.** To study the impact of various abiotic factors on the population phenology of *D. citri* during the experimental time duration of 25 months on an individual and cumulative basis for different citrus species, correlation coefficient values by using Minitab software were calculated (Table 1). In the case of the ACP-eggs' population, host plant flush growth patterns and average weekly relative humidity have a significant and positive relationship with ACP-eggs' production and growth as  $R=0.44$  and  $0.247$  and  $p \leq 0.05$ ). ACP-nymphs' growth was found to be positively correlated with input variable branches with flush as  $R=0.48$  and



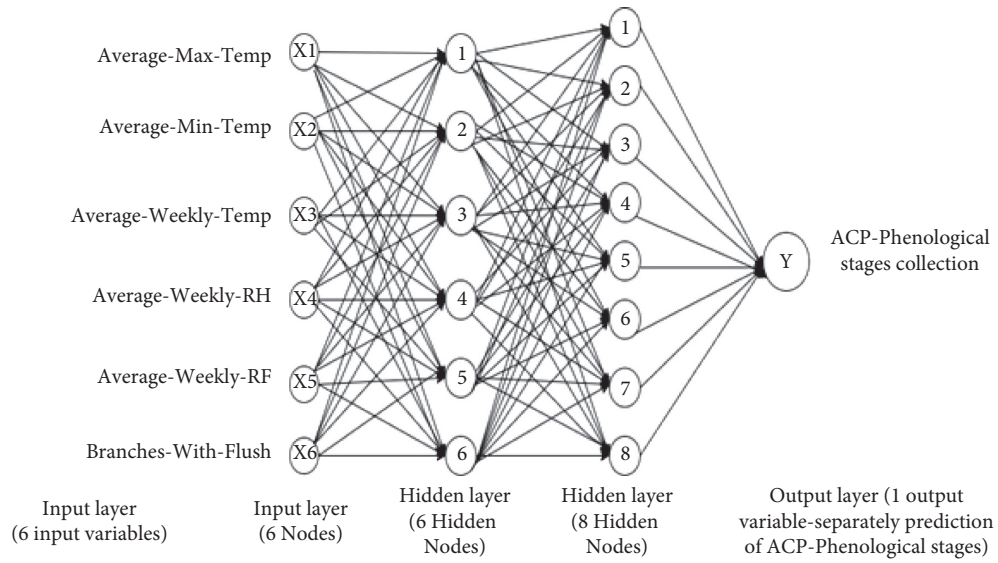


FIGURE 1: Deep neural network architecture for predicting ACP population dynamics.

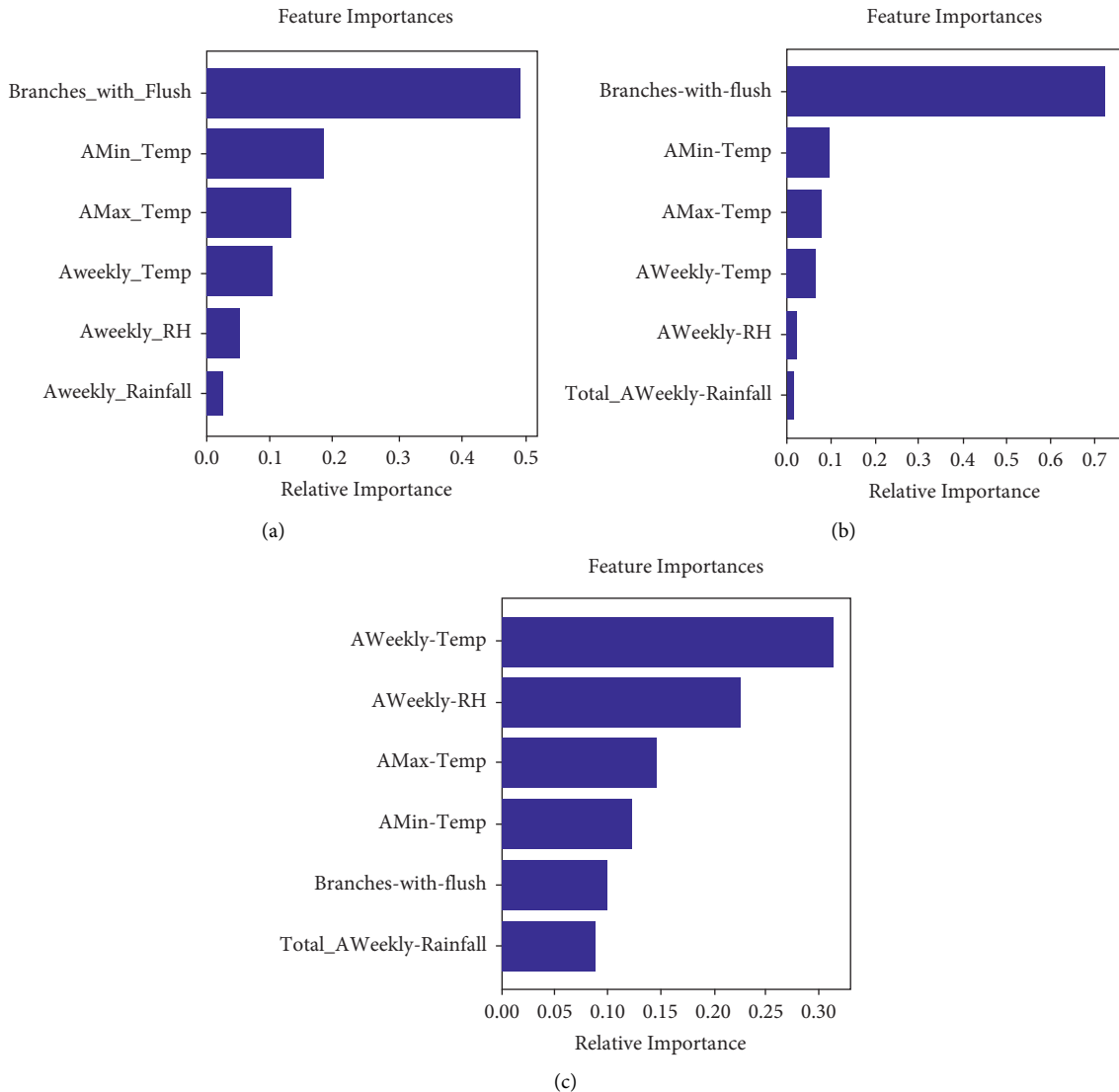


FIGURE 2: Feature importance graphs for different ACP phenological stages. (a) Input variables importance for ACP-egg's growth. (b) Input variables importance for ACP-nymph's growth. (c) Input variables importance for ACP-adult's growth.

TABLE 1: Pearson’s correlation coefficient values between different ACP phenological stages and different abiotic variables and host plant flush growth patterns.

Predictor variables	Phonological stage	Correlation coefficient (R)	Sig.
Average Max Temp	Eggs	-0.052	0.663
Average Min Temp		0.011	0.929
Average Weekly Temp		-0.019	0.871
Average Weekly RH		0.247	0.035**
Average Weekly Rainfall		0.121	0.308
Branches with flush		0.44	0.000**
Average Max Temp	Nymphs	0.065	0.583
Average Min Temp		0.043	0.715
Average Weekly Temp		0.053	0.654
Average Weekly RH		-0.028	0.817
Average Weekly Rainfall		-0.036	0.762
Branches with flush		0.48	0.000**
Average Max Temp	Adults	0.233	0.047**
Average Min Temp		0.25	0.033**
Average Weekly Temp		0.244	0.038**
Average Weekly RH		-0.013	0.914
Average Weekly Rainfall		0.051	0.667
Branches with flush		-0.053	0.655

$p \leq 0.05$ . Average weekly rainfall and relative humidity were nonsignificant and negatively correlated with ACP-nymphs’ abundance.

It is clear from the results that average minimum temperature, average maximum temperature, and average weekly temperature showed positive and significant impact on the population of ACP-adults ( $R = 0.233, 0.25, \text{ and } 0.244$  and  $p \leq 0.05$ ). Cumulatively, from the time course of March 2011 to April 2013, rainfall and relative humidity exerted a significant but negative impact on the ACP population. Meanwhile, all three temperatures exerted a positive but not nonsignificant correlation with population of ACP (Table 1).

**3.2. Comparison of Different Regression Approaches to Predict ACP-Eggs’ Population.** To predict ACP-eggs’ population, all three employed models were fitted using training data. We experimented with these regression-based approaches on eight types of datasets (Figure 3). Figure 3 shows a comparison of actual and predicted values. We have categorized models from best to worst in the context of their performance in ACP-eggs’ population. In most cases, the DNN model resulted in the least RMSE value of 0.63925 while predicting the ACP-eggs’ population. The RMSE value was computed by taking the mean of RMSE values obtained from results of eight datasets. The RFR model was the other best regression approach which resulted in the second least RMSE value of 0.70375. RFR is an ensemble method which is much efficient in extracting meaningful information from the given data. It was found to be true in previous studies [18, 35]. The MLR model resulted in the RMSE value of 0.7935 as it could not perform well in comparison with other approaches deployed for ACP-eggs’ population prediction. These findings are consistent with result of [19, 42, 43].

In the case of ACP-nymphs’ population prediction, the DNN model performed better when compared with the

other competitive approaches (Figure 4). DNN has the least RMSE value. Before the training of a prediction model, hyperparameter tuning was performed to attain the best parameters for each model. Then, models were retrained using these best parameters to obtain minimum the loss function’s values. The residuals calculated by DNN, RFR, and MLR models were 1.1875, 1.38775, and 1.2715, respectively (Table 2).

ACP-adult stage considers to be more threatening for all ornamental and citrus plants. Timely identification and removal from citrus cultivars is a matter of great interest for citrus growers. While predicting the ACP-adults’ population in relation to different abiotic variables (Table 1) and flush growth patterns, the DNN model resulted in an RMSE value of 3.6776 which was the least RMSE value as compared to residuals computed by RFR and MLR models (Figure 5). RFR and MLR models resulted in 6.0553 and 8.6883 residuals while predicting ACP-adults’ population’s fluctuating trends, respectively (Table 2).

## 4. Discussion and Conclusion

Pest’s population prediction can be used as a tool for area-wide integrated pest management programs as it will help to reduce the applications of agrochemicals in fields [27]. Different abiotic factors can be used as independent variables for building a pest’s population prediction model [44]. Along with abiotic factors, there are also some biotic factors that can be used for predicting pest population abundance, e.g., host plant phenology [27, 45]. It was found that, during seasons of abundant flush growth, more infestation of ACP-eggs and ACP-nymphs were observed in citrus orchards, and the same effects were observed in [1, 8, 46]. Proper pest management strategies will help to conserve psyllid-natural enemies by minimizing pesticides applications in fields so that they can play their role as a biocontrol agent against ACP effectively. Optimum climatic conditions and host plant phenological

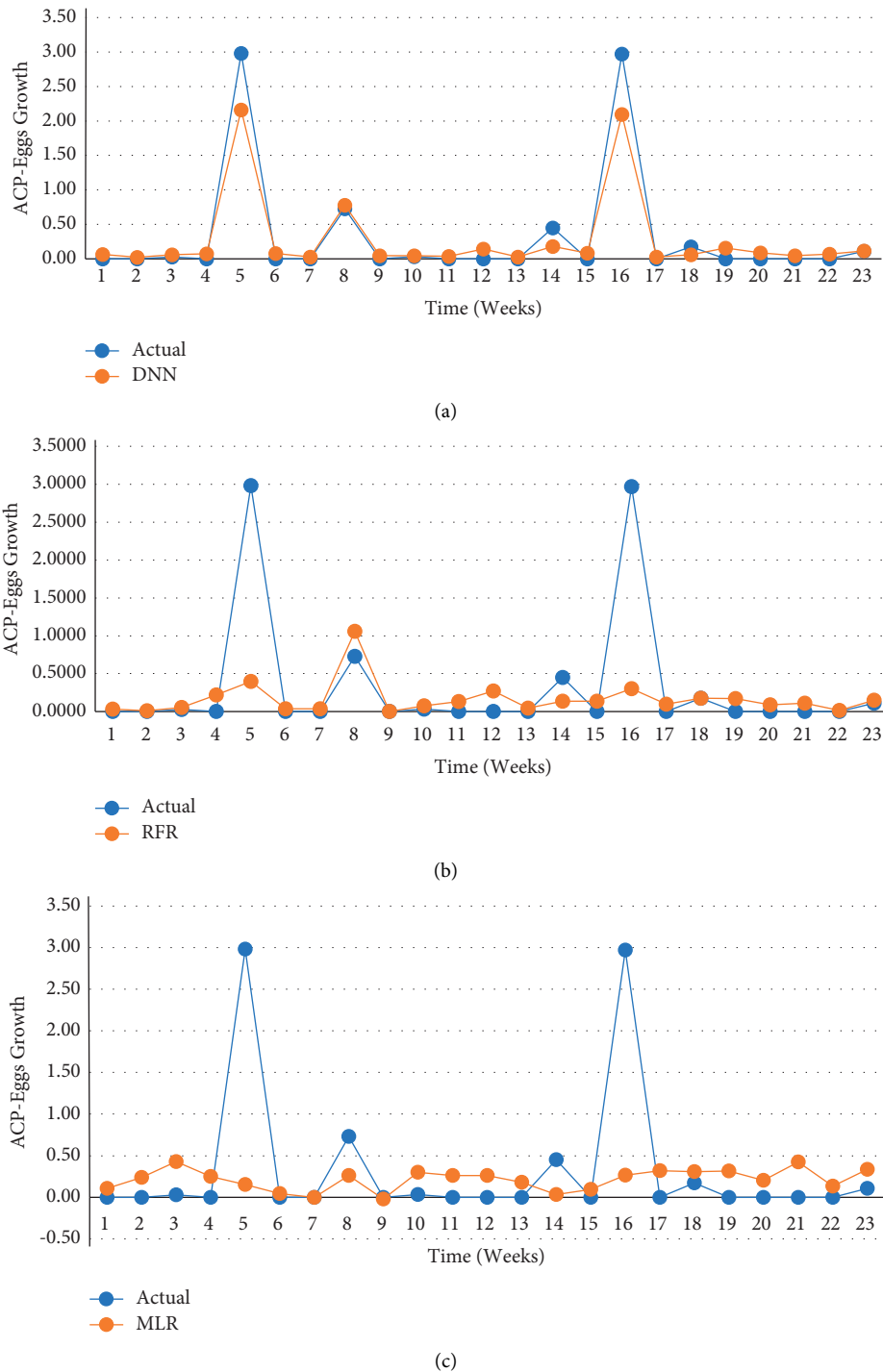


FIGURE 3: A comparison of actual and predicted ACP-eggs' values using different models. (a) DNN, (b) RFR, and (c) MLR comparison of different regression approaches to predict ACP-nymphs' population.

patterns have a great impact on ACP's survival, biology, and resultant pest abundance. The findings of this work are consistent with studies of [4, 5, 47]. ACP's population was found to decrease significantly with rainfall and relative humidity and increased with temperature. The ACP-adults' population was seen at its peak from March to April and September to October where maximum ACP-adults' population was observed in former study duration.

DNN is an appropriate choice for modeling the ACP population dynamics prediction problems as it has the potential to model complex data [35]. DL, an emerging and powerful evolution in ML, can become a powerful tool for ecologists because of its quantitative and predictive nature [48–50]. Because of generalizability of DL algorithms, they are competent models for prediction problems specifically in ecology and generally in all domains of research related to

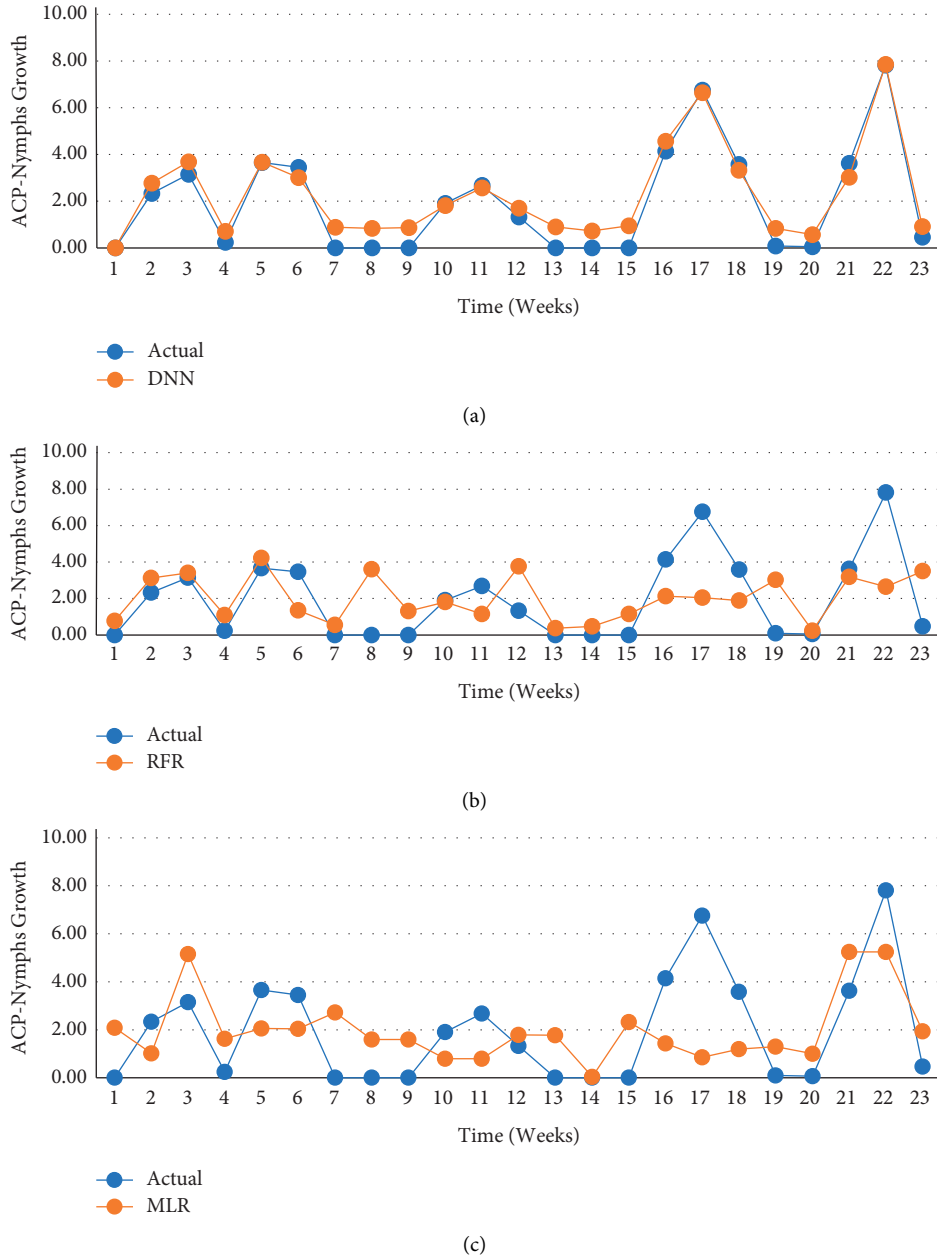


FIGURE 4: A comparison of actual and predicted ACP-nymphs' values using different models. (a) DNN, (b) RFR, and (c) MLR comparison of different regression approaches to predict ACP-adults' population.

TABLE 2: A comparison of RMSE values computed by different regression techniques for ACP-egg, ACP-nymph, and ACP-adult populations.

Models	RMSE-eggs	RMSE-nymphs	RMSE-adults
RFR	0.7037	1.3877	6.0553
MLR	0.7935	1.2715	8.6883
DNN	0.63925	1.1875	3.6776

forecasting problems. It was concluded here that DNN outperformed other classical and statistical regression techniques while modeling ACP population fluctuating trends, and we can deploy it in the future for modeling complex forecasting problems. The RFR techniques also

performed better than the statistical model MLR while predicting ACP-egg and ACP-adult population. According to [18], RFR can be more robust for prediction by acting upon some factors such as proper adjustment of hyper-parameter values and larger datasets.

In this study, various regression-based models, ranging from classical regression to deep learning-based regression, were employed for predicting the population dynamics of ACP. Current study compared predicting of the performance of various models by comparing and evaluating their resultant RMSE values. Different input variables, i.e., Average Max Temp, Average Min Temp, Average Weekly Temp, Average Weekly RH, Average Weekly Rainfall, and

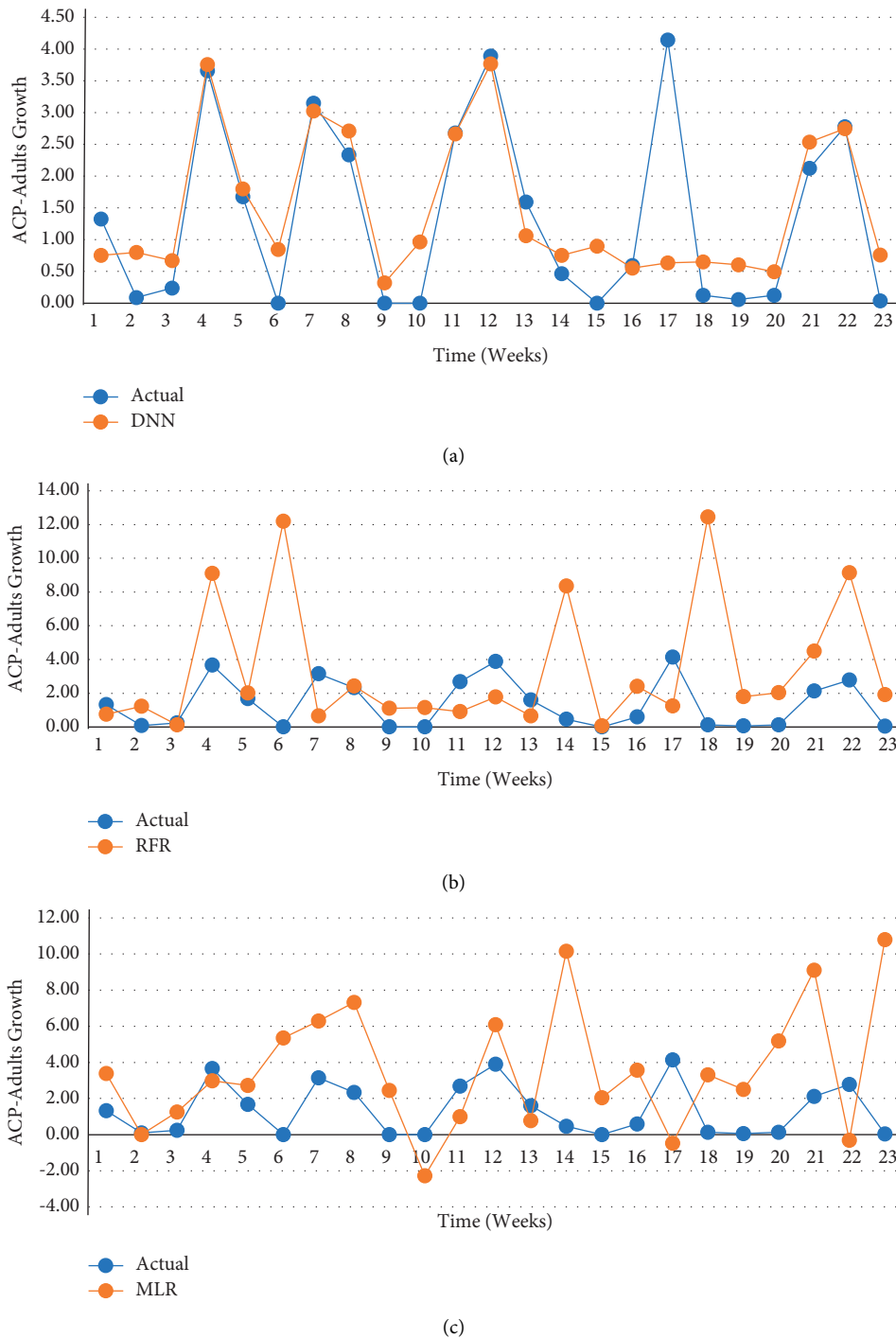


FIGURE 5: A comparison of actual and predicted ACP-adults' values computed by different regression techniques. (a) DNN. (b) RFR. (c) MLR.

Branches with flush, were used in present regression-based models. The key findings of this research can be summarized as follows. (1) The DNN model with differently tuned hyperparameters (Input, hidden, output layers, activation functions, and optimizer) is best suited for predicting population phenology of ACP. (2) A comparison of RMSE values computed by different regression-based models, configurations are also crucial. (5) The model which resulted in the smallest mean-RMSE value for the

model time-series forecasting problems. (3) The RFR model was another effective regression-based model and a good choice for predicting ACP-population dynamics as it resulted in the second least RMSE values for different ACP-phenological stages' population prediction. (4) For reliable predictions and optimization of different regression-based models, configurations are also crucial. (5) The model which resulted in the smallest mean-RMSE value for the

corresponding ACP-phenological stage was considered as the best prediction model.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

## References

- [1] D. G. Hall, M. L. Richardson, E.-D. Ammar, and S. E. Halbert, "Asian citrus psyllid, *Diaphorina citri*, vector of citrus huanglongbing disease," *Entomologia Experimentalis et Applicata*, vol. 146, no. 2, pp. 207–223, 2013.
- [2] J. M. Bove, "Huanglongbing: a destructive, newly emerging, century-old disease of citrus," *Journal of Plant Pathology*, vol. 88, pp. 7–37, 2006.
- [3] L. B. Kumagai, C. S. LeVesque, C. L. Blomquist et al., "First report of *Candidatus Liberibacter asiaticus* associated with citrus huanglongbing in California," *Plant Disease*, vol. 97, no. 2, p. 283, 2013.
- [4] F. L. Russell and S. M. Louda, "Phenological synchrony affects interaction strength of an exotic weevil with *Platte* thistle, a native host plant," *Oecologia*, vol. 139, no. 4, pp. 525–534, 2004.
- [5] S. Mopper, "Adaptive genetic structure in phytophagous insect populations," *Trends in Ecology & Evolution*, vol. 11, no. 6, pp. 235–238, 1996.
- [6] Y. H. Liu and J. H. Tsai, "Effects of temperature on biology and life table parameters of the Asian citrus psyllid, *Diaphorina citri* Kuwayama (Homoptera: psyllidae)," *Annals of Applied Biology*, vol. 137, no. 3, pp. 201–206, 2000.
- [7] J. R. Stedinger, C. A. Shoemaker, and R. F. Tenga, "A stochastic model of insect phenology for a population with spatially variable development rates," *Biometrics*, vol. 41, no. 3, pp. 691–701, 1985.
- [8] M. S. Hoddle, C. D. Hoddle, S. V. Triapitsyn, S. Z. Khan, and M. J. Arif, "How many primary parasitoid species attack nymphs of *Diaphorina citri* (Hemiptera: Liviidae) in Punjab, Pakistan?" *Florida Entomologist*, vol. 97, no. 4, pp. 1825–1828, 2014.
- [9] Q. Zeb, I. Khan, M. Inayatullah, P. Agriculture, Y. Hayat, and S. Ahmad-ur-Rehman, "Population dynamics of citrus whiteflies, aphids, citrus psylla leaf miner and their biocontrol agents in Khyber Pakhtunkhwa," *Sarhad Journal of Agriculture*, vol. 27, no. 3, pp. 451–457, 2011.
- [10] A. Chlingaryan, S. Sukkariyah, and B. Whelan, "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: a review," *Computers and Electronics in Agriculture*, vol. 151, pp. 61–69, 2018.
- [11] J. D. Olden, J. J. Lawler, and N. L. Poff, "Machine learning methods without tears: a primer for ecologists," *The Quarterly Review of Biology*, vol. 83, no. 2, pp. 171–193, 2008.
- [12] M. A. Nielsen, *Neural Networks and Deep Learning*. Vol. 2018, Determination press, San Francisco, CA, USA, 2015.
- [13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [14] L.-n. Yang, L. Peng, L.-m. Zhang, L.-l. Zhang, and S.-s. Yang, "A prediction model for population occurrence of paddy stem borer (*Scirpophaga incertulas*), based on back propagation artificial neural network and principal components analysis," *Computers and Electronics in Agriculture*, vol. 68, no. 2, pp. 200–206, 2009.
- [15] C.-N. Lin, M.-Y. Wei, N.-T. Chang, and Y.-Y. Chuang, "The occurrence of *Scirtothrips dorsalis* Hood in mango orchards and factors influencing its population dynamics in Taiwan," *Journal of Asia-Pacific Entomology*, vol. 18, no. 3, pp. 361–367, 2015.
- [16] Y. Yan, C.-C. Feng, M. P.-H. Wan, and K. T. T. Chang, "Multiple regression and artificial neural network for the prediction of crop pest risks," in *Proceedings of the International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries*, pp. 73–84, Springer, Tunisia, May 2015.
- [17] W. Y. Zhang, T. Z. Jing, and S. C. Yan, "Studies on prediction models of *dendrolimus superans* occurrence area based on machine learning," *Journal of Beijing Forestry University*, vol. 39, no. 1, pp. 85–93, 2017.
- [18] S. Skawsang, M. Nagai, N. K. Tripathi, and P. Soni, "Predicting rice pest population occurrence with satellite-derived crop phenology, ground meteorological observation, and machine learning: a case study for the central plain of Thailand," *Applied Sciences*, vol. 9, no. 22, p. 4846, 2019.
- [19] S. Chen, A. Whiteman, A. Li et al., "An operational machine learning approach to predict mosquito abundance based on socioeconomic and landscape patterns," *Landscape Ecology*, vol. 34, no. 6, pp. 1295–1311, 2019.
- [20] T. Nyabako, B. M. Mvumi, T. Stathers, S. Mlambo, and M. Mubayiwa, "Predicting *Prostephanus truncatus* (Horn) (Coleoptera: bostrichidae) populations and associated grain damage in smallholder farmers' maize stores: a machine learning approach," *Journal of Stored Products Research*, vol. 87, pp. 101–592, 2020.
- [21] J. Patil and V. D. Mytri, "A prediction model for population dynamics of cotton pest (*Thrips tabaci* Linde) using multi-layer-perceptron neural network," *International Journal of Computer Application*, vol. 67, no. 4, pp. 19–26, 2013.
- [22] M. J. Watts and S. P. Worner, "Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species," *Ecological Informatics*, vol. 3, no. 1, pp. 64–74, 2008.
- [23] T. M. Carvajal, K. M. Viacrusis, L. F. T. Hernandez, H. T. Ho, D. M. Amalin, and K. Watanabe, "Machine learning methods reveal the temporal pattern of dengue incidence using meteorological factors in metropolitan Manila, Philippines," *BMC Infectious Diseases*, vol. 18, no. 1, pp. 1–15, 2018.
- [24] D. S. Lee, Y. S. Bae, B. K. Byun, and S. Lee, "Occurrence prediction of the citrus flatid planthopper (*Metcalfa pruinosa* (Say, 1830)) in South Korea using a random forest model," *Forests*, vol. 10, no. 7, p. 583, 2019.
- [25] İ. Balaban, F. Acun, O. Y. Arpalı, and F. Murat, "Development of a forecasting and warning system on the ecological life-cycle of sunn pest," 2019, <https://arxiv.org/abs/1905.01640>.
- [26] D. S. Yadav, S. Chander, and K. Selvaraj, "Agro-ecological zoning of brown planthopper [*Nilaparvata lugens* (Stal)] incidence on rice (*Oryza sativa* L.)," *Journal of Scientific & Industrial Research*, vol. 69, pp. 818–822, 2010.
- [27] P. K. Jayanthi and V. A. Erghese, "Host-plant phenology and weather-based forecasting models for population prediction

- of the oriental fruit fly, *Bactrocera dorsalis* Hendel,” *Crop Protection*, vol. 30, no. 12, pp. 1557–1562, 2011.
- [28] K. Fang, C. Shen, D. Kifer, and X. Yang, “Prolongation of SMAP to spatiotemporally seamless coverage of continental US using a deep learning neural network,” *Geophysical Research Letters*, vol. 44, no. 21, pp. 11–30, 2017.
- [29] O. Adeyemi, I. Grove, S. Peets, Y. Domun, and T. Norton, “Dynamic neural network modelling of soil moisture content for predictive irrigation scheduling,” *Sensors*, vol. 18, no. 10, p. 3408, 2018.
- [30] S. Scher, “Toward data-driven weather and climate forecasting: approximating a simple general circulation model with deep learning,” *Geophysical Research Letters*, vol. 45, no. 22, pp. 12–616, 2018.
- [31] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.
- [32] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, “Machine learning in geosciences and remote sensing,” *Geoscience Frontiers*, vol. 7, no. 1, pp. 3–10, 2016.
- [33] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, “Deep learning for plant identification using vein morphological patterns,” *Computers and Electronics in Agriculture*, vol. 127, pp. 418–424, 2016.
- [34] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, “Machine learning for high-throughput stress phenotyping in plants,” *Trends in plant science*, vol. 21, no. 2, pp. 110–124, 2016.
- [35] W. Rammer and R. Seidl, “Harnessing deep learning in ecology: an example predicting bark beetle outbreaks,” *Frontiers of Plant Science*, vol. 10, p. 1327, 2019.
- [36] L. Breiman, “Randomizing outputs to increase prediction accuracy,” *Machine Learning*, vol. 40, no. 3, pp. 229–242, 2000.
- [37] S. Nawar and A. M. Mouazen, “Comparison between random forests, artificial neural networks and gradient boosted machines methods of on-line Vis-NIR spectroscopy measurements of soil total nitrogen and total carbon,” *Sensors*, vol. 17, no. 10, p. 2428, 2017.
- [38] K. Arun and C. J. Langmead, “Structure based chemical shift prediction using Random Forests non-linear regression,” in *Proceedings of the 4th Asia-Pacific Bioinformatics Conference*, pp. 317–326, Taipei, Taiwan, February 2006.
- [39] M. Meng and C. Song, “Daily photovoltaic power generation forecasting model based on random forest algorithm for north China in winter,” *Sustainability*, vol. 12, no. 6, p. 2247, 2020.
- [40] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagation errors,” *Nature*, vol. 323, pp. 533–536, 1986.
- [41] I. M. M. Ghani and S. Ahmad, “Stepwise multiple regression method to forecast fish landing,” *Procedia-Social and Behavioral Sciences*, vol. 8, pp. 549–554, 2010.
- [42] S. Z. Khan, M. J. Arif, C. D. Hoddle, and M. S. Hoddle, “Phenology of Asian citrus psyllid (Hemiptera: liviidae) and associated parasitoids on two species of Citrus, kinnow Mandarin and sweet orange, in Punjab Pakistan,” *Environmental Entomology*, vol. 43, no. 5, pp. 1145–1156, 2014.
- [43] L. E. de Oliveira Aparecido, G. de Souza Rolim, J. R. D. S. C. De Moraes, C. T. S. Costa, and P. S. D. Souza, “Machine learning algorithms for forecasting the incidence of *Coffea arabica* pests and diseases,” *International Journal of Biometeorology*, vol. 64, pp. 1–18, 2020.
- [44] M. K. Sharma, A. Atsedewoin, and S. Fanta, “Forewarning models of the insects of paddy crop,” *International Journal of Biodiversity and Conservation*, vol. 3, no. 8, pp. 367–375, 2011.
- [45] G. Hu, X. N. Cheng, G. J. Qi et al., “Rice planting systems, global warming and outbreaks of *Nilaparvata lugens* (Stål),” *Bulletin of Entomological Research*, vol. 101, no. 2, pp. 187–199, 2011.
- [46] H. A. Narouei-Khandan, S. E. Halbert, S. P. Worner, and A. H. van Bruggen, “Global climate suitability of citrus huanglongbing and its vector, the Asian citrus psyllid, using two correlative species distribution modeling approaches, with emphasis on the USA,” *European Journal of Plant Pathology*, vol. 144, no. 3, pp. 655–670, 2016.
- [47] M. Fiaz, M. Afzal, and M. Z. Majeed, “Influence of abiotic weather factors on population dynamics of asian citrus psyllid, *Diuraphis citri* Kuwayama (Hemiptera: psyllidae) in central Punjab,” *Pakistan Journal of Agricultural Research*, vol. 56, no. 1, pp. 35–40, 2018.
- [48] M. R. Evans, K. J. Norris, and T. G. Benton, “Predictive ecology: systems approaches,” *Philosophical Transactions of the Royal Society of London B Biological Sciences*, vol. 367, pp. 163–169, 2012.
- [49] J. S. Clark, S. R. Carpenter, and M. Barber, S. Collins, A. Dobson, J. A. Foley et al., *Ecological Forecasts: An Emerging Imperative*. Science, Beck-Johns, Cumbria, England GB, 2001.
- [50] M. Reichstein, G. Camps-Valls, B. Stevens, J. Denzler, and N. Carvalhais Prabhat, “Deep learning and process understanding for data-driven Earth system science,” *Nature* vol. 566, pp. 195–204, 2019.

## Research Article

# Price Risk Measurement of China's Soybean Futures Market Based on the VAR-GJR-GARCH Model

Chuan-hui Wang <sup>1</sup>, Li-ping Wang,<sup>1</sup> Wei-feng Gong <sup>1,2</sup>, Hai-xia Zhang <sup>1</sup> and Xia Liu <sup>1</sup>

<sup>1</sup>School of Economics, QuFu Normal University, Rizhao 276826, China

<sup>2</sup>School of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211006, China

Correspondence should be addressed to Wei-feng Gong; [gongweifeng0539@163.com](mailto:gongweifeng0539@163.com)

Received 25 June 2021; Revised 10 October 2021; Accepted 8 November 2021; Published 10 December 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Chuan-hui Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As one of the main forces in the futures market, agricultural product futures occupy an important position in China's market. As China's futures market started late and its maturity was low, there are many risks. This study focuses on the Dalian soybean futures market. Dynamic risk measurement models were established to empirically analyze risk measurement problems under different confidence levels. Then, the conditional variance calculated by the volatility model was introduced into the value-at-risk model, and the accuracy of the risk measurement was tested using the failure rate test model. The empirical results show that the risk values calculated by the established models at the 99% and 95% confidence levels are more valuable through the failure rate test, and the risk of China's soybean futures market can be measured more accurately. The characteristics of "peak thick tail" and "leverage effect" are added to the combination model to calculate the conditional variance more accurately. The failure rate test method is used to test the model, which enriches the research problem of risk measurement.

## 1. Introduction

In the bulk commodity trade, soybeans, as an agricultural product with a large demand in China, have a prominent price fluctuation problem. In addition, China's soybean imports are at a disadvantage. China has become the world's largest importer. China's soybean futures market started late, the maturity was low, and the degree of risk was high, so the risk problem was a concern. Since 1992, the import of soybeans in China has been increasing greatly, accelerating even more in 1995. Since 1996, China has become a major net importer of soybeans worldwide. To optimize the soybean supply structure, China has implemented the agricultural policy of "reducing corn and increasing soybeans" since 2016. When trade frictions between China and the United States intensified in 2018, the United States increased import tariffs by 25%, increasing import costs, and Russia and Canada increased exports to China. In 2019, the Ministry of Agriculture and Rural Affairs of China decided to implement the Soybean Revitalisation Plan and put

forward six subsidy policies to support soybean development. In 2020, the central government of China stated that China should increase its support for the promotion of high-yield soybean varieties and the new agronomic promotion of maize and soybean intercropping. Agricultural futures occupy an important position in China's market as one of the main forces of the futures market. In the early days, agricultural products were traded on the spot, with extremely violent price fluctuations. Later, futures were adopted to regulate the market system and avoid risks to a certain extent. The Chicago Board of Trade was formally established in 1848. Standard futures contracts replaced forward contracts. The margin system was implemented, and agricultural product trading entered the era of standardized futures trading. In 1993, China established the Dalian Commodity Exchange (DCE). As the main force of agricultural products, soybean futures are listed in China's first batch. Agricultural futures not only affect the development of the national economy but also have a decisive significance in downstream futures. The maturity of the soybean futures market in China



is not very high, and many problems still exist. The factors affecting the fluctuation of soybean futures prices should be comprehensively discussed, and the risk measurement of the soybean futures market in China should be systematically studied. All countries have taken effective measures to protect the soybean industry to solve the problem of food security. This approach avoids system risk to a certain extent while alleviating the Chinese economic changes caused by international soybean price changes. The research on China's soybean futures market can not only deeply analyze the problems existing in the soybean futures market but also adopt the corresponding control policies in a timely manner to avoid certain risks. The research presented in this paper has great practical significance for solving the problem of soybean futures risk measurement.

With the continuous development of economic integration, the scale of the financial market continues to expand, and its risk increases. Some foreign financial markets have matured earlier. In the early stage, qualitative analysis was mainly used to study risk measurement from a macro-perspective, and in the 1930s, models were adopted to carry out quantitative research. Markowitz first proposed a method of sample variance to measure risks. With the rapid development of financial derivatives, the value-at-risk (VAR) model has become one of the main models for measuring risks [1]. The main methods to measure the VAR value include the covariance method, the historical simulation method, and the Monte Carlo method.

The variance-covariance method is a risk-metrics model proposed by the J.P. Morgan Group. The basic idea is to assign different weights according to the distance between the time series and the current time. The closer the distance is to the current event, the richer the market information will be, the greater the weight will be, and the greater the impact on the current price will be. This model assumes that the rate of return must follow a normal distribution, whereas in practical applications, the rate of return often has the characteristics of sharp peaks and thick tails. If there is a significant increase or decrease in the value of the VAR, the current risk level will be underestimated. To remedy this defect, Bollerslev used the generalized autoregressive conditional heteroskedasticity (GARCH) model to calculate the value of VAR [2]. All such methods use variance to represent the volatility of the market, and they are collectively known as the variance-covariance method.

In recent years, the VAR combination model has been greatly developed. Many scholars have applied this model to securities, funds, futures, and other fields. Zou et al. showed that the VAR-GARCH (1,1) model can be used to accurately estimate the risk of the Shanghai stock market [3]. Liu and Yu concluded that China can effectively avoid risks by learning from the price avoidance methods of the American agricultural futures market when studying the risk avoidance measures of foreign futures markets [4]. Jiang and Qu used the VAR and GARCH models to study the trading data of simulated stock index futures and showed that this method could effectively estimate the value of risk, with only a small error between the method and the actual risk [5]. Based on the GARCH model, Yang discussed the VAR calculation

method of logarithmic return under generalized error distribution (GED), t-distribution, and normal distribution, and the research results showed that the VAR value of the GARCH model under GED-distribution reflects the fund risk more accurately [6]. Zhang et al. compared the different performance of risk measurement of the conditional autoregressive value at risk (CAVIAR) and GARCH-GED models under different prediction intervals and confidence levels, indicating that the combined extreme value theory (EVT)-CAVIAR model is more robust and accurate for carbon market risk prediction [7]. Tao and Chen built a VAR-GARCH model to explore the pressure degree of RMB currency on foreign exchange in different states [8]. Yang and Yang conducted an empirical analysis of the relationship between the interest rate of net loans, Shibor, and private lending rates by constructing the VAR-GARCH-BEKK (BEKK is named after Baba, Engle, Kraft, and Kroner) model to provide suggestions for financial risk prevention [9]. Karmakar and Paul predicted the VAR and conditional VAR (CVAR) of three market portfolios using the CGARCH-EVT-copula model to determine the optimal portfolio model [10]. Lin et al. made risk measurement more accurate based on the VAR-GARCH model family [11]. Gong et al. studied the tail dependence of VAR and portfolio risk for the crude oil futures market [12]. Liu and Liu studied the tail risk spillover effect of different types of financial institutions [13]. Liang introduced the GJR model with a negative information correction term to fit the expected volatility and expected drift of the historical return rates of rebar spot and futures, used the Monte Carlo method to calculate the VAR value of the hedging portfolio, and established the hedging ratio VAR value surface model to solve the optimal hedging ratio in reverse [14]. Alexander et al. believe that when the prediction range is larger than the frequency of the GARCH model, it usually requires a time-consuming simulation of summary income distribution, which can be achieved quickly based on the new GARCH-VAR formula, and the data from different financial assets can be accurately analyzed using the analysis moments of symmetric and asymmetric GJR-GARCH processes to obtain accurate GARCH-VAR predictions at multiple significance levels [15].

According to the above analysis, domestic and foreign scholars have mainly studied the risk measurement VAR model of the soybean futures market in recent years and have achieved rich research results. The risk research of the soybean futures market has focused on qualitative analysis, but now it begins to change to quantitative analysis. GARCH models with different distributions are mostly used to measure conditional standard deviations without considering the characteristics of price fluctuations. However, the VAR measurement using combined models will improve the accuracy. The use of the model is relatively simple and requires less analysis, which is not currently involved in innovative methods at the present stage. However, using a combined model to measure the VAR will improve the accuracy. In this study, the quantitative analysis method was adopted, considering the characteristics of price volatility clustering, leverage effect, peak, and thick tail; the combined VAR-GARCH model and VAR-GJR-GARCH model were

constructed under different confidence levels for risk measurement research, and the accuracy of the model using the failure rate test method was verified.

## 2. Statistical Analysis of Basic Characteristics of Chinese Soybean Futures Market

*2.1. Data Collection and Selection.* In recent years, China's soybean futures contracts have become more active on the DCE, so this study chooses those contracts for research. The volatility and risks in China's soybean futures market are mainly caused by fluctuations in price and yield rate, so price is selected as the research object. Because several contracts with different delivery months will participate in trading on the same trading day, the same futures product will have different trading prices on the same trading day. Therefore, in the selection of data, both the discontinuity of futures market prices and the price of futures contracts should be considered.

To make the trading data more representative and the research on the problem more convincing, this study chooses contracts with large settlement prices, trading volumes, and open positions in trading contracts, among which the contract data of A1505, A1605, A1705, A1805, A1905, and A2005 are selected. If there are two contracts in a day at the same time, according to the above choices, the paper chooses a contract with a large trading volume and open position to constitute the main contract in China's soybean futures market. The settlement price of the contract as transaction data was selected to form a continuous time series. The soybean futures available for delivery on the DCE of China include Yellow Soybean No. 1 and Soybean No. 2. As a non-GM soybean, Yellow Soybean No. 1 has a good representation in studying the price fluctuation of China's soybean futures market, so the main contract of Yellow Soybean No. 1 is chosen as the research object. In this study, a total of 1,219 sample data were selected from 31 December 2014 to 31 December 2019. A model was established and analyzed for the above data. Descriptive statistical analysis was conducted on the time-series data of the Chinese soybean futures yield. The return rate series data were obtained by logarithmic processing of the soybean futures' settlement price data in China. The formula used to calculate the rate of return was as follows:

$$R_t = \ln P_t - \ln P_{t-1}, \quad (1)$$

where  $R_t$  represents the yield of the day,  $P_t$  represents the settlement price of the day, and  $P_{t-1}$  represents the settlement price of the previous day. The time series of the rate of return was obtained and analyzed.

*2.2. Statistical Characteristics of Yield Rate.* Through logarithmic processing of the sample data, basic statistical results were obtained. Figure 1 shows a line graph of the daily return rate fluctuation of the main soybean futures contract in China.

As shown in Figure 1, the yield series of the soybean futures market in China has a fluctuation agglomeration

effect, which usually means that the yield series will have an autoregressive conditional heteroscedasticity (ARCH) effect.

The statistics of the daily return series of China's soybean futures contracts are obtained by a statistical test of the return series of China's soybean futures market. This is shown in Table 1.

As shown in Table 1, the skewness of the daily return series of Chinese soybean futures is  $S = 1.436 > 0$  and kurtosis  $K = 14.74 > 3$ . This skewness is skewed to the right, relative to the normal distribution. Therefore, the yield sequence of Chinese soybean futures shows the characteristics of "sharp peak and thick tail." In addition, the Jarque-Bera statistic was 7417, and the corresponding  $P$  value was 0. Therefore, the null hypothesis is rejected, indicating that the return rate series does not follow a standard normal distribution. Therefore, the  $t$ -test,  $F$ -test, and other tests suitable for normal distribution cannot be used to test the rate of return.

The futures market is a financial market, and the volatility of the financial market is relatively complex. The time series of the financial market is independent, and the variance is a fixed constant series. Volatility persistence is also known as the long-term memory of volatility. It refers to a long-term financial time series that has a special non-linear relationship. The longer the time interval, the stronger the correlation between the numerical values. This characteristic makes continuous movement of the sequence in the same direction, through a waning change direction. The long-term accumulation of time is caused by fluctuations in sustainability. Therefore, in the financial market, transaction information in the past period will have a long-term and lasting impact on information in the future. The financial time series is characterized by sharp peaks and thick tails and does not obey the standard normal distribution. Therefore,  $t$ -tests,  $F$ -tests, and other tests suitable for normal distribution cannot be used to carry out relevant tests, meaning that the traditional linear regression method cannot be used to solve this problem. As financial time series generally have heteroscedasticity, the ARCH model is considered for analysis. Therefore, stationarity tests and autocorrelation tests are also required for the sample time return series.

*2.3. Test of Yield Series.* Before establishing the model for the return rate data, it is necessary to conduct stationarity, autocorrelation, partial autocorrelation, and ARCH effect tests to confirm the practical significance of the subsequent model.

*2.3.1. Rate of Return Stability Test.* If the sample sequence data are non-stationary, even if there is no correlation between variables, the regression results will have a high degree of fit and even lead to pseudo-regression. Therefore, a stationarity test should be conducted before establishing the model. There are many methods for the sequence stationarity test; the Dickey-Fuller (DF) and augmented Dickey-Fuller (ADF) tests are widely used. When the sequence has a high-order lag correlation, the DF test violates the assumption that the random disturbance terms are

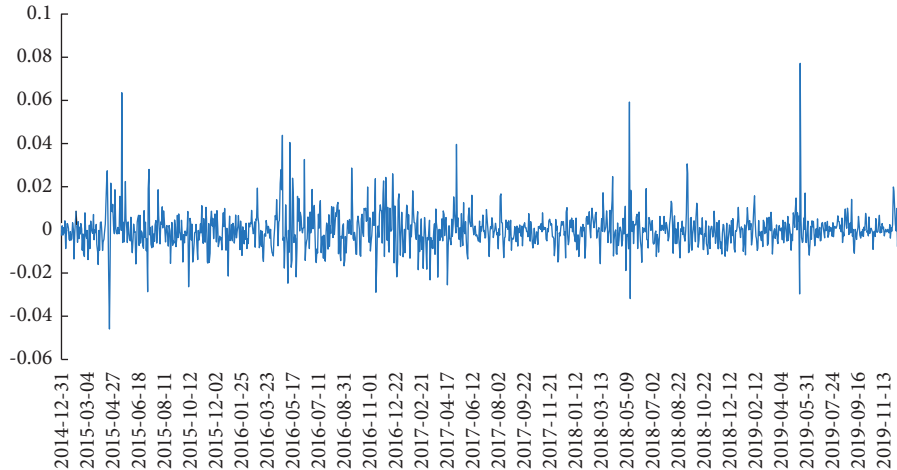


FIGURE 1: The daily yield fluctuation of China's soybean futures market.

TABLE 1: Daily yield series of China's soybean futures' main continuous contracts.

Statistic	Standard deviations	Skewness	Kurtosis	Jarque-Bera statistics	<i>P</i> value
Soybean futures daily return series	0.0086258	1.436	14.74	7417	$\leq 0.0000$

independently and identically distributed. Therefore, the ADF unit root test was used to test the stationarity of high-order sequences. According to the critical value of the *t*-statistic at the confidence levels of 1%, 5%, and 10%, if the test *t*-statistic is less than the critical value, the null hypothesis is rejected, and the sample sequence does not have a unit root and is stable. Otherwise, the null hypothesis is accepted, and the sample has a non-stationary sequence. Simultaneously, the size of the *P* value in the ADF test can be analyzed. When *P* is greater than 0.05, the null hypothesis is accepted, which means that the sample sequence has a unit root and a non-stationary sequence. Otherwise, it is a stationary sequence, and the closer the *P* value is to zero, the higher the data stationarity is, and the more accurate the result is.

The unit root test was performed on the time-series data, and the results are shown in Table 2.

As shown in Table 2, the *t*-statistic value of the ADF test is  $-30.923$ , which is far less than the critical value at the significance level of 1%, 5%, and 10%. The *P* value corresponding to the yield series is 0, so the null hypothesis is rejected. There is no unit root in the daily yield series of Chinese soybean futures, which is a stationary series that is suitable for empirical research.

**2.3.2. Test of Autocorrelation and Partial Autocorrelation of Return Rate.** The series of returns changes with time, presenting kurtosis and skewness. The returns of successive adjacent periods are not independently and identically distributed, which is called the autocorrelation of returns. If the rate of return is autocorrelated, residual autocorrelation will result in inaccurate empirical results. Therefore, it is necessary to conduct an autocorrelation test on the rate of return series. If the autocorrelation and partial

TABLE 2: Stability test of China's soybean futures yield series.

	<i>t</i> -statistic	Prob.*
Augmented Dickey-Fuller test statistic	$-30.923$	$\leq 0.0000$
	1% level	$-3.430$
Test critical values	5% level	$-2.860$
	10% level	$-2.570$

autocorrelation graphs have no obvious truncation or trailing phenomena, there will be no autocorrelation or partial autocorrelation.

Autocorrelation and partial autocorrelation tests were carried out on the yield data of the Chinese soybean futures market. The results show that there is high-order non-autocorrelation in the daily yield series, and there is no obvious truncation or trailing phenomenon in the selected yield series. Therefore, there is no autocorrelation or partial autocorrelation in the yield series of the Chinese soybean futures market.

**2.3.3. ARCH Effect Test.** To test whether the residual series of return rate has an ARCH effect, that is, to test whether the residual has conditional heteroscedasticity, this study chose the commonly used Lagrange multiplier (LM) test. Engel proposed the LM test to test whether the ARCH effect exists in residual sequences. The original hypothesis of the test was that there was no ARCH effect in residual sequences up to order *Q*, and the regression equation was expressed as follows:

$$VaR = -Z_c \sigma_\rho \sqrt{\Delta t}. \quad (2)$$

Two statistics are output through this test: the *F* statistic is used to test the joint significance for all lagging residual

square terms; the  $T \times R^2$  statistic represents the product of sample size and goodness of fit of the test regression equation. Under the condition that the null hypothesis holds, the sample distribution of the F statistic is not clear, but the LM test statistic asymptotically obeys the distribution  $\chi^2(q)$ . The ARCH-LM test results of the residual sequences are shown in Table 3.

As shown in Table 3, the  $P$  value corresponding to each statistic of the residual squared lag of order 1–5 is 0, indicating that all the lagging residual squares are jointly significant. The  $P$  value of the ARCH effect test is 0, so the null hypothesis is rejected. The residual sequence has conditional heteroscedasticity and an ARCH effect.

### 3. Empirical Study on the Price Risk Measurement of China's Soybean Futures Market

The price of China's soybean futures fluctuated greatly, the uncertainty was strong, and the risk was high. Therefore, the risk measurement of the Chinese soybean futures market prices was studied by establishing VAR-GARCH and VAR-GJR-GARCH portfolio models. First, the GARCH and GJR-GARCH models were constructed to measure the conditional variance. Then, a dynamic risk measurement model was constructed, and the variance values calculated by the GARCH and GJR-GARCH models were introduced into the VAR model to measure the risk value. Finally, the failure rate test of the VAR was conducted to observe whether the risk value was effective mainly through the failure rate. The main process is illustrated in Figure 2.

**3.1. Empirical Analysis of Yield Characteristics Based on GARCH Model.** To reduce the error and variability of the data, the yield series is standardized, and the GARCH model is constructed to depict the volatility of soybean futures prices in China.

**3.1.1. GARCH Model Construction.** In the GARCH ( $p, q$ ) model,  $P$  is the order of the autoregressive GARCH term and  $q$  is the order of the ARCH term, in which GARCH (1, 1) has the highest frequency of application, mainly because GARCH (1, 1) is relatively simple and has a great advantage in calculating the conditional variance and conditional mean of the time series. In the process of practical application, it is found that GARCH (1, 1) does not exhibit high-order risks, so GARCH (1, 1) is selected as the return rate sequence (Lin et al.). The regression equation is expressed as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \quad (3)$$

**3.1.2. Empirical Results and Analysis of Yield Characteristics of GARCH Model.** The GARCH (1, 1) model is constructed for China's soybean futures price, and the results are shown in Table 4.

The variance of the conditional equation in the GARCH (1, 1) model is

TABLE 3: ARCH-LM test results of residual series.

Lags ( $P$ )	Chi <sup>2</sup>	df	Prob > chi <sup>2</sup>
1	24.996	1	≤0.0000
2	25.293	2	≤0.0000
3	26.459	3	≤0.0000
4	26.457	4	≤0.0000
5	28.198	5	≤0.0000

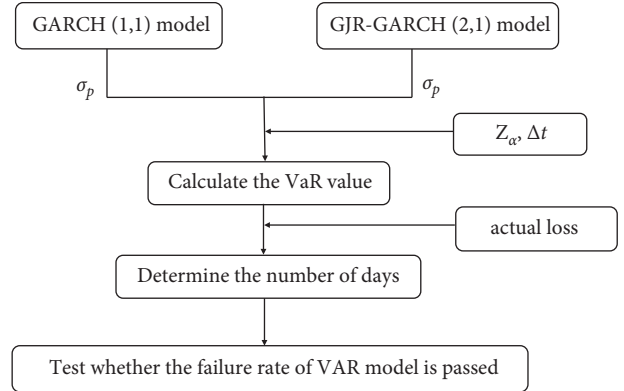


FIGURE 2: Process chart of dynamic risk measurement of China's soybean futures price.

TABLE 4: Estimation of GARCH (1, 1) model.

	$\alpha_0$	$\alpha_1$	$\beta_1$	$\alpha_1 + \beta_1$
Coefficient estimation	0.0001624	0.1424091	0.8435093	0.9859184
Z value	8.37	15.71	110.39	
P value	0.000	0.000	0.000	

$$\sigma_t^2 = 0.0001624 + 0.1424091 \varepsilon_{t-1}^2 + 0.8435093 \sigma_{t-1}^2. \quad (4)$$

The Akaike information criterion (AIC) value is  $-3217.4$ , the Bayesian information criterion (BIC) value is  $-3197$ , and the logarithmic likelihood value is  $1612.7$ .

As can be seen from Tables 3 and 4, the parameter estimates of ARCH and GARCH terms in the conditional variance equation are highly significant at the 5% significance level; the parameter estimates are all greater than zero, which meets the requirement of non-negative conditional variance, indicating that the price fluctuations in China's soybean futures market are characterized by volatility aggregation; the estimated coefficients of the ARCH and GARCH terms are 0.1424091 and 0.8435093, respectively, and the sum of coefficients is less than 1, which satisfies the constraint conditions of model parameters. This indicates that the conditional variance of the random error term can converge to unconditional variance, and the fluctuation process is stationary; as the sum of the coefficients is very close to 1, it indicates that the impact of the early shock on the subsequent conditional variance is lasting and will affect future fluctuations for a long time. Therefore, we believe that China's soybean futures market has high volatility and high market risk. This also describes the current situation of China's soybean futures: China's soybeans are mainly

imported, and the change of international political situation will cause a huge impact on China's soybean futures market.

The ARCH-LM test was performed on the GARCH (1,1) model with a lag of 10 orders, and the results are shown in Table 5.

As can be seen from Table 5, the R-squared equals 14.58331. The  $P$  value of the 10-order residual lag is 0.1479 and is greater than 0.05. This means that the ARCH effect does not exist in the conditional variance. The model passed the test, so the conditional heteroscedasticity of the daily return series of soybean futures contracts was eliminated after the application of the GARCH model.

**3.2. Empirical Analysis of Yield Characteristics Based on GJR-GARCH Model.** The leverage effect is common in financial time series, so the asymmetric ARCH model and GJR-GARCH model are constructed to describe the fluctuation characteristics of soybean futures prices.

**3.2.1. GJR-GARCH Model Construction.** As financial time series are prone to leverage effect, this study selects the GJR-GARCH model with asymmetry to measure the advantages and disadvantages of the model according to AIC and BIC and maximum likelihood value. The commonly used GJR-GARCH model is GJR-GARCH (1,1). The smaller the value of GJR-GARCH (1,2), GJR-GARCH (2,1), and GJR-GARCH (2,2), the smaller the model error. The fitting effect of the model is better when the maximum likelihood value is larger, and the results of the GJR-GARCH ( $p, q$ ) model are shown in Table 6.

As shown in Table 6, the GJR-GARCH (2,1) model has the smallest AIC and BIC values, and the GJR-GARCH (2,1) model has the largest maximum likelihood value. Therefore, it is believed that the GJR-GARCH (2,1) model has the advantages of small error and good fitting effect.

**3.2.2. Empirical Results and Analysis of Yield Characteristics of GJR-GARCH Model.** The GJR-GARCH (2,1) model was constructed to estimate Chinese soybean futures price. The regression equation is expressed as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \gamma \varepsilon_{t-1}^2 I_{t-1}, \quad (5)$$

where  $I_{t-1}$  is the dummy variable.

$$I_{t-1} = \begin{cases} 1, & \varepsilon_{t-1} \geq 0, \\ 0, & \varepsilon_{t-1} < 0. \end{cases} \quad (6)$$

In (6), good news or positive shock is  $\varepsilon_{t-1} > 0$ , and bad news or negative shock is  $\varepsilon_{t-1} < 0$ . It has different impact on conditional variance  $\sigma_t^2$ , the impact of the front is  $\alpha_1 \varepsilon_{t-1}^2$ , and the impact of the latter is  $(\alpha_1 + \gamma) \varepsilon_{t-1}^2$ . When  $\gamma < 0$ , it indicates the existence of asymmetric effect. When  $\gamma = 0$ , it indicates that there is no asymmetric effect.

The results of estimation of the GJR-GARCH (2,1) model are shown in Table 7.

By constructing the GARCH and GJR-GARCH models, the price fluctuation of China's soybean futures market has

TABLE 5: ARCH-LM test of GARCH (1, 1).

$F$ value	1.462706	$P$ value	0.1479
Obs * R-squared	14.58331	Prob. chi-square (10)	0.148

the characteristics of fluctuation aggregation and persistence. Both models can describe these two characteristics of price fluctuation well, indicating that the two models have the same information-fitting ability. Compared with the GARCH model, the GJR-GARCH model shows that negative information has a stronger impact on price fluctuations. The GJR-GARCH model has a stronger ability to capture information. The fitting ability of the two models is similar, and the GJR-GARCH model has a stronger ability to capture information. Therefore, the GJR-GARCH model is more suitable for depicting the price volatility characteristics of China's soybean futures market.

**3.3. Empirical Study on Dynamic Risk Based on VAR-GARCH Measure Model.** According to the above research, the variance-covariance method was adopted to measure risk. First, it is assumed that the series of return rates on assets obey a certain distribution, such as a normal distribution. Then, the parameter value of the return rate distribution is calculated by using the data of a certain period in the past, and finally, the asset value of the quantile is calculated at a certain confidence level. The calculation formula is as follows:

$$\text{VAR} = -Z_c \sigma_p \sqrt{\Delta t}, \quad (7)$$

where  $Z_c$  represents the quantile of confidence level  $C$ ,  $\sigma_p$  represents the standard deviation of assets, and  $t$  represents the holding period (i.e., the daily rate of return,  $\Delta t = 1$ ). When calculating the VAR value, we should first determine the  $\sigma_p$  value. Owing to the characteristics of financial return series, such as sharp peaks and thick tails, volatility agglomeration, persistence, and leverage effect, we only process the mean value of the time series in the process of general simple model data processing and ignore the characteristics of the return series. Therefore, the GARCH (1,1) model was selected according to the AIC and BIC after calculating the yield rate of the soybean futures A1505 contract, and the estimated  $\sigma_p$  value in the model was put into the calculation formula of the VAR model. At this time, the  $\sigma_p$  value can better reflect the characteristics of futures return rate, and the empirical analysis results are more reasonable and accurate.

According to the above analysis, the quantile, probability distribution, and conditional standard deviation under confidence must be obtained before calculating the value of VAR. The quantiles under 99%, 95%, and 90% confidence were selected to calculate the risk value of the VAR-GARCH based on normal distribution. Under normal distribution, the daily yield sequences of China's soybean futures market are 2.3263, 1.6449, and 2.3263 with 99%, 95%, and 90% confidence levels, respectively.

$\sigma_p$  of the GARCH (1,1) model and the quantile of different confidence degrees were put into the VAR

TABLE 6: Comparison of GJR-GARCH ( $p, q$ ) model.

	GJR-GARCH (1, 1)	GJR-GARCH (1, 2)	GJR-GARCH (2, 1)	GJR-GARCH (2, 2)
r	0.371*** (192.62)	0.371*** (192.95)	0.369*** (246.26)	0.371*** (196.42)
ARCH				
L.arch	0.378*** (10.33)	0.378* (10.31)	0.419*** (13.24)	0.364** (9.31)
L2.arch			0.356*** (-10.82)	0.234** (5.12)
L.gjr-garch	-0.0841 (-1.38)	-0.0788 (-1.28)	0.0453*** (-5.38)	-0.0529 (-0.84)
L2.gjr-garch		0.0133 (0.59)		-0.0107 (-0.20)
L.garch	-0.0237 (-0.81)	-0.0663 (-0.92)	0.962*** (153.91)	-0.759*** (-10.14)
Cons	0.00345*** (22.61)	0.00361*** (13.52)	0.00000252*** (3.05)	0.00592*** (16.89)
N	1219	1219	1219	1219
Log likelihood	1589.9	1590.1	1646.4	1595.4
AIC	-3169.7	-3168.1	-3280.9	-3176.8
BIC	-3144.2	-3137.5	-3250.3	-3141.1

$t$ -statistics are given in parentheses. \* $P < 0.05$ , \*\* $P < 0.01$ , and \*\*\* $P < 0.001$ .

TABLE 7: Estimation of GJR-GARCH (2, 1) model.

	$\alpha_0$	$\alpha_1$	$\alpha_2$	$\beta_1$	$\gamma$
Coefficient estimation	0.0000252	0.4193364	-0.3562949	0.9615618	-0.0453458
Z value	3.05	13.24	-10.82	153.91	-5.38
P value	0.002	0.000	0.000	0.000	0.000

calculation formula to calculate the risk value. The results are presented in Table 8.

According to Table 8, the mean, maximum, and minimum values of risks of different confidence levels in the two models can achieve a quantitative analysis of risks. In the GARCH (1,1) model, at the 99% confidence level, the average risk is  $-1.91\%$ , the maximum daily loss is not more than  $-0.90\%$ , and the minimum loss is not less than  $-11.20\%$ . At the 95% confidence level, the average risk is  $-1.35\%$ , the maximum daily loss is not more than  $-0.64\%$ , and the minimum daily loss is not less than  $-7.92\%$ . At the 90% confidence level, the average risk is  $-1.05\%$ , with a maximum daily loss of  $-0.50\%$  and a minimum daily loss of  $-6.17\%$ . The following comparison diagram of VAR and return rate under different confidence levels of different models shows the comparison between VAR and actual returns more intuitively and clearly (Figure 3).

In Figure 3, k99 represents the VAR-GARCH model under 99% confidence, k95 represents the VAR-GARCH model under 95% confidence, k90 represents the VAR-GARCH model under 90% confidence, and  $r$  represents the yield rate of China's soybean futures market. The comparison between risk volatility and the return rate at different confidence levels is shown in Figure 3. The risk value at the 90% confidence level was greater than that at the 95% confidence level and at the 99% confidence level. Therefore, the failure rate was highest at the 90% confidence level. At the same time, the VAR value of China's

soybean futures market also has a continuity and fluctuation agglomeration effect.

*3.4. Empirical Study on Dynamic Risk Based on VAR-GJR-GARCH Measure Model.* The GJR-GARCH (2,1) model was selected based on the AIC and BIC. The  $\sigma_p$  value estimated in the model was incorporated into the calculation formula of the VAR model. The  $\sigma_p$  value can better reflect the characteristics of the futures return rate, and the empirical analysis results are more reasonable and accurate.

According to the above analysis, the quantile, probability distribution, and conditional standard deviation under confidence should be obtained before calculating the value of VAR. The quantiles under 99%, 95%, and 90% confidence, respectively, were selected to calculate the risk value of VAR-GJR-GARCH based on the normal distribution.  $\sigma_p$  calculated by the GJR-GARCH (2, 1) model and the quantile under different confidence degrees were put into the VAR calculation formula to calculate the risk value. The results are presented in Table 9.

According to Table 9, we can see the mean value, maximum value, and minimum value of risks under different confidence levels in the two models, in order to achieve quantitative analysis of risks. In the GJR-GARCH (2, 1) model, at the 99% confidence level, the average risk is  $-1.97\%$ , the maximum daily loss is not more than  $-0.99\%$ , and the minimum loss is not less than  $-7.39\%$ . At the 95%

TABLE 8: Risk value calculation of the two models at different confidence levels.

Model	Degree of confidence (%)	Mean	Standard deviations	Minimum	Maximum
VAR-GARCH	99	-0.0190769	0.0077497	-0.111965	-0.0090189
	95	-0.013489	0.0054798	-0.0791691	-0.0063772
	90	-0.0105098	0.0042695	-0.0616835	-0.0049687

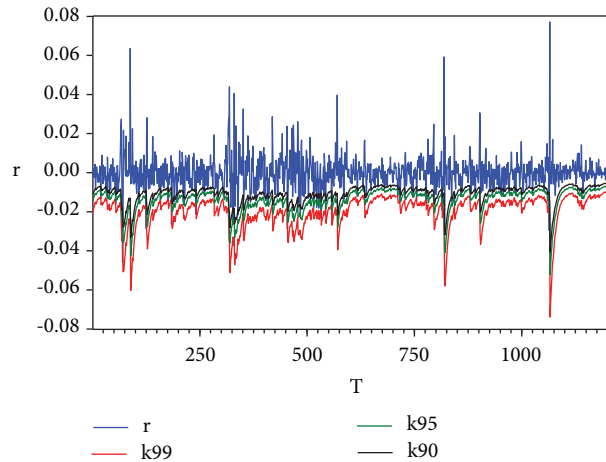


FIGURE 3: Comparison of VAR-GARCH model risk value and return rate at different confidence levels.

TABLE 9: Risk value calculation of the two models at different confidence levels.

Model	Degree of confidence (%)	Mean	Standard deviations	Minimum	Maximum
VAR-GJR-GARCH	99	-0.0196874	0.0080489	-0.0738947	-0.0098829
	95	-0.0139207	0.0056913	-0.0522501	-0.0069881
	90	-0.0108461	0.0044343	-0.0407099	-0.0054447

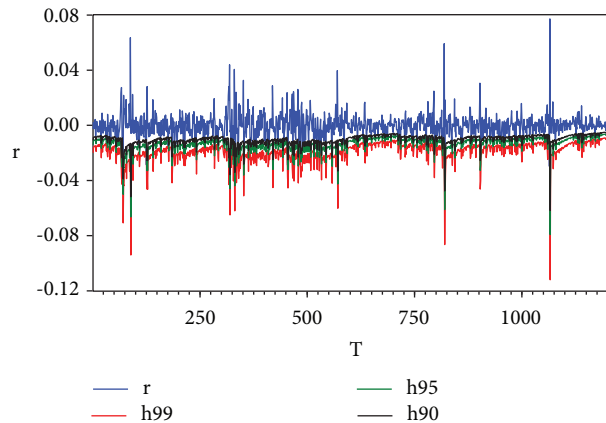


FIGURE 4: Comparison of VAR-GJR-GARCH model risk value and return rate at different confidence levels.

confidence level, the average risk is  $-1.35\%$ , the maximum daily loss is not more than  $-0.64\%$ , and the minimum daily loss is not less than  $-7.92\%$ . At the  $90\%$  confidence level, the average risk is  $-1.08\%$ , with a maximum daily loss of no more than  $-0.54\%$  and a minimum daily loss of no less than  $-4.07\%$ . To observe the comparison between VAR and actual returns more intuitively and clearly, a comparison diagram of VAR and return rate under different models and different confidence levels was drawn (Figure 4).

In Figure 4,  $h_{99}$  represents the VAR of the VAR-GJR-GARCH model with  $99\%$  confidence.  $h_{95}$  represents the VAR of the VAR-GJR-GARCH model with  $95\%$  confidence.  $h_{90}$  represents the VAR of the VAR-GJR-GARCH model with  $90\%$  confidence.  $r$  represents the rate of return on China's soybean futures market. A comparison between risk volatility and return rate at different confidence levels is clearly shown. The risk value at the  $90\%$  confidence level is greater than that at the  $95\%$  confidence level and at the  $99\%$

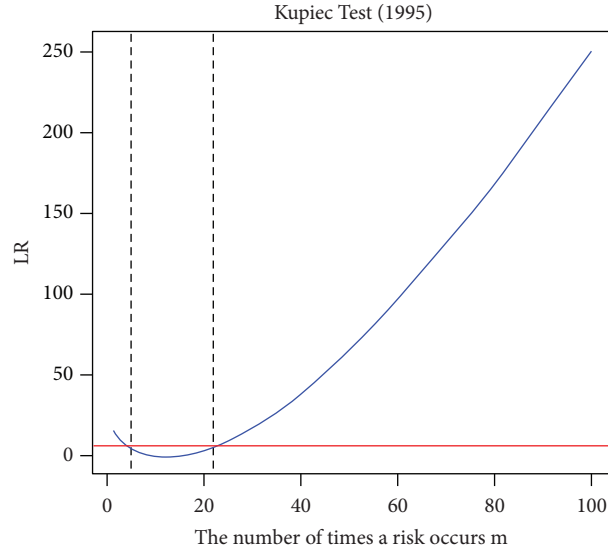


FIGURE 5: Interval chart of risk days at 99% confidence level.

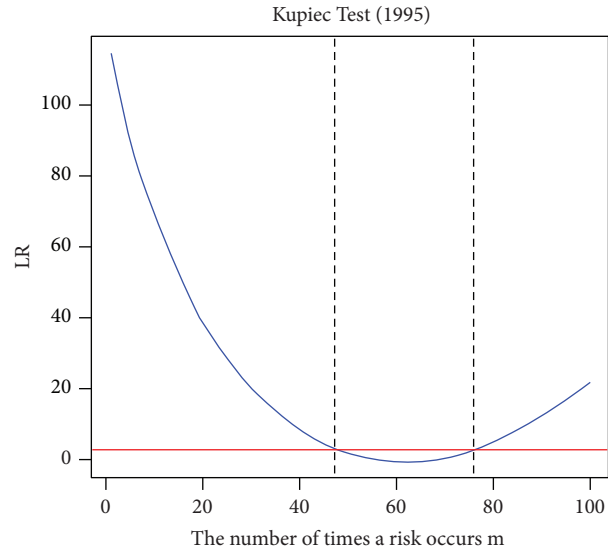


FIGURE 6: Interval chart of risk days at 95% confidence level.

confidence level. Therefore, the failure rate was the highest at a 90% confidence level. However, the failure rate cannot directly reflect the rationality of the model, and the failure days under the condition of the confidence level should be measured. Therefore, the failure rate test of the VAR model was used to verify the rationality of the model.

**3.5. Dynamic Risk Model Measure Effect.** Many risk assessment methods have been developed. The failure rate test method, which mainly tests the probability that the actual loss exceeds the VAR, was adopted. Kupiec proposed the failure rate test method. Its main idea is to record days in which the actual loss value exceeds the VAR estimate as failure days and then calculate the failure rate [16]. Because the estimation is assumed to be time-independent, the test of the binomial result under the failure condition can be

regarded as an independent Bernoulli experiment; therefore, the expected failure probability is denoted as  $P^*$ . The interval of failure days is different at different confidence levels, and the failure days within the interval of failure days at a given confidence level indicate that the VAR value is valid. If the observation days in the model are  $T$  and the failure days are  $N$ , then the failure rate is  $p = N/T$ , and the null hypothesis is  $p = p^*$ . The most appropriate method for testing A is the maximum likelihood ratio test. The formula is as follows:

$$LR = -2\ln[(1 - p^*)^{T-N}(p^*)^N] + 2\ln\left[\left(1 - \frac{N}{T}\right)^{T-N} \left(\frac{N}{T}\right)^N\right]. \quad (8)$$

Under the null hypothesis, the statistic obeys  $LR \sim \chi^2(1)$ . This test method tends to underestimate potential risks in small sample data. Therefore, this study



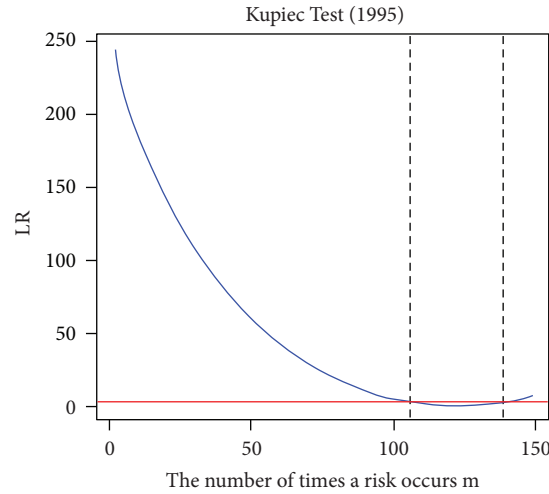


FIGURE 7: Interval chart of risk days at 90% confidence level.

TABLE 10: Failure rate test of the two models at different confidence levels.

Degree of confidence (%)	Failure day interval	Model	Actual failure days	Failure rate	Whether it passes the test
99	[4, 22]	VAR-GARCH	17	0.013945857	Yes
		VAR-GJR-GARCH	14	0.011484824	Yes
95	[48, 77]	VAR-GARCH	52	0.042657916	Yes
		VAR-GJR-GARCH	55	0.04511895	Yes
90	[106, 139]	VAR-GARCH	100	0.082034454	Yes
		VAR-GJR-GARCH	100	0.082034454	Yes

adopts a large sample of data, which can effectively avoid this situation.

According to the failure rate test formula and method, the failure day intervals were calculated at 99%, 95%, and 90% confidence levels. The corresponding graphs of the failure days  $M$  and  $LR$  were drawn. The interval at which the curve intersects the line is the confidence interval of the number of failure days. When the failure days are within the interval under a certain confidence degree, the null hypothesis is accepted, indicating that the risk measure is reasonable. If the number of failure days is lower than the minimum value of the interval, the model estimate is too conservative. If the number of failure days is greater than the maximum value of the model, the model underestimates the risk. The results are shown in Figures 5–7.

In Figures 5–7, the days of risk represent the days of risk under 99%, 95%, and 90% confidence, respectively. The dynamic VAR values obtained by the two models at different confidence levels were compared with the daily actual return rate. The part where the daily real return rate is lower than the VAR value is recorded as the actual failure days. The accuracy of the model was judged according to the interval values under different degrees of confidence. The results are presented in Table 10.

According to Table 10, the following conclusions can be drawn:

- (1) At the 99% confidence level, the actual failure days of the VAR-GARCH model and the VAR-GJR-

GARCH model both fall within the range of the failure days estimated by the model, and the failure rates pass the test, indicating that the two models can better measure the risk at the 99% confidence level.

- (2) At the 95% confidence level, the actual failure days of the VAR-GARCH model and the VAR-GJR-GARCH model both fall within the range of the failure days estimated by the model, and the failure rates pass the test, indicating that the two models can better measure the risk at the 95% confidence level. There is little difference between the two models in the measurement of risk, and at the 90% confidence level, the failure rates of the two models are the same, so neither model can estimate the value of risk. Therefore, participants in China's soybean futures market can conduct risk measurement research at 99% and 95% confidence levels according to the two models to estimate the value of risk and effectively avoid non-systemic risks.

#### 4. Conclusions

China's soybean futures market started late, and the price fluctuated violently, so there was a large risk. In this study, the VAR-GARCH and VAR-GJR-GARCH models were first constructed to measure the risk value at 99%, 95%, and 90% confidence levels, and then the risk was measured by quantitative analysis. Next, the accuracy and rationality of

the model estimation were measured using the failure rate test method. Finally, the risk value calculated by the two constructed models at 99% and 95% confidence levels is more valuable, which can better measure the risk of the Chinese soybean futures market. For futures markets with higher risk tolerance or when the risk-bearing ability is low for participants, the combined model can undertake the risk measure to avoid risks, to some extent, for futures market participants. Empirical research on the two types of combination models can be better applied to calculate risk measures in the soybean futures market. At the same time, the government can formulate corresponding countermeasures according to risk value to effectively avoid risks and promote the stable development of China's soybean futures market.

Although this paper studies the price risk measurement of China's soybean futures from a variety of perspectives and has made some achievements, there is still a lot of work to be done to further understand the price risk. The specific shortcomings and prospects are as follows:

- (1) Although compared with other risk measurement models, the VAR method can better measure the risk value, this paper describes more accurately through the GARCH class model, but there exist still shortcomings. Compared with futures markets with greater risks, extreme situations occur in a high frequency. It is difficult to calculate the future VAT risk by using the VAR method in advance and predict the future VAT risk by using the historical rate of return, which cannot reflect the mood changes of market traders. Therefore, in our next step of research, macro-environmental changes will be taken into account in time, and the model established above should be adjusted in time. Traders' emotions should be taken into account in order to carry out more accurate risk measurement.
- (2) The daily high-frequency data selected in this paper can better reflect the volatility. For in-sample data, the prediction accuracy is high. For out-of-sample prediction, only a few data can be predicted to ensure the data accuracy, and the daily high-frequency data can only be predicted for a few days. Therefore, monthly data and quarterly data will be considered in the future research, compared with daily data for comparative analysis. This work will characterize price risk more comprehensively and profoundly.

## Appendix

R programming:

```
n <- -1219; m <- seq (from = 1, to = 100, length = 100);
p <- -0.01; #The probability of exceptions occurring.
conf <- -0.99; # The confidence level of the chi^2
distribution.
LR = -2*log(((1-p)^(n-m))* (p^m)) + 2*log(((1-(m/n))^(n-m))* ((m/n)^m)); critical_value <- -qchisq (conf, 1);
```

```
plot (m, LR, type = "l," col = "blue," lwd = 2, main = "Kupiec
Test (1995)," xlab = "The number of times a risk occurs m")
abline (h = critical_value, col = "red," lwd = 3).
LR_C <- -which (LR <= critical_value).
abline (v = c(LR_C [1], LR_C [length (LR_C)]),
lty = "dashed")
LR_C.
n <- -1219; m <- seq (from = 1, to = 100, length = 100);
P <- -0.05; # The probability of exceptions occurring.
conf <- -0.95; # The confidence level of the chi^2
distribution.
LR = -2*log(((1-p)^(n-m))* (p^m)) + 2*log(((1-(m/n))^(n-m))* ((m/n)^m)); critical_value <- -qchisq (conf, 1);
plot (m, LR, type = "l," col = "blue," lwd = 2, main = "Kupiec
Test (1995)," xlab = "The number of times a risk occurs m")
abline (h = critical_value, col = "red," lwd = 3).
LR_C <- -which(LR <= critical_value).
abline (v = c (LR_C[1], LR_C[length(LR_C)]),
lty = "dashed")
LR_C.
n <- -1219; m <- seq (from = 1, to = 150, length = 150);
P <- -0.10; # The probability of exceptions occurring.
conf <- -0.90; # The confidence level of the chi^2
distribution.
LR = -2*log(((1-p)^(n-m))* (p^m)) + 2*log(((1-(m/n))^(n-m))* ((m/n)^m)); critical_value <- -qchisq (conf, 1);
plot (m, LR, type = "l," col = "blue," lwd = 2, main = "Kupiec
Test (1995)," xlab = "The number of times a risk occurs m")
abline (h = critical_value, col = "red," lwd = 3).
LR_C <- -which(LR <= critical_value).
abline (v = c(LR_C[1], LR_C[length(LR_C)]),
lty = "dashed")
LR_C.
```

## Data Availability

The data used to support the findings of this study are available at <https://www.gtarsc.com/>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was funded and supported by the Science Foundation of Ministry of Education of China (grant no. 19YJC790128). The authors are grateful to all the individuals who conducted data collection and data entry. The authors would like to thank Editage (<http://www.editage.cn>) for English language editing.

## References

- [1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [2] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

- [3] J. J. Zou, Z. Y. Zhang, and Z. Qin, "Research on the application of GARCH model to calculate the value at risk of stock market in China," *Systems Engineering Theory and Practice*, vol. 05, pp. 20–25+135, 2003.
- [4] Y. Liu and Z. Yu, "Experience and reference of agricultural product price risk management using futures market in the United States," *Chinese Rural Economy*, vol. 05, pp. 65–72, 2008.
- [5] H. Jiang and D. D. Qu, "An empirical study on risk management of CSI 300 stock index futures based on VaR," *Economic Problems*, vol. 12, pp. 119–122, 2008.
- [6] F. L. Yang, "An empirical study on VaR calculation of securities Investment fund based on GARCH Model," *Economic Problems*, vol. 394, no. 6, pp. 87–91, 2012.
- [7] C. Zhang, Y. Ding, and W. J. Wang, "An innovation of estimating value at risk of international carbon market: conditional autoregressive value at risk model with refinements from extreme value theory," *Chinese Journal of Management Science*, vol. 23, no. 11, pp. 12–20, 2015.
- [8] S. G. Tao and J. Y. Chen, "China's Foreign exchange reserve safety management stress test and Empirical Analysis," *Finance and Economics*, vol. 346, no. 1, pp. 1–11, 2017.
- [9] Q. Yang and J. J. Yang, "The ripple spillover effect between Shibor online lending rate and private lending rate," *Research in Financial Economics*, vol. 33, no. 4, pp. 14–23, 2018.
- [10] M. Karmakar and S. Paul, "Intraday portfolio risk management using VaR and CVaR: A CGARCH-EVT-Copula approach," *International Journal of Forecasting*, vol. 35, no. 2, pp. 699–709, 2019.
- [11] W. H. Lin, M. Q. Chen, L. G. Zhou, and X. X. Meng, "Analysis of risk prediction ability of Chinese stock market based on Var-GARCH model family," *Statistics & Decisions*, vol. 35, no. 21, pp. 151–155, 2019.
- [12] X.-L. Gong, X.-H. Liu, and X. Xiong, "Measuring tail risk with GAS time varying copula, fat tailed GARCH model and hedging for crude oil futures," *Pacific-Basin Finance Journal*, vol. 55, no. 55, pp. 95–109, 2019.
- [13] C. Liu and B. B. Liu, "Spillover effects of financial institutions' tail risks: an improved asymmetric CoVaR model," *Statistical Research*, vol. 37, no. 12, pp. 58–74, 2020.
- [14] Y. Liang, "Innovation and application of rebar futures hedging method -- analysis based on GJR-GARCH-VAR model," *Price Theory and Practice*, vol. 434, no. 8, pp. 108–111, 2020.
- [15] C. Alexander, E. Lazar, and S. Stanescu, "Forecasting VaR using analytic higher moments for GARCH processes," *International Review of Financial Analysis*, vol. 30, no. 30, pp. 36–45, 2013.
- [16] P. H. Kupiec, "Techniques for verifying the accuracy of risk measurement models," *Journal of Derivatives*, vol. 3, no. 2, pp. 73–84, 1995.

## Research Article

# Using Proximity Graph Cut for Fast and Robust Instance-Based Classification in Large Datasets

Stanislav Protasov  and Adil Mehmood Khan 

*Machine Learning and Knowledge Representation Lab, Innopolis University, Innopolis 420500, Innopolis, Russia*

Correspondence should be addressed to Stanislav Protasov; [s.protasov@innopolis.ru](mailto:s.protasov@innopolis.ru)

Received 17 May 2021; Accepted 29 October 2021; Published 29 November 2021

Academic Editor: Siew Ann Cheong

Copyright © 2021 Stanislav Protasov and Adil Mehmood Khan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

K-nearest neighbours (kNN) is a very popular instance-based classifier due to its simplicity and good empirical performance. However, large-scale datasets are a big problem for building fast and compact neighbourhood-based classifiers. This work presents the design and implementation of a classification algorithm with index data structures, which would allow us to build fast and scalable solutions for large multidimensional datasets. We propose a novel approach that uses navigable small-world (NSW) proximity graph representation of large-scale datasets. Our approach shows 2–4 times classification speedup for both average and 99th percentile time with asymptotically close classification accuracy compared to the 1-NN method. We observe two orders of magnitude better classification time in cases when method uses swap memory. We show that NSW graph used in our method outperforms other proximity graphs in classification accuracy. Our results suggest that the algorithm can be used in large-scale applications for fast and robust classification, especially when the search index is already constructed for the data.

## 1. Introduction

Proximity graphs are a practical class of graphs with applications in multiple areas. For example, they are used for motion planning, as rapidly exploring random trees in [1, 2] and minimum spanning trees in clustering [3]. Most importantly, they lay in the core of  $\log(|V|)$  search time data structures for large-scale multidimensional data indexing, where  $|V|$  stands for dataset cardinality.

Instance-based classification (IbC) methods store items (instances) from the training dataset as part of the classifier. Unlike other methods such as decision trees and artificial neural networks, the IbC algorithms do not estimate the classifier function from the training data in advance; instead, they store training data and derive a class label from an examination of the unseen sample's nearest neighbours at test time [4]. Such methods easily adopt to unseen data by extending the list of stored samples.

Among pure IbC methods, we can identify k-nearest neighbours (kNN) with different variations [5–7], piecewise functions (e.g., splines [8]), and kernel approximators, such as

radial basis function (RBF) interpolation methods. Splines and kernel approximation are frequently used in numerical methods for equation solving. At the same time, kNN is considered both a good basis for novel machine learning approaches [5] and useful tool for complex applied machine learning tasks [9].

Decision trees, support vector machines (SVM), self-organizing maps [10], learning vector quantization [11], and RBF networks [12] can also be attributed to instance-based methods. However, we avoid such wide interpretation, as these methods do not require storing original samples for classification.

In this paper, we address the problem of classification speed in the context of IbC using proximity graphs. Large datasets often appear in content recommendation tasks of internet services: search, online shops, streaming, or social networks. Classification accompanies recommendations in such tasks as sentiment analysis [13] or auto-labelling [14]. As such systems serve millions or even billions of requests per day, this makes a millisecond algorithm overhead scale into hours and days of CPU time every day. This is a notable financial load.

The K-nearest neighbours method estimates the class label of a test sample based on the labels of its closest neighbours from the training set. Distance is defined with some metric function. To avoid computing the distance of the test sample to every item in the training data, indexing is employed. This allows achieving sublinear classification time with various data structures such as trees, graphs, and inverted indices.

Graph-based indexing utilizes the idea that a dataset can often be represented in a metric space. Thus, adding a distance metric for nodes and requiring edges to represent close neighbourhoods, we can benefit from greedy-like search algorithms, traversing the graph with preliminary knowledge about the desired direction towards a query sample.

In instance-based methods, algorithm execution time depends on the number of stored instances, while model-based methods depend on the number of model parameters. Thus, IbC should offer both asymptotically and practically fast methods even for very large datasets, which requires constructing additional structures to navigate the data, such as search indices. We consider the case where index creation is indispensable and try to reduce the classification wall time. More specifically, we show how navigable small-world (NSW) [15] and hierarchical navigable small-world (HNSW) graphs [16] properties can be utilized in machine learning. We propose an improvement to NSW and HNSW index data structures, which results in 2–4 times sustainable speedup on average compared to 1-NN classification baseline.

The contribution of this work can be summarized as follows:

- (i) We propose a new instance-based classification approach, which utilizes properties of NSW and HNSW index data structures to achieve 2–4 times 1-NN classification speedup.
- (ii) Our proposed methods show a 2-order time improvement when used with a memory swap file.

The rest of the paper is organized as follows. Section 2 discusses different indexing strategies for large multidimensional datasets. Section 3 covers both algorithm construction and theoretical justification of the proposed idea. Section 4 describes the experimental setup, datasets, hardware, and ways of comparison of our method to other approaches. Section 5 is devoted to the numerical results of our experiments. In this section, a proposed classifier is assessed in terms of speed and accuracy, and the NSW graph is compared to other proximity graphs. Section 6 analyses obtained numbers and state conditions in which using our method is beneficial and discusses interesting properties. Section 7 closes the paper with a highlight of major outcomes.

Our experiments, results, and code are available in the GitHub repository (<https://github.com/IUCVLab/proximity-cut>).

## 2. Related Work

This section overviews how a problem of large dataset indexing is solved in the industry right now and considers indexing application to instance-based learning.

The problem of large-scale indexing for multidimensional data arose together with efficient methods of document embedding using artificial neural networks [17–19]. The Internet became an endless source of data, including web pages, Wikipedia articles (<https://dumps.wikimedia.org/>), scientific papers ([https://en.wikipedia.org/wiki/Web\\_of\\_Science](https://en.wikipedia.org/wiki/Web_of_Science)) and images ([https://en.wikipedia.org/wiki/Google\\_Photos](https://en.wikipedia.org/wiki/Google_Photos)) which form collections with  $10^5$  to  $10^{10}$  items in each. Contemporary research in natural language processing also requires bigger datasets to prove robustness [20]. As there is no exact borderline, we address these sizes as large. Search on such a scale can no longer be exhaustive. To be practical, it requires sublinear time. For an unsorted collection, this means that, on classical computers, we need to use approximate methods, also known as approximate nearest neighbour search (ANNS). Exact and approximate nearest neighbour searches are the core tool of many metric-based machine learning algorithms, including kNN classification, k-means, k-centroids, and DBSCAN clustering. We discuss three approaches to building large dataset indices to guarantee fast ANNS. In this paper, we assume that the data can be represented in a metric or in a vector space depending on indexing method.

*2.1. Tree Based.* The invention of AVL-trees and B-trees made search trees a powerful tool to build  $\mathcal{O}(\log(|V|))$  indices for numerical data. Quad-trees [21] and KD-trees [22] have been used for indexing multidimensional vector data. Unfortunately, their usage is limited to low dimensions because they suffer from the curse of dimensionality. For example, indexing  $10^9$  items with KD-tree will utilize at most  $\log_2(10^9) \approx 30$  first dimensions of the vector, while contemporary deep models produce 100–1000 dimensional vectors, such as 768-dimensional BERT embeddings as in [17]. For such big vectors, the search procedure will not account the majority of dimensions. Thus, it cannot guarantee a low distance from the query to the obtained “neighbours.” To solve this problem, authors of annoy (<https://github.com/spotify/annoy>) apply random projections instead of predefined vector dimensions and multiple search trees, which is proven an efficient way of reducing data dimensions for large datasets [23]. A collection of trees can achieve high ANNS accuracy with a small search time. However, it comes with a significant memory overhead as each tree consumes memory proportional to dataset size.

*2.2. Inverted Index Based.* An inverted index file (IVF) is an efficient method for text indexing, as it utilizes statistical properties of human language and discrete word representation. Since multidimensional vector data is continuous, various metric-based discretization approaches, such as vector quantization and vector clustering, are used to prepare the so-called vocabularies—finite collections of vectors, representing data clusters [24, 25]. Current works discuss methods to avoid the problem from which IVF suffers in natural language processing. Word frequencies in natural language are different, leading to a skewed index. The proposed methods include different k-means implementations to form a vocabulary and product quantization

technique for a better space partitioning as in [24]. Though IVF is a fast and scalable method with promising search speed and ANNS accuracy, it requires significant additional memory [26].

*2.2.1. Proximity Graph-Based.* A proximity graph is a graph with a distance metric defined for vertices [27]. In practice, the metric can be defined not for all pairs of vertices, and edge in such a graph exists if and only if (or with higher probability if) its vertices satisfy particular geometric requirements; for example, if they are close in metric space. Building a proximity graph with dataset items as vertices can be understood as building a road network. It allows the search algorithm to travel starting from the arbitrary vertex in the direction of the search query by following some greedy strategy.

There are multiple types of deterministic and probabilistic proximity graphs, including minimum spanning trees (MST), relative neighbourhood graphs (RNG), Gabriel graphs, and Delaunay triangulations [28]. Among them, there are a group of data structures based on the idea of small-world graphs. A major feature of small-world (SW) networks [29] compared to other graphs is that together with edges connecting tight neighbourhoods (compare with local roads), they also include “distant” edges (compare with flights). In this example, “distant” means the edge which connects near-clique clusters which do not share any nodes. The existence of such “distant” edges leads to  $\mathbb{E}(\log(|V|))$  expected shortest path length (edges count) between arbitrary pair of vertices which was proven in [30]. NSW and HNSW graphs [16] place graph vertices into a metric space, introducing a highly efficient greedy-like algorithm to traverse the graph. The authors claim that their data structure approximates Delaunay triangulations in high dimensions and propose a novel method of constructing SW graphs in metric space, which has  $\mathcal{O}(|V|\log|V|)$  construction time complexity and  $|V| * d$  memory overhead, where  $d$  represents the number of dimensions in data vectors.

### 3. Methodology

This work is dedicated to an improvement of the IbC methods. Given a big multidimensional dataset, we can achieve good results with the kNN classifier: using existing search indices, we can guarantee  $\log(|V|)$  search time without sacrificing accuracy. These methods are competitive and are used in recent applied research as in [5, 9]. Still, we must consider speed in terms of both theoretical complexity and wall time, as large-scale services are sensitive to even a millisecond overhead in a single function. Proximity graphs (NSW and HNSW in our case) built upon the unlabelled collection achieve expected logarithmic nearest neighbours search time with a greedy-like algorithm. Therefore using a graph-based index, the kNN classifier can run in logarithmic time, which can hardly be improved in terms of theoretical complexity. On the contrary, we concentrate our efforts on utilizing label information to reduce practical computation time and preserve classification accuracy.

Our work is limited to the assumption that the dataset has a property of a metric space, with high probability nearest neighbours (in terms of the metric) of an item belonging to the same class as the item. This assumption is sometimes called the compactness hypothesis [31]. This assumption is general for all metric-based machine learning methods, including both unsupervised (e.g., k-means and DBSCAN) and supervised (e.g., kNN and linear models) approaches.

The core theoretical idea of the proposed method lies in the fact that a proximity *graph cut* can be used to approximate the class boundaries. A graph cut is a *set of edges where the source and destination vertices belong to different classes*. A graph cut example is given in Figure 1.

The outline of our methodology is the following:

- (i) The same as in kNN classification, we accept the compactness hypothesis. This allows making an assumption that classes are closed volumes in  $\mathbb{R}^n$ .
- (ii) kNN classifier assigns a class to an unseen sample based on implicit class border estimation with neighbours voting. Border estimation can be replaced with faster border crossing detection, based on The Jordan Curve theorem [32] and its extension [33].
- (iii) The proposed border crossing detection technique is based on traversing NSW and HNSW graphs with a greedy algorithm (beam search). This algorithm produces the near-shortest path between the starting point and unseen sample, which is shown to be  $\log(|V|)$  long [30].

Further paragraphs expand the listed ideas.

The Jordan curve theorem guarantees that if there are two classes in  $\mathbb{R}^2$  where one class is surrounded by a closed curve, then an arbitrary path between two points belonging to different classes intersects this border an odd number of times and at least once. We apply multidimensional consequence [33] of the theorem following the compactness hypothesis: for an arbitrary path (edges sequence) in proximity graph, single class border crossing can be used to indicate class change. An exact crossing point location is not needed. It is enough to account edges where vertices have different labels, that is, which belong to the graph cut. The method also works even if the class is not a single cluster but a set of disconnected clusters.

Speed characteristics of our implementation are derived from two properties of NSW and HNSW graphs. Firstly, in small-world graphs (by definition), the shortest path between two arbitrary vertices has expected logarithmic length. Thus, any query search algorithm can start from a random graph node and find the nodes closest to the query node in  $\log(|V|)$  time on average if the shortest path is known. Secondly, the greedy beam search algorithm in a dense enough NSW graph produces a path with  $\mathcal{O}(\log(|V|))$  edges with a probability of  $1 - o(1)$ , which is shown by the experiments in [15] and theoretical proof in [30]. Greediness here is defined as selecting the next vertex from the neighbours, such that it is the closest (in metric space) to the

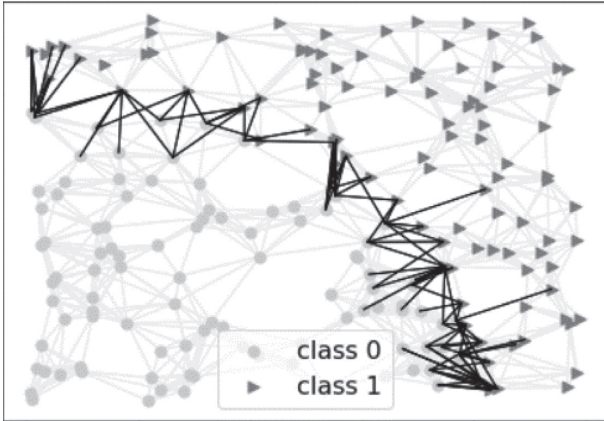


FIGURE 1: NSW index graph is built on 2 data classes (circles and triangles) in metric space. Black edges represent graph cut. They have source and destination vertices in different classes.

destination. Euclidean and angular metrics are the most popular for vector space datasets. In other words, if algorithms search for query node  $q$  starting in node  $n$ , at each step, it should move to such a neighbor  $n'$  of  $n$ , which has the smallest distance to  $q$  in terms of metric. The aforementioned properties guarantee that this search, on average, will successfully converge in  $\log(|V|)$  time.

To sum up, for a classification task, class boundary estimation is not needed. Instead, it is enough to detect the event of boundary-crossing. This useful observation allows reducing computational overhead, which is valuable for large systems. For implementation, we use both properties of NSW graphs to efficiently obtain a path in a graph and combine them with the Jordan curve theorem.

A formal description of our method is as follows: let a class be a set of volumes in a multidimensional metric space. As we mentioned earlier, we approximate class boundaries with a graph cut. Boundary-crossing occurs if an edge in a path belongs to a cut. Thus, we propose the following algorithm of classification. Given a sample vector that needs to be classified, a search is started from a graph vertex with a random index taken from a uniform distribution. A vertex choice procedure does not influence the result as any shorted path in the NSW graph has a logarithmic length. Then, graph nodes are greedily traversed towards the given vector, and the algorithm stops if it cannot find a closer neighbour. If class labels are available for all vertices, only the last border crossing is needed (if any happened) to assign a class to the sample.

The algorithm works for both binary and multiclass problems. Generalization to multiclass comes from applying the one-vs-all technique: the last border crossing can be considered as moving from the united “all” class to “one” class.

The proposed approach for NSW graphs is summarized in algorithm 1, which is an approximate equivalent to 1-NN classification. The only difference for HNSW implementation requires to start at the top level of the graph and repeat the same algorithm at the lower levels until  $0^{th}$  convergence. This also means that the choice of vertex  $v$  for HNSW graphs is deterministic.

The Euclidean distance for normalized features is used as a metric in tests if other is not mentioned explicitly. This choice is reasonable in many practical applications as it captures the human perception of “closeness”: a significant change in the value of one feature or insignificant changes in multiple features should not dramatically influence the distance metric.

NSW graph allows using the proposed algorithm together with the kNN voting procedure. That is, classification can be run multiple times to achieve better accuracy. In the case of the HNSW graph, the search procedure always uses a predefined starting point. Thus applying voting will not bring any benefit.

## 4. Experiments

We implement our method to improve the original NSW and HNSW graph search procedure. Our experiments study our method from three points of view:

- (i) NSW graph ANNS quality compared to other proximity graphs,
- (ii) Classification accuracy compared to 1-NN,
- (iii) Time improvement compared to baseline 1-NN classification with HNSW.

We understand *graph quality* criterion as an ability to provide a better approximation for the ANNS problem. Application of proximity graphs is always a trade-off between speed of neighbourhood exploration and percentage of actual neighbours retrieved (which can be referred to as recall metric). Experiments show that graph choice is good. The other two criteria are devoted to the method assessment for both accuracy and time. 1-NN classification is used as a baseline. The first reason is that a proposed method is an approximation for this classification technique, so we assess our solution compared to the best achievable nonexhaustive 1-NN classification method done with HNSW. Our target is to achieve better practical time with acceptable accuracy loss. The second reason is that based on 1-NN results, one can easily extrapolate the time cost for an arbitrary kNN classification method.

To compare a *graph type* used in our method with all graph types presented in survey [28], we run experiments with 3 UCI datasets mentioned in a paper: Dermatology (<https://archive.ics.uci.edu/ml/datasets/dermatology>), Isolet (<https://archive.ics.uci.edu/ml/datasets/isolet>), and Image Segmentation ([https://archive.ics.uci.edu/ml/datasets/Image + Segmentation](https://archive.ics.uci.edu/ml/datasets/Image+Segmentation)). Non-normalized Euclidean distance is used for Isolet and Image Segmentation to reproduce original accuracy results. For the Dermatology dataset, as it contains both categorical and numerical features, we implement and use the Heterogeneous Value Difference Metric (HVDM) defined in [34] (implemented in <https://github.com/IUCVLab/proximity-cut/blob/master/modules/tools/hvdm.py>). For this set of experiments, NSW implementation from our repository was used.

By construction, the HNSW graph contains the NSW graph as a subset on level 0. Thus, all remaining experiments are conducted with hnsplib implementation where NSW graphs are extracted from the parent HNSW graph.

The *speed and accuracy* of classification are compared to the 1-NN classifier on the medium-size road signs dataset [35] with 43 classes (images are resized to 256-dimensional representation, 10% test set) and two large binary classification datasets HIGGS ( $1.1 * 10^7$  items) and SUSY ( $5 * 10^6$  items) [36] (5% test set). Detailed speedup statistics are measured using another medium-size Cover Type dataset [37].

In this work, we do not claim to invent or improve existing classification algorithm(s). These types of work require exhaustive testing for all marginal cases. We aim to apply state-of-the-art indexing infrastructure and show what can we gain from it (in speedup) and at what cost (in accuracy). As NSW and HNSW's time complexity characteristics have already been studied [15, 16] and proven [30], in this paper, we focus on practical improvement. Since algorithm time is shown to depend on dataset dimensionality and size, we cover both aspects.

All experiments were conducted at a 64-bit Windows 10 laptop using a single CPU core. The laptop has AMD Rizen 3 3200U chip with 2.6 GHz frequency and 2 physical cores. 6 GB RAM is installed in the machine with 3.5 GB available for experiments. Python implementations were launched at Jupyter notebooks with Python 3.7.4. C++ implementation was compiled with GCC 7.4.0 using Windows Subsystem for Linux (Ubuntu 18.04).

## 5. Results

*5.1. Graphs Comparison.* The choice of NSW graph was validated by comparing accuracy results with other proximity graphs, namely, relative neighbourhood graphs (RNG), Gabriel graphs, and minimum spanning trees (MST). We compare our results against implementations from [28] on UCI datasets proposed in the paper. The authors intentionally focused only on classification accuracy and omitted speed comparison. Thus, we can compare our results by accuracy only. On Isolet, our method outperformed RNG graphs classification with 88.5% accuracy against 88.1%. With Dermatology data, it achieved the same 95.65% accuracy as RNG graphs, which can be the result of very small dataset and almost complete graph. With Image Segmentation data, our method achieved 87.5% accuracy which is only slightly worse than 1-NN (90.3%) and RNG (88.8%). Detailed results are given in Table 1.

*5.2. Average Classification Accuracy and Time.* In NSW and HNSW graphs, the construction phase depends on hyperparameter  $M$ , which linearly influences the number of graph edges. According to original papers, increasing this parameter can bring better accuracy results paying with additional index memory. We compared how this parameter influences baseline 1-NN classification and the proposed method on two large datasets. Results are provided in Figure 2.

We also studied how dataset size and graph edges density controlled by NSW hyperparameter  $M$  influence average classification time and accuracy. We compared baseline 1-NN classification with the proposed method on three

datasets with different hyperparameter values. With comparable accuracy number, our method showed sustainable speedup on both graphs. Time and accuracy results are given in Table 2.

NSW and HNSW graphs are built by a deterministic procedure, but their properties depend on the order of inserting and the structure of the dataset itself. Search and classification time for such graphs can only be estimated in terms of expected values. We used a medium-size Cover Type dataset to compare classification times distributions. While both baseline 1-NN classification and NSW-based proposed method show comparable time spread growth, the HNSW-based method shows extremely good numbers. For visual comparison, please refer to Figure 3.

*5.3. Service Reliability Comparison.* Indexing structures are used in different search tasks to improve the quality of service. Service reliability is frequently assessed in terms of 95<sup>th</sup> or 99<sup>th</sup> percentiles. Thus, we prepared a percentile-based comparison of the proposed method to the 1-NN baseline, which shows 1.5–2 times speedup for NSW-based implementation and 4+ times comparison for NSW-based. The numbers are given in Table 3.

## 6. Discussion

The experiments on graph comparison show that NSW-based classification accuracy outperforms sparse Gabriel and MST graphs in all experiments. Also, the resulting classifier shows a behavior very similar to RNG-based implementation for each of the experiments. Considering this, we refer to [38], which states that although 2-dimensional case RNG construction requires  $\Theta(|V|\log|V|)$  operations, the  $k$ -dimensional and non-Euclidean metric spaces will require  $\mathcal{O}(|V|^3)$  operations. NSW graphs are constructed in  $\mathbb{E}(|V|\log|V|)$  which is a significant gain for large-scale datasets.

Comparison of the proposed method accuracy to the baseline 1-NN classifier shows that the proposed method is slightly under the baseline, HNSW-based implementation in all tests 0–7 percent points worse than 1-NN. But we also observed that NSW-based implementation asymptotically tends to the baseline (see Figure 2) with the growth of graph density, defined by  $M$  hyperparameter. For the small size of the dataset, the NSW-based implementation showed significant speedup, thus using  $k$ -NN classifiers where  $k < \text{speedup}$  built upon the proposed method will achieve asymptotically better accuracy for the smaller time. HNSW-based implementation at the same time shows consistent speedup for all graph sizes and densities. The loss in accuracy can be explained by the fact that the final step of our method returns only an approximation of the nearest neighbour. Thus, in future studies, improvement of the algorithm for a better approximation at the last search iterations can be addressed to compete with the baseline in accuracy while preserving similar time. We can also say that speedup was observed at all experiments for all sizes of the datasets and graph densities. A general observation is that speedup tends



```

Input: NSW - dataset index;  $x$  - sample to be classified
Result: class label
 $v \leftarrow$  random vertex from NSW;
 $d_{new} \leftarrow distance(v, x)$ ;
 $class_{new} \leftarrow v.class$ 
repeat
   $d \leftarrow d_{new}$ ;
   $class \leftarrow class_{new}$ ;
  // closest to  $x$  neighbours of  $d$ 
   $v \leftarrow closest(v.neighbours | x)$ ;
   $d_{new} \leftarrow distance(v, x)$ ;
   $class_{new} \leftarrow v.class$ ;
  // until we can't get closer to  $x$ 
until  $d_{new} > d$ ;
return class

```

ALGORITHM 1: Greedy graph-based in NSW for multiclass classification.

TABLE 1: Comparison of classification accuracy for different proximity graphs against SVM classifier with polynomial kernel.

Dataset	Accuracy (%)					
	SVM	kNN	RNG	MST	Gabriel	NSW
Dermatology ( $k = 10$ )	96.9	98.6	95.6	86.7	65.4	<b>95.65</b>
Isolet ( $k = 7$ )	96.0	91.9	88.1	N/A	N/A	<b>88.5</b>
Image Segmentation ( $k = 1$ )	92.9	90.3	88.8	76.7	82.1	<b>87.5</b>

$k$  refers to the best kNN hyperparameter. NSW is our choice. Bold shows our study implementations.

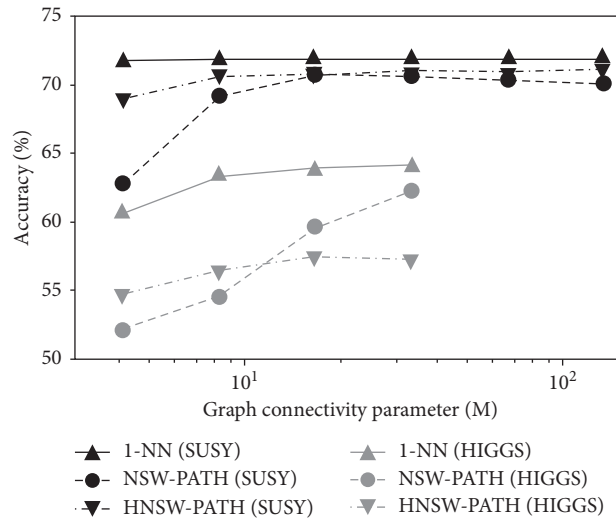


FIGURE 2: Accuracy comparison of the proposed NSW-PATH and HNSW-PATH classifiers with 1-NN baseline at two large-scale datasets SUSY (5M items) and HIGGS (11M items).

to be bigger for smaller datasets, but for  $10^7$ -scale datasets, it remains on a significant level.

We separately stop on the speedup value for the SUSY dataset with high density ( $M = 128$ ), which shows 1.17 times improvement for NSW and enormous 148.1 times improvement for HNSW (see Table 2). The experiment was conducted multiple times with different system parameters showing the same result. We found out that this behavior fully depends on using swap memory. For the small Road Signs dataset, doubling graph density (parameter  $M$ ) implies

linear absolute time growth for 1-NN classification, whereas for SUSY, we observe 3-order time growth compared to 16 times density growth ( $M = 8, 128$ ). We detected that, at the test machine, the process could only allocate  $\sim 3.5$  GB of physical RAM, while the data structure required almost 5 GB of virtual memory. Thus, a significant part of the data was dumped into HDD. This slowed down both the index construction phase and classification. The HNSW graph architecture uses exponentially smaller parts of memory for higher graph levels according to the construction process.

TABLE 2: Time and accuracy comparison of a proposed graph-based method implemented with NSW and HNSW graphs. Graph parameters are given as follows: dataset name (number of items, HNSW connectivity parameter  $M$ ).

Graph parameters	Time (ms), accuracy (%)						Speedup NSW vs. 1-NN	Speedup HNSW vs. 1-NN
	1-NN		NSW-PATH		HNSW-PATH			
Road signs (10K, $M = 8$ )	209	92.57%	25.2	92.57%	72.4	92.57%	<b>8.3</b>	2.89
Road signs (10K, $M = 64$ )	1615	82.57%	151	82.57%	186	82.57%	<b>10.7</b>	8.68
SUSY (5M, $M = 4$ )	78.16	71.74%	22.41	62.74%	15.01	69.02%	3.49	<b>5.21</b>
SUSY (5M, $M = 8$ )	89.5	71.87%	49.43	69.19%	21.43	70.58%	1.81	<b>4.18</b>
SUSY (5M, $M = 128$ )	8538	71.86%	7271	71.12%	57.66	70.1%	1.17	<b>148.1</b>
HIGGS (11M, $M = 32$ )	596.08	64.09%	253.61	62.27%	218.11	57.14%	2.35	<b>2.73</b>

Bold values refer to the best achieved speedup.

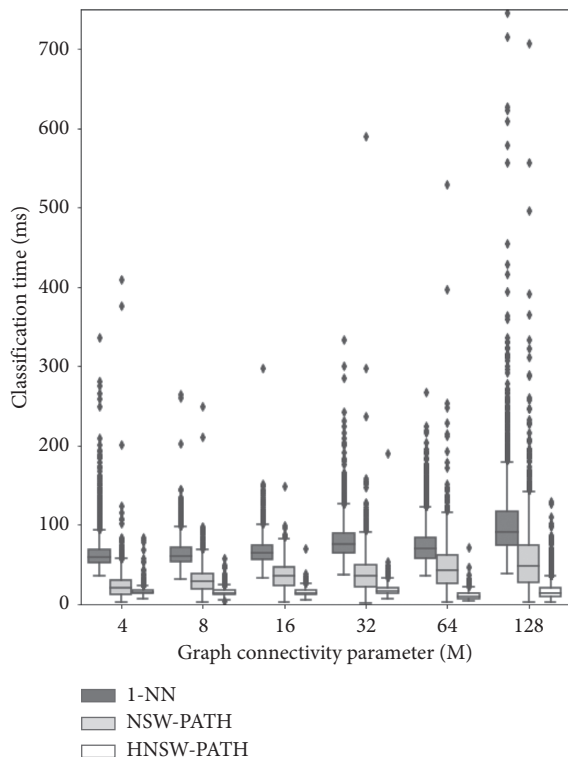


FIGURE 3: Comparison of classification time distribution on medium-size Cover Type dataset.

TABLE 3: Algorithm speedup comparison with respect to different graph densities (tuned by NSW  $M$  hyperparameter) and percentile. Speedup values compared to baseline HNSW-based 1-NN classification are provided for 99<sup>th</sup> percentile (p99), 95<sup>th</sup> percentile (p95), and average classification time (Avg).

M	NSW-PATH			HNSW-PATH		
	p99	p95	Avg	p99	p95	Avg
4	2.44	2.17	2.85	5.59	4.55	4.16
8	1.56	1.71	2.12	4.07	4.36	4.34
16	1.47	1.42	1.84	4.31	4.18	4.45
32	1.55	1.55	2.10	4.51	4.14	4.62
64	1.32	1.26	1.61	<b>6.44</b>	<b>6.50</b>	<b>7.10</b>
128	1.45	1.42	1.82	4.11	3.18	5.30

Thus, only the last steps of the algorithm require 0-level graph traversal, while the level with higher numbers can

easily fit into physical RAM. This makes HNSW-based classification very promising in cases of low physical RAM.

Considering potential service quality, we state that both NSW and HNSW-based implementations of the proposed method show comparable speed improvement for the average and high-percentile classification time. We also observe that this speedup does not depend on graph density as shown in Table 3.

To sum up, we define two potential applications for our proposed method. Firstly, both NSW- and HNSW-based algorithms can be used as dedicated classifiers for datasets of all sizes to improve absolute classification time still sacrificing a few percent points of accuracy. NSW-based classifier, in this case, will offer asymptotically growing accuracy for denser graphs, HNSW-based version will be extremely time-efficient when RAM does not fit the dataset, and a swap file is used. Secondly, for small- and medium-sized datasets, NSW-based kNN implementation can offer better accuracy for the same classification time.

We want to mention that although we discussed competing indexing approaches (IVF, trees) in Section 2, we cannot implement our methods with these data structures, as they do not produce the graph cut we use in both our methods.

## 7. Conclusion

In this paper, we introduced the novel approach to instance-based classification. The approach improves existing NSW and HNSW data structures for faster classification of unseen items. It simplifies the original search algorithm and connects it with the Jordan curve theorem. The method achieved sustainable 4x speedup at real medium-scale datasets and more than 2x speedup for a large dataset using production hswlib C++ library while preserving asymptotically close accuracy. It also showed extremely good time improvement if used with swap file.

We analysed our solution execution time, and we can say that it provides significantly better reliability in terms of 95<sup>th</sup> and 99<sup>th</sup> classification time percentile compared to the 1-NN classification baseline.

Our future research can target improvement of nearest neighbour estimation in the end of the proposed search algorithm which will improve classification accuracy while keeping time smaller.

## Data Availability

The machine learning classification datasets (Dermatology, Isolet, Image Segmentation, Cover Type, SUSY, and HIGGS) used to support the findings of this study have been deposited in the UCI repository (<https://archive.ics.uci.edu/ml/datasets/dermatology>, <https://archive.ics.uci.edu/ml/datasets/isolet>, [https://archive.ics.uci.edu/ml/datasets/Image + Segmentation](https://archive.ics.uci.edu/ml/datasets/Image+Segmentation), <https://archive.ics.uci.edu/ml/datasets/HIGGS>, <https://archive.ics.uci.edu/ml/datasets/SUSY>, <https://archive.ics.uci.edu/ml/datasets/covertypes>). These prior studies (and datasets) are cited at relevant places within the text as references [28, 36, 37]. The machine learning classification dataset with road signs used to support the findings of this study has been deposited in the INI Benchmark Website <https://benchmark.ini.rub.de>. These prior studies (and datasets) are cited at relevant places within the text as references [35]. The code related to synthetic dataset generation used to support the findings of this study has been deposited in the GitHub repository <https://github.com/IUCVLab/proximity-cut>.

## Conflicts of Interest

Adil Mehmood Khan acts as an editor of Complexity and Robustness Trade-Off for Traditional and Deep Models special issue.

## Acknowledgments

This research has been financially supported by The Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK 000000D730321P5Q0002).

## References

- [1] S. M. Lavalley, "Rapidly-exploring random trees: A new tool for path planning," Technical Report, Iowa State University, Ames, Iowa, 1998.
- [2] V. Massagué Respal, D. Devitt, and R. Fedorenko, "Unmanned aerial vehicle path planning for exploration mapping," Technical Report, Innopolis University, Innopolis, Russia, 2020.
- [3] M. A. Carreira-Perpiñán and R. S. Zemel, "Proximity graphs for clustering and manifold learning," in *Proceedings of the 17th International Conference on Neural Information Processing Systems, NIPS'04*, pp. 225–232, MIT Press, Cambridge, MA, USA, December 2004.
- [4] E. Keogh, *Instance-Based Learning*, pp. 549–550, Springer US, Boston, MA, USA, 2010.
- [5] O. Anava and K. Levy, "k\*-nearest neighbors: from global to local," in *Advances in Neural Information Processing Systems*, pp. 4916–4924, 2016.
- [6] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based K -nearest neighbor classifier," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 3, pp. 1–25, 2019.
- [7] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.
- [8] L. Bookstein Fred, "Principal warps: thin-plate spline and decomposition of deformations," *Transactions on pattern analysis and machine intelligence*, vol. 11, 1992.
- [9] C. Sitawarin and D. Wágner, "Defending against adversarial examples with k-nearest neighbor," 2019, <https://arxiv.org/abs/1906.09525>.
- [10] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.
- [11] T. Kohonen, "Learning vector quantization," in *Self-organizing Maps*, pp. 175–189, Springer, Berlin, Germany, 1995.
- [12] D. Lowe, D. Broomhead, R. Signals, and M. U. K. Radar Establishment, *Radial Basis Functions Multivariable Functional Interpolation and Adaptive Networks. No. RSRE-MEMO-4148*, Royal Signals And Radar Establishment, Malvern, UK, 1988.
- [13] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment analysis on twitter data using knn and svm," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017.
- [14] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and knn models for the text classification," *Augmented Human Research*, vol. 5, no. 1, 2020.
- [15] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," *Information Systems*, vol. 45, pp. 61–68, 2014.
- [16] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," 2018, <http://arxiv.org/abs/1810.04805>.
- [18] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13*, pp. 2333–2338, Association for Computing Machinery, California, CA, USA, November 2013.
- [19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, <https://arxiv.org/abs/1301.3781>.
- [20] A. Onan, "Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach," *Computer Applications in Engineering Education*, vol. 29, no. 3, pp. 572–589, 2020.
- [21] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1–9, 1974.
- [22] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [23] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, no. 189–206, p. 1, 1984.
- [24] A. Babenko and V. Lempitsky, "The inverted multi-index," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1247–1260, 2015.
- [25] D. Baranchuk, A. Babenko, and Y. Malkov, "Revisiting the inverted indices for billion-scale approximate nearest neighbors," in *Computer Vision – ECCV 2018*, V. Ferrari,

- M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Springer International Publishing, Cham, Switzerland, pp. 209–224, 2018.
- [26] H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg, “Searching in one billion vectors: re-rank with source coding,” 2011, <https://arxiv.org/abs/1102.3828>.
- [27] L. Mathieson and P. Moscato, “An introduction to proximity graphs,” Springer International Publishing, Cham, Switzerland, pp. 213–233, 2019.
- [28] G. T. Toussaint and C. Berzan, “Proximity-graph instance-based learning, support vector machines, and high dimensionality: an empirical comparison,” in *Machine Learning and Data Mining in Pattern Recognition*, P. Perner, Ed., Springer, Berlin, Heidelberg, pp. 222–236, 2012.
- [29] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [30] L. Prokhorenkova, “Graph-based nearest neighbor search: From practice to theory,” 2019, <http://arxiv.org/abs/1907.00845>.
- [31] A. Arkadev and E. Braverman, *Computers and Pattern Recognition*, Thompson Book Co. Inc., Washington, DC, USA, 1966.
- [32] C. Jordan, *Cours d’analyse de l’École polytechnique*, Gauthier-Villars et fils, vol. 1, pp. 587–594, 1887.
- [33] E. Spanier, *Algebraic Topology*, p. 198, McGraw-Hill, New York, NY, USA, 1966.
- [34] D. R. Wilson and T. R. Martinez, “Reduction techniques for instance-based learning algorithms,” *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.
- [35] V. N. Sichkar and S. A. Kolyubin, “Effect of various dimension convolutional layer filters on traffic sign classification accuracy,” *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol. 19, no. 3, pp. 546–552, 2019.
- [36] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” *Nature Communications*, vol. 5, no. 1, p. 4308, 2014.
- [37] J. A. Blackard and D. J. Dean, “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables,” *Computers and Electronics in Agriculture*, vol. 24, no. 3, pp. 131–151, 1999.
- [38] G. T. Toussaint, “The relative neighbourhood graph of a finite planar set,” *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

## Research Article

# Equipment Maintenance Support Effectiveness Evaluation Based on Improved Generative Adversarial Network and Radial Basis Function Network

Zhen Li , Jianping Hao, and Cuijuan Gao

Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang 050003, China

Correspondence should be addressed to Zhen Li; [liyuchen88@outlook.com](mailto:liyuchen88@outlook.com)

Received 27 April 2021; Revised 22 May 2021; Accepted 6 July 2021; Published 8 November 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Zhen Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the lack of maintenance support samples, maintenance support effectiveness evaluation based on the deep neural network often faces the problem of small sample overfitting and low generalization ability. In this paper, a neural network evaluation model based on an improved generative adversarial network (GAN) and radial basis function (RBF) network is proposed to amplify maintenance support samples. It adds category constraint based on category probability vector reordering function to GAN loss function, avoids the simplification of generated sample categories, and enhances the quality of generated samples. It also designs a parameter initialization method based on parameter components equidistant variation for RBF network, which enhances the response of correct feature information and reduces the risk of training overfitting. The comparison results show that the mean square error (MSE) of the improved GAN-RBF model is  $5.921 \times 10^{-4}$ , which is approximately 1/2 of the RBF model, 1/3 of the Elman model, and 1/5 of the BP model, while its complexity remains at a reasonable level. Compared with traditional neural network evaluation methods, the improved GAN-RBF model has higher evaluation accuracy, better solves the problem of poor generalization ability caused by insufficient training samples, and can be more effectively applied to maintenance support effectiveness evaluation. At the same time, it also provides a good reference for evaluation research in other fields.

## 1. Introduction

As an important part of equipment maintenance support work, equipment maintenance support effectiveness evaluation has always been the research focus in the field of equipment support, which is mainly used to evaluate maintenance support strength, find out weaknesses in time, and formulate improvement plans, so as to improve the maintenance support effectiveness. In recent years, with the rapid development of machine learning technology, maintenance support effectiveness evaluation based on the neural network has become a research hotspot. The neural network has good characteristics of self-adaptive, self-learning, and strong fault tolerance, which can recognize new samples by learning database samples, and is widely used in prediction and evaluation [1–3]. Some scholars have carried out relevant research on

this method. For example, Wang et al. [2] proposed the joint operation effectiveness evaluation algorithm based on an adaptive wavelet neural network in combination with specific expert experience and an intelligent algorithm, aiming at the problems of subjectivity and uncertainty in joint operation effectiveness evaluation. To solve the effectiveness evaluation problem of complex, multifunctional, and poor samples, Liu et al. [4] constructed a system effectiveness evaluation model based on grey RBF neural network according to the three-layer structure of system effectiveness evaluation index system. In order to effectively evaluate and predict combat effectiveness of surface to air missile weapon system, Qiao and Zhao [5] reduced and decorrelated the original data through the principal component analysis method and then trained the BP neural network with the principal component as the input.

These researches based on a deep neural network greatly enhance the diversity of equipment maintenance support effectiveness evaluation methods and have foreseeable research prospects, but there is always a pain point: due to imperfect data collection system and nonstandard data collection procedures in the army for a long time, the current maintenance support data quality is generally poor and the type is relatively single, so it is impossible to form complete and sufficient training samples for neural network input [6, 7]. For this reason, the neural network evaluation of maintenance support effectiveness always faces problems such as overfitting of small samples and low generalization ability, which makes it difficult to achieve the ideal application effect in the fields of data classification and pattern recognition [8, 9]. The insufficiency of maintenance support training samples greatly limits the application of neural networks in maintenance support effectiveness evaluation.

Generative adversarial network (GAN) is a popular unsupervised learning algorithm in recent years. Compared with other generation models, it can avoid complex calculations and generate better image quality. Some scholars use its data generation characteristics to carry out relevant research in combination with their own fields. For example, Yin et al. [10] proposed a smoke feature extraction and detection model combining deep convolution generative adversarial network and convolution neural network to effectively reduce the false alarm rate of smoke detection. Yao et al. [11] put forward a fault diagnosis method based on generative adversarial network and residual network, which uses GAN to track the distribution of rail fastener failure data, balancing and expanding the existing data set. Luo and Wang [12] proposed a demosaicing image generation method based on a generative adversarial network in order to reduce noise and artifacts when reconstructing digital camera color images, which effectively eliminated image artifacts. These studies show that GAN can play a better role in dealing with problems such as insufficient data, unbalanced data, and more noise in some fields. Therefore, it is envisaged to apply the data generation characteristics of GAN to the field of equipment maintenance support to solve the problem of insufficient samples in neural network evaluation of maintenance support effectiveness. Based on the optimization of GAN and RBF, respectively, this paper combines the data generation of GAN with the training of RBF, constructs an improved GAN-RBF evaluation model, and compares it with other common models to verify its validity. In order to facilitate theoretical analysis and practical operation, this paper does not use complex deep convolution generative adversarial network and convolution neural network but chooses original GAN and RBF network with simple structure and concise training.

The arrangement of this paper is as follows. In Section 2, based on the brief introduction of GAN and RBF, the improved GAN-RBF evaluation model is constructed; the index system, data processing, evaluation criteria, and evaluation steps of maintenance support effectiveness evaluation are introduced in turn. In Section 3, the training results of the improved GAN-RBF model and the other three common models are shown in the form of graphs. In

Section 4, by analyzing the accuracy and complexity of four models, and comparing them with similar research, the validity of the improved model is verified. Finally, Section 5 concludes this paper with a discussion of future research extensions.

## 2. Method

### 2.1. Main Algorithms

*2.1.1. Generative Adversarial Network.* GAN is a deep learning model, which can automatically define potential loss function and learn data distribution of original sample set. It has become one of the most promising unsupervised learning methods on complex distribution in recent years, which has been successfully applied in image generation, image inpainting, 3D object generation, and other fields.

(1) *Basic Idea.* Based on the idea of game theory, GAN regards generative model and discriminative model as both sides of the game and trains the generative model and discriminative model alternately, so that the two models are both optimized and finally reach the Nash equilibrium: the samples generated by the generator are infinitely close to the real sample distribution, while the discriminator cannot distinguish the true from the false and predict the true probability for a given sample is 0.5.

(2) *Network Structure.* GAN is mainly composed of a generator and a discriminator. The structure is shown in Figure 1.

The generator is actually a kind of maximum likelihood estimation. After capturing the target sample distribution, the generator converts the original input into target samples of the specified distribution by converting the parameters. The discriminant is actually a dichotomy, which determines whether the data generated by the generated model is a true sample.

In practice, the fully connected neural network is generally used as a generator and discriminator.

(3) *Objective Function.* The objective function of GAN is as follows:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where  $G$  is the generator and  $D$  is the discriminator;  $D(x)$  is the probability that the discriminator judges that the sample is true, which is a real number in the range of 0–1;  $G(z)$  is the sample generated by the generator after receiving random noise;  $p_{\text{data}}(x)$  is the real sample distribution and  $p_z(z)$  is the generated sample distribution.

Formula (1) is a minimax optimization problem, which is essentially two optimization problems, respectively, corresponding to the minimization of the generative model and the maximization of the discriminative model in the alternating iterative training process [13]. The minimax objective function of GAN integrates the objective function of the

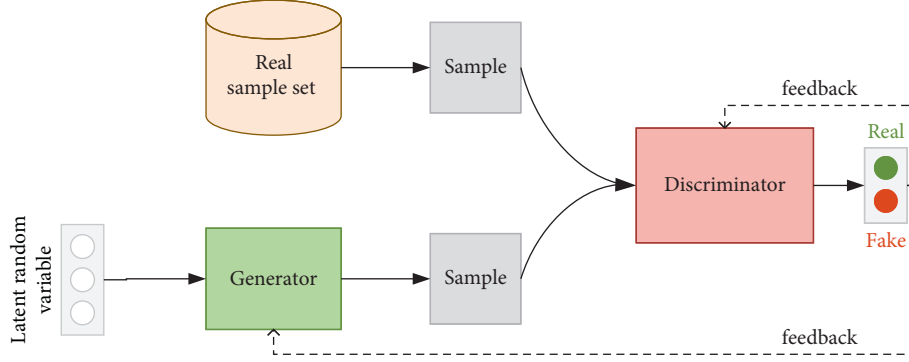


FIGURE 1: The topology of the generative adversarial network.

generative model and discriminative model and describes the alternating iterative training process, which achieves the perfect unity of mathematical form.

**2.1.2. Radial Basis Function Network.** RBF network is a kind of feedforward neural network with radial basis function as activation function, which has the characteristics of data-driven, independent data center searching and fast learning speed. It has a uniform approximation to nonlinear continuous function and avoids the emergence of local minimum problem, which is widely used in time series analysis, data classification, pattern recognition, and other fields [1, 14].

(1) *Basic Idea.* By taking the radial basis function (such as Gaussian function) as the base of the hidden unit to form the hidden layer, RBF can directly map input vectors to the hidden space without connecting by weight. The hidden layer transforms low-dimensional input into high-dimensional space, which makes a linear nonseparable problem in low-dimensional space linearly separable in high-dimensional space. The network output can be obtained by linear weighted summation of hidden cell output.

(2) *Network Structure.* RBF network usually has three layers, including the input layer, hidden layer, and output layer. The structure is shown in Figure 2.

The number of input layers is  $n$ , which is composed of signal source nodes and only transmits data information without any transformation of input information; the number of hidden layers is  $m$ , and the vector is mapped from the low dimension  $n$  to the high dimension space  $m$  by using radial basis function, so that linear separability in low dimension is transformed into linear separability in high dimension; the number of output layers is  $l$ , and the network output is obtained by linear weighted summation of hidden units outputs.

(3) *Activation Function.* The hidden layer neurons of the RBF network generally use the distance between the input vector and central vector (such as Euclidean distance) as the function independent variable and use radial basis function (such as Gaussian function) as activation function.

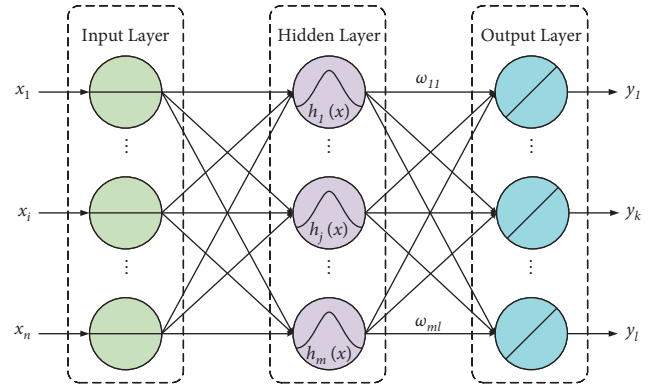


FIGURE 2: RBF network topology based on Gaussian kernel.

The activation function of the RBF network based on Gaussian kernel is as follows:

$$h_j(x) = \exp\left(-\frac{1}{2\sigma_j^2}\|X - c_j\|^2\right), \quad (2)$$

where  $X = [x_1, x_2, \dots, x_i, \dots, x_n]$ ;  $c_j$  is the central parameter of the  $j$ -th basis function,  $j = 1, 2, \dots, m$ ;  $\sigma_j$  is the width parameter of the  $j$ -th basis function, which is actually the variance of Gaussian function;  $\|X - c_j\|$  is the norm of vector  $X - c_j$ , which represents the distance between  $X$  and  $c_j$ .

The output of RBF network is as follows:

$$y_k = \sum_{j=1}^m \omega_{jk} h_j(x), \quad (3)$$

where  $\omega_{jk}$  is the weight of the basis function.

**2.2. Construction of Improved GAN-RBF Evaluation Model.** In this compound model, the maintenance support training samples are amplified by GAN, and then, the samples are used as the input of the RBF network. It is not a simple integration of the two but needs to increase relevant optimization according to the characteristics of maintenance support samples in order to ensure that GAN is suitable for the expansion of maintenance support samples and ensure that the RBF network can effectively train maintenance support samples.

First of all, the optimization of GAN is discussed. The composition of maintenance support data is complex; for example, it is divided by the functional system, including maintenance support command system, maintenance equipment support system, and maintenance equipment transportation delivery system; divided by operational level, including group army, integrated brigade, and integrated battalion; and divided by equipment type, including vehicles, ships, aircraft, ordnance, and armor. Complex categories bring some difficulties to maintenance support samples amplification based on GAN, so it is difficult to ensure the consistency between generated sample category and original category, which leads to the distortion of generated samples. Zhu et al. [15] pointed out that the L2 loss function can effectively improve the distortion of samples generated by the generator due to the inaccurate gradient of GAN's discriminative model. Therefore, this paper introduces the L2 loss function, defines a new class constraint on the basis of original antagonism constraint to determine the loss function of GAN synthetically, and ensures that GAN can generate ideal class samples without obvious distortion.

Secondly, the optimization of RBF is discussed. The parameter initialization of the RBF network has a great influence on its training. The traditional methods include the self-organizing selection center method and random selection method, both of which have some limitations. The former is more complex in the calculation, while the latter is generally applicable to the representative sample distribution [16]. They are not suitable for maintenance support samples with large volumes and complex categories. Therefore, aiming at the initialization of center parameter,

width parameter, and weight, an optimization method is designed with parameter components changing at equal intervals: define a center parameter  $C_j = [c_{j1}, c_{j2}, \dots, c_{ji}, \dots, c_{jn}]$ , width parameter  $\sigma_j = [\sigma_{j1}, \sigma_{j2}, \dots, \sigma_{ji}, \dots, \sigma_{jn}]$ , and weight  $\omega_j = [\omega_{j1}, \omega_{j2}, \dots, \omega_{jk}, \dots, \omega_{jl}]$ . The components of each center parameter are changed from small to large with equal spacing, and the spacing is adjusted by neurons number in the hidden layer so that weak input information can produce a strong response near the smaller center, and different input characteristics can be reflected more clearly in different centers, so that the center initialization is as reasonable as possible, reflecting the characteristics of Gaussian kernel.

*2.2.1. Amplification of Maintenance Support Samples.* The determination of the loss function of GAN is the most important work. Considering the optimization of GAN, two kinds of constraints are determined, which make the loss function of GAN more reasonable.

*(1) Adversarial Constraint.* Adversarial constraint refers to the following: in the alternating iterative training, generative model and discriminative model play games with each other, confront each other, constrain each other, and finally reach the state of mutual optimization where it contains the idea of a zero-sum game between generative model and discriminative model [17]. The constraint is expressed as the objective function of GAN. According to the objective function, the adversarial constraint is defined as follows [18]:

$$\min_G \max_D L^{\text{adv}} = E_{x \sim P_{\text{data}}(x)} [\log D(x)] + E_{z \sim P_z(z)} [\log(1 - D(G(z)))]. \quad (4)$$

According to formula (4), the adversarial loss function of the discriminative model can be easily determined as

$$L_G^{\text{adv}} = -E_{x \sim P_{\text{data}}(x)} [\log D(x)] - E_{z \sim P_z(z)} [\log(1 - D(G(z)))]. \quad (5)$$

Correspondingly, the adversarial loss function of the generative model is

$$L_G^{\text{adv}} = -E_{z \sim P_z(z)} [\log(1 - D(G(z)))]. \quad (6)$$

*(2) Category Constraint.* When the category of generated samples deviates greatly from the original category, serious distortion will occur to the generated samples, resulting in convergence failure in the whole training process. The ideal representation of sample generation is that the sample has the highest probability in the target category and the second probability in a similar category [19]. Therefore, category constraint is defined as follows:

$$\min_G L^{\text{cat}} = E_{x \sim P_{\text{data}}(x)} [\text{CE}(f(G(z)), h_d(f(x)))]. \quad (7)$$

where CE is the cross-entropy function,  $f$  is the depth neural network to be attacked, whose output is the class probability vector of the sample, and  $h_d$  is the class probability vector reordering function of the original sample and target class  $d$ .

According to formula (7), it is easy to determine the class loss function of the generative model as follows:

$$L_G^{\text{cat}} = -E_{x \sim P_{\text{data}}(x)} [\text{CE}(f(G(z)), h_d(f(x)))]. \quad (8)$$

There is no class constraint in the discriminative model, so it has no class loss function.

Considering the above two constraints, the final loss functions of discriminator and generator are as follows:

$$\begin{aligned} L_D &= -E_{x \sim P_{\text{data}}(x)} [\log D(x)] \\ &\quad - E_{z \sim P_z(z)} [\log(1 - D(G(z)))], \quad (9) \\ L_G &= \alpha L_D^{\text{adv}} + \beta L_G^{\text{cat}}, \end{aligned}$$

where  $\alpha$  and  $\beta$  are balance factors to adjust the two constraints.

When the loss function is determined, the GAN training is started, and iteration number is set as  $K$ , original sample



and generated sample set are  $X$  and  $Z$  respectively, the sample size is  $p$ , and sample size taken out in a single time is  $q$ . Usually, generator parameters are fixed first [20], the  $q$  sample is taken from the sample set  $X$  and  $Z$ , respectively, and discriminator parameter  $\theta_d$  is updated by random gradient rise:

$$\nabla_{\theta_d} \frac{1}{p} \sum_{i=1}^p [\log D(x_i) + \log(1 - D(G(z_i)))]. \quad (10)$$

Then, fix the discriminator parameters, take the  $q$  sample from the sample set  $Z$ , and use random gradient descent to update generator parameter  $\theta_g$ :

$$\nabla_{\theta_g} \frac{1}{p} \sum_{i=1}^p [\alpha \log(1 - D(G(z_i))) + \beta \text{CE}(f(G(z_i)), h_d(f(x_i)))]. \quad (11)$$

Repeat the above process until the specified iteration number is reached and the training is over [21]. The amplified sample set  $R$  is obtained, whose sample number is  $a$ .

**2.2.2. Training of Evaluation Model.** This paper uses the RBF network model based on Gaussian kernel (Figure 2) to train samples. The amplified sample set generated by GAN is used as the input, the input vector is  $X = [x_1, x_2, \dots, x_i, \dots, x_n]$ , the output is  $y$ , and the expected output is  $\hat{y}$ .

Firstly, three parameters of the RBF network are initialized by the optimization method. Calculated as follows:

$$c_{ji} = \min i + \frac{\max i - \min i}{2m} + (j-1) \frac{\max i - \min i}{m}, \quad (12)$$

where  $\min i$  and  $\max i$  are the minimum and maximum values of the samples received by the  $i$ -th neuron in the input layer, respectively, and  $m$  is the total number of neurons in the hidden layer.

$$\sigma_{ji} = \sigma_f \sqrt{\frac{1}{a} \sum_{q=1}^a (x_i^q - c_{ji})^2}, \quad (13)$$

where  $d_f$  is width adjustment coefficient, whose value is less than 1;  $x_i^q - c_{ji}$  is the distance between the  $i$ -th sample received by the  $q$ -th neuron in the input layer and the corresponding center.

$$\omega_{jk} = \min k + j \frac{\max k - \min k}{l+1}, \quad (14)$$

where  $\min k$  and  $\max k$  are the minimum and maximum values of all expected outputs in the  $k$ -th output neuron and  $l$  is total neurons number in the output layer.

When the three parameters are initialized, the neural network begins to train where the random gradient descent method is used to update it iteratively [22]. The iterative calculation is as follows:

$$\begin{aligned} c_{ji}(t) &= c_{ji}(t-1) - \eta \frac{\partial E}{\partial c_{ji}(t-1)} \\ &\quad + \alpha [c_{ji}(t-1) - c_{ji}(t-2)], \\ \sigma_{ji}(t) &= \sigma_{ji}(t-1) - \eta \frac{\partial E}{\partial \sigma_{ji}(t-1)} \\ &\quad + \alpha [\sigma_{ji}(t-1) - \sigma_{ji}(t-2)], \\ \omega_{jk}(t) &= \omega_{jk}(t-1) - \eta \frac{\partial E}{\partial \omega_{jk}(t-1)} \\ &\quad + \alpha [\omega_{jk}(t-1) - \omega_{jk}(t-2)], \end{aligned} \quad (15)$$

where  $c_{ji}(t)$  is the central component of the  $j$ -th hidden layer neuron for the  $i$ -th input neuron in the  $t$ -th iteration,  $\sigma_{ji}(t)$  is the width component corresponding to the central component,  $\omega_{jk}(t)$  is the adjustment weight of the  $j$ -th hidden layer neuron for the  $k$ -th output neuron in the  $t$ -th iteration,  $\eta$  is the learning factor, and  $\alpha$  is the momentum factor.

The loss function is defined as

$$E = \frac{1}{2} \sum_{q=1}^a (y_q - \hat{y}_q)^2, \quad (16)$$

where  $y_q$  is the actual output of the  $q$ -th sample of the neural network,  $\hat{y}_q$  is the expected output of the corresponding sample, and  $a$  is the total number of amplified samples.

Iteratively update center parameter, width parameter, and weight, and then calculate the loss. When the loss is within the acceptable range, stop training [23].

Summarize the operation process of the GAN-RBF evaluation model, as shown in Figure 3.

**2.3. Index System of Maintenance Support Effectiveness.** On the basis of field investigation and expert opinions [1, 2], combined with the current operational development trend, and considering main operation factors of equipment maintenance support system, the evaluation index system of equipment maintenance support effectiveness is established, as shown in Table 1.

**2.4. Data Processing.** The types and dimensions of maintenance support effectiveness evaluation indexes are different, so it will be difficult to balance the importance of input components when inputting them into neural networks, which will not only reduce the training speed of GAN and RBF networks but also affect the accuracy of sample generation and weight iteration. [24]. Therefore, before sample training, the data must be normalized and transformed into a standardized interval [0,1]. The maintenance support effectiveness index is divided into benefit-index and

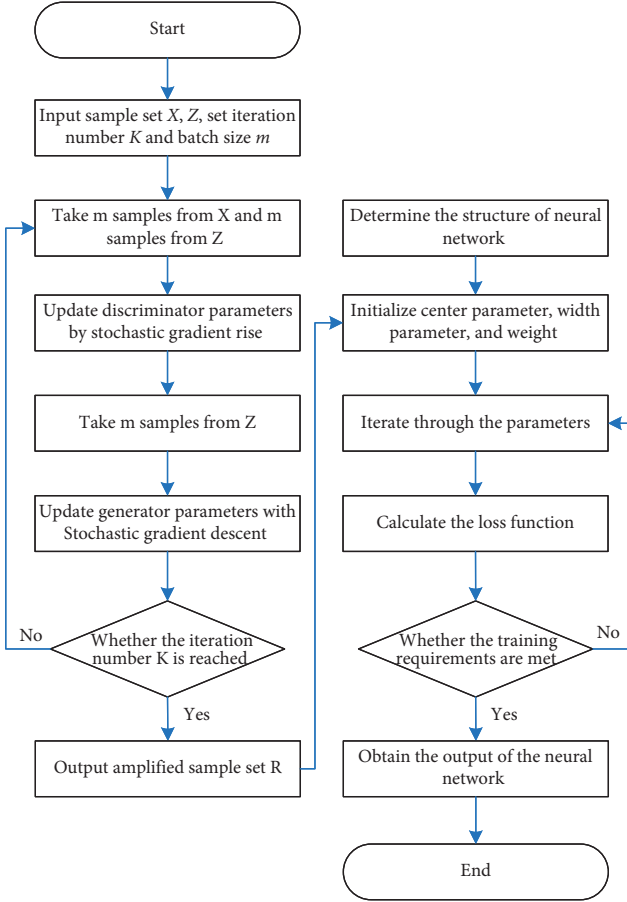


FIGURE 3: Operation process of GAN-RBF evaluation model.

cost-index, which have the characteristics of the bigger the better and the smaller the better [25]. This paper uses the range method to normalize these two types of indexes.

Benefit-index:

$$Y = \frac{X - X_{\min}}{X_{\max} - X_{\min}}. \quad (17)$$

Cost-index:

$$Y = \frac{X_{\max} - X}{X_{\max} - X_{\min}}. \quad (18)$$

### 2.5. Evaluation Step

- (1) Construct evaluation index system of equipment maintenance support effectiveness
- (2) Select original data and random noise of maintenance support for GAN training, and normalize them
- (3) Sent original data and random noise into GAN for training to generate amplified samples
- (4) Train RBF network by amplified samples, and obtain network output, that is, the predictive value of maintenance support effectiveness

- (5) Calculate the accuracy and complexity of the evaluation model and compare it with other evaluation methods to verify its effectiveness

**2.6. Model Evaluation Criteria.** The maintenance support effectiveness evaluation is generally carried out by the test base, aiming at accurately estimating the maintenance support effectiveness of a certain support unit, so as to help improve the maintenance support strength [1]. Obviously, accuracy is the most important index of maintenance support effectiveness evaluation, so this paper takes accuracy as the primary evaluation criterion for models. Secondly, the complexity of the general composite model is higher than that of the previous single model. If the complexity of the improved GAN-RBF model is too high, even if its accuracy is high, its performance is still low, so the model complexity should be taken as an auxiliary reference index. Because the test equipment in the test base has sufficient memory and the test time is relatively abundant, the evaluator is tolerant of model complexity. If the complexity of the improved GAN-RBF model is not much different from that of the common model, it is completely acceptable.

**2.6.1. Accuracy.** Firstly, when the model output is obtained, a set of real effectiveness sets is used as the baseline for comparison, and the fitting effect of model-predicted values to real values is observed from a macroperspective.

Then, the mean square error (MSE) of the models is calculated, and the accuracy of each model is compared from a microperspective. MSE is the square's expected value of the difference between the predicted value and real value for evaluation models, which is used to measure data deviation degree and reflect the actual situation of error. The smaller the value is, the higher the mode accuracy is [26]. MSE can show the advantages and disadvantages of the four models more clearly. The calculation formula is as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - r_i)^2, \quad (19)$$

where  $y_i$  is the model-predicted value and  $r_i$  is the model real value.

**2.6.2. Complexity.** The algorithm complexity is an index to measure the execution of the program, which consists of time complexity and space complexity. Time complexity determines the training/prediction time of the model, while space complexity determines the number of the parameters of models. Because algorithm design is more in pursuit of high efficiency, and the increasing computer storage capacity can basically meet algorithm space need, the algorithm complexity is generally dominated by time complexity [27]. There are two methods to calculate the time complexity: preanalysis method and poststatistics method. The former uses the big O notation method, which is suitable for predicting the complexity when the algorithm program is not written. The four models in this paper have corresponding programs. It is obviously more convenient and quicker to

TABLE 1: Index system of equipment maintenance support effectiveness.

	First-level index	Second-level index
Evaluation index system of equipment maintenance support effectiveness	Mobility support effectiveness	Mobile arrival rate
		Maneuvering average speed
	Equipment repair effectiveness	Mobile consumption rate
		Equipment repair rate
		Equipment repair time
	Maintenance management effectiveness	Equipment repair loss rate
		Maintenance pass rate
		Maintenance safety rate
	Resource support effectiveness	Management personnel matching rate
		Equipment matching rate
Personnel support effectiveness	Funding availability	
	Completion rate of technical data	
		Personnel competence rate
		Average delay time

estimate the algorithm complexity by directly measuring running time by using the postevent statistics method.

### 3. Result

The hardware configuration of this experiment is as follows: the running environment is Windows 7 operating system, the CPU is Intel (R) Core (TM) i7-4790k CPU @ 4.00 GHz processor, the running memory is 16 G, the graphics card is NVIDIA GeForce RTX 2080Ti, and the programming environment is PyTorch 3.7.3.

Maintenance support data is usually collected at the lowest level of indicators [28], and the data in this paper is collected at the second-level index. According to the 14 second-level indexes of maintenance support effectiveness index system (Table 1), 28 groups of 392 original samples were selected from the database, and 100 groups of 1400 amplified samples were generated by GAN. The original sample set and amplification sample set are shown in Table 2. After inputting these samples, the RBF network will output five first-level indexes values, which are predicted values of maintenance support effectiveness components. Compared with the direct output of final effectiveness value, the former method enriches the analysis at the level of effectiveness components. If the final effectiveness value is wanted, it can be realized by changing the activation function and units' number in the output layer of the RBF network.

In order to verify the validity of the improved GAN-RBF model, this paper compares three common neural network evaluation models of Elman neural network, BP neural network, and ordinary RBF network. To ensure the contrast effect, the same general network parameters are set for the four models. According to 14 second-level indexes and 5 effectiveness components, 14 nodes are set for the input layer and 5 nodes for the output layer. According to experience, the hidden layer is set to 75% of the nodes number in the input layer and set to 10 nodes. The maximum number of iterations is 100, the learning rate is 0.1, and the allowable error is 0.001. According to the requirement of the test base for maintenance support effectiveness accuracy, the target value of the iterative loss is set to 10–15.

The original sample set is used to train four models. For the improved GAN-RBF model, the input of the RBF network is actually the amplified sample set generated by GAN. Test results are as follows.

*3.1. Iteration Curve of Models.* The iterative process of four models is shown in Figure 4.

*3.2. Predicted Value and MSE of Models.* Twenty-eight groups were randomly selected from the amplified sample set as the test set to test the four trained models. The output of evaluation models is the predicted value of five effectiveness components E1, E2, E3, E4, and E5. Taking a group of real effectiveness sets as a baseline, we observe the fitting effect of predicted values to real values, as shown in Figure 5.

According to formula (19), the single group MSE and the mean value of 28 MSEs are calculated. MSE mean values of four models are shown in Figure 6.

In order to observe the fluctuations of four model MSEs, a single group of MSE was made into a line chart, as shown in Figure 7.

*3.3. Running Time of Models.* Through real-time calculation, the training/prediction time of the four models is obtained, as shown in Table 3.

## 4. Discussion

*4.1. Model Iterative Analysis.* Figure 4 shows that, in the initial stage of iteration, the convergence speed of the improved GAN-RBF model is faster than that of the BP model and lower than that of the Elman model and RBF model. However, after about 40 iterations, the BP model and Elman model have fallen into local optimum, and the loss function values are 103 and 10<sup>-5</sup>, respectively, which do not reach the target value of the loss. At this time, the convergence speed of the RBF model is obviously reduced to that of the improved GAN-RBF model. It is surpassed by the improved GAN-RBF model after about 70 iterations, and the target

TABLE 2: Normalized original sample set and amplified sample set.

Index	Group								
	Original samples				Amplified samples				
	1	2	...	28	1	2	...	99	100
Index 1	0.6845	0.6329	...	0.7023	0.6522	0.6316	...	0.6473	0.6832
Index 2	0.9327	0.8719	...	0.9236	0.9176	0.9431	...	0.9418	0.9673
Index 3	0.4423	0.4226	...	0.4653	0.5031	0.4943	...	0.5186	0.4823
...	...	...	...	...	...	...	...	...	...
Index 13	0.7032	0.7523	...	0.7421	0.6832	0.7186	...	0.7342	0.7143
Index 14	0.4232	0.4621	...	0.4312	0.4539	0.4622	...	0.4548	0.4463

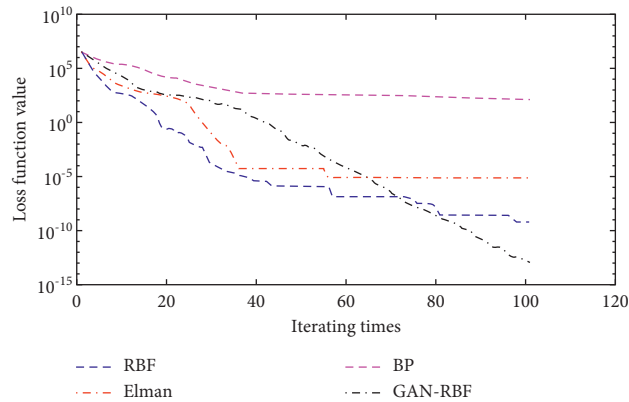


FIGURE 4: Comparison of four models' iteration curve.

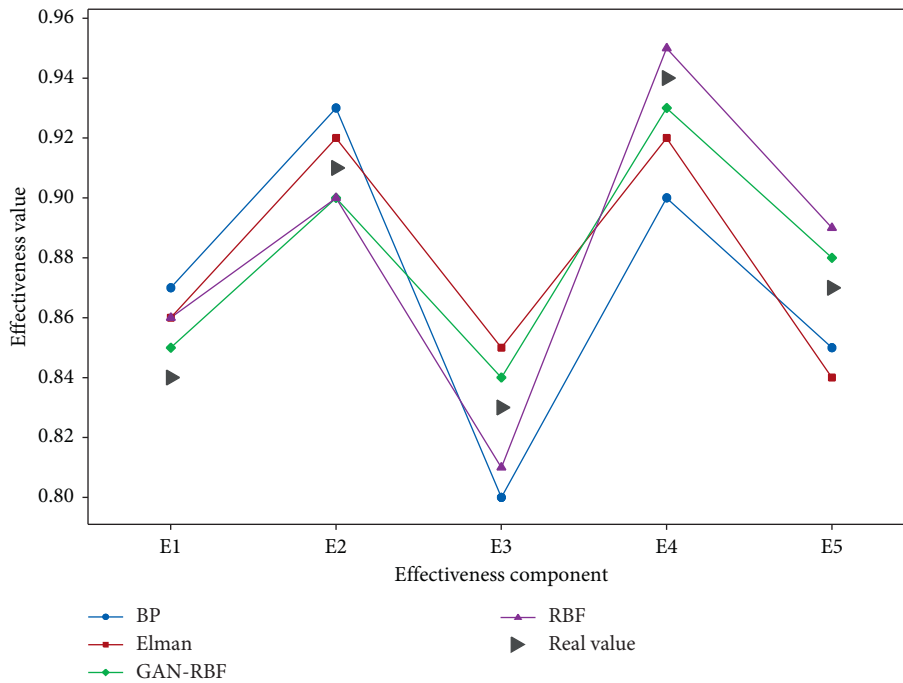


FIGURE 5: Comparison of the predicted value of four models' effectiveness components and real value.

value of the loss is never reached in subsequent iterations. The improved GAN-RBF model always keeps a stable and fast convergence speed, does not fall into local optimum, and achieves the ideal loss value.

From the perspective of model iteration, the improved GAN-RBF model can not only maintain a stable and fast convergence speed but also achieve a minimum loss value, with high prediction accuracy and no fitting problem.

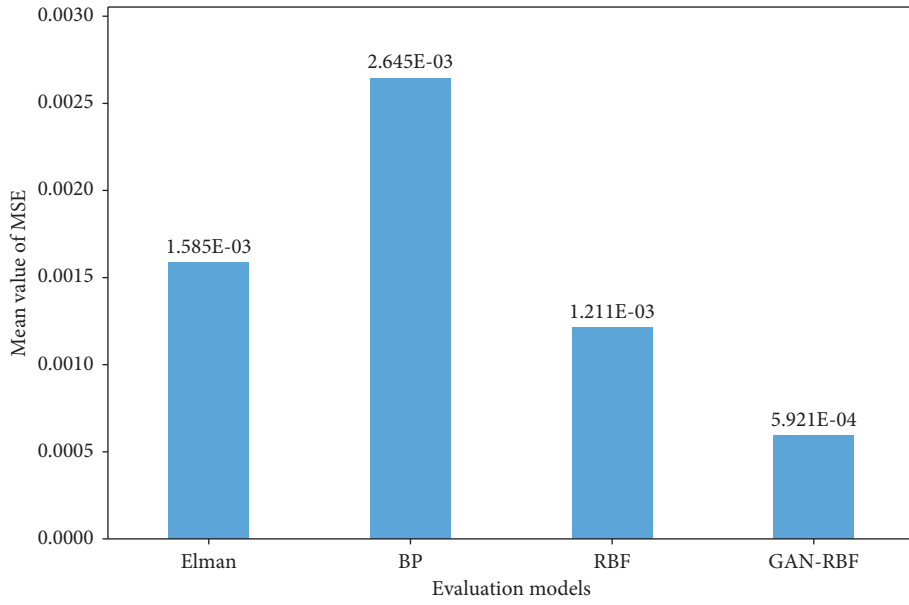


FIGURE 6: Comparison of MSE mean value.

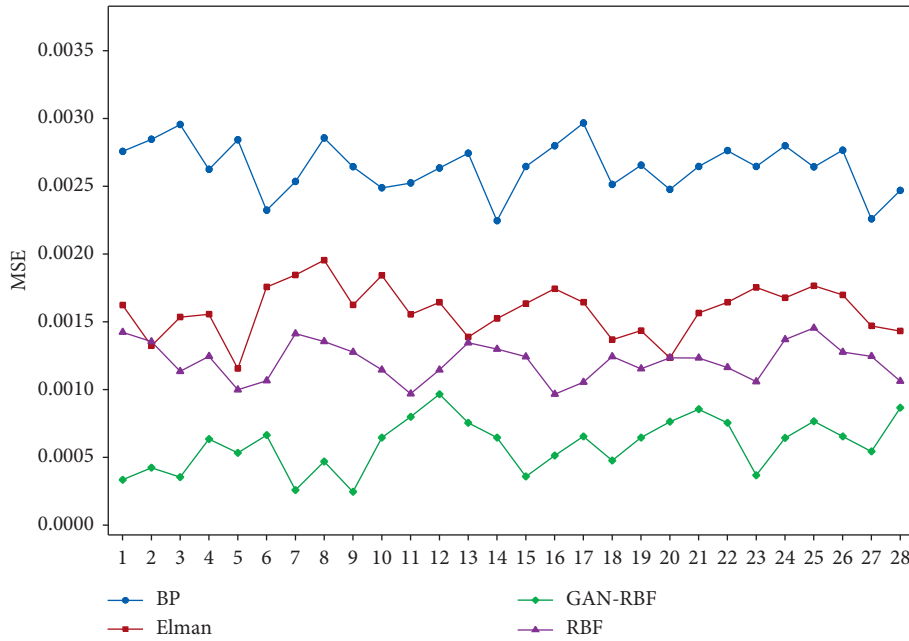


FIGURE 7: MSE Comparison of a single sample.

TABLE 3: Running times of four models.

	Training time (s)	Prediction time (s)
BP	639	9.45
Elman	325	4.59
GAN-RBF	142	2.01
RBF	118	2.37

Although the complexity of the improved GAN-RBF model is increased after adding GAN, and the initial convergence speed is slightly slower than the Elman model and RBF

model, it can always maintain a stable and fast convergence speed and, finally, achieve convergence accuracy better than other models.

**4.2. Model Accuracy Analysis.** Firstly, the output fitting effect of each model is discussed. Observing the position of effectiveness component predicted value relative to real value in Figure 5, the improved GAN-RBF model is the closest among the four models, whose prediction curve is obviously more in line with effectiveness real value. From the perspective of the output fitting effect, the improved GAN-RBF model is superior to the traditional three neural network evaluation models. In addition, the E1 and E2 predicted values of the four models are more aggregated and the differences are smaller. It can be concluded that the prediction effects of E1 and E2 are generally better than those of E3, E4, and E5. As the two most basic and important maintenance support effectiveness components, E1 (mobility support effectiveness) and E2 (equipment repair effectiveness) are more complete than other effectiveness components in data collection and processing equipment and procedures, and the sample quality is relatively high [1]. Therefore, its prediction effect is relatively good. This shows the importance of sample quality for the neural network evaluation model from another aspect.

Secondly, the MSE of each model is discussed. Figure 6 shows that MSE of Elman, BP, RBF, and improved GAN-RBF models are, respectively,  $1.585 \times 10^{-3}$ ,  $2.645 \times 10^{-3}$ ,  $1.211 \times 10^{-3}$ , and  $5.921 \times 10^{-4}$ . MSE of the improved GAN-RBF model is the smallest, and the prediction accuracy is undoubtedly the best. The height difference of the four bars in Figure 6 is obvious. The bar height of the improved GAN-RBF model is approximately 1/2 of the RBF model, 1/3 of the Elman model, and 1/5 of the BP model, and the prediction accuracy is far superior to the other three evaluation models. It can be roughly estimated from Figure 7 that the volatility order of a single MSE is BP > Elman > improved GAN-RBF  $\geq$  RBF, and the volatility of the improved GAN-RBF model can be called optimal together with the RBF model after excluding some abnormal points.

It can be seen that the prediction accuracy of the improved GAN-RBF model is the best whether it is evaluated from the fitting effect of model output or the size and fluctuation of model MSE.

**4.3. Model Complexity Analysis.** When the accuracy of the improved GAN-RBF model is verified to be optimal, it is still necessary to analyze its complexity. If its complexity is not too high, and it is within the acceptance of maintenance support effectiveness evaluators, it can be considered that the performance of the improved model is really good. If its complexity is too high, its performance cannot be considered as good.

Table 3 shows that the training time of the improved GAN-RBF model is 142 s, which is better than the BP model and Elman model and weaker than the RBF model. Its prediction time is 2.01 s, which is the best among the four models. After adding GAN to RBF, the complexity of this composite model has been improved to some extent, and training time has been increased. However, this increase is relatively small and completely within the acceptable range. On the other hand, through the optimization of GAN and RBF, the prediction

efficiency of this composite model is improved, which is 0.36 s faster than the fastest RBF model among the other three models. Under the effect of optimal design, the complexity of the improved GAN-RBF model has no negative effect and even has higher efficiency in the prediction scenario.

On the whole, among four evaluation models, the improved GAN-RBF model ranks first in the most important index "accuracy", and the model complexity is reasonable, which achieves the ideal evaluation effect. Li et al. [24] also used GAN to amplify the sample of the deep neural network and applied it to evaluate weapon system effectiveness, whose MSE is about  $1.6 \times 10^{-3}$ . Apart from different research objects, the MSE of the GAN-RBF model in this paper reaches  $5.921 \times 10^{-4}$ , which has higher accuracy than that model from the numerical point of view. The above discussion illustrates the following: (1) using amplified samples generated by GAN to train the RBF network can alleviate the problem of poor generalization ability caused by small training samples and effectively improve the training effect. It is a feasible way to combine GAN with a general neural network to solve the problem of small sample size in effectiveness evaluation or other research. (2) The optimization of two key points, the determination of GAN loss function and the initialization of RBF parameters, is also very effective. The category constraint based on the category probability vector reordering function is introduced to avoid the simplification of sample category and enhance sample quality. The parameter initialization method based on parameter components equidistant variation is simple to calculate, enhances the response of correct feature information, and reduces the risk of training overfitting.

## 5. Conclusion

Small sample evaluation of equipment maintenance support is a difficult problem in the equipment support field. The neural network evaluation model based on amplified samples proposed in this paper exploratively combines the optimized GAN and RBF, respectively, to evaluate maintenance support effectiveness. The experimental results show that, compared with the traditional neural network evaluation method, the improved GAN-RBF evaluation model has higher evaluation accuracy and can evaluate effectiveness value more accurately. This method not only solves the problem of insufficient maintenance support training samples but also provides a good reference for evaluation research in other fields. However, there are still some deficiencies in this paper. For example, in the maintenance support data, text data occupies a certain proportion. The quality of maintenance support data is not high because GAN cannot directly backpropagate the discrete sequence samples, which affects the training effect of the RBF network. Therefore, how to improve generating quality of maintenance support text data will be the focus of the next step.

## Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Acknowledgments

The authors thank Professor Xingxin Li for his help in collecting experimental data. This work was supported in part by the National Defense Pre-Research Fund (9140A27010215)B34422).

## References

- [1] X. Du, M. Zhou, and Y. Lv, "Effectiveness evaluation of equipment support system based on optimized RBF neural network," *Computer Engineering*, pp. 1–8, 2020.
- [2] L. Wang, D. Zhou, and Z. Yang, "Effectiveness evaluation of joint operations based on adaptive wavelet neural network," *Firepower and Command Control*, vol. 45, no. 12, pp. 73–79, 2020.
- [3] M. Zhu, Q. Lu, and Y. Ding, "Effectiveness evaluation of underwater cluster operation based on GA Elman neural network," *Firepower and Command Control*, vol. 45, no. 7, pp. 115–119, 2020.
- [4] J. Liu, J. Liu, and L. Xiao, "A system effectiveness evaluation method based on Grey RBF neural network," *Application of Electronic Technology*, vol. 46, no. 12, pp. 107–110, 2020.
- [5] R. Qiao and F. Zhou, "Operational effectiveness evaluation of surface to air missile weapon system based on PCA-BP neural network," *Military Operations and Systems Engineering*, vol. 34, no. 04, pp. 38–43+67, 2020.
- [6] L. Wang, K. Wang, S. Liu, Y. Liu, and J. Yuan, "Research on evaluation model of wartime maintenance support capability of vehicle equipment," *Journal of Military Transportation Academy*, vol. 21, no. 5, pp. 37–41, 2019.
- [7] M. Chen, Y. Zhang, S. Ma, and Y. Chen, "Research on airport support capability assessment method based on operation data," in *proceedings of the 2019 International Conference on Aviation Safety and Information Technology*, Kunming, China, October 2019.
- [8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," 2014, <https://arxiv.org/abs/1406.2661>.
- [9] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: algorithms, theory, and applications," 2020, <https://arxiv.org/abs/2001.06937>.
- [10] H. Yin, Y. Wei, H. Liu, S. Liu, C. Liu, and Y. Gao, "Deep convolutional generative adversarial network and convolutional neural network for smoke detection," *Complexity*, vol. 2020, Article ID 6843869, 12 pages, 2020.
- [11] D. Yao, Q. Sun, J. Yang, H. Liu, and J. Zhang, "Railway fastener fault diagnosis based on generative adversarial network and residual network model," *Shock and Vibration*, vol. 2020, Article ID 8823050, 15 pages, 2020.
- [12] J. Luo and J. Wang, "Image demosaicing based on generative adversarial network," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7367608, 13 pages, 2020.
- [13] T. Tsukahara, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Collaborative learning of generative adversarial networks," in *Proceedings of the 16th International Conference on Computer Vision Theory and Applications*, Online, October 2021.
- [14] A. Gholamzadeh Chitgar and J. Berenjian, "Performance evaluation of RBF networks with various variables to forecast the properties of SCCs," *Civil Engineering Infrastructures Journal*, 2020.
- [15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, Venice, Italy, October 2017.
- [16] S. Xie, G. Ren, and J. Zhu, "Application of a new one-dimensional deep convolutional neural network for intelligent fault diagnosis of rolling bearings," *Science Progress*, vol. 103, no. 3, 2020.
- [17] A. R. Sittikova, A. M. Gusenkov, D. N. Galanin, and N. R. Bukharaev, "Using generative adversarial networks for relevance evaluation of search engine results," in *Proceedings of the 2020 IEEE East-West Design & Test Symposium (EWDTS)*, Varna, Bulgaria, September 2020.
- [18] H. Jo, J. E. Santos, and M. J. Pyrcz, "Conditioning well data to rule-based lobe model by machine learning with a generative adversarial network," *Energy Exploration & Exploitation*, vol. 38, no. 6, pp. 2558–2578, 2020.
- [19] G. Hu, J. Huang, Q. Wang, J. Li, Z. Xu, and X. Huang, "Unsupervised fabric defect detection based on a deep convolutional generative adversarial network," *Textile Research Journal*, vol. 90, no. 3–4, pp. 247–270, 2020.
- [20] Q. Gao and Z. Jiang, "Amplification of small sample library based on GAN equivalent model," *Electrical Measurement and Instrument*, vol. 56, no. 6, pp. 76–81, 2019.
- [21] L. Cai and Q. Sun, "Multiautonomous underwater vehicle consistent collaborative hunting method based on generative adversarial network," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, Article ID 1729881420925233, 2020.
- [22] N. Li, Y. Li, and G. Gong, "Intelligent evaluation and optimization of system of systems combat effectiveness based on deep learning," *Journal of System Simulation*, vol. 32, no. 8, pp. 1425–1435, 2020.
- [23] J. Yin, J. Gu, Y. Chen, W. Tang, and F. Zhang, "Method for determining load magnitude and location from the plastic deformation of fixed beams using a neural network," *Science Progress*, vol. 104, no. 1, Article ID 00368504211003385, 2021.
- [24] J. Li, H. Liu, and D. Hu, "Effectiveness evaluation of weapon system based on generative countermeasure network and deep neural network," *Computer Applications and Software*, vol. 37, no. 02, pp. 253–258, 2020.
- [25] Y. He, F. Hu, L. Lv, C. Zhou, X. Xiao, and L. Wang, "Research on effectiveness evaluation of equipment maintenance support based on simulation," in *Proceedings of the 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, pp. 339–343, IEEE, Chongqing, China, December 2019.
- [26] J. Gu, G. Ding, and C. Liu, "Research on effectiveness evaluation method of electronic countermeasures based on genetic algorithm and neural network," *Communication Countermeasure*, vol. 36, no. 2, pp. 45–49, 2017.
- [27] M. Morzy, T. Kajdanowicz, and P. Kazienko, "On measuring the complexity of networks: kolmogorov complexity versus entropy," *Complexity*, vol. 2017, Article ID 3250301, 12 pages, 2017.
- [28] M. Hu, G. Li, B. Lei, G. Li, X. Jia, and Y. Bao, "Research on evaluation of equipment maintenance support capability of shortwave fixed communication stations," *Journal of Ordnance Equipment Engineering*, vol. 37, no. 8, pp. 47–51, 2016.

## Research Article

# Development of ANPR Framework for Pakistani Vehicle Number Plates Using Object Detection and OCR

**Salma, Maham Saeed, Rauf ur Rahim, Muhammad Gufran Khan , Adil Zulfiqar, and Muhammad Tahir Bhatti**

*Department of Electrical Engineering, National University of Computer and Emerging Science, Islamabad (Chiniot-Faisalabad Campus), Islamabad, Pakistan*

Correspondence should be addressed to Muhammad Gufran Khan; [m.gufran@nu.edu.pk](mailto:m.gufran@nu.edu.pk)

Received 12 February 2021; Accepted 27 September 2021; Published 19 October 2021

Academic Editor: Atila Bueno

Copyright © 2021 Salma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The metropolis of the future demands an efficient Automatic Number Plate Recognition (ANPR) system. Since every region has a distinct number plate format and style, an unconstrained ANPR system is still not available. There is not much work done on Pakistani number plates because of the unavailability of the data and heterogeneous plate formations. Addressing this issue, we have collected a Pakistani vehicle dataset having various plate configurations and developed a novel ANPR framework using the dataset. The proposed framework localizes the number plate region using the YOLO (You Only Look Once) object detection model, applies robust preprocessing techniques on the extracted plate region, and finally recognizes the plate label using OCR (optical character recognition) Tesseract. The obtained mAP score of the YOLOv3 is 94.3% and the YOLOv4 model is 99.5% on the 0.50 threshold, whereas the average accuracy score of our framework is found to be 73%. For comparison and validation, we implemented a LeNet Convolutional Neural Network (CNN) architecture which uses the segmented image as an input. The comparative analysis shows that the proposed ANPR framework comprising the YOLOv4 and OCR Tesseract has good accuracy and inference time for a wide variation of illumination and style of Pakistani number plates and can be used to develop a real-time system. The proposed ANPR framework will be helpful for researchers developing ANPR for countries having similar challenging vehicle number plate formats and styles.

## 1. Introduction

Vehicle ownership is increasing proportionally with the economic growth that makes the management and governance of the transportation system complicated. Violation of traffic rules, overspeeding, and car theft are common practices. The detection and retrieval of number plates from fast-moving vehicles make it hard to catch and penalize the culprit. The traffic congestion and unavailability of parking slots lead to the problem of time, fuel consumption, and air pollution. Till today, the vehicle number plate is often noted manually, and human errors in record keeping are unavoidable. Indeed, there is a need to have an automatic and efficient device for detecting, collecting, and managing car information.

In the era of the fourth industrial revolution, an Intelligent Transport System (ITS) is a necessity in which the

ability to share information without any human intervention is possible by the use of Artificial Intelligence (AI) and the Internet of Things (IoT). The most important subsystem of an ITS is Automatic Number Plate Recognition (ANPR). The ANPR system reads the image, preprocesses it, and recognizes the vehicle number plate characters independent of human involvement. It helps to identify potential risks, prevent crime, improve reliability, develop barrier-free infrastructure, and provide location information. The Global Automatic Number Plate Recognition System Market is forecasted to increase with a ratio of 9.63% from 2017 to 2025 [1].

The studies show that existing ANPR methods are not viable, and it is hard to find a single efficient approach for different regions due to the unique format and style of each region. Several parameters, namely, vehicle pace,



illumination, image dimension, and resolution affect the rate and accuracy of recognition. Considering the case of Pakistani vehicle number plates, another challenge is that no standard formulation of plates is followed in Pakistan. Unlike other region's plates, plates in Pakistan are both double and single rows and have no fixed number of characters. Moreover, there is no standard Pakistani vehicle number plate dataset available. To address these issues, we collected our dataset having heterogeneous plate formations, i.e., double or single row, varying font styles, namely, Charles's wright, Barcelona, indigo, nonconforming background having stickers, and orientation, which is followed in various areas of Pakistan. We reviewed the existing methods applied in different parts of the world and implemented few methods on Pakistani number plates. Our analysis showed that the factors affecting the ANPR performance are ghastly plates, low illumination, plate orientation, diverse fonts and languages, and imperfect hardware vision. We developed a novel ANPR framework that works efficiently for various plate configurations and addresses the challenges to a large extent. It has been examined on several images with low resolution ( $416 \times 416$ ). The performance of our proposed OCR method is discussed and further compared with existing methods to help develop an efficient ANPR for Pakistani number plates.

The remaining paper is structured as follows. A detailed review of previously applied techniques is presented in Section 2, whereas in Section 3 the proposed Framework of ANPR is discussed. In Section 4, the proposed methodology using OCR Tesseract for recognition is implemented. In Section 5, results and a comparative study of the proposed methods in tabular form are discussed. Finally, Section 6 concludes the paper and presents the future work.

## 2. Review of Previous Work

Recently, Automatic Number Plate Recognition (ANPR) has turned out a primary mechanism for security and traffic administrations that spans from congestion supervision to vehicle parking leading to data maintenance for surveillance causes. ANPR offers a simple description of vehicles and a guide for more vehicle tracking and behavior analysis. The big range of number plates, different in color, form, size, and pattern, is one of the main LPR challenges. Such challenges include adverse weather, bad lighting, and a limited camera resolution, as they affect image clarity in real time as a camera captures the plate. In the field of ANPR, a detailed review/comparison of formerly existing systems, their performance, their inferiorities, and their limitations are discussed in the following.

In 1998, PC-based number plate recognition systems were common. C. Coetzee proposed the system [2] that uses the Niblack thresholding algorithm over digital gray level car images that transcend all binarization methods used in comparable systems. The position and size of the number plates are detected by an efficient rule-based algorithm. The segmentation of character is done with blob-coloring and is transferred to the neural network optical character recognition (OCR) method as a 15/spl times/15-pixel bitmap. The

proposed feature reduction method lessens the input features from 255 to 50. Six characters are recognized using six small networks running in parallel. The system is performed with an accuracy of 86.1 percent for single- and double-line plates under variable brightness and annotations.

As time passed, many improved ANPR systems were developed to identify the vehicle number plates, but recognition remained a challenge due to the diverse formation of plates in different regions. The work [3] focuses on Nepali vehicle number plates for noting the performance of automobiles license plate identification system based on template matching. The digital camera captures the images that are processed to obtain the vehicle plate information. Different algorithms are applicable for plate detection, including morphological operations, edge detection, smoothing, and filtering. Characters are segregated for identification by using a template matching standardized interrelation and phases correlation, and the result is differentiated into  $70 \times 70$  blocks and correspondence with the repository templet. For consistency, the characters are separated. Ninety patterns under several conditions are assessed using the following approach. After application on several images, the study exhibits cross-correlation with greater precision than phase correlation. The exactness of normalized cross-correlation and phase correlation was 67.98% and 63.46%, respectively.

Real-time identification of number plates using Raspberry Pi is also considered as detection and classification of characters that are additionally utilized in surveillance, congestion, and access control applications. The common ALPR systems that come under image processing uses MATLAB as an exclusive tool. As vehicle recognition systems have complex features, a substitute and more systematic method for automated number plate identification is Open Computer Vision (OpenCV). OpenCV is used by the OCR method in [4]. The significant deviation between the MATLAB and OpenCV is a mode of the procedure: online and offline. For online processes, MATLAB cannot be done, while OpenCV is ideal for real time. Another quicker way of using Machine Language python and the Open Vision Library is provided in the paper.

All formerly means of ALPR systems work under constraints like frontal view, standard font, and good illumination, but the introduced novel Convolutional Neural Network (CNN) is efficient for skewed and dis-oriented depictions in [5]. YOLOv2 is used as a black box for car detection with a precision of 76.8 percent. The suggested Warped Planner Object Detection Network (WPOD-NET) explores the number plate (NP), resized to keep the NP identifiable, and creates rectangular bounding boxes. The CNN architecture comprises 21 convolution layers with 14 inside residual blocks,  $3 \times 3$  ReLU activation, and 4 max-pooling layers sized  $2 \times 2$ . Annotated NPs are accomplished by 100k iterations and batch-size 32, where the learning rate is set to 0.001 for training. OCR is utilized for character segmentation and recognition giving an accuracy of up to 75 percent. The unhackneyed methodology is best for oblique images but more affine transformations are yet to explore.

In [6], Aishwarya considered vehicle images with low resolution for automated detection and recognition of NP using a support vector machine (SVM). The model was evaluated for 150 car number plates under particular circumstances. The images are manipulated into grayscale, binarized, and filtered to remove noise. The masking technique assists in extracting the required input image. Lastly, recognition is done by distinguishing definite numbers and characters. The system is deployed and synthesized in MATLAB 2010a. The accuracy of identification is 92.0%. The false results arise because of overlapped, blurred, and slang vehicle or variable style and font plates.

The system in [7] initially captures the license plate's image, pretreat, and scan every single character available on the number plate separately for their absolute identification. The OCR usage is the most significant phase; here, the letterings on the plate's portrayal are converted to the textual format which is lately decoded. The specified paper encapsulates an overall computation and organized stream for ANPR and its effective applications. The following method established an exactness of 75 to 85 percent for Indian Number Plate after the concept of template matching for ANPR systems.

Researchers proposed four algorithms for the OCR phase of real-time high definition (HD) ALPR models, in [8]. HD images are preferred for the proposed complex over standard definition (SD) as the precision of recognition improves. HD images take more computing time which is countered by executing the system on heterogeneous Systems on Chip (SoC) and Field Programmable Gate Array (FPGA). The input HD image undergoes resizing and morphological operations. With several promising solutions, Automatic Number Plate Recognition (ANPR) is a reasonably well-explored problem. However, owing to the differences in the characteristics of number plates around the world, these solutions are usually tuned to a specific area. Such characteristics are based on algorithms written for number plate identification, but it will be impossible to realize a universal solution since the image processing methods used to develop these algorithms cannot themselves boast a hundred percent accuracy. An algorithm proposed by Andrew et al. [9] that is optimized to work with Ghanaian vehicle number plates is the subject of this article. The algorithm, written with the OpenCV library in C++, uses edge detection and feature detection techniques for locating the plate combined with mathematical morphology. For recognition, they used OCR Tesseract.

Fakhar et al. proposed an affordable mobile ANPR system using Raspberry Pi [10]. The model utilizes a real-time image taken from the camera. The captured image is desaturated, filtered, and segmented, and the plate's characters are recognized. All computation complexity is handled on the Raspberry Pi, and a noticeable 3-second delay is observed before achieving the final output on the LCD. The resultant label is also stored in the database.

In [11], a plate recognition framework employing deep learning is designed. The image dataset consists of 6500 Indian car number plates that are split into 90 and 10 percent for training and testing, respectively. The images have

diverse qualities because these are captured from 3 cameras having different specifications, like bit rate and focal length. For character recognition 37-class CNN model is trained. The model comprises 126 filters. YOLO v3 is utilized for the detection and identification of car plates. An accuracy of 100% and 91% is achieved for plate detection and number plate recognition accordingly.

Moreover, in [12], an embedded system based on KNN Machine Learning for number plates following Polri regulations is developed. The attained model exactness using KNN for character recognition is 84%. The images are taken by webcam with a resolution of  $640 \times 480$  pixels. They undergo cropping, gray scaling, inversion, thresholding, edge detection, and morphological operations before contouring in segmentation. Character improvement increased the accuracy of the recognized number plate model to 98%.

Ariff et al. in [13] explore variant segmentation techniques for processing a noisy number plate image. Hundred Malaysian car plates of  $1932 \times 2576$  pixels are passed to threshold techniques, namely, Savoula and Niblack method for removal of undesired pixels. In this, Savoula segmentation gives an average accuracy of 83.17% which compromises the accuracy of recognition. The classification of characters is done by the template matching technique.

In [14], a speedy and more accurate vehicle nameplate recognition system is proposed based on optical character recognition (OCR) engine "Tesseract". Raw images go undersampling and quantization. Then, region of interest (ROI) is obtained by a binary mask. The input plate image goes through grayscale, global image thresholding by the Otsu method, and noise removal. The processed image is passed to OCR where character segmentation takes place. The segmented decision is a string that is stored in a text file. The overall precision of the model lies between 90 and 100%.

In [15], using the ANPR idea for smart check-in and check-out is narrated to minimize waiting time and keep records of vehicle entries. Images are captured from CCTV cameras, so ANPR instantly identifies the plate number via image processing and stores vehicle registration and security information in a database to access it electronically through a web application. The protection and approved staff may monitor the progress of vehicles with the details of their owners through this device. The proposed solution would minimize check-in time while bringing additional advantages in terms of parking and traffic surveillance.

A state of art image processing technique for dealing with noisy and substandard Indian number plates, in [16], as previously presented models, performs well under controlled conditions such as clear text, fine illustration, and perfect angles. The shoddy images undergo morphological operations and Gaussian rectification as preprocessing. Plate detection is done by applying contours formulated on character cartography and proportionality. Then, the extracted region of interest is filtered and deskewed to pass to the KNN model for character identification. The following work gives an accuracy of 96.22%.

Virakwan and Nui Din in [17] suggested Automatic Vehicle Number Plate Recognition System in POLIMAS to verify that only permitted automobiles set foot on the

campus. A webcam is installed considering four orientations: front side, rear side, front top, and rear top. The seized image is converted to grayscale; then histogram equalization technique helps to adjust the intensity and contrast. Two combinations which are Canny and Laplacian edge detector and Sobel edge and Laplacian edge detectors are introduced. The best one will be contemplated in the process. Succeeding plate discrimination, the bounding box method is used for identifying and cropping characters. Each character is compared in OCR using eigenvector and correlation. The outcomes are accumulated in string configuration and differentiated with the reserved entries in the database. Barrier unbolts if the plate matches the database; else, entrance is opposed.

A 2020 study published by Shivani et al. [18] reviews ANPR systems for vehicular applications deployed to track the traffic jam, intelligent parking complex, toll fee collection and protection of the drivers, etc. Numerous propositions are used to build mechanized vehicle plates, namely, Cloud-Computing based, Artificial Neural Network (NN) based, morphological based, composite based, template matching based, etc. A study shows 97.7% of precision where OCR, NN, Support Vector Machine (SVM), and Hierarchical Temporal Memory (HTM) are applied. Another research indicates that the histogram approach with Machine Learning (ML) fails to detect vehicle number plates under diverse illumination. Further, genetic NN and forward backpropagation NN are capable of variable segmentation and recognition. OCR has the lion's share when evaluating techniques for ANPR, but more coherent systems are still to be presented.

Agrawals in [19] portrayed a Cognition Number Plate Recognition System centered on Machine Learning (ML) and Data Visualization techniques. Loaded images passed through the minimum filter and changed to grayscale for getting a subimage of the localized number plate. Further characters are circumscribed and parsed for character identification using ML methods. 20 images for a total of 36 characters are converted to a one-dimensional array. These arrays are considered a feature for model training. The trained model recognizes segmented characters and passes across template matching. The outcome is exhibited to the user and saved in the database.  $K$  means clustering on a dataset is performed for number plates considering car brands, namely Nissan, Ford, Honda, etc. The plate localization accuracy is 97.2% while it is 73.4% for character identification. The average precision of the proposed model that efficiently tackles adverse conditions is 88.38%.

Although there are a lot of vehicle recognition systems in Europe, the USA, China, Korea, India, Singapore, and several other well-developed countries, hence there is no such ANPR system for Pakistani number plates. Nevertheless, little work is available for Pakistani number plates recognition, due to the variety of nonstandard number plates. An attempt to develop a usable and efficient ANPR system for the multiscrypt Pakistani number plate is done in [20]. The method used distinct feature matching acceptable for Pakistani plate conditions. The introduced method was tested on 50 images and had a margin of improvement for

further implementation in a real-time environment. A rational methodology especially for Pakistani number plates recognition is designed in [21]. A dataset consists of 900 images collected from six different regions of Pakistan. 80% standard and 20% nonstandard number plates following some constraints of distance, brightness, and pixels. The captured image is transfigured to grayscale, and for noise removal, it is passed to the Gaussian filter. As to approach plate localization Canny edge detector and contouring are carried out. Subsequently, every distinct character is fragmented into distinct images. The KNN technique is utilized for the recognition of characters. The average performance is 93 percent.

Table 1 depicts the summary of the comparative study of the previously presented methods for ANPR. After analyzing the performance metrics of the proposed methods, we inferred that their localization techniques for number plates would not generate the best results. The performance rate column defines the existing method's performance rates in terms of accuracy and mean average precision score. After this, the discussion and remarks column shows some major points which are discussed in previously proposed papers and are affecting their performance rate.

### 3. Proposed ANPR Framework

After reviewing the previous papers, we observed that everyone followed the typical ANPR framework [22–24] in which they usually used character segmentation or template matching having average results.

In the United Kingdom, one of the cop stations invented the primary Automatic Vehicle plate recognition system [25]. Moreover, many hardware and software-related ANPRs have been developed using evolving technology to improve accuracy. But unfortunately, corresponding to traffic concerns, adequate precision is not attained [4]. Vehicle plate localization and identification become a daring assignment [26]. An ANPR [27] is a framework that scans the alphabets and digits on the vehicle number plate.

We proposed a novel ANPR framework in Figure 1 that gives better results in less computation time. The five phases that comprise our ANPR framework are as follows: number plate image capture, plate extraction, image preprocessing, character recognition, and number plate label management.

**3.1. Image Acquisition.** Image acquisition is the fundamental stage of any vision system. The following stage, known as image acquisition, is to transform an optical image (real-world features) into a numerical data array for future manipulation. Pakistani vehicles images are considered for further preprocessing. We used low-resolution images of Pakistani number plates having various formats in terms of illumination, orientation, brightness, etc.

**3.2. Number Plate Localization.** The number plate can be detected anywhere in the vehicle. The image is devised for robust plate detection. ANPR performed under constraints as brightness, angle of the plate, and resolution can use basic

TABLE 1: Summary of the comparative study of the existing methods.

Ref no.	Plate's type	Proposed method	Performance rate/accuracy	Discussion/remarks
[2]	Real-time images	Niblack threshold, blob-coloring, neural network -based OCR	86.1% for recognition	Addressing low-resolution images with an average computation time is 1.5 seconds
[3]	Nepali number plates	Grayscale, morphological operation, median filter, phase correlation, cross-correlation in template matching	67.98% for cross-correlation, 63.46% for phase correlation	Due to template matching the average accuracy is low
[4]	Indian number plates	Basic preprocessing, PCA for feature extraction, CNN classifier for recognition	Successful execution is done by using Raspberry Pi	Suitable resources are discussed in it
[5]	Real-time images	YOLOv2, Warped Planner Object Detection Network (WPOD-NET) for detection, OCR for recognition	For detection is 76.8% and for recognition is 75%	Focus on unconstrained images having single-row number plates
[6]	Real-time images	Grayscale, binarization, masking for plate detection, distinguishing definite characters by SVM (deployed using MATLAB 2010a)	92% accuracy for recognition	Cannot recognize motion blurred, overlapped, skewed, and plate with a different language
[7]	Indian number plates	Grayscale, binarization, contrast extension, median filter, MATLAB region props function for segmentation, zonal function for feature extraction, template matching for recognition	The recognition rate lies between 75% and 85%	Addressing low resolution, unskew and clear images
[8]	Qatar number plates	Rescaling, morphological operation, connected component analysis (CCA), vector crossing, zoning, template matching	Recognition rate is 99.5% with 0.63 ms computation time	High-resolution and single-row images are addressed
[9]	Ghanaian number plates	Grayscale, Gaussian kernel, Sobel edge detector, CCA on a binarized image, Tesseract OCR for character recognition	Recognition rate is 60% with 0.2 s computation time	Up to a distance of 5 meters, the detection algorithm performs fairly efficiently
[10]	Real-time images	Desaturation, segmentation, plate recognition using Raspberry Pi	Recognition rate is 85% with a 3-second delay	The system manages to deliver good results when the subject is within 2 meters from the camera
[11]	Indian number plates	YOLOv3 for detecting and recognition	100% for detection, 91% for recognition	High-resolution and single-row number plates are focused
[12]	Indonesian number plates	Thresholding, morphological operations, KNN for recognition	98% accuracy for recognition	Damage and cut-off characters are unidentifiable
[13]	Malaysian number plates	Sauvola threshold, template matching for character identification	83.17% average accuracy for recognition	High-resolution images capture from the distance of 1.5 meters to 2 meters
[14]	European number plates	Undersampling, quantization, binary masking, Tesseract OCR for character segmentation and recognition	90–100% average precision for recognition	High-resolution single-row number plates are focused
[15]	Real-time images	Binarization, minimum filter to enhance dark values, Roberts edge detection, bounding boxes, template matching	No accuracy information because they preferred plate label management	Suitable resources are discussed in it
[16]	Indian number plates	Morphological operations, Gaussian filter, deskewing, KNN	96.22% for recognition	Focused on the single-row number plate
[17]	Indonesian number plates	Histogram equalization, canny and Laplacian or Sobel and Laplacian edge detection, bounding box, cropping, OCR using eigenvector	No accuracy information because they preferred hardware configuration	Suitable resources are discussed in it
[20]	Pakistani number plates	Histogram equalization, distinct feature matching	93% accuracy for recognition	Medium resolution with the single-row number plate
[21]	Pakistani number plates	Grayscale, Gaussian filter, canny edge detector, KNN	93% accuracy for recognition	Mainly focus on a car number plate

image processing techniques discussed in [23, 28–30]. Contouring is a method used for shape scanning and entity localization and identification but it would not give us accurate results in all situations; for example, the image is tilted. For the identification of number plates, a deep neural network-based YOLO (You Only Look Once) model is recommended. To analyze the performance of the YOLO family on proposed ANPR methods, we have

trained YOLOv3 and YOLOv4 to localize the number plate region.

*3.3. Image Preprocessing.* Image preprocessing [31] is substituted for functionalities performed on nonfigurative images. The objective is the augmentation of image information that subdues the undesirable deformation or

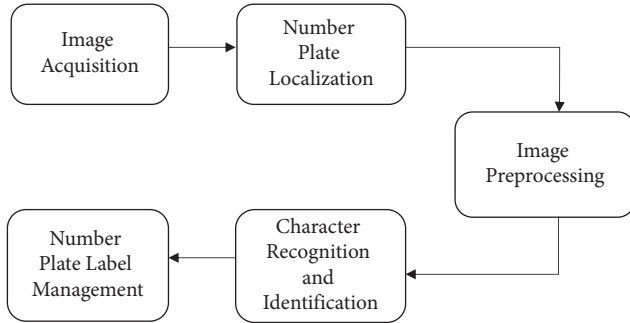


FIGURE 1: Proposed ANPR framework.

amplifies features for further process. The methods involved in the paper for image preprocessing comprise gray scaling, binarization, thresholding, and histogram equalization, etc. During this stage, such images are obtained which are passed to algorithms for further training and predictions.

**3.4. Character Recognition and Identification.** In the recognition stage, the characters from the image are recognized by using different approaches. A commonly used technique behind character recognition is the Artificial Neural Network (ANN). ANN is considered a script since the neural network comprises interconnecting artificial neurons and a mathematical definition by using feedforward back-propagation (BP). In [25], BP is proposed as a neural network-based method. The most important neural network for character recognition is the Convolutional Neural Network (CNN) which is used in Section 4 as an implemented method of the previous ANPR system. Our proposed ANPR method for recognition involves OCR Tesseract, which is the time-efficient method.

**3.5. Number Plate Label Management.** At this stage, the number plate's labels are stored in a file to keep a record. We have reconstructed the recognized characters into plate label strings and stored them in the excel file which is further utilized for scoring.

## 4. Implementation of ANPR Methods

According to the above-proposed ANPR framework, we implemented two pipelines having different recognition techniques; the first one includes the deep learning method (CNN) and the second one is OCR Tesseract. These methods being used are applied in different phases from image number plate localization to recognition of plate's characters.

**4.1. Dataset Acquisition.** Due to commercialization and privacy concerns, there was no publicly available dataset of Pakistani number plates. We, hence, prepared our real-time dataset by capturing 1000 images of vehicles having various formats of number plates in terms of size, fonts, plate color, and orientation practiced in different regions of Pakistan; see Figure 2 which depicts some examples of our vehicle dataset.

From two different cameras, i.e., 900 images were captured by the Samsung J series cell phone, and 100 images were captured by Oppo mobile phone. All images were taken from different angles considering environmental conditions like illumination, dust, and fog, etc. The vehicle number plates are comprised of a subset of 36 characters set in which 26 characters are alphabets and 10 are numbers. We resized all high-resolution images into low-resolution fixed-size ( $416 \times 416$ ) images using an online web service given by Roboflow [32]. This helps to make the dataset realistic as the images received from CCTV cameras mostly have low resolution. Further, reduced frame size decreases inference time as the YOLO network runs faster on small-size images. For training the Darknet object detection model, without using any augmentation technique, we split our dataset into 900 training images or 100 validation images. Further, the performance of the model has been examined on 200 images out of these resized 1000 images.

**4.2. Number Plate Localization on Pakistani Vehicle Dataset Using Object Detection.** The inspiration to prefer YOLO (You Only Look Once) [33] model can be found from the literature review where the number plate localization using YOLO models exhibits greater precision. YOLO is a quick and efficient real-time open-source system trained to object detection. To localize the object, the basic working of the YOLO model is applied to an image at multiple scale and location, puts some scores on multiple regions, and then selects only prioritized high scoring of the region as a detection region. We have used two versions of the YOLO family; the first is YOLOv3 and the second is YOLOv4.

Table 2 depicts the parameters which we have set for training the YOLOv3 and YOLOv4 models on images. The learning rate for both models is 0.001. All parameters are the same as for the YOLO family except the maximum batches because it depends on the number of classes.

Figures 3 and 4 depict the average loss and mAP score of YOLOv3 and YOLOv4. The dotted line shows the average loss rate throughout all iterations and batches. While training, the graph shows the average loss under 2.0 which makes the model efficient. The solid line shows the mean average precision score. We achieved 94.3% mAP using YOLOv3 and 99.5% mAP using YOLOv4 on the 0.5 thresholds. The best mAP score is achieved by YOLO v4. The results of localization using both versions are discussed in Section 5.

**4.3. ANPR Method Using CNN.** The CNN is a deep learning algorithm which is a special kind of multilayer perceptron using Gradient Descent Backpropagation learning algorithm. It is proved to be the best neural network in recognition and prediction. To compare the performance criteria of previous methods, we have also trained a CNN-based model by using TensorFlow and Keras. TensorFlow and Keras are python libraries that are mostly used to train models based on large datasets. The dataset which we are using for training purposes is collected from 36 classes of characters in which 864 are training images and 216 are



FIGURE 2: Pakistani vehicle dataset.

TABLE 2: YOLO parameters used for model training.

Parameter	Value
Batch	1
Subdivisions	64
Width	416
Height	416
Channels	3
Momentum	0.949
Decay	0.0005
Angle	0
Saturation	1.5
Exposure	1.5
Hue	0.1
Learning rate	0.001
Burn_in	1000
Maximum batches	2000
Policy	Steps
Steps	4800, 5400

validation images. We gathered a collection of images of characters and then performed rotational and brightness techniques. The dataset which we are using for testing purposes is collected after training the YOLO Number Plate Detection model in which all image resolution is  $416 \times 416$ . We trained 6 layers' sequential CNN model. The modified LeNet architecture of our CNN model is discussed as follows. The model comprises a convolution layer with 32 filters, kernel size (5, 5), and ReLU (Rectified Linear Unit) activation function with an additional max pooling layer with pool size (2, 2). To avoid overfitting, a drop-out of 0.4 is utilized which drops 40% of neurons, while to flatten the node, the information flatten layer is added. Then, a dense layer with 128 outputs and ReLU activation function is added. And the last layer comprises a dense layer with 36 outputs and Softmax (probabilistic final decision) activation function. The 36 neurons are for the total number of outputs as 26 alphabets and 10 numbers. The learning rate we used for training is 0.001 with 50 epochs, categorical\_crossentropy loss function, and Adam optimization function. The CNN model is making predictions with 97.89% accuracy, but after reconstructing the number plate label, the accuracy decreases.

Figure 5 illustrates the overall pipeline of the CNN model, while Figure 6 portrays the resultants of the CNN method series. First, the segmentation of the cropped number plate is done using the CNN model. The CNN generates a string of recognized characters which are ordered before saving into records. The reason behind not considering the CNN for further processing is mainly related to time resources.

*4.4. Proposed ANPR Method Using OCR.* Character recognition from the processed ROIs is done using Tesseract. Tesseract is an open-source API used by the optical character recognition engine with the ability to identify and recognize more than 100 languages. We used pytesseract (Python-Tesseract) as an open-source module for python which recognizes the read the text. It is a wrapper of Google Tesseract OCR Engine which can read all image types and returned the recognized text instead of writing it to a file. Pytesseract is easily used as we simply install the Tesseract OCR engine into our system and then install this library and used it. This library used page segmentation and OCR engine mode. To run OCR on a small region of an image with a different orientation, the page segmentation mode is used. It has 13 engines. The OCR engine mode has a 2 OCR engine which is used for LSTM and legacy engines.

As we know, the OCR gives precise results when a standard image (Perfect Resolution and White background with black characters) is passed. We have taken pre-processing steps for this as it offers an outstanding input option for the OCR stage. Figure 7 portrays the proposed OCR Tesseract-based ANPR pipeline in which the following steps applied on each frame are described in detail. To start with, we have done gray scaling of the extracted image for a better result. It removes all color information leaving the brightness of each pixel. Bilateral filtering is selected which is a denoising technique for smoothing the image. This reduces the redundancy while preserving edges through a nonlinear grouping of adjacent pixel values. The intensities are adjusted to improve the global contrast of the image using histogram equalization. It also enhances the edges of each object region of an image. The conditions of lighting differ

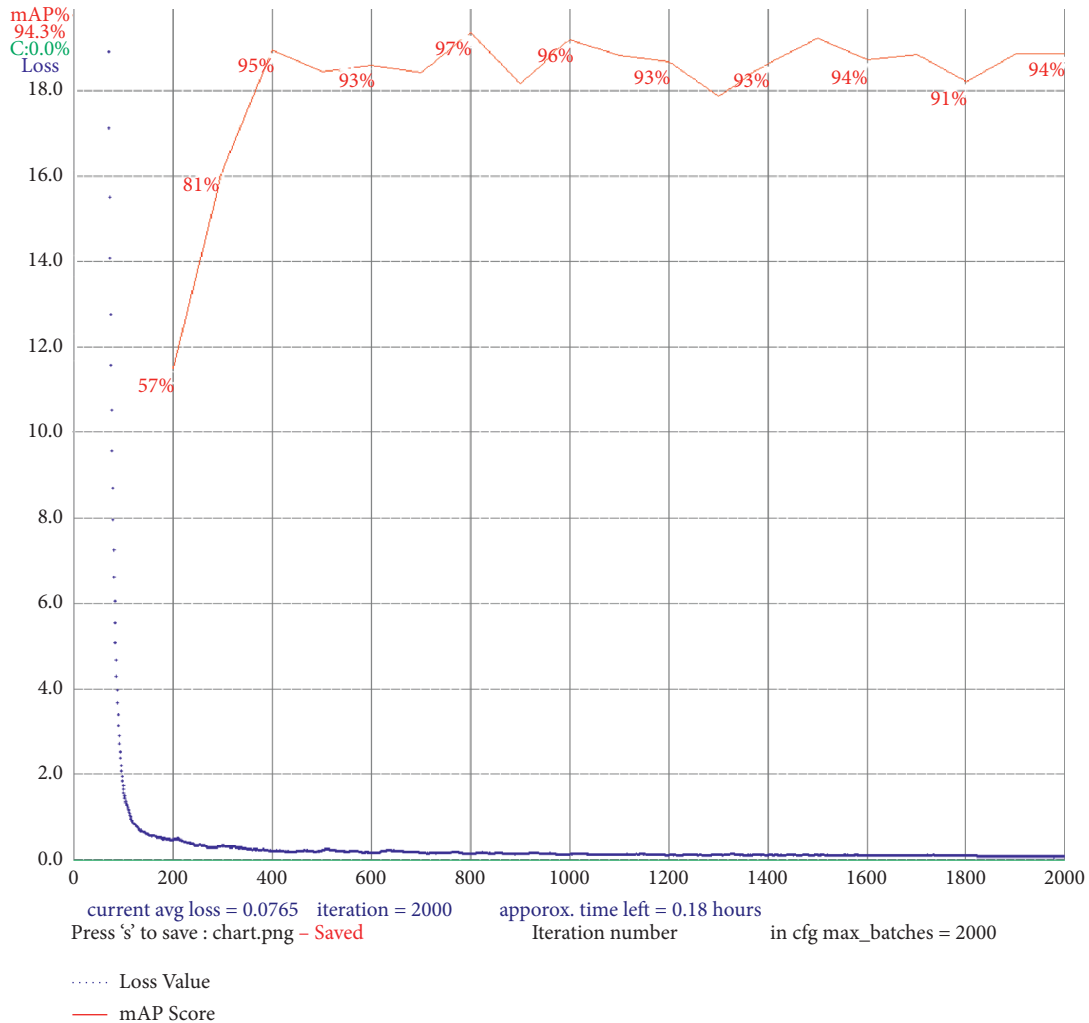


FIGURE 3: Loss and mAP graph of YOLOv3 training.

widely over a plate for which thresholding is important. The binarization process is used to convert grayscale images into black and white pixels (0 bits for black and 1 bit for white). The pixels which have less value than a threshold are converted to 0, and the above threshold image is converted into 1. We used morphology to exclude small objects from the image. The opening function of morphology not only excludes the small objects from the image but also retains the shape and scale of the larger object in the image. The performance of the OCR engine is further enhanced by sharpening. This increases the visibility level of characters and makes it readable for the OCR engine. The results of our proposed ANPR framework are also discussed in Section 5.

## 5. Results and Discussion

In this section, the results of the proposed framework have been presented. We have verified and validated our proposed framework by experimenting on a real dataset and comparing it with the previously proposed approaches using the same accuracy metric.

*5.1. Results of Proposed Pipeline on Single- or Double-Row Vehicle Number Plates.* The raw image is passed to the YOLO model for plate detection. YOLO Darknet neural network framework is applied on a single neural network to image, divides the image into the region, and predicts bounding boxes and scoring probabilities for each region. The highest probability is selected as an object detected area. It can easily localize every type of number plates of vehicles including cars, busses, trucks, and bikes.

To localize the number plate area, we have trained YOLOv3 and YOLOv4. After observing their mAP score, we prefer the YOLOv4 as it gives us the best mAP score. Figure 8 depicts the localization results of YOLOv3 on a double-row number plate. Figure 9 represents the OCR-based ANPR pipeline using YOLO v3.

Figure 10 and Figure 11 exhibit the localization results of the YOLOv4 Darknet model on single-row and double-row number plates, respectively, by creating the boundary around it. The vehicle plate is cropped according to the drawn outline. Extraction of cropped images is done by using the coordinates of bounding boxes.

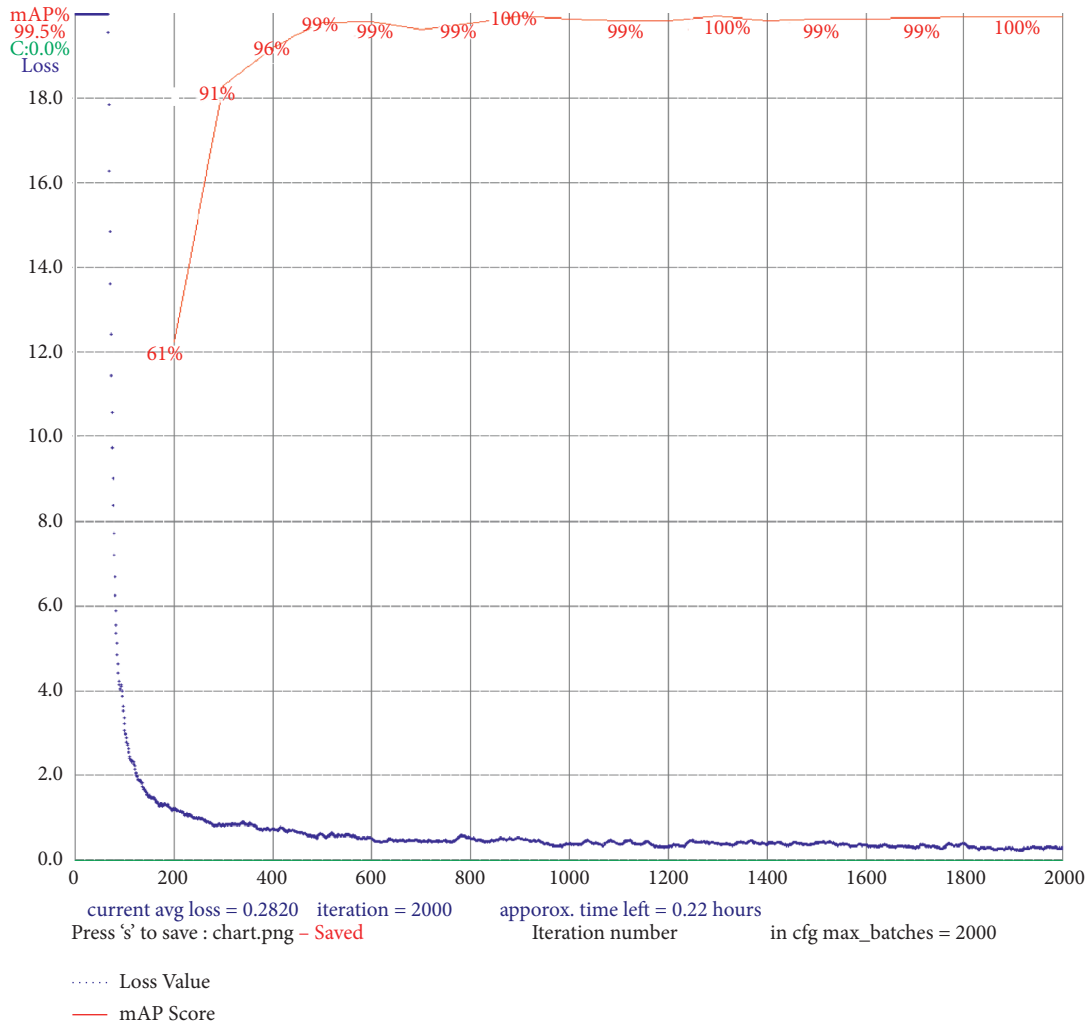


FIGURE 4: Loss and mAP graph of YOLOv4 training.

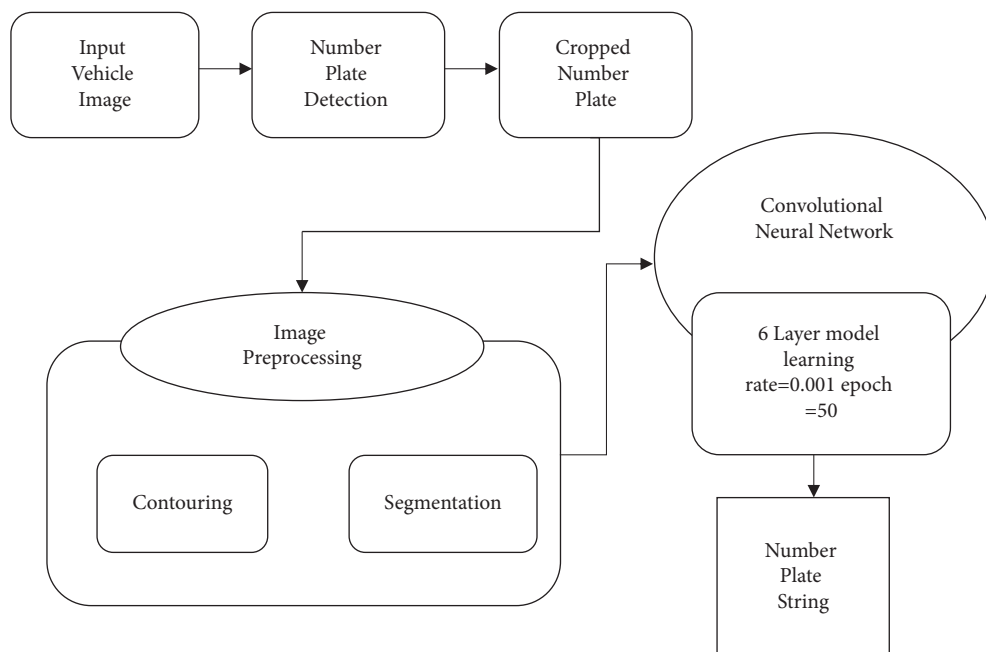


FIGURE 5: CNN-based ANPR pipeline.



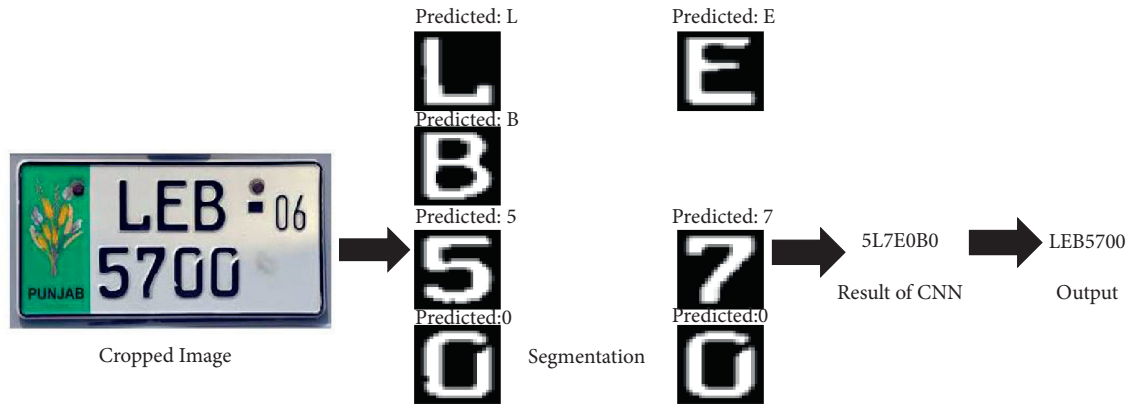


FIGURE 6: Resultant pipeline of CNN-based ANPR.

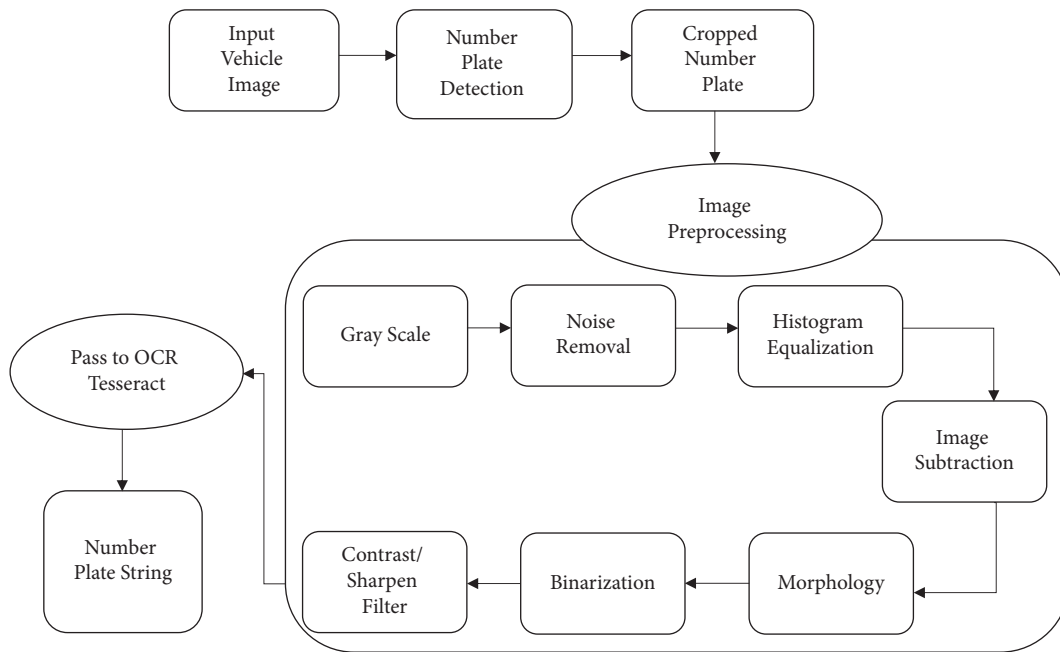


FIGURE 7: OCR Tesseract-based ANPR pipeline.



FIGURE 8: Localization result of YOLOv3 on the double-row image.

The localized cropped image is passed to the chain of preprocessing techniques like gray scaling, noise removal, morphology, and binarization before proceeding to the OCR engine for best recognition of the car’s plate label in Figure 12.

Table 3 summarizes the performance of implemented pipelines involving CNN and OCR Tesseract. To compare uniformly, we used the same metric for measuring the performance of the OCR Tesseract and CNN model. We have calculated the average score using fuzzywuzzy python library in which every predicted string is compared with the actual string and generate a score on the base of correct occurrences of string character. For the evaluation of the proposed method, we used a 64-bit system with a Core i5 CPU having a  $3.00 \times 6$  Gigahertz processing speed. After analyzing our pipeline computation time on CPU, our proposed method works well with less computation time of 0.8seconds per image. We prefer the YOLOv4 for localization as it is efficient and has more precision rate than the YOLOv3.

5.2. Comparative Analysis with Previous Papers. In spite of the fact that there are a lot of vehicle number plate

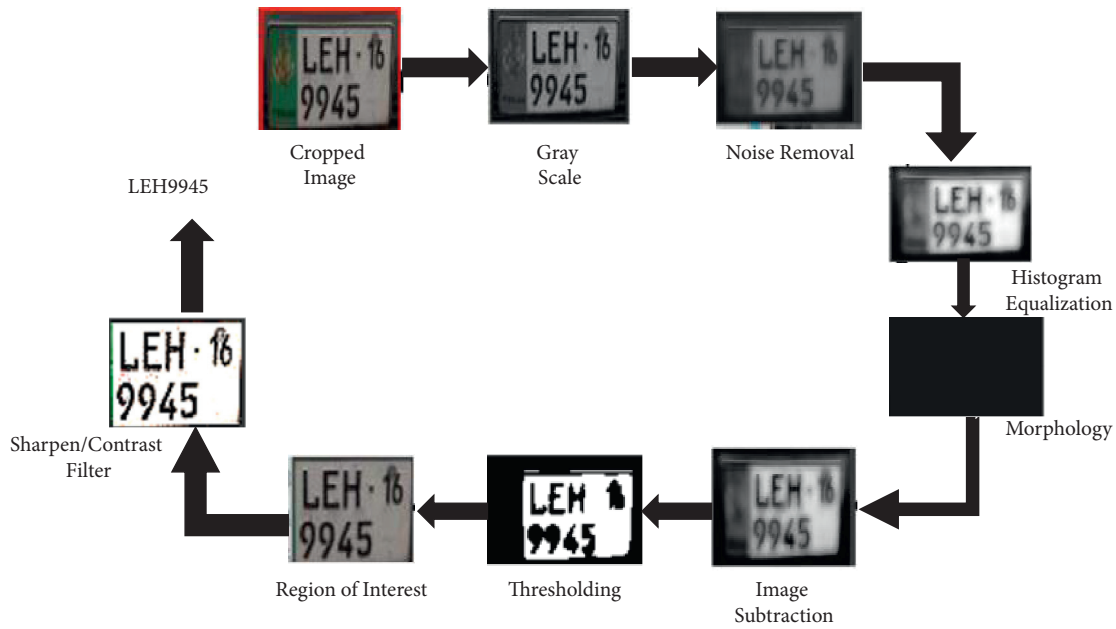


FIGURE 9: OCR Tesseract-based ANPR pipeline results on the double-row plate using YOLOv3.



FIGURE 10: Localization result of YOLOv4 on the single-row image.



FIGURE 11: Localization using YOLOv4 on the double-row image.

recognition systems for Europe, the USA, China, Turkey, Korea, India, Singapore, Malaysia, and many other well-developed countries, for Pakistani number plate identification, little work is present and it still needs more consideration due to the diversity of nonstandard number

plates. The proposed research work is targeting the identification and visualization of Pakistani number plates by using the OCR Tesseract library. We compared our ANPR framework with an existing ANPR method involving the KNN approach proposed for Pakistani number plates [21].

In Figure 13, we inferred by using the preprocessing methods of [21] on our images that their pipeline did not recognize the characters well due to the poor localization. Further, their preprocessing techniques are only applicable to car vehicle types.

In the view of Table 4, when we applied their [21] ANPR framework on our dataset, the processing computation time taken per image is approximately 44 seconds with an average score of 9% using the fuzzywuzzy library. It did not recognize the plate characters well, due to poor localization done by edge detection. Moreover, after applying our ANPR approach on their given dataset of resolution (1140 × 641), we achieved a better outcome with a 71% average accuracy score in less computation time as the localization is improved by the use of YOLOv4.

From the comparative analysis in Table 5, we can easily observe that our proposed ANPR system achieved the best results. The previous ANPR system used different techniques, namely, edge detection, histogram equalization, and CCA for localization which is not an efficient way to handle every type of vehicle number plate. Their preprocessing techniques did not work well on double-row number plates due to the poor localization approach. Our employed YOLO-based ANPR framework performs fast and more efficiently for detecting such challenging plates. They used different metrics to evaluate their performance rate; hence, we used the average score metric for our proposed method by using the fuzzywuzzy python library.

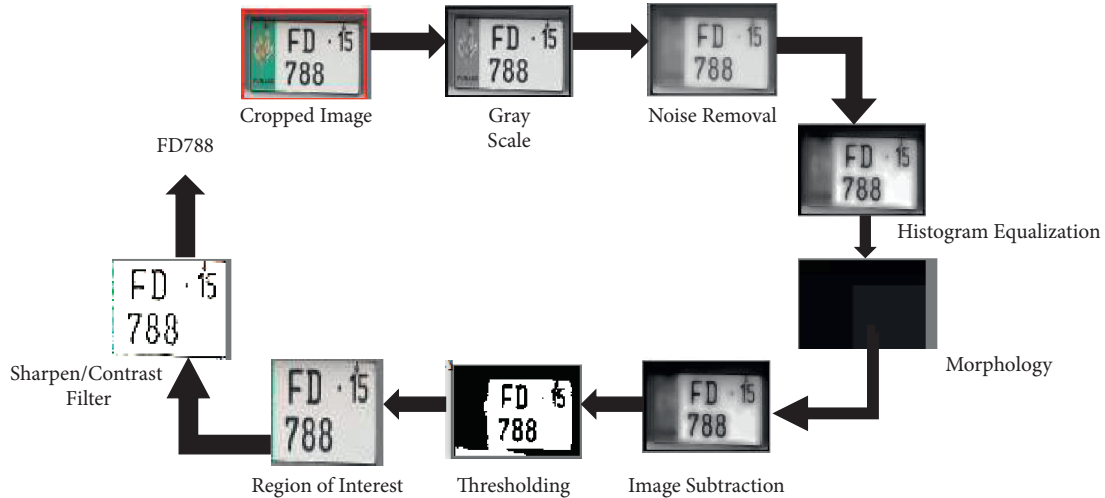


FIGURE 12: OCR Tesseract-based ANPR pipeline results on the double-row plate using YOLOv4.

TABLE 3: Performance metrics of pipeline methods.

Method	Average score on our dataset (%)	Inference time per image (on CPU)
CNN (YOLOv3)	68.3	1.89 seconds
CNN (YOLOv4)	67.76	1.9 seconds
OCR Tesseract (YOLOv3)	70	0.79 seconds
OCR Tesseract (YOLOv4)	73	0.80 seconds



FIGURE 13: Localization result of the proposed method in [21].

TABLE 4: Comparative analysis with proposed method [21].

	Number of testing images	Resolution (pixel)	Computation time on one image	Average score (%)	Localization result
Pipeline in [21] on our dataset	90	416 × 416	44 seconds	9	Poor localization because they used edge detection
Our pipeline on the dataset in [21]	35	1140 × 641	2.7 seconds	71	Best localization because of YOLOv4

Therefore, considering the comparative study of our framework with the existing methodologies, we concluded that, by using our proposed pipeline on the Pakistani dataset, a better result would be achieved. Our

proposed pipeline strategies are universal and applicable on every type of Pakistani vehicle datasets such as cars, bikes, and busses having double and/or single-row number plates.

TABLE 5: Comparative analysis with previous ANPR methods based on Pakistani dataset.

ANPR methods	Vehicle types	Resolution	Localization technique	Character/plate recognition	Segmentation	Testing dataset	Overall accuracy/ average score	Can be implemented in real time?
[20]	Car, bike, and bus	Medium	Vertical and horizontal histogram	OCR	Yes	50	93%	No
[21]	Only car	High (1140 × 641)	Edge detection	KNN	Yes	900	93%	No
The proposed ANPR	Car, bike, and bus	Low (416 × 416)	YOLOv3 or YOLOv4	OCR Tesseract	No	200	70% or 73%	Yes

## 6. Conclusions

This paper presents a novel ANPR framework using OCR Tesseract to address the vast heterogeneity and assortment of number plate genres across the provinces in Pakistan. The proposed OCR pipeline is tested on low-resolution images which are collected by considering the variations in the environment illumination and image orientation. The obtained mAP score for the plate extraction using YOLOv3 is 94.3% and YOLOv4 is 99.5% on 0.50 thresholds. The robust preprocessing techniques are applied to the localized plate image, and finally, the processed image is passed to the OCR Tesseract to recognize the number plate. Our introduced pipeline with the YOLOv4 model takes a 0.80-second average computation time per image and gives an average accuracy score of 73% for recognition. We have also processed our dataset using a previously proposed ANPR framework based on KNN and got an average score of 9% due to the poor plate localization technique. Moreover, we found that LeNet CNN architecture gives 97.89% accuracy for character recognition on our dataset but an average score of 67.76% for complete plate recognition. Furthermore, we experimented with an available constrained high-resolution dataset on our proposed framework and gained an accuracy of 71%. The comparison between our proposed method and the existing ANPR approaches shows that our proposed framework using YOLOv4, preprocessing steps, and OCR Tesseract is applicable in real time. It can be concluded that, to the best of our knowledge, it is the first robust and low complexity ANPR framework for Pakistani number plates that gives reasonable accuracy for unconstrained plates. Finally, this paper can be useful for researchers to develop an ANPR framework for the countries having similar challenging vehicle number plate formats and styles. A promising direction for future research is to test our proposed framework on real-time scenarios that can be installed commercially and improving the pipeline to counter unconstrained scenarios like the distance between camera and vehicle.

## Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National University of Computer and Emerging Science, Chiniot Campus Pakistan. The research work was funded by an Internal National University of Computer and Emerging Science (NUCES) University research grant for faculty.

## References

- [1] Global Automatic Number Plate Recognition (ANPR) System Market Size, 2021 <https://www.bccresearch.com/partners/verified-market-research/global-automatic-number-plate-recognition-anpr-system-market.html/>.
- [2] C. Coetzee, C. Botha, and D. Weber, "PC based number plate recognition system," in *Proceedings of the IEEE International Symposium on Industrial Electronics ISIE'98 (Cat. No. 98TH8357)*, vol. 2, pp. 605–610, Pretoria, South Africa, July 1998.
- [3] G. Sharma, "Performance analysis of vehicle number plate recognition system using template matching techniques," *Journal of Information Technology & Software Engineering*, vol. 8, 2018.
- [4] N. Abirami, "Accurate vehicle number plate recognition and real-time identification using raspberry pi," *International Research Journal of Engineering and Technology IRJET*, vol. 5, no. 4, p. 7, 2018.
- [5] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, pp. 580–596, Munich, Germany, September 2018.
- [6] B. Aishwarya, "Automatic detection and recognition of vehicle plate numbers using svm," *International Journal of Advanced Research and Development*, vol. 5, no. 6, 2018.
- [7] A. Kashyap, B. Suresh, A. Patil, S. Sharma, and A. Jaiswal, "Automatic number plate recognition," in *Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 838–843, Greater Noida, India, October 2018.
- [8] A. Farhat, O. Hommos, A. Al-Zawqari et al., "Optical character recognition on heterogeneous SoC for HD automatic number plate recognition system," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, p. 58, 2018.

- [9] A. Agbemenu, J. Yankey, and E. O. “An automatic number plate recognition system using OpenCV and tesseract OCR engine,” *International Journal of Computers and Applications*, vol. 180, pp. 1–5, 2018.
- [10] A. G. S. Fakhari, M. S. Hamid, A. F. Kadman, and R. Hamzah, “Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi,” *International Journal of Electrical and Computer Engineering IJECE*, vol. 9, p. 1805, 2019.
- [11] R. N. Babu, V. Sowmya, and K. P. Soman, “Indian car number plate recognition using deep learning,” *Instrumentation and Control Technologies (ICICT)*, vol. 1, pp. 1269–1272, 2019.
- [12] S. Sugeng and E. Y. Syamsuddin, “Designing automatic number plate recognition (ANPR) systems based on K-NN machine learning on the raspberry pi embedded system,” *JTEV J. Tek. Elektro Dan Vokasional*, vol. 51 page, 2019.
- [13] F. N. M. Ariff, A. S. A. Nasir, H. Jaafar, and A. Zulkifli, “Sauvola and Niblack techniques analysis for segmentation of vehicle license plate,” *IOP Conference Series: Materials Science and Engineering*, vol. 864, Article ID 012136, 2020.
- [14] V. Nayak, “Automatic number plate recognition,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3783–3787, 2020.
- [15] S. Tenzin, P. Dorji, B. Subba, and T. Tobgay, “Smart check-in check-out system for vehicles using automatic number plate recognition,” in *Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, Kharagpur, India, July 2020.
- [16] R. K. Varma, S. Ganta, and p. Svsrk, “A novel method for Indian vehicle registration number plate detection and recognition using image processing techniques,” *Procedia Computer Science*, vol. 167, pp. 2623–2633, 2020.
- [17] N. D. Kerah, P. Ehkan, V. H. Kelian, N. M. Noor, H. Darus, and E. Loke, “Automatic vehicle identification system using number plate recognition in POLIMAS,” *IOP Conference Series: Materials Science and Engineering*, vol. 767, Article ID 012056, 2020.
- [18] S. Bansal, M. Gupta, and A. K. Tyagi, “A necessary review on optical character recognition (OCR) system for vehicular applications,” in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 918–922, Piscataway, NJ, USA, July 2020.
- [19] R. Agrawal, M. Agarwal, and R. Krishnamurthi, “Cognitive number plate recognition using machine learning and data visualization techniques,” in *Proceedings of the 2020 6th International Conference on Signal Processing and Communication (ICSC)*, pp. 101–107, Noida, India, March 2020.
- [20] S. A. Haider and K. Khurshid, “An implementable system for detection and recognition of license plates in Pakistan,” in *Proceedings of the 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pp. 1–5, Karachi, Pakistan, April 2017.
- [21] S. U. Rehman, M. Ahmad, A. Nawaz, and T. Ali, “An efficient approach for vehicle number plate recognition in Pakistan,” *The Open Artificial Intelligence Journal*, vol. 6, no. 1, 2020.
- [22] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (ALPR): a state-of-the-art review,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013.
- [23] S. Azam and M. M. Islam, “Automatic license plate detection in hazardous condition,” *Journal of Visual Communication and Image Representation*, vol. 36, pp. 172–186, 2016.
- [24] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, “A robust and efficient approach to license plate detection,” *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.
- [25] K. Sonavane, B. Soni, and U. Majhi, “Survey on automatic number plate recognition (ANR),” *International Journal of Computer Application*, vol. 125, no. 6, pp. 1–4, 2015.
- [26] S. N. Hashmi, K. Kumar, S. Khandelwal, D. Lochan, and S. Mittal, “Real time license plate recognition from video streams using deep learning,” *International Journal of Information Retrieval Research*, 2019.
- [27] R. Jogekar, A. Dhoble, S. Kakde, P. Taklikar, and D. Larokar, “Automatic number plate recognition system through smart phone using image processing,” vol. 04, no. 3, 2017.
- [28] N. Saleem, H. Muazzam, H. M. Tahir, and U. Farooq, “Automatic license plate recognition using extracted features,” in *Proceedings of the 2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 221–225, Olten, Switzerland, September 2016.
- [29] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, “Automatic license plate recognition with convolutional neural networks trained on synthetic data,” in *Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSp)*, pp. 1–6, London, UK, October 2017.
- [30] J. A. Khan, M. A. Shah, A. Wahid, M. H. Khan, and M. B. Shahid, “Enhanced car number plate recognition (ECNPR) system by improving efficiency in preprocessing steps,” in *Proceedings of the 2017 International Conference on Communication Technologies (ComTech)*, pp. 156–161, Rawalpindi, Pakistan, April 2017.
- [31] K. Sato, “Image processing device and method,” *US10755444B2*, 2020.
- [32] “Roboflow: everything you need to start building computer vision into your applications,” <https://roboflow.ai>.
- [33] “YOLO: real-time object detection,” <https://pjreddie.com/darknet/yolo/>.

## Research Article

# Sublemma-Based Neural Machine Translation

Thien Nguyen <sup>1</sup>, Huu Nguyen <sup>2</sup>, and Phuoc Tran <sup>1</sup>

<sup>1</sup>Natural Language Processing and Knowledge Discovery Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

<sup>2</sup>Faculty of Information Technology, Ho Chi Minh City University of Food Industry, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Thien Nguyen; [nguyenchithien@tdtu.edu.vn](mailto:nguyenchithien@tdtu.edu.vn)

Received 14 May 2021; Revised 15 June 2021; Accepted 24 September 2021; Published 8 October 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Thien Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Powerful deep learning approach frees us from feature engineering in many artificial intelligence tasks. The approach is able to extract efficient representations from the input data, if the data are large enough. Unfortunately, it is not always possible to collect large and quality data. For tasks in low-resource contexts, such as the Russian → Vietnamese machine translation, insights into the data can compensate for their humble size. In this study of modelling Russian → Vietnamese translation, we leverage the input Russian words by decomposing them into not only features but also subfeatures. First, we break down a Russian word into a set of linguistic features: part-of-speech, morphology, dependency labels, and lemma. Second, the lemma feature is further divided into subfeatures labelled with tags corresponding to their positions in the lemma. Being consistent with the source side, Vietnamese target sentences are represented as sequences of subtokens. Sublemma-based neural machine translation proves itself in our experiments on Russian-Vietnamese bilingual data collected from TED talks. Experiment results reveal that the proposed model outperforms the best available Russian → Vietnamese model by 0.97 BLEU. In addition, automatic machine judgment on the experiment results is verified by human judgment. The proposed sublemma-based model provides an alternative to existing models when we build translation systems from an inflectionally rich language, such as Russian, Czech, or Bulgarian, in low-resource contexts.

## 1. Introduction

Many neural models have been introduced for machine translation [1–5]. Although they have different architectures, they all follow the sequence-to-sequence pattern. Source sentences are represented as sequences of source units. The source sequences are processed by the neural models; then the models generate corresponding sequences of target units. The target sequences are then concatenated to form target sentences. The most intuitive representation of source/target units is words. If the bilingual datasets used to train neural machine translation (NMT) models are large enough, the models will be able to learn reliable statistics of source/target words. Unfortunately, in practice, there are many cases of scarce data, such as Russian → Vietnamese translation tasks. The language pair is of low resource. Moreover, Russian is a highly inflectional language. A word can have different forms according to its grammatical role in

sentences. The property leads to a high chance that we will meet word forms which do not occur frequently enough in humble-size training datasets.

The first attempt to solve the scarce data problem for Russian → Vietnamese translation tasks can be found in the work of Nguyen et al. [6]. The authors use a mixed-level representation system, where Russian source units are subwords, and Vietnamese target units are words. Due to the division of Russian words, rare words are replaced by more common subwords; therefore, the severity of the rare-word problem is reduced. Another solution to the scarce data problem for Russian → Vietnamese translation tasks is proposed by Nguyen et al. [7]. They decompose a Russian source word into a set of linguistic features: part-of-speech, morphology, dependency label, and lemma.

We have assessed the available approaches of unit representation on a Russian-Vietnamese bilingual data collected from TED talks [8]. Experiment results show that

the decomposition approach significantly outperforms mixed-level representation. Nevertheless, we still believe in the effectiveness of subword representation, which has become a default part of many NMT models [9–13]. Therefore, we experiment combining source-word decomposition and subword representation. Specifically, we perform a two-step procedure. First, we decompose a Russian source word into a set of features as in source-word decomposition approach. After that, we continue to divide lemmas into sublemmas using BPE algorithm [14]. Since many Russian lemmas are derived from the same root with different prefixes or suffixes, it makes sense to divide them into smaller parts. For example, the verbs “Приходить” (to arrive), “входить” (to enter), “Пройти” (to go by), “Подходить” (to approach), “Выходить” (to leave), “Достигать” (to reach), and “Уходить” (to leave) have the same root part “ходить” (to go), with a prefix added to modify their meaning. Sometimes, both prefixes and suffixes modify the same root to create different lemmas; for example, the verbs “являть” (show), “Появлять” (appear), “являться” (to be shown), and “Появляться” (to be appeared) have the same root “являть.”

In total, we propose a sublemma-based NMT model for Russian  $\rightarrow$  Vietnamese translation. On the Russian source side, we represent a translation unit as a combination of part-of-speech tag, morphology, dependency labels, and a list of sublemmas with their corresponding tags informing that a sublemma is the beginning, middle, or final part of a lemma. On the Vietnamese target side, we tokenize sentences into sequences of subtokens with BPE algorithm. A token is a sequence of characters delimited by space. In Russian, a token is a word. In Vietnamese, there are few cases when a token is a word. Usually, a Vietnamese token is a syllable. In this work, we use the term “subtoken” to indicate a part of a token regardless of whether it is a word or syllable.

This work is composed of six sections. This first section introduces our study. The second section reviews related works. The third section describes our proposed sublemma-based NMT model revised from the state-of-the-art Transformer NMT model. The fourth section describes materials and methods. The fifth section presents the experiment results and analysis. Conclusions from this work are given in the final section.

## 2. Related Works

In this section, we briefly describe the approaches of translation unit representation in NMT models which influence our study.

While the use of linguistic features as part of a translation unit is widespread in traditional factored statistical models [15–18], it is only recently that Sennrich and Haddow [19] has applied it in a modern deep model. The authors complement a source word with its features. As a result, they represent a source unit as a combination of a source word and its linguistic input features. Their approach performs well for English  $\leftrightarrow$  German and English  $\rightarrow$  Romanian translation tasks. For their Russian  $\rightarrow$  Vietnamese translation system, Nguyen et al. [7] made a step further by

removing source words in the list of features. They represented a source translation unit as a combination of linguistic features: part-of-speech, morphology tag, dependency label, and lemma. On the target Vietnamese side, they simply used words as translation units. Their NMT model with source-word decomposition outperformed baseline NMT models including the one by Sennrich and Haddow [19]. Their source-word decomposition is the first processing step in our two-stage procedure to represent a source translation unit.

To handle the rare-word problem, Kudo and Richardson [20] created a language-independent word segmentation algorithm, SentencePiece, to divide words into subwords. Their work comes from an intuition that smaller units of rare words, such as compounds, are easier to translate. They demonstrated the quality of their algorithm in an English  $\rightarrow$  Japanese translation task. As in the work of Kudo and Richardson [20], Sennrich et al. [14] adapted byte pair encoding (BPE) algorithm originally used for compression to divide words into subwords. First, they considered characters as translation units. Considering words as sequences of translation units, they merged their frequent pairs to form new translation units. They repeated the merging process for a predefined number of times. Clearly, their approach is also language-independent. They reported improvements in translation quality for English  $\rightarrow$  German and English  $\rightarrow$  Russian translation tasks. In this work, we actually apply BPE algorithm for representing source translation units. Instead of word segmentation in the original work, we use the algorithm to divide lemmas into sublemmas, since we have already decomposed Russian words into features including the lemma in the first place.

Being language-independent tools, BPE and SentencePiece algorithms are really popular, since they can operate for all languages. However, these wonderful tools should not be utilized blindly. In a Russian  $\rightarrow$  Vietnamese news translation task, Nguyen et al. [6] showed that an NMT model with mixed-level representation outperformed a baseline NMT model where BPE algorithm was applied on both translation sides. Influenced by a work on a traditional statistical machine translation model for Chinese  $\rightarrow$  Vietnamese [21], the authors only applied BPE algorithm on Russian source side, while using words on Vietnamese target side, considering the different effects of BPE algorithm on each side of their bilingual corpus. Although their approach is interesting, it fails to take into account rare foreign named entities, which are commonly found in Vietnamese texts translated from a foreign language. Since our bilingual corpus contains many foreign named entities on both sides and we already apply BPE algorithm on the source side, we opt to use BPE method to tokenize Vietnamese target sentences into sequences of subtokens.

*2.1. Sublemma-Based Transformer Model.* Following the recommendation of Nguyen et al. [22], our sublemma-based NMT model is based on the state-of-the-art model Transformer [4]. The proposed model has a similar architecture

except for the embedding layer of the encoder of Transformer. In this section, we describe the source and target translation unit representation and the encoder of Transformer model which is revised to adopt the proposed translation unit representation.

## 2.2. Translation Unit Representation

### 2.2.1. Sublemma-Based Representation of Source Translation Units.

We represent a source translation unit as a combination of sublemma-based features, following a two-step procedure.

In the first step, we transform a Russian source sentence into a sequence of linguistic features: part-of-speech (POS), morphology (MOR), dependency label (DEP), and lemma (LEM), following source-word decomposition approach [7]. The grammatical parsing is performed with the help of a natural language processing toolkit, Stanza [23]. Typical part-of-speech tags of Russian words are shown in Universal Dependencies treebank [24], such as nouns, pronouns, verbs, auxiliary, numerals, particles, determiners, adjective, and adverbs. Russian has a rich morphology. A Russian word is inflected from an original lemma, depending on its part-of-speech and grammatical role in sentence. A word’s grammatical role in a sentence is denoted with a dependency label [25]. An example of a short Russian sentence being transformed into a sequence of linguistic features is presented in Table 1.

In the second step, we apply BPE method, segmenting lemmas into sublemmas. After the segmentation, the sequence of sublemmas is longer than its corresponding sequences of other features. Following the work of Sennrich and Haddow [19], we broadcast the sequences of other features, so that they have the same length as the sequence of sublemmas. Specifically, all sublemmas extracted from a lemma will have the same labels of features corresponding to the lemma. Moreover, using their subword notation, we assign a tag to each sublemma (TAG), depending on the position of the sublemma relative to the initial lemma. A sublemma can be the beginning (B), inside (I), ending (E), or the full lemma (O). In addition, the beginning and inside sublemmas are suffixed with characters “@@” to inform their roles. An example of sublemma-based sequences of linguistic features is shown in Table 2.

In total, we represent a Russian source sentence as a sequence of collections of sublemma-based features: sublemma, sublemma tag, part-of-speech tag, morphology label, and dependency label. Each source translation unit is represented as a collection of its features.

### 2.2.2. Target Translation Unit Representation.

Applying BPE algorithm [14], we segment Vietnamese target sentences into sequences of subtokens. The algorithm appends characters “@@” to the beginning and inside subtokens for later merge operations. Sequences of target subtokens are used to train the translation model. Generated sequences of target subtokens are merged to form target sentences, based on the

characters “@@.” A Vietnamese sentence and its corresponding sequence of subtokens are shown in Table 3.

In Table 3, we can see that BPE algorithm focuses on tokens which are the foreign named entity “Geographic Society.” It segments the entity into a sequence of subtokens “Geo@@ graphic So@@ ci@@@ e@@@ ty.”

### 2.3. Embedding Layer in the Encoder of Sublemma-Based Transformer Model.

As in [7, 19], we consider all features  $x_{ij}$  from the  $i$ -th source translation unit in a source sequence as strings in their respective domains  $x_{ij} \in \mathcal{S}_j$ , where  $\mathcal{S}_j$ ,  $j = 0, \dots, 4$ , is the set of sublemmas, sublemma tags, part-of-speech tags, morphology labels, and dependency labels, respectively. The trainable embedding  $\mathbf{e}_{ij}$  of a feature  $j$  is extracted from a corresponding dictionary  $f_j: x_{ij} \mapsto \mathbf{e}_{ij} \in \mathbb{R}^{d_j \times |\mathcal{S}_j|}$ , where  $d_j$  is a predefined size of embeddings of the feature  $j$  (equation (1)).

$$\mathbf{e}_{ij} = f_j(x_{ij}), \quad (1)$$

and the embedding of a source translation unit is represented as the concatenation of embeddings of its features (equation (2)).

$$\mathbf{e}_i = \text{concat}(\mathbf{e}_{ij}, \text{ for } j = 0, \dots, 4). \quad (2)$$

Since Transformer model does not leverage the order of translation units in its core layer, it deploys a positional embedding principle, such as sinusoidal positional embedding  $\mathbf{p}_i$  [4]. In total, the  $i$ -th source translation unit in a source sequence has the overall embedding computed as in the following equation:

$$\mathbf{o}_i = \sqrt{d} \times \mathbf{e}_i + \mathbf{p}_i, \quad (3)$$

where  $d = \sum_{j=0}^4 d_j$ .

## 3. Materials and Methods

### 3.1. Materials.

To assess NMT models, we used a bilingual Russian-Vietnamese corpus consisting of sentence pairs of length in the range (10 tokens, 30 tokens) extracted from TED talks [8]. The chosen sentences are ended with a punctuation mark and contain only word characters and punctuation. As in [26–28], we randomly divide the corpus into three datasets: training, development, and testing datasets. Specifically, a set of 47750 sentence pairs are randomly selected from the corpus and used as the training dataset. Furthermore, a set of 1500 sentence pairs are selected from the left corpus and used as the development dataset. The remaining 1500 sentence pairs are used as the testing dataset. Statistical summary of the datasets is presented in Table 4.

In Table 4, we use the term “token” to denote a sequence of characters delimited by space. Linguistically, it can be a Russian word, a Vietnamese syllable, or a punctuation.

## 4. Methods

We compared the proposed sublemma-based Transformer model with three baseline Transformer models. These



TABLE 1: A short Russian sentence with its corresponding sequence of linguistic features.

Words	POS	MOR	DEP	LEM
когда	SCONJ	–	Mark	когда
вода	NOUN	Animacy = Inan, Case = Nom, Gender = Fem, Number = Sing	Nsubj	вода
Поднимается	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 3, Tense = Pres, VerbForm = Fin, Voice = Mid	Advcl	Подниматься
,	PUNCT	–	Punct	,
Потом	ADV	Degree = Pos	Advmod	Потом
отстывает	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 3, Tense = Pres, VerbForm = Fin, Voice = Act	Root	отстывать
,	PUNCT	–	Punct	,
начодишь	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 2, Tense = Pres, VerbForm = Fin, Voice = Act	Conj	начодить
в	ADP	–	Case	в
нем	PRON	Case = Loc, Gender = Masc, Number = Sing, Person = 3	Obl	он
новые	ADJ	Animacy = Inan, Case = Acc, Degree = Pos, Number = Plur	Amod	новый
ракушки	NOUN	Animacy = Inan, Case = Acc, Gender = Fem, Number = Plur	Obj	ракушка
.	PUNCT	–	Punct	.

TABLE 2: Sublemma-based sequences of linguistic features

Sublemmas	TAG	POS	MOR	DEP	From lemma
когда	O	SCONJ	–	Mark	когда
вода	O	NOUN	Animacy = Inan, Case = Nom, Gender = Fem, Number = Sing	Nsubj	вода
Подниматься	O	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 3, Tense = Pres, VerbForm = Fin, Voice = Mid	Advcl	Подниматься
,	O	PUNCT	–	Punct	,
Потом	O	ADV	Degree = Pos	Advmod	Потом
от@@	B	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 3, Tense = Pres, VerbForm = Fin, Voice = Act	Root	отстывать
стывать	E	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 3, Tense = Pres, VerbForm = Fin, Voice = Act	Root	отстывать
,	O	PUNCT	–	Punct	,
начодить	O	VERB	Aspect = Imp, Mood = Ind, Number = Sing, Person = 2, Tense = Pres, VerbForm = Fin, Voice = Act	Conj	начодить
в	O	ADP	–	Case	в
он	O	PRON	Case = Loc, Gender = Masc, Number = Sing, Person = 3	Obl	он
новый	O	ADJ	Animacy = Inan, Case = Acc, Degree = Pos, Number = Plur	Amod	новый
ра@@	B	NOUN	Animacy = Inan, Case = Acc, Gender = Fem, Number = Plur	Obj	ракушка
ку@@	I	NOUN	Animacy = Inan, Case = Acc, Gender = Fem, Number = Plur	Obj	ракушка
шка	E	NOUN	Animacy = Inan, Case = Acc, Gender = Fem, Number = Plur	Obj	ракушка
.	O	PUNCT	–	Punct	.

TABLE 3: A Vietnamese sentence and its corresponding sequence of subtokens.

Vietnamese sentence	“Vì vậy tôi bắt đầu làm việc với tạp chí National Geographic Society cùng các báo khác và dẫn các cuộc thám hiểm tới Nam Cực.”
Sequence of subtokens	“Vì vậy tôi bắt đầu làm việc với tạp chí National Geo@@ graphic so@@ ci@@ e@@ ty cùng các báo khác và dẫn các cuộc thám hiểm tới nam Cực.”

TABLE 4: Statistical summary of the datasets.

Russian/Vietnamese	Training	Development	Testing
Average sentence length	16.1/18.1	16.2/21.2	16.2/21.3
Unique tokens	73205/25939	7202/2646	7120/2692
All tokens	766446/ 866175	24257/31741	24363/ 31948

models are the foundations from which our model is derived. The first baseline model is mixed-level Transformer model [6]. The second baseline model is a subtoken-based Transformer model [14]. The third baseline model is Transformer model with source-word decomposition [7]. We create all models with an open-source library, OpenNMT-tf [29, 30]. The architecture and hyperparameters of the baseline models can be found in the

respective works. Here, we only describe how we build our proposed model.

As reported in the description of the proposed model, we use Stanza natural language processing tool [23] to decompose Russian words into sets of features. Then, we use BPE algorithm [14] with 10,000 merge operations to divide lemmas into sublemmas. We also use the algorithm to divide Vietnamese target sentences into sequences of subtokens. The number of items in each feature domain is presented in Table 5.

We apply the sizes of 179, 11, 22, 22, and 22 for embeddings of sublemmas, sublemma tags, part-of-speech tags, morphology labels, and dependency labels, respectively. In total, we use 256 dimensions for concatenated embeddings of source translation units.

On the Vietnamese target side, we also use 256 dimensions to represent the embeddings of target units.

In addition to embedding layers, the proposed sublemma-based Transformer model consists of 6 hidden layers. The hidden layers contain 8-head attention sublayers and feedforward neural networks of 512 dimensions. Hidden states of the model are comprised of 256 values. To prevent the overfitting problem, we apply a dropout of 0.1 in all hidden layers. To generate translations, the model contains an inference module implementing a beam search algorithm with beam width = 5 [31].

For all models, the training procedure is as follows:

- (1) First, we train the model in 15,000 steps. In each training step, we use 64 sentence pairs from the training dataset to optimize the cross-entropy criterion described in the work of Muller et al. [32]. While there are many efficient algorithms for optimization, we choose to apply LazyAdam optimizer [33], as it is available in the chosen OpenNMT-tf library. We employ the optimizer with  $\beta_1 = 0.9$  and  $\beta_2 = 0.998$  and learning rate = 2.
- (2) Second, we save the values of the model parameters when we complete  $n \times 10^3$  training steps, where  $n \in \{11, \dots, 15\}$ . We use the development dataset to validate the translation quality of all candidate values. The values giving the best translation quality in the development dataset are chosen for the model parameters.

We validate translation quality of the models with the BLEU score [34]. The BLEU scores are computed with the script multi-bleu.perl [35]. BLEU is the abbreviation of “bilingual evaluation understudy,” measuring the similarity of candidate translations to their corresponding references. It is the geometric mean of constituent  $n$ -gram scores, where  $n = 1, \dots, 4$ . All  $n$ -grams are extracted from the candidate translations. While unigrams are individual words, bigrams, trigrams, and four grams are phrases of two, three, and four neighboring words, respectively. We calculate a constituent  $n$ -gram score by dividing the number of the  $n$ -grams appearances in the references by the total number of the  $n$ -grams in the candidate translations.

TABLE 5: Size of vocabulary in sublemma-based Transformer model.

Language side	Vocabulary	Size
Source	Sublemmas	9417
Source	Sublemma tags	4
Source	Part-of-speech tags	15
Source	Morphology labels	484
Source	Dependency labels	38
Target	Subtokens	8628

After training the models, we assess their translation quality using the testing dataset. To have a complete assessment, we employ not only the automatic BLEU scores but also limited human judgment on translation results. We accompany the BLEU score with human judgment, since it has an obvious pitfall. It only measures total matching of  $n$ -grams in the candidate translations and the references regardless of their meaning. To solve the problem, we compare the meanings of the candidate translations and their references, considering synonyms, as well as the similarity of meanings. We do this for all levels, from individual words to phrases and complete sentences.

## 5. Results and Analysis

BLEU scores of the comparative Transformer models are shown in Figure 1.

Among the baseline models, the model with source-word decomposition provides the best scores of 13.52 and 13.84 BLEU in the development and testing datasets, respectively. Fortunately, our proposed sublemma-based Transformer model outperforms the best baseline model in both development and testing datasets, delivering improved BLEU scores of 14.46 and 14.81, respectively. The improvements of 0.94 and 0.97 BLEU are recorded.

The performance order of the models for the development dataset is maintained for the testing dataset: mixed-level model < subtoken-based model < model with source-word decomposition < the proposed sublemma-based model. This consistency makes us more confident about the effectiveness of our proposed sublemma-based model.

In addition to machine judgment with automatic BLEU scores, we semantically studied a limit number of translation results by the two best models: the model with source-word decomposition (from now on, we call it “baseline” model) and the proposed sublemma-based model (from now on, we call it “proposed” model). Five cases in the testing dataset were randomly chosen and studied.

Table 6 shows the source, its meaning, the target, and the predicted sentences by the baseline and proposed models in the first case. The first case seems easy, since both models provide correct translations. Although the models literally choose words different from the reference, the meanings are the same. For example, the verb phrase “phủ nhận” (negate) by the models is similar in meaning to the reference “chối bỏ” (deny).

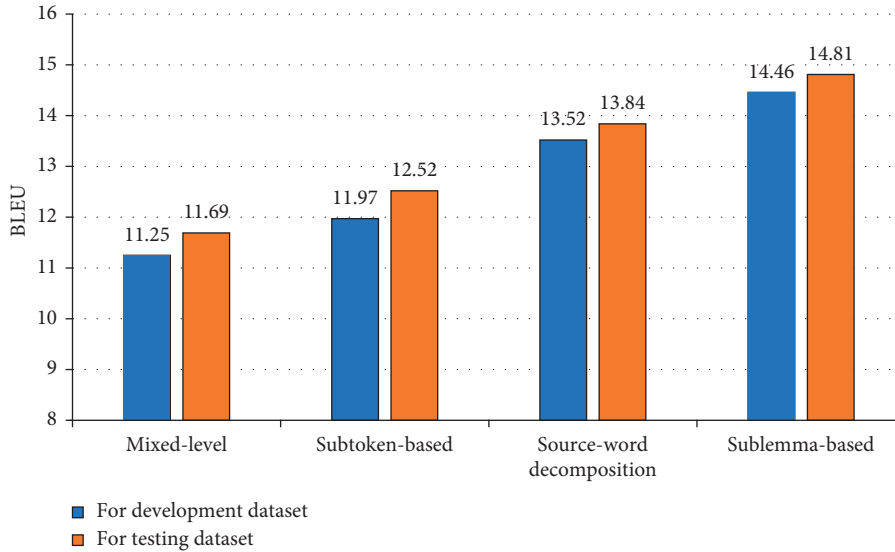


FIGURE 1: BLEU scores of comparative Transformer models.

TABLE 6: First case study with correct translations.

Source	“Мы не можем отрицать факт того, что все Потребление Пищи человечеством имеет Глобальные Последствия.”
Meaning	“We cannot deny the fact that all human consumption of food has global implications.”
Reference	“Không có cách gì để chối bỏ sự thật là những gì chúng ta ăn có ảnh hưởng đến toàn cầu.”
Baseline model	“Chúng ta không thể phủ nhận thực tế rằng tiêu thụ lương thực của loài người có tác động toàn cầu.”
Proposed model	“Chúng ta không thể phủ nhận rằng mọi thứ tiêu thụ thực phẩm trên toàn thế giới đều có hậu quả toàn cầu.”

Table 7 presents the second case study. The general meaning of the reference is found in the translations by the two models, except for a keyword “Phượng hoàng” (Phoenix). The corresponding source named entity “Феникс” (Phoenix) is a rare word; hence both models fail to translate the named entity. Nevertheless, the proposed model semantically performs better than the baseline model in this case. Although the phrase “các hòn đảo” (islands) by the proposed model and the phrase “hòn đảo”(island) by the baseline model are distinct from the reference “Quần đảo” (Archipelago), we think that the former translation is conceptually closer to the reference than the latter translation.

Table 8 demonstrates the third case study. Although the translations by the models contain many reference words, their meanings are not accurate. The key source phrase “с остальным миром” (with the rest of the world) is incorrectly translated into the phrase “với thế giới ngoài không gian” (with the world in outer space) and the phrase “với một thế giới khác” (with another world) by the baseline and proposed models, respectively. Comparing the models with each other, we think that the proposed model is better than the baseline model in this case. The phrase “trong không gian” (in space) by the proposed model better reflects meaning of the source “в Пространственном смысле” (in a spatial sense) than the phrase “ngoài không gian” (in outer space) by the baseline model.

Table 9 shows the fourth case study. Although the proposed model does not generate a translation completely reflecting the meaning of the source, it outshines the baseline model. It even successfully translates the rare named entity “Дубай” (Dubai). At the same time, the baseline model completely fails in this case with an incorrect translation which contains unknown words <unk>.

Table 10 displays the fifth case study. This case again proves the power of the proposed model in translating rare words. It successfully translates the rare source word “биоразнообразия” (biodiversity) into the phrase “sự đa dạng sinh học” as in the reference. The rare word is a keyword in the source sentence. Due to the ability to handle rare words, the proposed model finds itself superior to the baseline model. The translation by the proposed model keeps the meaning of the source sentence. On the other hand, the baseline model misses the key source word and hence provides an incomplete translation.

After semantically studying the test cases, we found that the proposed sublemma-based model tends to provide longer and better translations than the best baseline model. The similarity between manual evaluation and automatic assessment consolidates our proposal of using the sublemma-based Transformer model in place of the model with source-word decomposition.

TABLE 7: Second case study with inaccurate translations.

Source	“Но вернемся обратно к островам Феникс, которые являются темой сегодняшнего выступления.”
Meaning	“But back to the Phoenix Islands, which are the topic of today’s talk.”
Reference	“Nhưng hãy quay lại với Quần đảo Phượng hoàng, đó là chủ đề của bài nói chuyện này.”
Baseline model	“Nhưng quay trở lại với hòn đảo, đó là chủ đề của bài thuyết trình hôm nay.”
Proposed model	“Nhưng hãy quay trở lại các hòn đảo Erex, chủ đề của buổi nói chuyện hôm nay.”

TABLE 8: Third case study with incorrect translations.

Source	“И давайте сравним её с остальным миром в Пространственном смысле.”
Meaning	“And let’s compare it with the rest of the world in a spatial sense.”
Reference	“Và hãy so sánh nó với phần còn lại của thế giới theo giới hạn không gian.”
Baseline model	“Hãy so sánh nó với thế giới ngoài không gian.”
Proposed model	“Hãy so sánh nó với một thế giới khác trong không gian.”

TABLE 9: Fourth case study with a rare named entity.

Source	“Я Переехал в Дубаи на Пост лидера разработки содержания Программ для Западной телевизионной сети.”
Meaning	“I moved to Dubai as the content development leader for a Western television network.”
Reference	“Tôi chuyển đến Dubai với vai trò là người chịu trách nhiệm về nội dung cho một đài TV của phương Tây.”
Baseline model	“Tôi chuyển sang <unk> để nghiên cứu về các phần mềm ở <unk>.”
Proposed model	“Tôi chuyển tới gần Dubai, một nhà lãnh đạo những chương trình xây dựng chương trình tại Bờ Tây.”

TABLE 10: Fifth case study with a rare compound.

Source	“Лти места наиболее боГаты с точки зрения биоразнообразия и наиболее важны с точки зрения функционирования экосистемы.”
Meaning	“These sites are the richest in terms of biodiversity and the most important in terms of ecosystem functioning.”
Reference	“Đó là những nơi giàu nhất trong đa dạng sinh học và là quan trọng nhất từ quan điểm chức năng hệ sinh thái.”
Baseline model	“Những nơi này giàu về sự đa dạng và quan trọng nhất so với cách hệ gen.”
Proposed model	“Những nơi này rất phong phú với sự đa dạng sinh học và quan trọng nhất với phương diện hoạt động của hệ sinh thái.”

## 6. Conclusions

In this study, we have proposed a sublemma-based Transformer model for translation from Russian into Vietnamese. It is a derivation from the model with source-word decomposition and models with subword representation. In the proposed model, a source unit is represented as a combination of a sublemma, its tag, part-of-speech tag, dependency label, and morphology label, while a target unit is a subtoken. Experimental results show that our proposed model surpasses all available models for Russian  $\rightarrow$  Vietnamese translation task. Human judgment on the translation quality of the models has validated the comparison in terms of BLEU score.

Standing on the results of this study, we recommend our sublemma-based Transformer model for translation from a highly inflectional language, such as Russian, Bulgarian, or Czech.

## Data Availability

The datasets used in this study are accessible upon request to the corresponding author Thien Nguyen via e-mail: nguyenchithien@tdtu.edu.vn.

## Conflicts of Interest

The authors declare that there are no conflicts of interest in this paper.

## References

- [1] K. Cho, B. Merriënboer, C. Gulcehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1724–1734, Doha, Qatar, October 2014.
- [2] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, September 2015.
- [3] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proceedings of the International Conference on Machine Learning*, pp. 1243–1252, Ho Chi Minh City, Vietnam, January 2017.
- [4] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the Advances in neural information*

- processing systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.
- [5] S. Garg, S. Peitz, U. Nallasamy, and M. Paulik, “Jointly learning to align and translate with transformer models,” in *Proceedings of the EMNLP-IJCNLP 2019-2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pp. 4453–4462, Hong Kong, China, November 2020.
  - [6] T. Nguyen, H. Nguyen, and P. Tran, “Mixed-level neural machine translation,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8859452, 7 pages, 2020.
  - [7] T. Nguyen, H. Le, and V.-H. Pham, “Source-word decomposition for neural machine translation,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 4795187, 10 pages, 2020.
  - [8] N. Reimers and I. Gurevych, “Making monolingual sentence embeddings multilingual using knowledge distillation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4512–4525, November 2020.
  - [9] S. Ding, A. Renduchintala, and K. Duh, “A call for prudent choice of subword merge operations in neural machine translation,” in *Proceedings of the Machine Translation Summit XVII*, pp. 204–213, Dublin, Ireland, August 2019.
  - [10] Y. Wu and H. Zhao, “Finding better subword segmentation for neural machine translation,” in *Proceedings of the Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp. 53–64, Springer, Changsha, China, October 2018.
  - [11] C. Wang, K. Cho, and J. Gu, “Neural machine translation with byte-level subwords,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9154–9160, New York, NY, USA, February 2020.
  - [12] M. Pinnis, R. Krišlauskas, D. Dekšne, and T. Miks, “Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data,” in *Proceedings of the International Conference on Text, Speech, and Dialogue*, pp. 237–245, Prague, Czech Republic, August 2017.
  - [13] H. Deguchi, M. Utiyama, A. Tamura, T. Ninomiya, and E. Sumita, “Bilingual subword segmentation for neural machine translation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4287–4297, Barcelona, Spain, September 2020.
  - [14] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, Berlin, Germany, August 2016.
  - [15] S. Huet, E. Manishina, and F. Lefèvre, “Factored machine translation systems for Russian-English,” 2013.
  - [16] A. Birch, M. Osborne, and P. Koehn, “CCG supertags in factored statistical machine translation,” in *Proceedings of the second workshop on Statistical Machine Translation*, pp. 9–16, Prague, Czech Republic, June 2007.
  - [17] P. Koehn and H. Hoang, “Factored translation models,” in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 868–876, Prague, Czech Republic, June 2007.
  - [18] Y. Wang, L. Wang, X. Zeng, D. F. Wong, L. S. Chao, and Y. Lu, “Factored statistical machine translation for grammatical error correction,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 83–90, Baltimore, MD, USA, June 2014.
  - [19] R. Sennrich and B. Haddow, “Linguistic input features improve neural machine translation,” 2016, <http://arxiv.org/abs/1606.02892>.
  - [20] T. Kudo and J. Richardson, “SentencePiece: a simple and language independent subword tokenizer and detokenizer for Neural Text Processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018.
  - [21] P. Tran, D. Dinh, and H. T. Nguyen, “A character level based and word level based approach for Chinese-Vietnamese machine translation,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 9821608, 2016.
  - [22] T. Nguyen, H. Nguyen, and P. Tran, “Exploring neural machine translation on the Russian-Vietnamese language pair,” in *Proceedings of the Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pp. 393–400, Sendai, Japan, June 2021.
  - [23] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: a {Python} natural language processing toolkit for many human languages,” 2020, <https://nlp.stanford.edu/pubs/qi2020stanza.pdf>.
  - [24] J. Nivre, M.-C. de Marneffe, F. Ginter et al., “Universal dependencies v1: A multilingual treebank collection,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 1659–1666, Portorož, Slovenia, May 2016.
  - [25] M.-C. De Marneffe, T. Dozat, N. Silveira et al., “Universal Stanford dependencies: A cross-linguistic typology,” *LREC*, vol. 14, pp. 4585–4592, 2014.
  - [26] P. Tran, D. Dinh, and L. H. B. Nguyen, “Word re-segmentation in Chinese-Vietnamese machine translation,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 2, pp. 1–22, 2016.
  - [27] P. Tran, D. Dinh, T. Le, and L. H. B. Nguyen, “Linguistic-relationships-based approach for improving word alignment,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 17, no. 1, pp. 1–16, 2017.
  - [28] T. Nguyen, L. Nguyen, P. Tran, and H. Nguyen, “Improving transformer-based neural machine translation with prior alignments,” *Complexity*, vol. 2021, 2021.
  - [29] G. Klein, Y. Kim, Y. Deng, V. Nguyen, J. Senellart, and A. M. Rush, “OpenNMT: neural machine translation toolkit,” in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pp. 177–184, Boston, MA, USA, March 2018.
  - [30] G. Klein, F. Hernandez, V. Nguyen, and J. Senellart, “The OpenNMT neural machine translation toolkit: 2020 edition,” in *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pp. 102–109, Orlando, FL, USA, October 2020.
  - [31] M. Freitag and Y. Al-Onaizan, “Beam search strategies for neural machine translation,” in *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56–60, Melbourne, Australia, July 2017.
  - [32] R. Müller, S. Kornblith, and G. E. Hinton, “When does label smoothing help?” in *Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 4696–4705, December 2019, <https://proceedings.neurips.cc/paper/2019/hash/f1748d6b0fd9d439f71450117eba2725-Abstract.html>.
  - [33] D. P. Kingma and J. Ba, “Adam: {A} method for stochastic optimization,” in *Proceedings of the 3rd International*

*Conference on Learning Representations, {ICLR} 2015*, San Diego, CA, USA, May 2015, <http://arxiv.org/abs/1412.6980>.

- [34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Stroudsburg, PA, USA, July 2002.
- [35] P. Koehn, H. Hoang, A. Birch, and C. Callison-Burch, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 177–180, Stroudsburg, PA, USA, June 2007.

## Research Article

# COVID-19 Detection Using Deep Convolutional Neural Networks and Binary Differential Algorithm-Based Feature Selection from X-Ray Images

Mohammad Saber Iraji <sup>1,2</sup>, Mohammad-Reza Feizi-Derakhshi <sup>3</sup>, and Jafar Tanha <sup>2</sup>

<sup>1</sup>Department of Computer Engineering and Information Technology, Payame Noor University, Tehran, Iran

<sup>2</sup>Department of Computer Engineering, University of Tabriz, Tabriz, Iran

<sup>3</sup>Computerized Intelligence Systems Laboratory, Department of Computer Engineering, University of Tabriz, Tabriz, Iran

Correspondence should be addressed to Mohammad-Reza Feizi-Derakhshi; [mfeizi@tabrizu.ac.ir](mailto:mfeizi@tabrizu.ac.ir)

Received 4 April 2021; Revised 11 May 2021; Accepted 22 September 2021; Published 5 October 2021

Academic Editor: Adil Mehmood Khan

Copyright © 2021 Mohammad Saber Iraji et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The new COVID-19 is rapidly spreading and has already claimed the lives of numerous people. The virus is highly destructive to the human lungs, and early detection is critical. As a result, this paper presents a hybrid approach based on deep convolutional neural networks that are very effective tools for image classification. The feature vectors were extracted from the images using a deep convolutional neural network, and the binary differential metaheuristic algorithm was used to select the most valuable features. The SVM classifier was then given these optimized features. For the study, a database containing images from three categories, including COVID-19, pneumonia, and a healthy category, included 1092 X-ray samples, was used. The proposed method achieved a 99.43% accuracy, a 99.16% sensitivity, and a 99.57% specificity. Our findings indicate that the proposed method outperformed recent studies on COVID-19 detection using X-ray images.

## 1. Introduction

COVID-19's rapid spread has resulted in the death of numerous people worldwide. Muscle aches, cough, and fever are all symptoms of the virus, which can be detected through clinical trials and radiographic imaging. Medical imaging is critical for disease diagnosis, and disease X-rays and computed tomography (CT) scans can be used in the deep network to aid in the disease's diagnosis.

The process of classifying and diagnosing disease from an image using a neural network is divided into four steps: feature extraction, optimal feature selection, network training, and model performance test. The feature extraction step is divided into two types. In the first type, image processing techniques, algorithms, and filters extract the features. Among the features extracted from the images, the tissue shapes and textures are used to classify patients. In the second type, the original images and their actual output class are fed into the convolution network as input data, and the

features are extracted automatically in the final flattened layer following the network training process and weight adjustment.

Certain features extracted from the deep network may have a detrimental effect on classification accuracy [1]. As a result, effective feature selection methods are critical. There are three distinct types of feature selection methods. The filter method uses features' intrinsic properties and statistical indicators such as the fisher score, information gain, chi-square, and correlation coefficient. The wrapper method employs a learning algorithm that searches the feature space for a subset of features that optimize the classification accuracy. To this end, wrapper approaches employ metaheuristic methods for selecting feature subsets and performing cross-validation. Finally, the hybrid method employs both filter and wrapper methods [2]. Metaheuristic methods outperform other feature selection methods in applications where many features are required.

Classification performance is improved by analyzing extracted features from images and selecting the optimal features [3]. Numerous feature selection (FS) studies have been published in the field of medical imaging, including Robustness-Driven FS (RDFS) for lung CT images [4], Shearlet transform FS from brain MRI images [5], principal component analysis for lung X-ray images [6], genetic algorithm (GA) for lung nodules [7], bat algorithm (BA) versus particle swarm optimization (PSO) in lung X-ray images, and the flower pollination algorithm (FPA) from lung images [8].

The studies above propose that machine vision combined with metaheuristic algorithms can classify patients using lung images. On the other hand, existing diagnostic methods for the COVID-19 virus using X-ray images require a large amount of memory, ample time, and a large number of features. As a result, an intelligent system appears necessary to assist doctors and treatment staff in accurately and rapidly classifying COVID-19 patients in reducing disease-related mortality. This research aims to develop an efficient procedure utilizing artificial intelligence methods to assist doctors and patients in accurately predicting COVID-19. The research is novel in that it employs a binary differential evolution algorithm to design a deep learning structure based on feature selection for COVID-19 diagnosis. The contributions of the study include the following:

- (1) Using a deep convolutional neural network without a pretrained network to design an intelligent system based on lung X-ray images and extracting features with the least amount of memory required to create and train the network
- (2) Selecting the optimal features of the differential metaheuristic method that improves performance indexes
- (3) Increasing classification accuracy for multiclass problems, including patients with COVID-19, pneumonia, and the healthy group

The study is structured as follows. Section 2 reviews related works. Section 3 presents the proposed methodology and model for COVID-19 detection using deep convolution and binary differential algorithms. Section 4 contains the experimental results, and Section 5 discusses the method and compares this with prior works. Finally, the study concludes.

## 2. Related Works

Hemdan et al. used deep learning models to infer the positive or negative status of COVID-19 and reported that the VGG19 model performed better with an accuracy of 90% on 25 COVID-19 infected and 25 non-COVID-19 images [9]. Toğaçar et al. incorporated 295 COVID-19 images, 98 pneumonia images, and 65 normal images into MobileNet and SqueezeNet [10]. They extracted features from trained Net models and then used the SMO algorithm to select the features, with an overall accuracy of 99.27% reported for the SVM classifier. Zhang et al. investigated an 18-layer ResNet

model for 100 COVID-19 and 1431 pneumonia images and reported an accuracy of 95.18% [11]. Apostolopoulos and Mpesiana pretrained VGG19 on 224 COVID-19, 700 pneumonia, and 504 normal images, where the results demonstrated a 98.75% accuracy [12]. The authors of [13] evaluated the DarkNet with 17 convolutional layers using 127 COVID-19, 500 pneumonia, and 500 normal images and reported an accuracy of 98.08%. In [14], the performance of CNN was improved via preprocessing image algorithms, resulting in a model with 94.5% accuracy.

The authors of [1] developed a COVID-19 classification method based on two datasets that combined a CNN named Inception, a pretrained Imagenet as a feature extractor, the Marine Predators Algorithm as a feature selector, and a KNN as a classifier.

The first dataset contained 200 positive COVID-19 images and 1675 negative images, whereas the second dataset contained 219 positive COVID-19 images and 1341 negative images. Accuracy was reported as 98.7% for dataset 1 and 99.6% for dataset 2. Canayaz validated a COVID-19 diagnosis model by combining VGG19, ResNet, AlexNet, and GoogLeNet with two metaheuristic algorithms titled “binary particle swarm optimization” and “binary gray wolf optimization.” The highest overall classification accuracy of 99.38% after binary gray wolf optimization was used to select features from 1092 X-ray images from the COVID-19, pneumonia, and healthy category records [15].

One of the previous works’ limitations is their reliance on pretrained deep networks, which require a large amount of memory. Additionally, many input features plus a lengthy detection time are further drawbacks of these methods. In this study, a deep learning approach based on feature selection is proposed via the binary differential evolution algorithm to overcome these limitations and improve COVID-19 detection.

## 3. Methodology and Model

Figure 1 depicts the proposed model. Initially, the convolutional neural network is fed with lung images. After training the network, features are extracted from suboptimal images. The heuristic method is then used to extract the optimal features. As a result, the three classes of COVID-19, pneumonia, and healthy are classified with higher accuracy.

*3.1. Deep Convolution.* Convolutional neural networks are used in machine learning as a feature extractor and classification method. The input to a convolutional network is the original data, such as images. The network extracts the features automatically using the convolution function. After learning, rather than manually extracting the feature, the matrixes serve as filters that slide over the main input image, and the convolution operation is carried out via equation (1). Finally, after training and mapping the input images to the output labels, several convolution layers extract the features [16].



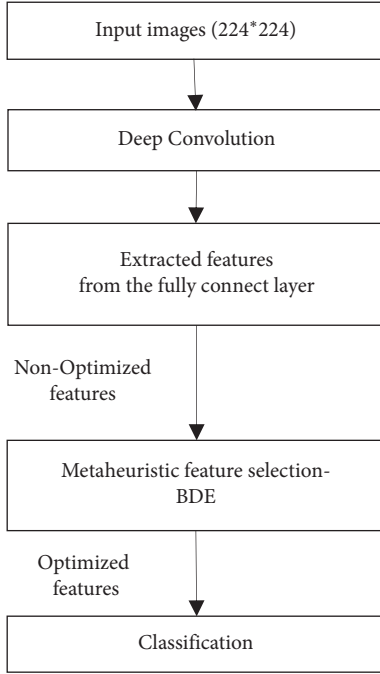


FIGURE 1: The proposed model for COVID-19.

$$(IMG * C)_{ij} = \sum_{p=0}^{c1-1} \sum_{q=0}^{c2-1} \sum_{c=1}^{tc} C_{p,q,c} \cdot IMG_{i+p,j+q,c} + bs, \quad (1)$$

where  $IMG$  denotes the input image with height =  $H$ , width =  $W$  dimensions, and  $tc$  is the number of image channels,  $C$  is the filter matrix with  $c1 * c2$  dimensions, and  $bs$  is a bias value for each filter  $C$ ,  $i = 0, \dots, H$  and  $j = 0, \dots, W$ .

Following convolution, the unwanted values are removed using the ReLU layer, and the input is then reduced using the pooling layer. The effective input vector is then passed to the fully connected layer, which functions similarly to the MLP. In the final section of the deep convolution layers, Softmax [17], classification layers perform classification using ADAM (adaptive moment optimizer) [18]; the lost function is shown in the following equation:

$$L(w, b) = -\frac{1}{M} \sum_{m=1}^M [y_m \log \hat{y}_m + (1 - y_m) \log (1 - \hat{y}_m)] + \Gamma \times \sum_{r=1}^M \|w^r\|_2, \quad (2)$$

where  $M$  denotes the sample size,  $y_m$  denotes the actual class for the  $m$ th sample,  $\hat{y}_m$  denotes the predicted output class for the  $m$ th input data, and  $\Gamma$  denotes the regularization coefficient.

ADAM is a gradient-based optimization algorithm that uses the exponential moving average of the gradient and the square of the gradient to update the neural network weights and solve deep network issues effectively. The deep neural network comprises numerous layers, each with its own set of

learning parameters, namely, weights and biases. Applying the optimal feature selection algorithm to the ADAM optimizer increases the optimization's speed and accuracy.

**3.2. Binary Differential Evolution.** Differential evolution (DE) [19] is a heuristic evolutionary method for minimizing the continuous problem. The concept of binary differential evolution (BDE) [20] is extended to address issues of feature selection. It is composed of three distinct builders, including mutation, crossover, and selection. Initially, dimensions  $D$  are used to generate the initial population, where  $D$  is the number of features to optimize. For the mutation operation, three random vectors  $p_{u1}$ ,  $p_{u2}$ , and  $p_{u3}$  are selected for vector  $p_k$  such that  $u1 \neq u2 \neq u3 \neq k$ , where  $k$  is a population vector arrangement.

If the  $d$ th dimensions of vectors  $p_{u1}$  and  $p_{u2}$  are equal, the  $d$ th feature of the difference vector (Equation (3)) is zero; otherwise, it has the same value as vector  $p_{u1}$ :

$$\text{difference vector}_k^d = \begin{cases} 0, & p_{u1}^d = p_{u2}^d, \\ p_{u1}^d, & \text{other.} \end{cases} \quad (3)$$

Afterward, the mutation and crossover operations are executed, as shown in the following equations:

$$\text{mute vector}_k^d = \begin{cases} 1, & \text{if difference vector}_k^d = 1, \\ p_{u3}^d, & \text{other,} \end{cases} \quad (4)$$

$$W_k^d = \begin{cases} \text{mute vector}_k^d, & \text{if } \gamma \leq CR \|d = d_{\text{random}}, \\ p_k^d, & \text{other,} \end{cases} \quad (5)$$

where  $W$  represents the try vector,  $CR \in (0, 1)$  represents the crossover amount, and  $\gamma \in (0, 1)$  represents a random number. In the selection procedure, if the fitness value of the try vector  $W_k$  is greater than that of the current vector  $p_k$ , it will be replaced. Otherwise, the current vector  $p_k$  is stored for the next generation.

## 4. Experimental Results

**4.1. Description of Data.** Canayaz developed a COVID-19 X-ray data set that included three subgroups of patients, including those with COVID-19, those with pneumonia, and those who were healthy [15]. By combining data from this database, a total of 364 images for each of the three categories were obtained as a balanced dataset [21–23]. The total number of images is equal to the number of classes multiplied by the number of class instances =  $(3 * 394)1092$ , with a  $224 \times 224$  dimension. The same data are used in this study to predict COVID-19 disease using a convolutional neural network and to select optimal features using the binary differential metaheuristic algorithm. Figure 2 illustrates a representative sample of three output classifications: COVID-19, pneumonia, and healthy.

**4.2. Performance Evaluation.** The proposed model was run in MATLAB version 9.1.0.441655 (R2018b) on a laptop computer equipped with a 1.8 GHz processor and 4

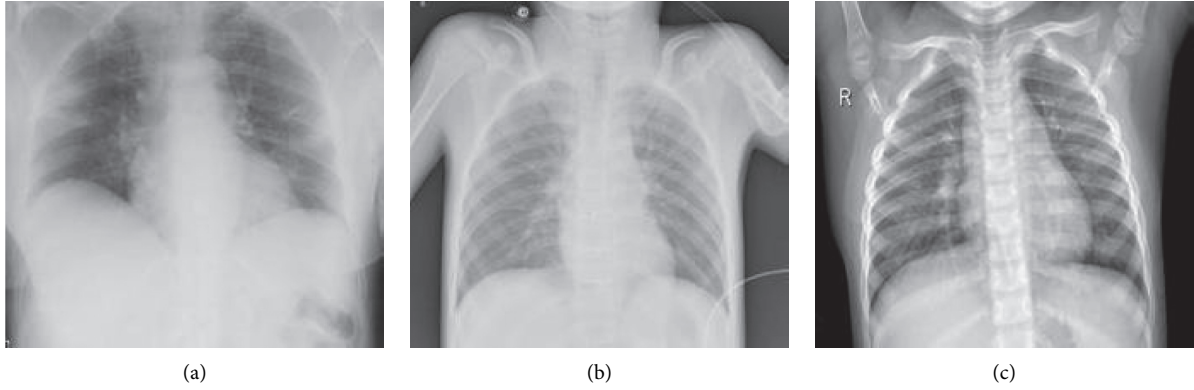


FIGURE 2: Chest X-ray images of the different conditions: (a) COVID-19; (b) pneumonia; (c) healthy.

Gigabytes of RAM. After training in the application phase, the proposed method took an average of 29 seconds per patient, which can be reduced by improving the hardware technology used. The COVID-19 prediction model was evaluated using the accuracy, sensitivity, specificity, geometric mean, and area under the curve (AUC-ROC) [24, 25] performance metrics (Equations (6)–(9)), where accuracy refers to the correctness of the classification. The proportion of correctly distinguished negative cases is referred to as “specificity,” while the proportion of correctly distinguished positive cases is referred to as “sensitivity.” The geometric mean is the second root of the sensitivity and specificity products. Higher values of the area under the curve (AUC) within the receiver operating characteristics indicate improved classification performance.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (6)$$

where

TP means true positives which is the number of samples that are correctly labeled as positive

FP means false positives which is the number of samples that are wrongly labeled as positive

TN means true negatives which is the number of samples that are correctly labeled as negative

FN means false negatives which is the number of samples that are wrongly labeled as negative

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \quad (7)$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{FP} + \text{TN})}, \quad (8)$$

$$\text{Geometric mean} = \sqrt{\text{sensitivity} * \text{specificity}}. \quad (9)$$

**4.3. Model Parameters.** Figure 3 illustrates the network structure of a deep convolutional neural network. Firstly, the image input layer dimensions were  $224 * 224$ , and the convolution operator used eight 3-by-3 filters. After

processing the first block of the network layers, i.e., image input, convolution, Batch Normalize, ReLU, max-pooling layers, fully connected layer 400, ReLU, and Drop out, the local features were automatically extracted. Finally, the second network block categorized the input images into three output classes by utilizing three fully connected layers, softmax, and classification. After 200 epochs, the validation accuracy was 97.25% when using the ADAM optimizer, and the minibatch size was 64 (Figure 4). Due to the neural network’s regularization and barricade overfitting, batch normalization and dropout were used.

Convolutional networks are used to transform data into feature vectors. Given that some network features may degrade the model’s performance [1], after extracting 400 features in the first fully connected layer, the binary differential metaheuristic algorithm was used to select the optimal feature subset and eliminate unnecessary features. The binary differential algorithm’s parameters were population = 20, iteration = 100 (Figure 5), and a crossover rate of 1. The amount  $(1 - (\text{geometric mean}))$  of the SVM classifier [26] was regarded as the population’s fitness values (Figure 5). Following the binary differential algorithm’s execution, 340 optimal features were selected.

**4.4. Performance Comparison.** The conventional validation (CV) method, which employs random sampling, is one of the training and testing protocols used to determine the model’s accuracy and validate the estimation results. According to the CV method, 70% of data were used for training, 15% for validation, and 15% for testing [27, 28]. The proposed method was applied to the data, and 100 runs were performed to prevent overfitting [29]. The optimally selected features from the differential algorithm and the initial extracted features from the deep convolutional neural network were entered into the SVM classifier.

The confusion matrix for the SVM classifier’s original and optimized features is demonstrated in Table 1 using training, testing, validation, and total data.

Table 1 depicts the confusion matrix using training, testing, validation, and total data for the SVM classifier’s original features and optimized features. The TP, TN, FP, FN, accuracy, sensitivity, specificity, geometric mean, and

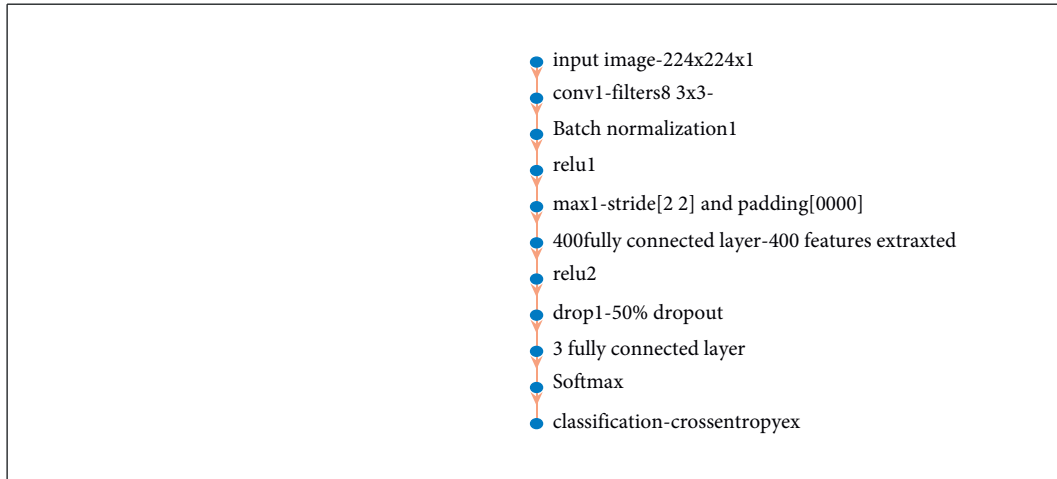


FIGURE 3: The proposed model layers' structure.

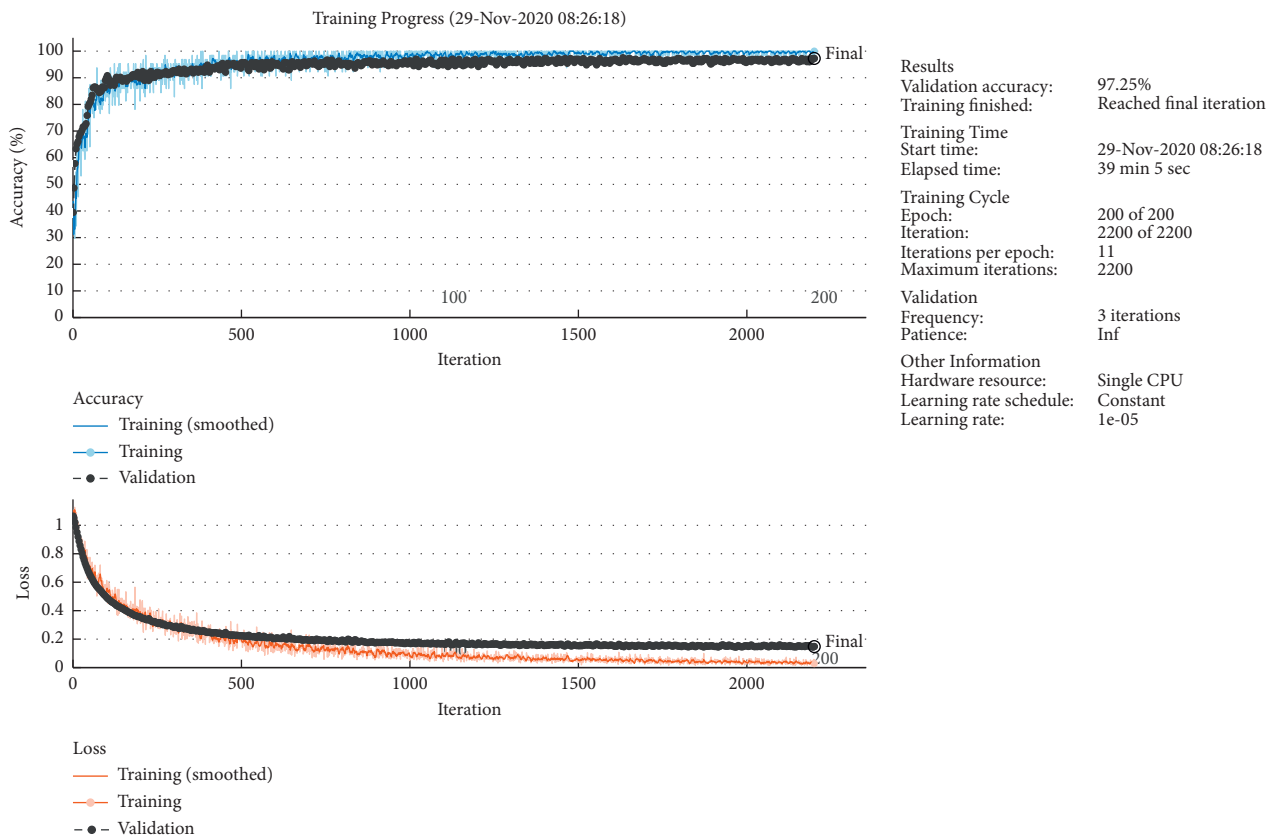


FIGURE 4: Accuracy and loss measurements for convolutional neural networks used for training and validation data.

AUC metrics for each of the three output classes plus each type of testing, training, validation, and total data were computed (Table 2).

### 5. Discussion

Predicting a disease can be accomplished by combining images with a deep neural network, where a deep neural network can be used as a feature extractor. The large size and

volume of images applied to the deep neural network result in numerous feature formations that increase the training and decision times of the predictive model.

The proposed model faced several design challenges, including collecting and improving lung images and the deep network architecture, in terms of the structure, number, and type of layers, plus the metaheuristic algorithm, the initial population, and the metaheuristic algorithm's objective function type. The presence of inefficient features

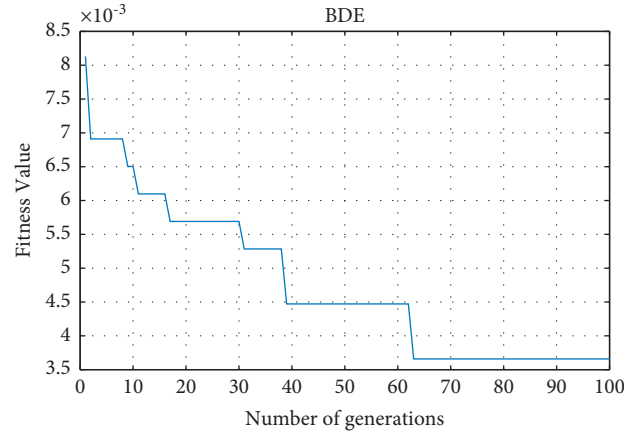


FIGURE 5: Fitness curve for the binary differential algorithm.

TABLE 1: Confusion matrix average after 100 runs with 3 classes using the (a) optimized features and (b) original features based on the training, test, validation, and total data.

(a) Optimized features			
Optimized features test		Predicted	
		COVID-19	Normal
Actual	COVID-19	51/75	0
	Normal	0/15	54/9
	Pneumonia	0/4	0/55
Pneumonia 55/95			
Optimized features validation		Predicted	
		COVID-19	Normal
Actual	COVID-19	55/05	0/05
	Normal	0/05	54/4
	Pneumonia	0/4	1
Pneumonia 52/3			
Optimized features training		Predicted	
		COVID-19	Normal
Actual	COVID-19	256/5	0
	Normal	0	254/1
	Pneumonia	0	0
Pneumonia 253/4			
Optimized features total		Predicted	
		COVID-19	Normal
Actual	COVID-19	363/3	0/05
	Normal	0/2	363/4
	Pneumonia	0/8	1/55
Pneumonia 361/65			
(b) Original features			
Original features test		Predicted	
		COVID-19	Normal
Actual	COVID-19	51/65	0
	Normal	0/25	54/65
	Pneumonia	0/5	0/95
Pneumonia 55/45			
Original features validation		Predicted	
		COVID-19	Normal
Actual	COVID-19	54/85	0
	Normal	0/15	54/2
	Pneumonia	0/55	1/2
Pneumonia 51/95			
Original features training		Predicted	
		COVID-19	Normal
Actual	COVID-19	256/45	0
	Normal	0	254/1
	Pneumonia	0	0
Pneumonia 253/4			
Original features total		Predicted	
		COVID-19	Normal
Actual	COVID-19	362/95	0
	Normal	0/4	362/95
	Pneumonia	1/05	2/15
Pneumonia 360/8			

TABLE 2: Comparison of indicators (TP, TN, FP, FN, accuracy, the area under curve, sensitivity, specificity, and geometric mean) for any output class based on (a) optimized features and (b) original features.

(a) Optimized features			
Optimized features-total	COVID-19	Normal	Pneumonia
TP	363/30	363/40	361/65
TN	727/00	726/40	726/95
FP	1/00	1/60	1/05
FN	0/70	0/60	2/35
Accuracy	99/84	99/80	99/69
Sensitivity	99/81	99/84	99/35
Specificity	99/86	99/78	99/86
Geometric mean	99/84	99/81	99/60
Area under curve	0/9984	0/9981	0/9961
Optimized features-training	COVID-19	Normal	Pneumonia
TP	256/50	254/10	253/40
TN	507/50	509/90	510/60
FP	0/00	0/00	0/00
FN	0/00	0/00	0/00
Accuracy	100/00	100/00	100/00
Sensitivity	100/00	100/00	100/00
Specificity	100/00	100/00	100/00
Geometric mean	100/00	100/00	100/00
Area under curve	1/0000	1/0000	1/0000
Optimized features-test	COVID-19	Normal	Pneumonia
TP	51/75	54/90	55/95
TN	111/50	108/30	106/80
FP	0/55	0/55	0/30
FN	0/20	0/25	0/95
Accuracy	<b>99/54</b>	<b>99/51</b>	<b>99/24</b>
Sensitivity	99/62	99/55	98/33
Specificity	99/51	99/49	99/72
Geometric mean	99/56	99/52	99/02
Area under curve	0/9956	0/9952	0/9903
Optimized features-validation	COVID-19	Normal	Pneumonia
TP	55/05	54/40	52/30
TN	108/00	108/20	109/55
FP	0/45	1/05	0/75
FN	0/50	0/35	1/40
Accuracy	99/42	99/15	98/69
Sensitivity	99/10	99/36	97/39
Specificity	99/59	99/04	99/32
Geometric mean	99/34	99/20	98/35
Area under curve	0/9934	0/9920	0/9840
(b) Original features			
Original features-total	COVID-19	Normal	Pneumonia
TP	362/95	362/95	360/80
TN	726/55	725/85	726/30
FP	1/45	2/15	1/70
FN	1/05	1/05	3/20
Accuracy	99/77	99/71	99/55
Sensitivity	99/71	99/71	99/12
Specificity	99/80	99/70	99/77
Geometric mean	99/76	99/71	99/44
Area under curve	0/9976	0/9971	0/9944
Original features-training	COVID-19	Normal	Pneumonia
TP	256/45	254/10	253/40
TN	507/50	509/90	510/55
FP	0/00	0/00	0/05
FN	0/05	0/00	0/00
Accuracy	99/99	100/00	99/99
Sensitivity	99/98	100/00	100/00
Specificity	100/00	100/00	99/99
Geometric mean	99/99	100/00	100/00
Area under curve	0/9999	1/0000	1/0000

TABLE 2: Continued.

(a) Optimized features			
Optimized features-total	COVID-19	Normal	Pneumonia
Original features-test	COVID-19	Normal	Pneumonia
TP	51/65	54/65	55/45
TN	111/30	107/90	106/55
FP	0/75	0/95	0/55
FN	0/30	0/50	1/45
Accuracy	99/36	99/12	98/78
Sensitivity	99/42	99/09	97/45
Specificity	99/33	99/13	99/49
Geometric mean	99/38	99/11	98/46
Area under curve	0/9937	0/9910	0/9847
Original features-valid	COVID-19	Normal	Pneumonia
TP	54/85	54/20	51/95
TN	107/75	108/05	109/20
FP	0/70	1/20	1/10
FN	0/70	0/55	1/75
Accuracy	99/15	98/93	98/26
Sensitivity	98/74	99/00	96/74
Specificity	99/35	98/90	99/00
Geometric mean	99/05	98/95	97/87
Area under curve	0/9904	0/9895	0/9789

TABLE 3: Average of the confusion matrix components after 100 runs using original and optimized features.

Method	TP	TN	FP	FN	Accuracy	Sensitivity	Specificity	Geometric mean	Area under curve	RMSE
Original features via deep convolution										
Training	254/65	509/32	0/02	0/02	1/0000	0/9999	1/0000	1/0000	0/9999	0/0036
Testing	53/92	108/58	0/75	0/75	0/9909	0/9866	0/9931	0/9898	0/9898	0/1533
Validation	53/67	108/33	1/00	1/00	0/9878	0/9816	0/9909	0/9862	0/9863	0/1905
Total	362/23	726/23	1/77	1/77	0/9968	0/9951	0/9976	0/9964	0/9964	1/1543
Optimized features via binary differential										
Training	254/67	509/33	0/00	0/00	1/0000	1/0000	1/0000	1/0000	1/0000	0/0000
Testing	54/20	108/87	0/47	0/47	0/9943	0/9916	0/9957	0/9937	0/9937	0/1133
Validation	53/92	108/58	0/75	0/75	0/9909	0/9862	0/9931	0/9896	0/9898	0/1592
Total	362/78	726/78	1/22	1/22	0/9978	0/9967	0/9983	0/9975	0/9975	1/1543

extracted from the deep network may reduce the predictive model's accuracy and efficiency; thus, using the meta-heuristic method to select the optimal features improved the model's memory, time, and accuracy.

According to Table 3, the proposed model achieved an accuracy of 99.43%, a sensitivity of 99.16%, a specificity of 99.57%, a geometric mean of 99.37%, an AUC of 0.99, and a root mean square error (RMSE) of 0.1133 using features extracted from the X-ray image via the CNN and features optimized using the binary differential metaheuristic algorithm. The accuracy of the classification of the COVID-19 problem was calculated to be 99.43% in this study, and the number of relevant features was 304 (Table 4), whereas, in a previous study [15], these figures were reported to be 99.38% and 448 features, respectively, based on the same data.

Transfer learning models are trained to classify 1,000 different types of object images and must be retrained to classify specific issues such as COVID-19 detection. Although the learning process is prompt in models like ResNet and SqueezeNet, they require preprocessing the input image,

sizing the data set, and setting multiple parameters. The upper layers extract color and edge features, while the deeper layers extract complex features. Process time increases as the number of layers in transfer learning models increases. The trained transfer learning model's feature map and activation layers must be customized for the specific COVID-19 problem, which requires a large amount of memory. After fine-tuning the pretrained model's principal component analysis (PCA), the optimal feature can be selected using heuristic methods, automated encoders, or variance-based selectors. Finally, ensemble methods, such as a combination of SVMs or other classifiers, can be used to predict COVID-19 disease diagnosis accuracy. Using semisupervised self-learning methods may result in acceptable accuracy and reduced labeling time.

In future work, a different feature selection algorithm and the application of additional learners may produce improved results. Along with the images, the parameters derived from clinical trials can create a new model with a novel combination of features for diagnosing the disease and possibly predicting mortality as a result.

TABLE 4: A comparison of the proposed method with prior research.

Research	Method	Number of features	Accuracy	Geometric mean	RAM (gigabyte)	Max computation time (s)
[15]	Binary particle swarm optimization-VGG19	448	99.38	-	16	2500
Proposed method	Binary differential-CNN	308	99.43	99.37	4	2300

## 6. Conclusion

The number of people infected with COVID-19 has risen rapidly. Machine vision techniques and artificial intelligence are critical in diagnosing and treating disease. The purpose of this paper was to propose a method for the “COVID-19” problem via a set of lung images that included three categories of pneumonia, COVID-19, and healthy.

A deep convolutional neural network consisting of 11 layers was applied to extract the features. The binary differential metaheuristic method was used to select relevant features and eliminate unrelated features. Lung X-ray images were classified using an SVM classifier based on these optimal features. This study demonstrated that the accuracy indicator and the number of relevant extracted features outperformed previous methods using the same data. Based on a deep neural network and a metaheuristic feature selection algorithm, the proposed model can be used in various other medical applications.

## Data Availability

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

## Ethical Approval

This paper contains no data or other information derived from studies or experiments involving human or animal subjects.

## Consent

Not applicable.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

Iraji and Feizi-Derakhshi proposed the image analysis algorithm; Iraji implemented the algorithm and analyzed the experimental results; Tanha provided clinical guidance; and Iraji, Feizi-Derakhshi, and Tanha validated the obtained results. The final manuscript was read and approved by all authors.

## Acknowledgments

The authors wish to express their gratitude to all study participants.

## References

- [1] A. T. Sahlol, D. Yousri, A. A. Ewees, M. A. A. Al-Qaness, R. Damasevicius, and M. A. Elaziz, “COVID-19 image classification using deep features and fractional-order marine predators algorithm,” *Scientific Reports*, vol. 10, no. 1, pp. 15364–15415, 2020.
- [2] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, “MIFS-ND: a mutual information-based feature selection method,” *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371–6385, 2014.
- [3] P. Lambin, E. Rios-Velazquez, R. Leijenaar et al., “Radiomics: extracting more information from medical images using advanced feature analysis,” *European Journal of Cancer*, vol. 48, no. 4, pp. 441–446, 2012.
- [4] D. Y. Chong, H. J. Kim, P. Lo et al., “Robustness-driven feature selection in classification of fibrotic interstitial lung disease patterns in computed tomography using 3D texture features,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, pp. 144–157, 2015.
- [5] U. R. Acharya, S. L. Fernandes, J. E. WeiKoh et al., “Automated detection of alzheimer’s disease using brain MRI images—a study with various feature extraction techniques,” *Journal of Medical Systems*, vol. 43, no. 9, Article ID 302, 2019.
- [6] A. Afzali, F. B. Mofrad, and M. Pouladian, “Feature selection for contour-based tuberculosis detection from chest X-ray images,” in *Book Feature Selection for Contour-Based Tuberculosis Detection from Chest X-Ray Images*, pp. 194–198, IEEE, Piscataway, NJ, USA, 2019.
- [7] S. F. Da Silva, M. X. Ribeiro, J. d. E. S. Batista Neto, C. Traina Jr, and A. J. M. Traina, “Improving the ranking quality of medical image retrieval using a genetic feature selection method,” *Decision Support Systems*, vol. 51, no. 4, pp. 810–820, 2011.
- [8] D. S. Johnson, D. L. L. Johnson, P. Elavarasan, and A. Karunanithi, “Feature selection using flower pollination optimization to diagnose lung cancer from CT images,” in *Book Feature Selection Using Flower Pollination Optimization to Diagnose Lung Cancer from CT Images*, pp. 604–620, Springer, Berlin, Germany, 2020.
- [9] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, “COVIDx-net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images,” 2020, <https://arxiv.org/abs/2003.11055>.
- [10] M. Toğaçar, B. Ergen, and Z. Cömert, “COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches,” *Computers in Biology and Medicine*, vol. 121, Article ID 103805, 2020.
- [11] J. Zhang, Y. Xie, Y. Li, C. Shen, and Y. Xia, “COVID-19 screening on chest x-ray images using deep learning based anomaly detection,” 2020, <https://arxiv.org/abs/2003.12338>.
- [12] I. D. Apostolopoulos and T. A. Mpesiana, “COVID-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks,” *Physical and*

- Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, 2020.
- [13] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” *Computers in Biology and Medicine*, vol. 121, Article ID 103792, 2020.
- [14] M. Heidari, S. Mirniaharikandehi, A. Z. Khuzani, G. Danala, Y. Qiu, and B. Zheng, “Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms,” *International Journal of Medical Informatics*, vol. 144, Article ID 104284, 2020.
- [15] M. Canayaz, “MH-COVIDNet: diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images,” *Biomedical Signal Processing and Control*, vol. 64, Article ID 102257, 2020.
- [16] Y. Gao, T. Zhu, and X. Xu, “Bone age assessment based on deep convolution neural network incorporated with segmentation,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 12, pp. 1951–1962, 2020.
- [17] K. Adem, S. Kiliçarslan, and O. Cömert, “Classification and diagnosis of cervical cancer with stacked autoencoder and softmax classification,” *Expert Systems with Applications*, vol. 115, pp. 557–564, 2019.
- [18] U. M. Khaire and R. Dhanalakshmi, “High-dimensional microarray dataset classification using an improved adam optimizer (iAdam),” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 5187–5204, 2020.
- [19] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [20] E. Zorarpacı and S. A. Özel, “A hybrid approach of differential evolution and artificial bee colony for feature selection,” *Expert Systems with Applications*, vol. 62, pp. 91–103, 2016.
- [21] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, “COVID-19 image data collection: prospective predictions are the future,” 2020, <https://arxiv.org/abs/2006.11988>.
- [22] M. E. Chowdhury, T. Rahman, A. Khandakar et al., “Can AI help in screening viral and COVID-19 pneumonia?” 2020, <https://arxiv.org/abs/2003.13145>.
- [23] D. Kermany, K. Zhang, and M. Goldbaum, “Labeled optical coherence tomography (OCT) and Chest X-Ray images for classification,” *Mendeley data*, vol. 2, 2018.
- [24] A. A. Farag, A. Ali, S. Elshazly, and A. A. Farag, “Feature fusion for lung nodule classification,” *International journal of computer assisted radiology and surgery*, vol. 12, no. 10, pp. 1809–1818, 2017.
- [25] Z. Li, J. Jiang, H. Zhou et al., “Development of a deep learning-based image eligibility verification system for detecting and filtering out ineligible fundus images: a multicentre study,” *International Journal of Medical Informatics*, vol. 147, Article ID 104363, 2021.
- [26] L. Xu, X. Wang, L. Bai et al., “Probabilistic SVM classifier ensemble selection based on GMDH-type neural network,” *Pattern Recognition*, vol. 106, Article ID 107373, 2020.
- [27] O. Er, A. C. Tanrikulu, A. Abakay, and F. Temurtas, “An approach based on probabilistic neural network for diagnosis of Mesothelioma’s disease,” *Computers & Electrical Engineering*, vol. 38, no. 1, pp. 75–81, 2012.
- [28] H. Temurtas, N. Yumusak, and F. Temurtas, “A comparative study on diabetes disease diagnosis using neural networks,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8610–8615, 2009.
- [29] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2014.



## Research Article

# A New Classification Method in Ultrasound Images of Benign and Malignant Thyroid Nodules Based on Transfer Learning and Deep Convolutional Neural Network

Weibin Chen <sup>1</sup>, Zhiyang Gu <sup>2</sup>, Zhimin Liu,<sup>3</sup> Yaoyao Fu,<sup>4</sup> Zhipeng Ye,<sup>4</sup> Xin Zhang <sup>4</sup>, and Lei Xiao <sup>1</sup>

<sup>1</sup>College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China

<sup>2</sup>Department of Electrical and Electronic Engineering, Wenzhou Polytechnic, Wenzhou, China

<sup>3</sup>Wenzhou Hospital of Traditional Chinese Medicine, Wenzhou, China

<sup>4</sup>School of Ophthalmology & Optometry, School of Biomedical Engineering, Wenzhou Medical University, Wenzhou, China

Correspondence should be addressed to Zhiyang Gu; [guzy@mail.nwpu.edu.cn](mailto:guzy@mail.nwpu.edu.cn) and Lei Xiao; [xiaolei@wzu.edu.cn](mailto:xiaolei@wzu.edu.cn)

Received 11 April 2021; Revised 10 August 2021; Accepted 21 August 2021; Published 13 September 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Weibin Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Thyroid nodule is a clinical disorder with a high incidence rate, with large number of cases being detected every year globally. Early analysis of a benign or malignant thyroid nodule using ultrasound imaging is of great importance in the diagnosis of thyroid cancer. Although the *b*-mode ultrasound can be used to find the presence of a nodule in the thyroid, there is no existing method for an accurate and automatic diagnosis of the ultrasound image. In this pursuit, the present study envisaged the development of an ultrasound diagnosis method for the accurate and efficient identification of thyroid nodules, based on transfer learning and deep convolutional neural network. Initially, the Total Variation- (TV-) based self-adaptive image restoration method was adopted to preprocess the thyroid ultrasound image and remove the boarder and marks. With data augmentation as a training set, transfer learning with the trained GoogLeNet convolutional neural network was performed to extract image features. Finally, joint training and secondary transfer learning were performed to improve the classification accuracy, based on the thyroid images from open source data sets and the thyroid images collected from local hospitals. The GoogLeNet model was established for the experiments on thyroid ultrasound image data sets. Compared with the network established with LeNet5, VGG16, GoogLeNet, and GoogLeNet (Improved), the results showed that using GoogLeNet (Improved) model enhanced the accuracy for the nodule classification. The joint training of different data sets and the secondary transfer learning further improved its accuracy. The results of experiments on the medical image data sets of various types of diseased and normal thyroids showed that the accuracy rate of classification and diagnosis of this method was 96.04%, with a significant clinical application value.

## 1. Introduction

In recent years, the incidence of thyroid cancer has continued to rise. As a malignant tumor of the head and neck, it continues to threaten people's health [1]. It is reported that, in the United States, thyroid carcinoma is expected to be the third disease threat among women, with approximately 37 cases per 100,000 people [2]. The thyroid nodule is a symptom of thyroid-related disease. The nodule may be caused by the growth of thyroid cells or thyroid cyst. The thyroid tissues around the scattered lesion of thyroid nodule

can be clearly distinguished through images [3, 4]. If the benign and malignant nodules can be judged earlier, even malignant nodules can be cured. In addition, the accurate distinguishing methods can provide an effective basis for the proper subsequent clinical treatments. Besides, the accurate diagnosis earlier can also reduce the medical risk to be suffered by patients and a large amount of health care costs caused by acupuncture detection.

Currently, there are two major methods for examining the nature of thyroid nodules: ultrasound image analysis and computer tomography imaging analysis. Between

them, ultrasound imaging is cheap and common in hospital. This is why the ultrasound image analysis is more common. However, in the ultrasound image, the malignant thyroid nodule with prominent histopathological components and blurred boundaries usually adhere to other tissues, difficult to distinguish the morphology. This requires an efficient image classification method to improve accuracy and reduce the misdiagnosis rate. In the past studies, radiographers have summarized thyroid nodules ultrasonographic features images according to their characteristics, which function as signs of cancer. However, thyroid nodules diagnosis relying on these characteristics is time-consuming and poorly robust. To this end, the accurate computer aided diagnosis system based on ultrasound images is still to be judged by doctors. A fully automatic computer aided diagnosis system consists of image preprocessing, such as denoising, ROI extraction, and classification. Nowadays, most researches mainly focus on image denoising and ROI extraction. At present, it is still difficult to judge by ultrasonic images alone. The low quality and noise pollution of the ultrasound image makes it extremely challenging to classify it. Tsantis et al. [5] proposed an SVM classifier to divide thyroid nodules into high-risk and low-risk malignant tumors. Ma et al. [6] present a noninvasive and automatic approach for differentiating benign and malignant thyroid nodules based on support vector machines (SVM). Acharya et al. [7] proposed a wavelet transform filter to classify it. Shukla et al. [8] utilize artificial neural network dealing with thyroid disease. Prochazka et al. [9] proposed a dual threshold binary decomposition method to classify it.

Rapid progress in the automatic classification of medical image data is also made by this method. Thandiackal et al. [10] identified skin lesions through some pretrained classical classification networks. Convolutional Neural Network (CNN) models are a type of deep learning architecture introduced to achieve the correct classification of breast cancer [11]. It proposed an in-depth model that uses limited chest CT data to distinguish malignant nodules and benign nodules [12–15]. It proposed a classification algorithm for thyroid nodule ultrasound images based on DCNN [16]. Nevertheless, these methods are defective in the following aspects at present:

Needless to say, transfer learning has played an important role in ultrasound imaging diagnosis of thyroid cancer. However, few-shot learning is the challenging problem of making predictions based on a limited number of samples [17–20]. Also, data labelling is a task that requires a lot of manual work [21–23]. Finally, the inappropriate model and imbalanced training data are difficult to get better classification accuracy [24–30].

Therefore, in view of the above problems, this paper discusses and does the following work:

In response to the abovementioned few-shot learning, in this paper, a TV model is introduced for the automated preprocessing of original data collected by various institutions. Some image marks made by doctors also need to be removed. The original image is then expanded by data

augmentation for the purpose of supplementing inadequate training samples. Also, in response to what is mentioned above to select suitable learning transfer model, the GoogLeNet model was established for the experiments on thyroid ultrasound image data sets. The results showed that the model enhanced the accuracy for the nodule classification. Finally, in response to the above imbalanced training data problem, this paper puts forward the secondary transfer learning conducted on public thyroid database and the actual data sets collected by hospitals, which improves the classification accuracy.

The structure of this paper is presented as follows. In Section 1, the writing motivation is given, and the relevant literature has been examined. Section 2 provides the traditional CNN structure and describes the Tv-Based Image Restoration. Section 3 describes the network structure of the proposed methods based on GoogLeNet. Section 4 shows the experimental results, including the application of the proposed method for diagnosis of thyroid cancer. The research results are summarized in Section 5.

## 2. Related Work

*2.1. Abbreviations and Acronyms.* This kind of network structure is usually called CNN, local connection, weight sharing, and other characteristics of feed-forward neural network [31]. It then inputs the extracted features into the fully connected network; thereby, the parameters are to be optimized. In their research, Moon et al. used ultrasound images for cancer diagnosis. The difference from the previous method is that they use a variety of data sets and combine different CNN algorithms for fusion diagnosis. It was found that the accuracy rates of different data sets were 91.1% and 94.62%, respectively [32]. Kim et al. used deep learning methods for intelligent diagnosis of breast ultrasound images. By calculating different performance standards, the AUC value was 89% [33]. There are also some methods that use a three-dimensional convolutional neural network structure. Through experiments, different performance standards have been found, and the accuracy rate can reach 96.7%.

It is an effective method to extract image features. The image input to the convolutional layer. In this layer, it can perform feature extraction tasks. Each feature can be extracted from each feature map through the convolutional layer. The weight is updated through continuous backward propagation during the training. The computing formula of the convolution layer is

$$X_j = f \left( \sum_i x_i k_{ij} + b \right), \quad (1)$$

where  $X_j$  is the output neuron cell,  $x_i$  is the input signal of each network cell,  $f$  is the activation function,  $K_{ij}$  is the convolution kernel, and  $b$  is the offset.

After the features are extracted through the convolution layer, the output feature map reaches the pooling layer for feature selection and information screening. The pooling formula is shown in the following equation:

$$X_j = f(\beta_j L(x_j) + b), \quad (2)$$

where  $\beta_j$  is the weight coefficient and  $L$  is the sampling function [34–36].

After pooling, the data is input to the fully connected layer that is equivalent to the traditional forward neural propagation. The connected end of convolutional neural only transmits signals to other fully connected layers. The traditional CNN Structure is shown in Figure 1.

In the traditional CNN structure, the forward propagation is adopted to build the network structure, and the backward propagation is adopted to train the network parameters. The loss function, learning rate, and moving average are used for network optimization. Regularization and cross entropy are the loss functions in CNN. The cross entropy formula is shown as

$$H = - \sum y * \log y', \quad (3)$$

where  $y$  is the standard answer and  $y'$  is the predicted value. The exponential decay learning rate is adopted, i.e., the magnitude of each parameter update. The formula of parameter update is given by

$$w_{n+1} = w_n - r * f', \quad (4)$$

where  $r$  is the learning rate and  $f'$  is the gradient of the loss function.

**2.2. TV-Based Image Restoration.** The data sets collected for this experiment were few and needed to be augmented. In the present study, the data set was augmented only by rotation and translation.

The current data contained manual marks, as shown in Figures 2(a) and 2(c). Manual marks mainly refer to the marks made by the professionals on the lesion area in the ultrasound image, which destroy the part of the texture and affect the accuracy and integrity of the image of the area to be analyzed. This also impacts the subsequent training. Therefore, restoring the image was essential. In 2002, Shen et al. [37] extended the TV model to image inpainting and proposed an image inpainting method based on the TV model. The Total Variation- (TV-) based self-adaptive image restoration was adopted for images to estimate the value after pixel restoration:

$$G_O^{(n)} = \sum_{p \in A} H_{Op}^{(n-1)} G_p^{(n-1)} + H_{OO}^{(n-1)} G_O^0, \quad (5)$$

where  $G_O$  represents the pixel of the current point  $O$  to be restored,  $G_p$  represents the pixel of neighboring points of the current point  $O$  at four directions,  $H_{Op}$  ( $P \in A, A = \{N, S, W, E\}$ ), and  $H_{OO}$  is the weight coefficient, which was mainly determined by  $W_p$ . It is defined in

$$W_p = \frac{1}{\sqrt{a^2 + |\nabla g_p|^2}}, \quad p \in A,$$

$$H_{Op} = \frac{W_p}{\sum W_p + \lambda(O)}, \quad (6)$$

$$H_{OO} = \frac{\lambda(O)}{\sum W_p + \lambda(O)},$$

where  $\nabla g_p$  is the divergence;

$$\nabla g_e \approx \frac{1}{h} \sqrt{(G_W - G_O)^2 + \left[ \frac{(G_{NW} - G_{SW})}{2} \right]^2}, \quad (7)$$

where  $G_W$ ,  $G_{NW}$ , and  $G_{SW}$  are the pixels of the left neighboring point, the upper left neighboring point, and the lower left neighboring point of the current pixel;  $\lambda(O)$  is the parameter of  $\lambda$  at point  $O$ .

Finally, as shown in Figure 2, the image was well restored to an extent that its texture was similar to the surrounding texture. The same method was applied to restore the pixel of Figures 2(a) and 2(c), from which Figures 2(b) and 2(d) were obtained.

### 3. Proposed Methods

**3.1. Proposed Network Structure.** The CNN model of GoogLeNet was established to realize the diagnosis of thyroid classification. The process is shown in Figure 3. Initially, the TV-based preprocessing was performed for the thyroid nodule image. Subsequently, the training of CNN model was conducted to extract the features of images of various sizes. Thereafter, the transfer learning was implemented based on the open source database and the database actually collected. The features were integrated, and the dual-softmax assisted forward propagation was conducted. In the end, a softmax classifier was adopted to classify features. The diagnosis of thyroid classification was thus completed.

**3.2. GoogLeNet CNN Structure.** GoogLeNet adopts the structure of the inception proposed in the Going Deeper with Convolutions [38]. Generally, a CNN structure just simply augments the network, with two disadvantages, namely, overfitting and increase in the computation amount. Generally, the network depth and width can be increased by reducing the parameters, while the reduction of parameters turns the full connection to a sparse connection. For the dense matrix optimization mode, the computation amount does not have a qualitative improvement with this kind of change. The inception structure has a sparse structure and high computing performance. The inception structure is shown in Figure 4.

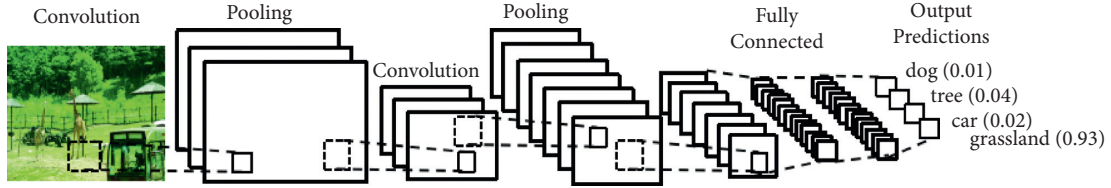


FIGURE 1: Traditional CNN structure.

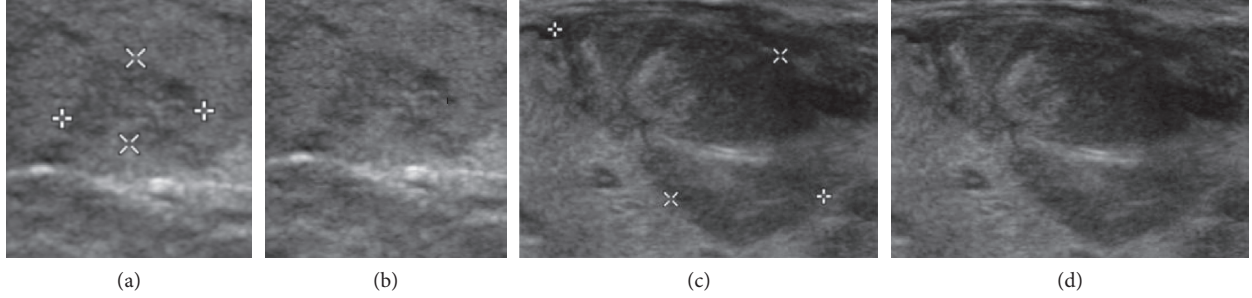


FIGURE 2: TV-based image restoration.

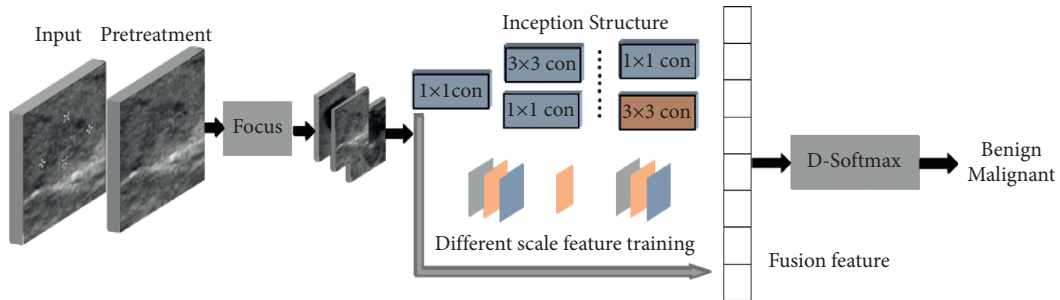


FIGURE 3: Proposed network model.

The use of various scale convolution kernels can get various sizes receptive fields. The final stitching refers to the integration of various scales. Different kernel sizes were set for alignment, such as  $1 \times 1$ ,  $3 \times 3$ . Also, the convolution stride = 1 and the pad = 0, 1, 2, respectively, which was directly stitched together later. However, as the use of  $5 \times 5$  convolution kernel still generated a large amount of computation, hence, the  $1 \times 1$  convolution kernel was utilized to reduce the dimension. The specifically improved inception structure is shown in Figure 5.

**3.3. Improved GoogLeNet Structure.** The GoogLeNet network model is stacked based on the Inception module. Being a network with a relatively large given depth, there is a problem with the backward propagation of effective communication gradient through all layers. For this task, the performance of the shallower network shows that the features generated by the intermediate layer of the network should be very discernible. The discriminative ability of classifiers at low stages can be expected to add complex classifiers. It is considered as a method that overcomes the problem of vanishing gradient. It can adopt the forms of small CNN that are placed above the output of the inception module. These auxiliary networks are discarded in

case of inference. The subsequent control experiment results show that the influence of the complex networks is almost the same. One of them is adequate to achieve the same effect.

Dropout determined what percentage of fully connected nodes was shut off for a training cycle. Dropout improved the model generalizability by preventing nodes from overlearning the training data. The average pooling was finally adopted for the network to replace the fully connected layer. Furthermore, in order to prevent the gradient from vanishing, the network was provided with two additional softmax for the forward propagation gradient. The structure of inception is shown in Figure 1. The computation was performed after the number of channels was reduced through the  $1 \times 1$  convolution to aggregate the information, effectively making use of computing power. The integration of multidimensional features by combining the convolution and pooling of different scales also contributed to a better effect in terms of recognition and classification. By changing the computing power from being deep to being wide, it avoided the problem of dispersion of the training gradient. The global average pooling adopted by GoogleNet solved the typical problem of the complicated and weakly generalized parameters of traditional CNN network.

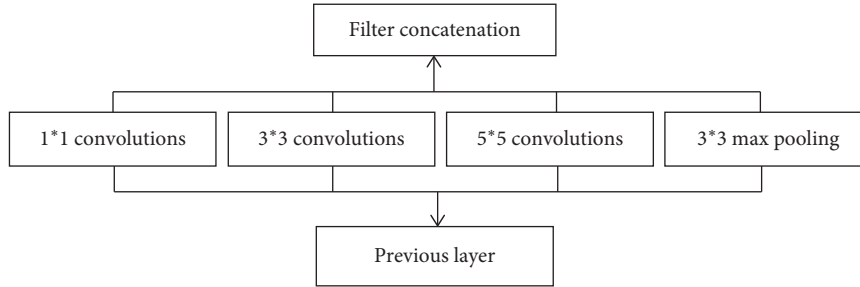


FIGURE 4: Inception module.

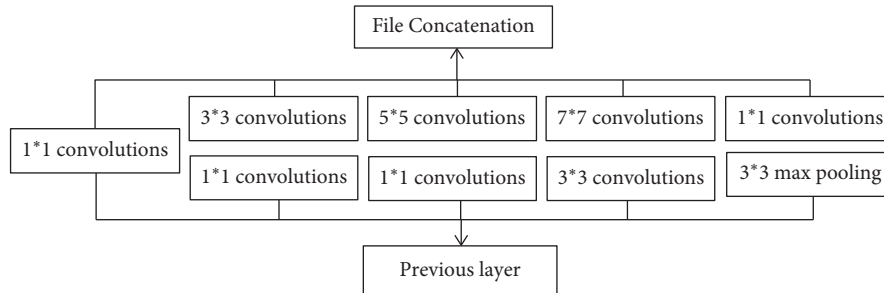


FIGURE 5: Improved inception module.

## 4. Experiments

**4.1. Selected Data Sets and Evaluation Indicators.** We verified through a lot of experiments that the accuracy for predicting the morphological classification of candidate star clusters depended on the following characteristics of the training sample:

- (i) Origin of classifications: Primary classification or the mode of two classifiers is shown in Table 1;
- (ii) Size of images used for training:  $240p * 240p$ ;
- (iii) Using a random selection of 80 percent of the samples described in Table 1 separately, and the remaining 20 percent was reserved for validation.

The thyroid nodule ultrasound image data used was obtained from the hospitals. After the data augmentation, there were 2,763 images of malignant cases and 541 images of benign cases, with a total of 3,304 images. All images were cropped into a size of  $240 * 240$ . The images were extracted from the thyroid ultrasound video sequence by the ultrasonic apparatus, at a frequency of 12 MHz. The TI-RADS score was given by a professional physician after the image diagnosis. 3,123 images of cases were used for the training of improved models. 541 images, as a test data set, were then randomly divided into 5 groups to test the above three models. Each of the benign and malignant samples is divided into a verification set, test set, and training set. The specific classification scheme is shown in Table 1. The overall condition is shown in Table 1 below.

**4.2. Comparative Analysis of Experimental Results.** The comparison of accuracy is important for different models. Table 2 shows that our mean (Improved Inception) was

TABLE 1: Distribution of samples in training and validation of database 1.

	Malignant	Benign	Normal	Total samples
Training	2210	433	500	3143
Validating	553	108	500	1161
Testing	50	20	500	570
Total samples	2813	561	1500	4874

improved in terms of accuracy than the common GoogLeNet model, and it exhibited the highest accuracy rate in determining whether a thyroid nodule changed pathologically.

The confusion matrix and performance standards obtained in LeNet5, VGG16, GoogLeNet, and GoogLeNet (improved) models are shown in Figure 6.

The LeNet5 architecture correctly predicted 860 out of 1000 images and incorrectly predicted 140. The VGG16 architecture correctly predicted 920 out of 1000 images and incorrectly predicted 80. Although the GoogLeNet architecture correctly predicted 960 of the 1,000 pictures, it incorrectly predicted 40 of them. The most successful class of the GoogLeNet (improved) architecture is the ordinary class. The GoogLeNet (improved) architecture correctly predicted 970 out of 1000 images and incorrectly predicted 30.

True Positive Rate (TPR) is shown on the vertical axis of Figure 6, and False Positive Rate (FPR) is shown on the horizontal axis of Figure 6. The entire graph is also called the ROC curve. Figure 6 shows the result of classification accuracy percentage of our proposed algorithm as 96.65%, 97.81%, 97.32%, 95.97%, and 0.97%, respectively.

The loss values of different CNN models are shown in Table 3, and the GoogLeNet (Improved Inception) model was relatively minimal. The change in the trend in

TABLE 2: Classification of testing samples by different models.

Sample no	LeNet5	VGG16	GoogLeNet	GoogLeNet (improved inception)
1	0.5427	0.3326	0.2907	0.2876
2	0.8965	0.5024	0.5041	0.4992
3	0.7129	0.5598	0.3716	0.3218
4	0.6872	0.4901	0.3524	0.3412
5	0.7428	0.4983	0.4033	0.3916
Average	0.7164	0.4766	0.3844	0.3683

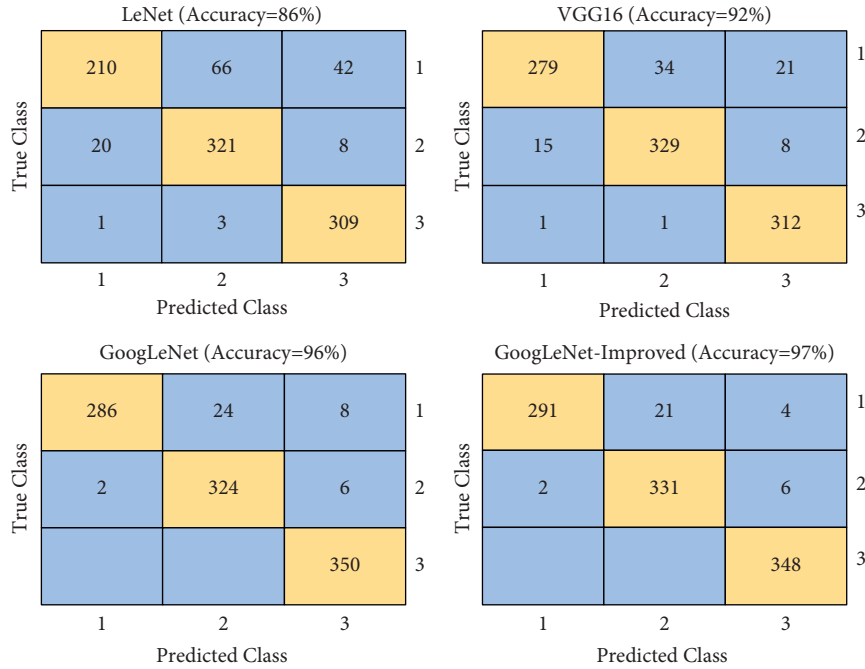


FIGURE 6: Confusion matrices obtained in the Softmax classifier (1 is benign, 2 is malignant, and 3 is normal).

TABLE 3: Loss values of different CNN models.

Sample no	LeNet5	VGG16	GoogLeNet	GoogLeNet (improved inception)
1	87.29	93.02	96.31	96.65
2	84.68	91.31	95.83	97.81
3	85.03	90.98	96.47	97.32
4	86.39	92.17	95.96	95.97
5	85.42	91.54	95.63	97.12
Average	85.76	91.80	96.04	96.97

continuous iteration is shown in Figure 7. Table 4 shows the results of the time consumed by the different models to diagnose the same test image.

As shown in Table 5, the LeNet5 model exhibited a shortest time to diagnose the thyroid ultrasound image, and the GoogLeNet model exhibited the second shortest.

**4.3. Joint Training and Secondary Transfer Learning.** In GoogLeNet, the transfer is from the MINIST data set to the thyroid image. Generally, it is believed that the transfer effect is worse than that of the two similar data sets, when the two data sets have a great difference. MINIST, as a natural image, greatly differs from the medical image. Therefore, the joint data training was conducted herein, based on the public

database, and it was provided by the cooperative organization of this paper. Because of the lack of samples, the joint database was deemed as a whole in the training, which further expanded the overall database.

In transfer learning, the database of small samples was used as the aiming field, and a great quantity of marked database was used as the source domain. In the previous experiments, 2,210 images of malignant and 553 images of benign cases, a total of 3,374 images, were collected from hospitals.

**4.4. Analysis of Experimental Results.** Table 4 shows the difference in the performance between the secondary transfer learning and the primary transfer, the data joint training, and the VGG16-based system. The results showed

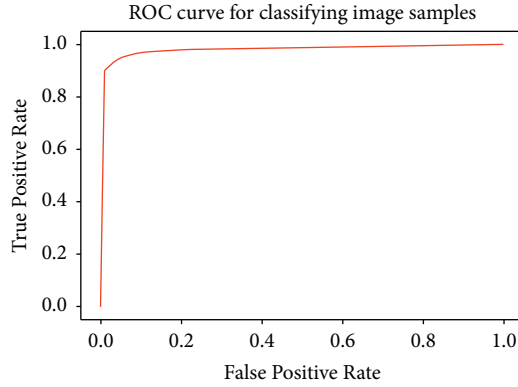


FIGURE 7: TPR/FPR curves.

TABLE 4: Comparison of evaluation index obtained by different methods.

Method	F1 score	Precision	Recall	Accuracy
Twice transfer learning based on LeNet5	0.9748	0.9678	0.9671	0.9649
Once transfer learning based on LeNet5	0.9584	0.9788	0.9685	0.9635
Twice transfer learning based on VGG16	0.9657	0.9718	0.9781	0.9742
Once transfer learning based on VGG16	0.9604	0.9548	0.9685	0.9692
Twice transfer learning based on inception V3 (our method)	0.9874	0.9842	0.9684	0.9885
Once transfer learning based on inception V3 (our method)	0.9674	0.9747	0.9774	0.9745

TABLE 5: Time consumed by different CNN models to diagnose thyroid ultrasound images.

Sample no	LeNet5	VGG16	GoogLeNet	Our method (improved inception)
1	14.65	19.23	15.69	16.70
2	13.97	20.56	16.53	16.87
3	15.56	18.59	17.59	18.47
4	15.55	21.95	18.81	18.91
5	15.13	22.37	17.03	17.05
Average	14.57	20.50	17.09	17.36

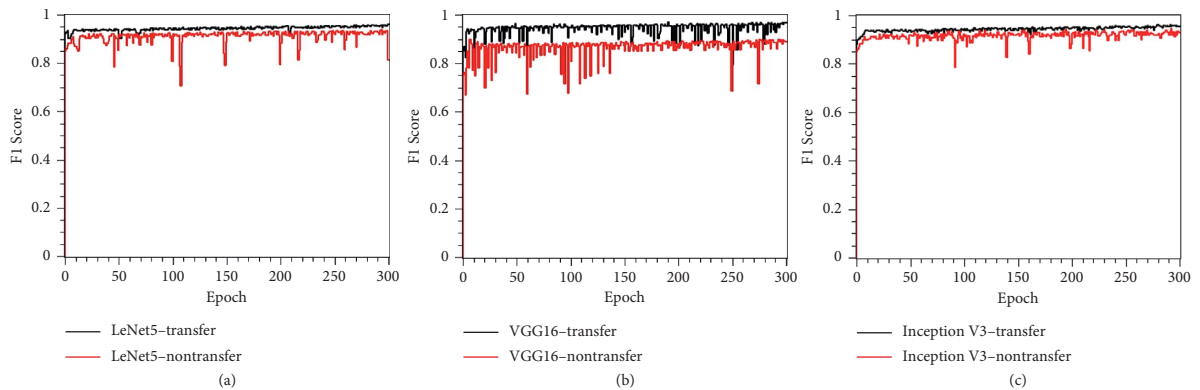


FIGURE 8: Performance comparison of system transfer and nontransfer learning based on (a) LeNet5, (b) VGG16, and (c) inception V3.

that, for small medical data sets, the secondary transfer significantly improved the system performance. Figure 8 shows the comparison of system transfer and non-transfer learning based on LeNet5, VGG16, Inception V3.

With  $\alpha = 0.05$ , the  $p$  value of the VGG16 model is less than 0.001, and the  $t$  value =  $-28.71$ . Because the  $p$  value is

less than  $\alpha$ , there is enough evidence to reject these invalid hypotheses. The  $p$  value of the LeNet model is 0.05 and the  $t$  value =  $-1.66$ , which shows that there is not enough evidence to reject the null hypothesis  $\alpha = 0.05$ . However, between LeNet and VGG16, the average value of VGG16 is higher, and the average value of the other two groups of GoogleNet

and GoogLeNet (improved) is higher than that of LeNet and VGG16.

In case of data sets with similar category, the data joint training also showed a close agreement to the experimental result of secondary transfer. The data joint training and secondary transfer were effective in further improving the system performance, while introducing the transfer learning. This provided a reference for the classification of small data sets and medical image data sets.

## 5. Conclusions

In the present study, the thyroid ultrasound image was preprocessed by the TV-based self-adaptive image restoration method. Subsequently, the CNN model was established using the corresponding loss function, learning rate, moving average, and optimization algorithm set for optimization. Three improved models, namely, the LeNet5 model, VGG16 model, and GoogLeNet model, were trained to diagnose the benign and malignant thyroid nodules. Thereafter, the accuracy rate of each model in terms of diagnosing results was obtained through the tests.

Although all of the three trained models completed the recognition, to verify the best CNN model for diagnosing such ultrasound images, we collected a large amount of image data for training and testing. In the comparison studies, it was found that the GoogLeNet (Improved) exhibited the relatively highest accuracy rate in determining whether a thyroid nodule changed pathologically. The average accuracy rate of the GoogLeNet model was up to 96.04%; furthermore, GoogLeNet (Improved) achieves classification accuracy of 97%, with a loss value of 0.3844. It explains that the GoogLeNet model can diagnose whether the patient's thyroid is in diseased state or is normal. In the end, the data joint training and secondary transfer learning were performed for the open source data sets, and the thyroid ultrasound image data was collected from the hospitals, which further improved the classification accuracy.

In the experiment in this paper, deep learning was applied to the auxiliary medical diagnosis. Our next step is to gradually optimize the model and study the improvement of model, so as to ensure a high accuracy rate of the results. The image classification and diagnosis method based on deep learning will provide a reference to the doctors to diagnose such diseases, help them improve diagnosis efficiency and accuracy, immensely save manpower, and provide new concepts for the ultrasound diagnosis of the thyroid nodules in future.

## Data Availability

The data used to support the findings of this study are available from Weibin Chen via e-mail: sun@wzu.edu.cn.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

WB.CHEN conceptualized the study; WB.CHEN and ZY.GU developed the methodology; WB.CHEN and ZY.GU

worked on software; ZM.LIU, YY.FU, and ZP.YE validated the study; XIN.ZHANG carried out formal analysis; L.XIAO investigated the study; L.XIAO helped with the resources; WB.CHEN and L.XIAO wrote, reviewed, and edited the manuscript; XIN.ZHANG visualized the study. All authors have read and agreed to the published version of the manuscript.

## Acknowledgments

This work was financially supported by Zhejiang Provincial Natural Science Foundation of China under Grant nos. LY21F020001 and LY19F030006, Wenzhou Science and Technology Bureau of China (Wenzhou major scientific and technological innovation project, under Grant nos. ZG2020026 and ZY2019019), and Science and Technology of Wenzhou (Y20180232).

## References





- [1] D. Chen, J. Niu, Q. Pan, Y. Li, and M. Wang, "A deep-learning based ultrasound text classifier for predicting benign and malignant thyroid nodules," in *Proceedings of the 2017 International Conference on Green Informatics (ICGI)*, Fuzhou, China, August 2017.
- [2] J. Chi, E. Walia, P. Babyn, J. Wang, G. Groot, and M. Eramian, "Thyroid nodule classification in ultrasound images by fine-tuning deep convolutional neural network," *Journal of Digital Imaging*, vol. 30, no. 3, pp. 477–486, 2017.
- [3] M. Halicek, G. Lu, J. V. Little et al., "Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging," *Journal of Biomedical Optics*, vol. 22, no. 6, Article ID 060503, 2017.
- [4] H. Li, J. Weng, Y. Shi et al., "An improved deep learning approach for detection of thyroid papillary cancer in ultrasound images," *Scientific Reports*, vol. 8, no. 1, p. 6600, 2018.
- [5] S. Tsantis, D. Cavouras, I. Kalatzis, N. Piliouras, N. Dimitropoulos, and G. Nikiforidis, "Development of a support vector machine-based image analysis system for assessing the thyroid nodule malignancy risk on ultrasound," *Ultrasound in Medicine and Biology*, vol. 31, no. 11, pp. 1451–1459, 2005.
- [6] J. Ma, S. Luo, M. Dighe, D.-J. Lim, and Y. Kim, "Differential diagnosis of thyroid nodules with ultrasound elastography based on support vector machines," in *Proceedings of the 2010 IEEE International Ultrasonics Symposium*, pp. 1372–1375, San Diego, CA, USA, October 2010.
- [7] U. R. Acharya, S. V. Sree, G. Swapna et al., "Effect of complex wavelet transform filter on thyroid tumor classification in three-dimensional ultrasound," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 227, no. 3, pp. 284–292, 2013.
- [8] A. Shukla, R. Tiwari, P. Kaur, and R. R. Janghel, "Diagnosis of thyroid disorders using artificial neural networks," in *Proceedings of the IEEE International Advance Computing Conference*, Patiala, India, March 2009.
- [9] A. Prochazka, S. Gulati, S. Holinka, and D. au, "Patch-based classification of thyroid nodules in ultrasound images using direction independent features extracted by two-threshold binary decomposition," *Computerized Medical Imaging and Graphics*, vol. 71, pp. 9–18, 2019.
- [10] R. Thandiackal, C. Brandle, D. Leach, A. Jafari, and F. Iida, "Exploiting passive dynamics for robot throwing task," in



- Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, October 2012.
- [11] N. Gessert, M. Bengs, L. Wittig et al., "Deep transfer learning methods for colon cancer classification in confocal laser microscopy images," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 11, pp. 1837–1845, 2019.
  - [12] T. Liu, S. Xie, J. Yu, L. Niu, and W. Sun, "Classification of thyroid nodules in ultrasound images using deep model based transfer learning and hybrid features," in *Proceedings of the IEEE International Conference on Acoustics*, New Orleans, LA, USA, March 2017.
  - [13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
  - [14] X. Zhang, H. Gu, L. Zhou, L. Chen, Q. Zheng, and W. Chen, "Improved dual-domain filtering and threshold function denoising method for ultrasound images based on non-subsampled contourlet transform," *Journal of Medical Imaging and Health Informatics*, vol. 7, no. 7, pp. 1624–1628, 2017.
  - [15] W.-B. Chen, L. Zhou, H.-B. Gu, and X. Zhang, "Ratio weighted and condition weighted medical image fusion based on contourlet transformation coefficients," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 19, no. 3, pp. 191–198, 2015.
  - [16] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, December 2012.
  - [17] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. De Vries, M. J. N. L. Benders, and I. Išgum, "Automatic segmentation of MR brain images with a convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1252–1261, 2016.
  - [18] W.-B. Chen, M. Hu, L. Zhou, H. Gu, and X. Zhang, "Fusion algorithm of multi-focus images with weighted ratios and weighted gradient based on wavelet transform," *Journal of Intelligent Systems*, vol. 8, no. 19, pp. 0077–0078, 2017.
  - [19] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, no. 15, pp. 42–49, 2016.
  - [20] X. Zhang, M. Fan, D. Wang, P. Zhou, and D. Tao, "Top- $k$  feature selection framework using robust 0-1 integer programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3005–3019, 2021.
  - [21] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 238–255, 2021.
  - [22] T. Williams and R. Li, "Advanced image classification using wavelets and convolutional neural networks," in *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, February 2017.
  - [23] Y. Tokuoka, S. Suzuki, and Y. Sugawara, "An inductive transfer learning approach using cycle-consistent adversarial domain adaptation with application to brain tumor segmentation," 2020, <https://arxiv.org/abs/2005.04906>.
  - [24] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, M. Grosse-Wentrup, and Moritz, "Transfer learning in brain-computer interfaces abstractuFFFFDThe performance of brain-computer interfaces (BCIs) improves with the amount of avail," *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 20–31, 2016.
  - [25] X. Zhang, T. Wang, J. Wang, G. Tang, and L. Zhao, "Pyramid channel-based feature attention network for image dehazing," *Computer Vision and Image Understanding*, vol. 197–198, Article ID 103003, 2020.
  - [26] X. Zhang, J. Zhang, W. Chen, M. Hu, and X. Zhang, "Double fidelity terms unidirectional variation model for single-image rain removal," *Jouanal of Electronic Imaging*, vol. 28, no. 2, pp. 15–23, 2019.
  - [27] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, no. 26, pp. 228–235, 2017.
  - [28] B. Zoph, D. Yuret, J. May, and K. K. Jonathan, "Transfer learning for low-resource neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, November 2016.
  - [29] P. Kulkarni, J. Zepeda, F. Jurie, P. Perez, and L. Chevallier, "Hybrid multi-layer deep CNN/aggregator feature for image classification," 2015, <https://arxiv.org/abs/1503.04065>.
  - [30] D. Mahapatra, P. K. Roy, S. Sedai, and R. Garnavi, "Retinal image quality classification using saliency maps and CNNs," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging*, pp. 172–179, Athens, Greece, October 2016.
  - [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
  - [32] W. K. Moon, Y.-W. Lee, H.-H. Ke, S. H. Lee, C.-S. Huang, and R.-F. Chang, "Computer-aided diagnosis of breast ultrasound images using ensemble learning from convolutional neural networks," *Computer Methods and Programs in Biomedicine*, vol. 190, Article ID 105361, 2020.
  - [33] S. Y. Kim, Y. Choi, E. K. Kim et al., "Deep learning-based computer-aided diagnosis in screening breast ultrasound to reduce false-positive diagnoses," *Scientific Reports*, vol. 11, no. 1, pp. 1–11, 2021.
  - [34] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018.
  - [35] T. O Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications & Networking*, vol. 3, no. 4, pp. 563–575, 2017.
  - [36] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and R. S. Tan, "A deep convolutional neural network model to classify heartbeats," *Computers in Biology and Medicine*, vol. 89, 2017.
  - [37] J. Shen, S. H. Kong, and T. F. Chan, "Euler's elastica and curvature-based inpainting," *SIAM Journal on Applied Mathematics*, vol. 63, no. 2, pp. 564–592, 2003.
  - [38] C. Szegedy, W. Wei Liu, Y. Yangqing Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

## Research Article

# Diagnosis of COVID-19 Using a Deep Learning Model in Various Radiology Domains

**Yousef Alhwaiti** <sup>1</sup>, **Muhammad Hameed Siddiqi** <sup>1</sup>, **Madallah Alruwaili** <sup>1</sup>,  
**Ibrahim Alrashdi**,<sup>1</sup> **Saad Alanazi** <sup>1</sup>, and **Muhammad Hasan Jamal**<sup>2</sup>

<sup>1</sup>College of Computer and Information Sciences, Jouf University, Sakaka, Aljouf, 2014, Saudi Arabia

<sup>2</sup>Department of Computer Science, COMSATS University, Islamabad, Lahore Campus, Pakistan

Correspondence should be addressed to Muhammad Hameed Siddiqi; mhsiddiqi@ju.edu.sa

Received 28 April 2021; Accepted 12 August 2021; Published 13 September 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Yousef Alhwaiti et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many countries are severely affected by COVID-19, and various casualties have been reported. Most countries have implemented full and partial lockdowns to control COVID-19. Paramedical employee infections are always a threatening discovery. Front-line paramedical employees might initially be at risk when observing and treating patients, who can contaminate them through respiratory secretions. If proper preventive measures are absent, front-line paramedical workers will be in danger of contamination and can become unintentional carriers to patients admitted in the hospital for other illnesses and treatments. Moreover, every country has limited testing capacity; therefore, a system is required which helps the doctor to directly check and analyze the patients' blood structure. This study proposes a generalized adaptive deep learning model that helps the front-line paramedical employees to easily detect COVID-19 in different radiology domains. In this work, we designed a model using convolutional neural network in order to detect COVID-19 from X-ray, Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) images. The proposed model has 27 layers (input, convolutional, max-pooling, dropout, flatten, dense, and output layers), which has been tested and validated on various radiology domains such as X-ray, CT, and MRI. For experiments, we utilized 70% of the dataset for training and 30% for testing against each dataset. The weighted average accuracies for the proposed model are 94%, 85%, and 86% on X-ray, CT, and MRI, respectively. The experiments show the significance of the model against state-of-the-art works.

## 1. Introduction

The rapid spread of COVID-19 has motivated scientists to quickly develop countermeasures using technologies such as cognitive computing, deep learning, artificial intelligence, machine intelligence, cloud-based collaboration, and wireless communication [1].

Cognitive computing simulates human thought processes and is extensively used in fields such as finance and investment, healthcare and veterinary medicine, travel, and mobile systems [2–4]. The Internet of things (IoT) is implemented on interconnected electronic devices with unique identifiers (UIDs), such as computers, smartphones, coffeemakers, washing machines, and wearable devices

[5, 6]. The IoT, along with cloud computing, Artificial Intelligence (AI), Machine Learning (ML), and deep learning, could be a powerful tool to combat COVID-19 [7, 8], and 4<sup>th</sup> generation (4G) and 5<sup>th</sup> generation (5G) wireless communication technologies have the potential to revolutionize many sectors, including healthcare [9–11]. China has already used 5G technology to fight the COVID-19 pandemic by monitoring patients, collecting and analyzing data, and tracking viruses [1].

Most developing countries utilize wireless technologies, laboratory-based trails, and radiological investigations in order to recognize and diagnose COVID-19 [12]. A standard method is real-time reverse transcription polymerase chain reaction (qRT-PCR), but false-negative results can occur due

to asymptomatic patients, and mistakes may also affect its role in identifying COVID-19 [13, 14]. In the early stages, imaging technologies such as CT scan, MRI, and X-rays might play a vital role in detecting COVID-19 patients [15–17].

Radiology-based chest scanning has been employed to investigate pneumonia [18]. An artificial intelligence- (AI-) based tool was developed [19] to automatically detect, quantify, and monitor COVID-19 and to differentiate affected and normal patients. A deep learning-based approach [20] was developed to automatically segment the entire lung with infection sites under a chest CT. Similarly, an early screening system based on deep learning can discriminate influenza (viral pneumonia) from vigorous cases and COVID-19 [21]. A deep learning-based approach can extract graphical features from CT images of COVID-19 [22]. These features deliver prior medical analysis pathogenic testing and have been claimed to save crucial time for disease investigation. However, most consider just one radiology domain, such as X-ray or CT.

This work builds a deep learning approach in order to notice COVID-19 from various radiological input images such as X-ray, CT, and MRI. The model is a convolutional neural network (CNN) whose 27 layers include input, convolutional, max-pooling, dropout, flatten, dense, and output. The input layer accepts input grayscale images of size  $128 \times 128$  and uses 64 filters of size  $3 \times 3$ . A ReLU activation function is employed in the input layer and all hidden layers. Following the max-pooling layer is a dropout layer to avoid overfitting. This drops out different neurons in the hidden layer. The percentage of neurons to drop should be specified when using the dropout function. We drop 30%. Next are two convolutional layers, both with 128 filters of size  $3 \times 3$ , then a  $2 \times 2$  max-pooling layer, and a dropout layer to drop 30% of neurons. We add three convolutional layers with 256 filters, each with size  $3 \times 3$ , followed by a max-pooling layer with the same parameters as the previous pooling layer. We also drop 30% of the output neurons. We continue increasing filters in more layers, adding three convolutional layers with 512 filters in each layer, with the same filter size as previous layers. A max-pooling layer follows this stack of layers, and 30% of the neurons are dropped. Two stacks, the same as previous layers, are added using the same parameters.

The rest of the paper is organized as follows. Section 2 summarizes state-of-the-art work in various radiology domains. Section 3 presents the proposed methodology. The datasets used in this research are described in Section 4. The experimental environment for the proposed approach is presented in Section 5, and the results are discussed in Section 6. Section 7 discusses conclusions and directions of future work.

## 2. Related Works

Various radiology techniques (e.g., X-ray, CT, and MRI) have been utilized as imaging modalities in the diagnosis of COVID-19, and research has proposed the identification of COVID-19 against different radiology methods, with various limitations.

An early-stage screening model [23] could differentiate COVID-19 patients from normal humans by employing deep learning techniques under pulmonary CT images, with 86.7% accuracy against 618 CT samples. However, the segmentation model employed before feeding it to the learning model could lose some important features and cause misclassification, and only limited radiology images were employed in experiments. An automatic deep CNN system [18] was based on pretrained models under chest X-ray images. This heuristic model utilized limited X-ray images in a controlled domain.

An integrated technique based on an artificial neural network and convolutional CapsNet [24] was developed to identify COVID-19 against chest X-ray images with pill networks. The performance was assessed with binary and multiclass classifications such as infected, normal, and pneumonia, indicating a 97% recognition rate on binary classification and 84% on multiclass classification. There was no rule to find the structure of the artificial neural network, which had no specific scheme to define the structure of neurons, which could be achieved by experience or trial-and-error [25]. When training of the neural network was completed, the network was reduced to a specific value of error on image samples; hence, it provided no optimum outcomes [25].

Some recent systems [18, 26–28] have utilized deep learning and artificial intelligence to identify COVID-19, but only on X-ray images. Similarly, a commercial platform was used to classify infected patients and normal humans [29], with limited contribution from the authors, who utilized only X-rays in experiments. Deep learning and CNN were used to classify positive patients with coronavirus and healthy patients [30]. A very small dataset of X-ray images was used, which might not be applicable in naturalistic domains.

An automated method [31] was proposed to detect COVID-19-positive patients from normal humans, employing a deep learning-based network coupled with gradient-weighted class activation mapping (Grad-CAM) for feature extraction under CT scan images. However, Grad-CAM-based methods require modification of the network architecture, which could degrade accuracy; computational Grad-CAM is expensive [32]; and a non-standard dataset was utilized. A deep CNN, called decompose, transfer, and compose (DeTraC), was used with principal component analysis (PCA) as a feature dimension reduction method to identify coronavirus against chest X-ray images [33]. However, PCA is problematic in the precise assessment of the covariance matrix [34]. Moreover, even modest invariance might not be taken by PCA unless the training data openly deliver this evidence [35]. A deep learning-based system to identify COVID-19 from normal humans had a recognition rate of up to 100% [36], which is not realistic. Only one radiology (X-ray) image was utilized. Similarly, a system was proposed to classify infected, normal, and pneumonia cases with significant accuracy [37], and a deep CNN-based system was proposed to identify patients with coronavirus and normal humans [38]. However, both systems utilized limited X-ray images and used one radiology images.

We develop a deep learning model to accurately categorize the infected patients with COVID-19 and normal humans. The model employs radiology input images such as X-ray, CT, and MRI, through which we can prove the robustness of the model, which is based on 27 layers of a CNN, including input, convolutional, max-pooling, dropout, flatten, dense, and output layers, and shows significant performance on various radiology images such as X-ray, CT scan, and MRI, compared to state-of-the-art methods.

### 3. Materials and Methods

We describe the proposed deep learning-based approach, whose flowchart against X-ray images is shown in Figure 1.

The model is based on a Convolutional Neural Network (CNN) with 27 layers. The input layer accepts a grayscale image of size  $128 \times 128$  and uses 64 filters of size  $3 \times 3$ . A rectified linear unit (ReLU) activation function is used in the input layer and all hidden layers, where ReLU is defined by the relation  $R(z) = \max(0, z)$ , as shown in Figure 2.

The ReLU activation function omits negative pixels in the input image. The second layer is a convolutional layer that has 64 filters of size  $3 \times 3$ . Next is a max-pooling layer that takes the maximum value for each patch of the feature map, with pool size and stride both  $2 \times 2$ .

Next is a dropout layer to avoid overfitting by dropping out different neurons in hidden layers. We drop 30% of the neurons to reduce overfitting. The dropout technique is shown in Figure 3.

The output of the previous layer is flattened to convert a matrix to a single layer. For instance, an output shape of (1, 128, 18) is flattened to (1, 16384). Then, two dense layers with 4096 units each are added. An ReLU activation function is used in both layers. The last layer is the output layer with two neurons, which is the number of classes (COVID-19 positive and COVID-19 negative). A soft-max activation function normalizes the input vector from the previous layer of real numbers to a probability distribution

$$\sigma(Z) \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad \text{for } i = 1, \dots, K \text{ and } Z = (z_1, \dots, z_k) \in R^K. \quad (1)$$

The proposed approach is described in Figure 4.

### 4. Datasets Used

We utilized the following datasets to show the efficacy of the developed approach.

**4.1. X-Ray Image Dataset.** We utilized a radiology dataset with 270 X-ray images from males and females of age 20–55 years, collected from various open sources (used to diagnose coronavirus). During implementation, we regularly updated the dataset to incorporate the latest complex chest X-ray images. The dataset was thoroughly checked by medical experts (physicians). We did not provide metadata for patients. Images were converted to a vector with dimensions  $1 \times 6400$  by decreasing the dimension of every input image to

$80 \times 80$ . To avoid imbalance, we utilized 135 normal patients' images and 135 COVID-19-positive images. The dataset was collected over a period of 3 months (June to August 2020).

**4.2. Computed Tomography (CT) Scan Image Dataset.** The CT image dataset contained 270 chest CT images. The dataset was built from open sources commonly used to diagnose COVID-19. The dataset incorporated new complex CT scan images that were systematically checked by doctors. The images were from males and females of age 35 to 55 years. For experiments, images in this dataset were transformed to a vector with dimensions  $1 \times 6400$  by decreasing the dimension of every input image to  $80 \times 80$ . To avoid imbalance, we utilized images of 135 normal patients and 135 images from COVID-19-positive patients. The dataset was collected over 3 months (June to August 2020).

**4.3. Magnetic Resonance Imaging (MRI) Image Dataset.** Another type of radiology dataset was of MRI scans, which generate two types of images. T1-weighted images highlight (brighten) lipids and fats by a radio-frequency pulse sequence, and T2-weighted images highlight also water. So, the timing of the radiofrequency pulse sequence highlights the target tissues. We included 270 MRI images of males and females of age 35 to 60 years. These were confirmed cases of COVID-19. We added controls with approximately similar ages and genders but without COVID-19. All images were converted to a vector of dimension  $1 \times 6400$  by decreasing the dimension of every input image to  $80 \times 80$ . To avoid imbalance, we utilized 135 images of normal patients and 135 COVID-19-positive images. The dataset was collected over 3 months (June to August 2020).

## 5. Experimental Setup

We performed many experiments to show the significance of the proposed model against each dataset; these were divided into 70% for training and 30% for testing for all tested algorithms. The same model architecture was used for each dataset, with different hyperparameters.

All experiments were performed using Python, TensorFlow, and Google Colab (for training) on an Intel Pentium Core i7-6700 (3.4 GHz) with 16 GB RAM. Experiments are described as follows:

- (i) The first experiment assessed the proposed model against chest X-ray, CT scan, and MRI datasets through an average cross-validation scheme.
- (ii) The second experiment included a set of sub-experiments performed under the absence of the developed approach against all three datasets. We utilized logistic regression, support vector machine, random forest, k-nearest neighbor, artificial neural network, Naive Bayes, decision tree, passive aggressive classifier, multilayer perceptron, and extra tree classifiers.
- (iii) The third experiment compared the proposed technique to the state of the art.

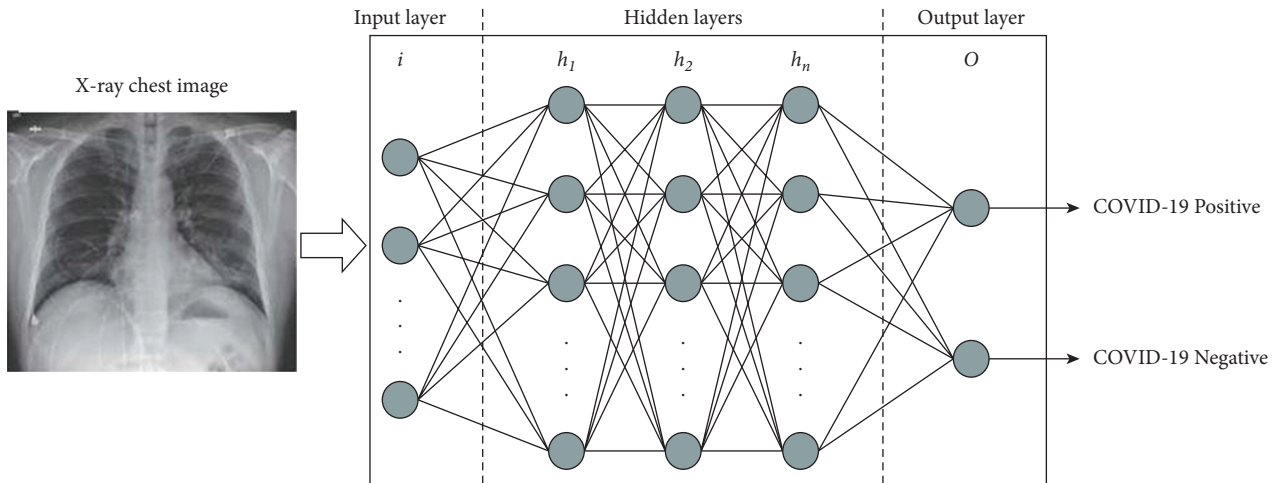


FIGURE 1: Flow diagram of the proposed approach against X-ray images.

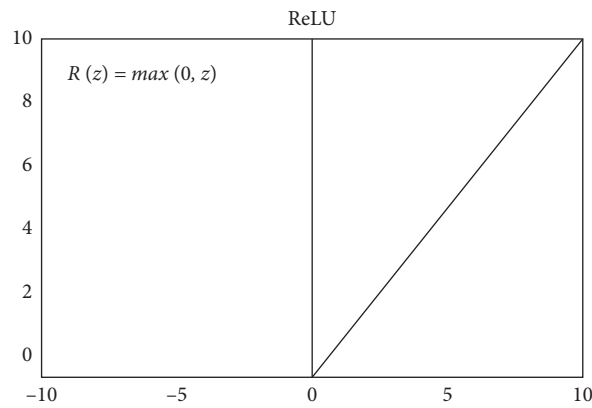


FIGURE 2: ReLU activation function [39].

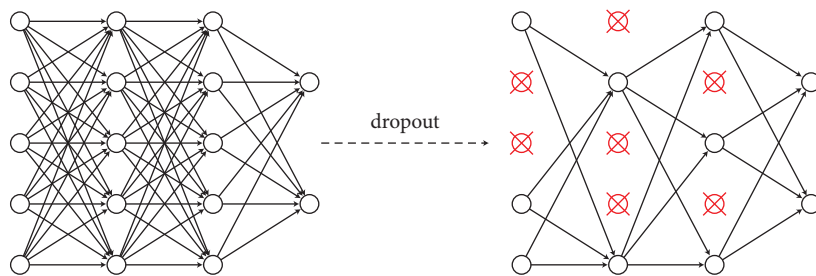


FIGURE 3: Dropout technique [40].

## 6. Results and Discussion

6.1. *First Experiment.* The results of the first experiment are shown in Table 1.

As can be seen from Table 1, the model got significant accuracy among state of the art. It trained in different datasets with different hyperparameters using same structure of the model. That indicates that the structure of the model is significant. Also, the model was compared with different machine

learning algorithms to show the difference in accuracy between deep learning and regular machine learning algorithms.

6.2. *Second Experiment.* The results of the second experiment are presented in Tables 2–11.

We observe from Tables 2–11 that all of the existing classifiers did not achieve better accuracy against the three datasets. This is because most of the medical images are

Layer (type)	Output Shape	Param #			
conv2d (Conv2D)	(None, 128, 128, 64)	640	conv2d_7 (Conv2D)	(None, 16, 16, 512)	1180160
conv2d_1 (Conv2D)	(None, 128, 128, 64)	36928	conv2d_8 (Conv2D)	(None, 16, 16, 512)	2359808
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0	conv2d_9 (Conv2D)	(None, 16, 16, 512)	2359808
dropout (Dropout)	(None, 64, 64, 64)	0	max_pooling2d_3 (MaxPooling2)	(None, 8, 8, 512)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856	dropout_3 (Dropout)	(None, 8, 8, 512)	0
conv2d_3 (Conv2D)	(None, 64, 64, 128)	147584	conv2d_10 (Conv2D)	(None, 8, 8, 512)	2359808
max_pooling2d_1 (MaxPooling2)	(None, 32, 32, 128)	0	conv2d_11 (Conv2D)	(None, 8, 8, 512)	2359808
dropout_1 (Dropout)	(None, 32, 32, 128)	0	conv2d_12 (Conv2D)	(None, 8, 8, 512)	2359808
conv2d_4 (Conv2D)	(None, 32, 32, 256)	295168	max_pooling2d_4 (MaxPooling2)	(None, 4, 4, 512)	0
conv2d_5 (Conv2D)	(None, 32, 32, 256)	590080	dropout_4 (Dropout)	(None, 4, 4, 512)	0
conv2d_6 (Conv2D)	(None, 32, 32, 256)	590080	flatten (Flatten)	(None, 8192)	0
max_pooling2d_2 (MaxPooling2)	(None, 16, 16, 256)	0	dense (Dense)	(None, 4096)	33558528
dropout_2 (Dropout)	(None, 16, 16, 256)	0	dense_1 (Dense)	(None, 4096)	16781312
			dense_2 (Dense)	(None, 2)	8194
			=====		
			Total params: 65,061,570		
			Trainable params: 65,061,570		
			Non-trainable params: 0		

(a)

(b)

FIGURE 4: (a), (b) The workflow of the proposed model.

TABLE 1: Recognition rates of the proposed approach against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	94%
	Misclassified	6%
CT scan	Classified	85%
	Misclassified	15%
MRI	Classified	86%
	Misclassified	16%

TABLE 2: Recognition rates of logistic regression (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	73%
	Misclassified	27%
CT scan	Classified	65%
	Misclassified	35%
MRI	Classified	74%
	Misclassified	26%

sensitive to noise and other environmental factors. The proposed approach can achieve significant accuracy under the presence of noise and other environmental factors, as shown in Table 1.

**6.3. Third Experiment.** The recognition rates of the proposed model and other models are shown in Tables 12–14, which present that the proposed model achieved higher accuracy on all three datasets.

As illustrated in Tables 12–14, the proposed approach achieved higher accuracy than other recent works under all the three radiology datasets.

**6.4. Discussion.** COVID-19 has affected millions of people around the world. The early detection of COVID-19 could help in stopping the spread. One of the most effective detections is screening the infected patients. Deep learning plays an affective role in this detection, and it is more

TABLE 3: Recognition rates of support vector machine (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	75%
	Misclassified	25%
CT scan	Classified	51%
	Misclassified	49%
MRI	Classified	78%
	Misclassified	22%

TABLE 4: Recognition rates of random forest (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	79%
	Misclassified	21%
CT scan	Classified	74%
	Misclassified	26%
MRI	Classified	77%
	Misclassified	23%

TABLE 5: Recognition rates of k-nearest neighbor (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	79%
	Misclassified	21%
CT scan	Classified	69%
	Misclassified	31%
MRI	Classified	81%
	Misclassified	19%

TABLE 6: Recognition rates of support vector machine (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	72%
	Misclassified	28%
CT scan	Classified	66%
	Misclassified	34%
MRI	Classified	79%
	Misclassified	21%

TABLE 7: Recognition rates of Naïve Bayes (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	80%
	Misclassified	20%
CT scan	Classified	58%
	Misclassified	42%
MRI	Classified	77%
	Misclassified	23%

TABLE 8: Recognition rates of decision tree (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	72%
	Misclassified	28%
CT scan	Classified	65%
	Misclassified	35%
MRI	Classified	77%
	Misclassified	23%

TABLE 9: Recognition rates of passive aggressive classifier (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	65%
	Misclassified	35%
CT scan	Classified	71%
	Misclassified	29%
MRI	Classified	70%
	Misclassified	30%

TABLE 10: Recognition rates of multilayer perceptron (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	60%
	Misclassified	40%
CT scan	Classified	66%
	Misclassified	34%
MRI	Classified	47%
	Misclassified	53%

TABLE 11: Recognition rates of extra tree classifier (under the absence of the proposed approach) against X-ray, CT, and MRI datasets.

Datasets	Recognition rates	
X-ray	Classified	81%
	Misclassified	19%
CT scan	Classified	79%
	Misclassified	21%
MRI	Classified	79%
	Misclassified	21%

TABLE 12: The comparison of the proposed approach along with the state-of-the-art methods on the X-ray dataset (unit: %).

State of the art	Weighted average recognition rates	Standard deviation
[28]	89.6	$\pm 2.9$
[30]	83.5	$\pm 3.7$
[41]	80.3	$\pm 2.6$
[42]	85.4	$\pm 1.2$
[43]	79.8	$\pm 3.8$
[44]	89.3	$\pm 2.3$
[45]	91.5	$\pm 1.9$
[46]	90.5	$\pm 2.7$
Proposed model	94.0	$\pm 3.5$

TABLE 13: The comparison of the proposed approach along with the state-of-the-art methods on the CT scan dataset (unit: %).

State of the art	Weighted average recognition rates	Standard deviation
[28]	82.3	$\pm 1.7$
[47]	75.2	$\pm 1.9$
[48]	84.7	$\pm 2.1$
[49]	82.4	$\pm 2.5$
[50]	83.4	$\pm 1.2$
[51]	79.2	$\pm 3.6$
Proposed model	85.0	$\pm 3.5$

accurate. There are many studies implemented in to detect COVID-19 using deep learning. In this research, we have used a deep learning neural network (convolutional neural network) to detect COVID-19 from different radiologists. Our model results are comparable to the state of the art

because we have built a robust model. What made our model a comparable result is that we build the model using different layers (27 layers), adjusting hyperparameters, using effective percentage in dropout, size of the filters in convolutional and pool layers, and using a suitable way in preprocessing. These



TABLE 14: The comparison of the proposed approach along with the state-of-the-art methods on the MRI dataset (unit: %).

State of the art	Weighted average recognition rates	Standard deviation
Logistic regression	74.0	$\pm 3.1$
Support vector machine	78.0	$\pm 1.8$
Random forest	77.0	$\pm 2.4$
k-nearest neighbor	81.0	$\pm 1.1$
Artificial neural network	79.0	$\pm 3.7$
Naïve Bayes	71.0	$\pm 5.2$
Decision tree	77.0	$\pm 2.6$
Passive aggressive classifier	70.0	$\pm 4.6$
Multilayer perceptron	47.0	$\pm 6.1$
Extra tree classifier	79.0	$\pm 4.4$
Proposed model	86.0	$\pm 3.5$

layers include input, convolutional, max-pooling, dropout, flatten, dense, and output. The input layer accepts input grayscale images of size  $128 \times 128$  and uses 64 filters of size  $3 \times 3$ . An ReLU activation function is used in the input layer and all hidden layers. Following the max-pooling layer is a dropout layer to avoid overfitting. This drops out different neurons in the hidden layer. The percentage of neurons to drop should be specified when using the dropout function. We drop 30%. Next are two convolutional layers, both with 128 filters of size  $3 \times 3$ , then a  $2 \times 2$  max-pooling layer, and a dropout layer to drop 30% of neurons. We add three convolutional layers with 256 filters, each with size  $3 \times 3$ , and then a max-pooling layer with the same parameters as the previous pooling layer. We also drop 30% of the output neurons. We continue increasing filters in more layers, adding three convolutional layers with 512 filters in each layer, with the same filter size as previous layers. A max-pooling layer follows this stack of layers, and 30% of the neurons are dropped. Two stacks, the same as previous layers, are added using the same parameters. Moreover, the model got significant accuracy among the state of the art. It trained in different datasets with different hyperparameters using same structure of the model. That indicates that the structure of the model is significant. Also, the model was compared with different machine learning techniques to show the difference in accuracy between deep learning and regular machine learning algorithms.

## 7. Conclusions

We developed a model to efficiently detect COVID-19 from different radiology techniques and showed its robustness on X-ray, CT, and MRI datasets. We used a CNN to build the deep learning model, which gives adequate image classification. To show the performance of the proposed model, many experiments were performed against each dataset. In the first experiment, we built a model using a CNN with different layers and trained it on the first dataset, and the same model constructor was used to train the other datasets. For each dataset, we adjusted the hyperparameters for the model to get a robust model. In the second experiment, we utilized different machine learning algorithms on each dataset (in the absence of the proposed model). This demonstrated the importance and significance of the proposed model. Regardless of the lack of instances in a dataset, our model had

high classification accuracy for COVID-19. Finally, the classification rates of our technique were compared to those of the previous work, and the developed approach presented the best performance on various radiology datasets.

The proposed system was tested and validated in a controlled environment. In future research, we will deploy the system in real healthcare systems in which COVID-19 is easily detected from real images.

## Data Availability

The data utilized in order to support the discoveries of this work are described in the paper and will be offered by the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the present study.

## Acknowledgments

This study was supported by Jouf University, Sakaka, Aljouf, Kingdom of Saudi Arabia, under grant no. DSR2020-06-3675.

## References





- [1] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "Comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90225–90265, 2020.
- [2] K. Lin, C. Li, D. Tian, A. Ghoneim, M. S. Hossain, and S. U. Amin, "Artificial-intelligence-based data analytics for cognitive communication in heterogeneous wireless networks," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 83–89, 2019.
- [3] Y. Zhang, Y. Qian, D. Wu, M. S. Hossain, and A. Ghoneim, M. Chen, "Emotion-aware multimedia systems security," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 617–624, 2018.
- [4] M. S. Hossain and G. Muhammad, "An audio-visual emotion recognition system using deep learning fusion for a cognitive wireless framework," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 62–68, 2019.
- [5] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive internet of

- things: improving sensitivity and interactivity,” *IEEE Network*, vol. 33, no. 3, pp. 58–64, 2019.
- [6] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, “Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city,” *IEEE Access*, vol. 7, pp. 18611–18621, 2019.
  - [7] S. A. Alanazi, M. M. Kamruzzaman, M. Alruwaili, N. Alshammari, S. A. Alqahtani, and A. Karime, “Measuring and preventing COVID-19 using the SIR model and machine learning in smart health care,” *Journal of Healthcare Engineering*, vol. 2020, Article ID 8857346, 12 pages, 2020.
  - [8] M. M. Kamruzzaman, “Architecture of smart health care system using artificial intelligence,” in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, pp. 1–6, London, UK, July 2020.
  - [9] Y. Zhang, M. S. Hossain, A. Ghoneim, and M. Guizani, “COcME: content-oriented caching on the mobile edge for wireless communications,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 26–31, 2019.
  - [10] J. Wang, Y. Miao, P. Zhou, M. S. Hossain, and S. M. M. Rahman, “A software defined network routing in wireless multihop network,” *Journal of Network and Computer Applications*, vol. 85, pp. 76–83, 2017.
  - [11] A. Alelaiwi, A. Alghamdi, M. Shorfuzzaman, M. Rawashdeh, M. S. Hossain, and G. Muhammad, “Enhanced engineering education using smart class environment,” *Computers in Human Behavior*, vol. 51, pp. 852–856, 2015.
  - [12] M. Shahzeb, A. Khan, and A. Muhammad, “Detection of coronavirus disease (COVID-19) using radiological examinations,” *Journal of Pure Applied Microbiology*, vol. 14, 2020.
  - [13] H. J. Shi, J. Yu, and C. S. Zheng, “Radiological diagnosis of new coronavirus infected pneumonitis: expert recommendation from the Chinese society of radiology,” *Zhonghua Fang She Xue Za Zhi*, vol. 54, pp. 279–285, 2020.
  - [14] J. F. W. Chan, S. Yuan, K. H. Kok et al., “A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster,” *The Lancet*, vol. 395, no. 10223, pp. 514–523, 2020.
  - [15] H. Fu, H. Xu, N. Zhang, H. Xu, and Z. Li, H. Chen, R. Xu, R. Sun et al., “Association between clinical, laboratory and CT characteristics and RT-PCR results in the follow-up of COVID-19 patients,” *MedRxiv*, 2020.
  - [16] General Office of National Health Committee and others, “Office of state administration of traditional Chinese medicine (2020),” *Notice on the Issuance of a Program for the Diagnosis and Treatment of Novel Coronavirus (2019-nCoV) Infected Pneumonia (Trial Fifth Edition)*, 2020.
  - [17] J. P. Kanne, “Chest CT findings in 2019 novel coronavirus (2019-nCoV) infections from Wuhan, China: key points for the radiologist,” *Radiology*, vol. 295, 2020.
  - [18] A. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (covid-19) using X-ray images and deep convolutional neural networks,” 2020, <https://arxiv.org/abs/2003.10849>.
  - [19] O. Gozes, M. Frid-Adar, H. Greenspan et al., “Rapid AI development cycle for the coronavirus (covid-19) pandemic: initial results for automated detection & patient monitoring using deep learning CT image analysis,” 2020, <https://arxiv.org/abs/2003.05037>.
  - [20] F. Shan, Y. Gao, J. Wang et al., “Lung infection quantification of covid-19 in CT images with deep learning,” 2020, <https://arxiv.org/abs/2003.04655>.
  - [21] C. Butt, J. Gill, D. Chun, and B. A. Babu, “Deep learning system to screen coronavirus disease 2019 pneumonia,” *Applied Intelligence*, p. 1, 2020.
  - [22] S. Wang, B. Kang, J. Ma, X. Zeng, and M. Xiao, “A deep learning algorithm using CT images to screen for coronavirus disease (COVID-19),” *MedRxiv*, 2020.
  - [23] X. Xu, X. Jiang, C. Ma et al., “Deep learning system to screen coronavirus disease 2019 pneumonia,” 2020, <https://arxiv.org/abs/2002.09334>.
  - [24] S. Toraman, T. B. Alakus, and I. Turkoglu, “Convolutional capsnet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks,” *Chaos, Solitons & Fractals*, vol. 140, pp. 110–122, 2020.
  - [25] M. M. Mijwel, “Artificial neural networks advantages and disadvantages,” 2018, <https://www.linkedin.com/pulse/artificial-neuralnet-works-advantages-disadvantages-maadm-mijwel>.
  - [26] M. K. Pandit and S. A. Banday, “SARS n-CoV2-19 detection from chest X-ray images using deep neural networks,” *International Journal of Pervasive Computing and Communications*, vol. 16, 2020.
  - [27] M. Ilyas, H. Rehman, and A. Nait-Ali, “Detection of COVID-19 from chest X-ray images using artificial intelligence: an early review,” 2020, <https://arxiv.org/abs/2004.05436>.
  - [28] S. H. Kassani, P. H. Kassasni, M. J. Wesolowski, K. A. Schneider, and R. Deters, “Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning-based approach,” 2020, <https://arxiv.org/abs/2004.10641>.
  - [29] A. A. Borkowski, N. A. Viswanadham, L. B. Thomas, R. D. Guzman, L. A. Deland, and S. M. Mastorides, “Using artificial intelligence for COVID-19 chest X-ray diagnosis,” *Medrxiv*, vol. 37, 2020.
  - [30] A. Makris, I. Kontopoulos, and K. Tserpes, “COVID-19 detection from chest X-ray images using deep learning and convolutional neural networks,” in *Proceedings of the 11th Hellenic Conferences on Artificial Intelligence*, pp. 60–66, Athens, Greece, September 2020.
  - [31] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, “A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset,” *MedRxiv*, 2020.
  - [32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conferences on Computer Vision*, pp. 618–626, Venice, Italy, October 2017.
  - [33] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, “Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network,” *Applied Intelligence*, 2020.
  - [34] P. J. Phillips, P. J. Flynn, T. Scruggs et al., “Overview of the face recognition grand challenge,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, pp. 947–954, San Diego, CA, USA, June 2005.
  - [35] C. Li, Y. Diao, H. Ma, and Y. Li, “A statistical PCA method for face recognition,” in *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application*, pp. 376–380, Shanghai, China, December 2008.
  - [36] J. Civit-Masot, F. Luna-Perejón, M. Domínguez Morales, and A. Civit, “Deep learning system for COVID-19 diagnosis aid using X-ray pulmonary images,” *Applied Sciences*, vol. 10, no. 13, p. 4640, 2020.
  - [37] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-

- 19 cases using deep neural networks with X-ray images,” *Computers in Biology and Medicine*, vol. 121, Article ID 103792, 2020.
- [38] A. K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, and A. Chakrabarti, “Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network,” *Pattern Analysis and Application*, vol. 24, 2020.
- [39] C. Dabakoglu, “Convolutional neural network,” 2020, <https://medium.com/@cdabakoglu/what-is-convolutional-neural-networkcnn-with-keras-cab447ad204c>.
- [40] J. Ricco, “Max-pooling/pooling,” 2020, [https://computer-science-wiki.org/index.php/Max-pooling/\\_Pooling](https://computer-science-wiki.org/index.php/Max-pooling/_Pooling).
- [41] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, “Deep learning based detection and analysis of COVID-19 on chest X-ray images,” *Applied Intelligence*, pp. 1–11, 2020.
- [42] G. Jain, D. Mittal, D. Thakur, and M. K. Mittal, “A deep learning approach to detect Covid-19 coronavirus with X-ray images,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 4, pp. 1391–1405, 2020.
- [43] M. Alazab, A. Awajan, A. Mesleh, A. Abraham, V. Jatana, and S. Alhyari, “COVID-19 prediction and detection using deep learning,” *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp. 168–181, 2020.
- [44] M. M. Rahaman, C. Li, Y. Yao et al., “Identification of COVID-19 samples from chest X-ray images using deep learning: a comparison of transfer learning approaches,” *Journal of X-Ray Science and Technology*, vol. 28, pp. 1–19, 2020.
- [45] B. Sekeroglu and I. Ozsahin, “Detection of COVID-19 from chest X-ray images using convolutional neural networks,” *SLAS Technology: Translating Life Sciences Innovation*, vol. 25, no. 6, pp. 553–565, 2020.
- [46] A. Mangal, S. Kalia, H. Rajgopal, K. Rangarajan, and V. Namboodiri, “CovidAID: COVID-19 detection using chest X-ray,” 2020, <https://arxiv.org/abs/2004.09803>.
- [47] P. Silva, E. Luz, G. Silva et al., “COVID-19 detection in CT images with deep learning: a voting-based scheme and cross-datasets analysis,” *Informatics in Medicine Unlocked*, vol. 20, Article ID 100427, 2020.
- [48] C. Zheng, X. Deng, Q. Fu et al., “Deep learning-based detection for COVID-19 from chest CT using weak label,” *MedRxiv*, 2020.
- [49] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, “Explainable COVID-19 detection using chest CT scans and deep learning,” 2020, <https://arxiv.org/abs/2011.05317>.
- [50] H. J. A. Adams, T. C. Kwee, D. Yakar, M. D. Hope, and R. M. Kwee, “Chest CT imaging signature of coronavirus disease 2019 infection: in pursuit of the scientific evidence,” *Chest*, vol. 158, no. 5, pp. 1885–1895, 2020.
- [51] K. Purohit, A. Kesarwani, D. R. Kisku, and M. Dalui, “Covid-19 detection on chest X-ray and CT scan images using multi-image augmented deep learning model,” *BioRxiv*, 2020.

## Research Article

# Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques

Muhammad Yaseen Khan <sup>1,2</sup>, Abdul Qayoom <sup>1</sup>, Muhammad Suffian Nizami <sup>3</sup>,  
Muhammad Shoaib Siddiqui <sup>4</sup>, Shaukat Wasi <sup>1</sup>,  
and Syed Muhammad Khaliq-ur-Rahman Raazi <sup>1</sup>

<sup>1</sup>Center for Language Computing, Mohammad Ali Jinnah University, Karachi 75400, Pakistan

<sup>2</sup>Research and Development, Love For Data, Karachi 75600, Pakistan

<sup>3</sup>Dept. of Pure and Applied Science, University of Urbino 'Carlo Bo', Urbino 61029, Italy

<sup>4</sup>Islamic University of Madinah, Madinah 42351, Saudi Arabia

Correspondence should be addressed to Muhammad Yaseen Khan; [yaseenkhan.yzai@gmail.com](mailto:yaseenkhan.yzai@gmail.com)

Received 25 June 2021; Accepted 14 August 2021; Published 13 September 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Muhammad Yaseen Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dictionaries not only are the source of getting meanings of the word but also serve the purpose of comprehending the context in which the words are used. For such purpose, we see a small sentence as an example for the very word in comprehensive book-dictionaries and more recently in online dictionaries. The lexicographers perform a very meticulous activity for the elicitation of Good Dictionary EXamples (GDEX)—a sentence that is best fit in a dictionary for the word's definition. The rules for the elicitation of GDEX are very strenuous and require a lot of time for committing the manual process. In this regard, this paper focuses on two major tasks, i.e., the development of labelled corpora for top 3K English words through the usage of distant supervision approach and devising a state-of-the-art artificial intelligence-based automated procedure for discriminating Good Dictionary EXamples from the bad ones. The proposed methodology involves a suite of five machine learning (ML) and five word embedding-based deep learning (DL) architectures. A thorough analysis of the results shows that GDEX elicitation can be done by both ML and DL models; however, DL-based models show a trivial improvement of 3.5% over the conventional ML models. We find that the random forests with parts-of-speech information and word2vec-based bidirectional LSTM are the most optimal ML and DL combinations for automated GDEX elicitation; on the test set, these models, respectively, secured a balanced accuracy of 73% and 77%.

## 1. Introduction

The comprehensive dictionary of any language provides the meaning of a word; at the same time, we find the correct usage of that word with an example of a sentence. Thus, when we can think of a *word*, a suite of multiple sentences can be set as examples to define it. All of these examples can be accurate w.r.t grammatical structure, the metaphor it delivers, and the context it is used into. In practice, with the corpus of these many (hundreds of thousands of) sentences

against a single word, the lexicographers, under the activity of considering Good Dictionary EXamples (GDEX), try to elicit one particular sentence which best defines the very word on the qualitative grounds of being typical, informative, and highly readable [1, 2]. There are certain rules that the lexicographers have to take care of during the elicitation process. On these rules, for example, Kilgarriff et al. [2] have maintained that a good sentence is *one*—in an adequate length of 10–25 words, *two*—comprised of words that are in the top 17,000, *three*—consisting of target collocation in the

main clause, *four*—not engaging pronouns and anaphors, *five*—provides a context, and *et cetera*. Overall, the activity is quite dawdling and sometimes it is converged into a compromising scenario when a good sentence is not good enough to be an example in the context of contemporary fashion. All of it eventually turns into a powerful need to substitute an automated GDEX elicitation process with artificial intelligence, which specifically deals with natural language processing (NLP) and natural language understanding (NLU).

The recent methods of automating such text classification tasks are based on supervised machine learning (ML) and Neural Network (NN) based deep learning (DL) techniques. These systems heavily rely on the prelabelled data, which mean, technically, a dataset that is labelled by humans. The accuracy of any such system is directly dependent on the size of data and quality of data labelling. However, recently, the researchers have produced abundant datasets for various classification tasks, but for the problem under study data is obscure, quite in deep relation to the fact that a lot of data is available over the Internet in the form of raw/unlabelled corpus; and if we aim to employ humans to do data labelling, a huge amount of time and labours efforts are required to complete it. In a parallel contrast, we have seen techniques such as distant supervision, which makes generalized assumptions for data labelling. For example instead of labelling a relation of Barack and Michelle marriage from the sentence “Michelle married Barack in 1992, and they have two daughters,” we consider every sentence for marriage-relation where the terms Obama and Michelle appear [3]. Similarly, for sentiment analysis of product reviews, we can have binary star ratings supplied to it (such as the reviews with 3 or above stars out of 5 are positive, otherwise negative [4]).

Thus, for automation of such manual procedure of GDEX, in this paper, we have contributed to

- (i) the development of a dataset using the distinct supervision technique for GDEX classification.
- (ii) the application of supervised ML and DL algorithms to predict whether, for a given word, a sentence in running English text is good or not.
- (iii) the comparative analysis on the robustness and trade-off between ML and DL approaches.
- (iv) the competitive analysis between manual GDEX elicitation routines and automated GDEX classification.

However, it does not mean that the proposed methodology explicitly examines the syntax and other linguistic elements of good writing, nor does it deal with the inference of polarity (under the computational study for effect) in the given text, which, in general convention, refers to the task of sentiment analysis. Instead, as prefatory research, it aims to verify whether a discriminative classifier can be sought for categorizing English sentences as either of the binary classes good and bad through the supervised ML algorithms.

This paper is systematically divided into 5 subsequent sections, where the related work done for the same problem

is given in Section 2. Section 3 provides details on the material and methods: data source, data labelling strategy, and approaches followed by maintaining information on ML and DL methods. The insights into the results, critique, and comparative and complete analyses on the results are presented in Section 4. The conclusion of the paper and future work are given at the end.

## 2. Literature Review

On the problem under study, there are many significant methodologies proposed by researchers; however, we maintain that, in comparison to other classification tasks in NLP, the amount of work for GDEX classification is small.

Pilán et al. [5] made the most relevant work for GDEX classification; they have developed a system to evaluate either the sentence suitability for dictionary examples or good examples for teaching purposes. They argue that a good example should be typical, informative, and intelligible and should be easily readable for the learners. The two techniques based on natural language processing and machine learning were used for sentence selection. The content has been taken from Swedish novels, newspapers, and blogs for applying both techniques. From this work, 70% of the total sentences were suitable for understanding by students and teachers. Srdanović and Kosem [6] presented GDEX classification for the Japanese language; it was designed mainly for the lexicography of the Japanese language and learning purposes. In this research, a randomly extracted list of lemmas was used for evaluating GDEX configurations.

Kilgarriff et al. [2] presented some rules and boundaries for a good sentence; according to the study, the sentence should hold the following characteristics (or comply with the following rules):

- (i) (Rule#1) A sentence consisting of 10 to 15 words will be preferred.
- (ii) (Rule#2) A sentence will be penalized when it does not lie among 17,000 commonest words in a language.
- (iii) (Rule#3) A sentence containing pronouns and anaphors will be penalized.
- (iv) (Rule#4) Target collocation should be in the main clause.
- (v) (Rule#5) A sentence should start with a capital letter and end with a full stop, exclamation sign, or question mark.

Moreover, for a GDEX, Kilgarriff et al. [2] eulogize the first two characteristics/features (sentence length and word frequency) should be given the highest weight as compared to other features. According to Kosem et al. [7], the most important characteristics of a GDEX are authenticity, typicality, informativeness, and intelligibility. The developers of Good Dictionary EXamples system and their configurations are often lexicographers and lack programming skills in many cases.

Geyken et al. [8] show that GDEX work can be extended through ML techniques for mapping example sentences to

dictionary sense. They performed the computations of all collocations sets and then maximum entropy [9] was used for learning the correct mapping between corpus sentences and their correct dictionary sense. Ljubešić and Peronja [10] presented another ML approach to extract GDEX. The dataset used in their experiment contains several examples of sentences with annotations of four classes/levels (i.e., very bad, bad, good, and very good). They used the Random Forest regressor algorithm [11] and secured an average precision of 90%.

Stanković et al. [12] gave a similar work for the selection of GDEX for Serbian and it was used for the development of preliminary components of the model. Their approach analyses the lexical and syntactic aspects of a corpus consisting of five digitized volumes of examples from the Serbian Academy of Sciences and Arts (SASA) dictionary. They compared the feature distribution of examples from their corpora with the feature distribution of sentence samples extracted from corpora comprising various other texts. This way, selected candidate 140 examples were represented as feature vectors, and supervised machine learning classifiers were used for standard and nonstandard Serbian sentences.

Koppel [13] presented work for GDEX classification in the Estonian language. The group used the web corpus of etTenTen13; in their approach, they focus on the sentence length, word length, the number of subordinate clauses, and keyword position. In another similar study, Uprety and Shakya [14] conducted a test to analyse the effectiveness of context clue sentences among Nepalese students. Their study results showed that context clue sentences were more useful in learning vocabulary words than GDEX sentences. Based on their research results they concluded and recommended that context clue sentences should be included in the Good Dictionary EXamples to help the new learners.

### 3. Materials and Methods

This section is divided into three subsections; each one is dealing with the focused methodology such as data gathering and labelling (Subsection 3.1), preprocessing and feature selection (Subsection 3.2), and an overview about experiment setup employing the suit of predictive algorithms for machine learning (Subsection 3.3).

**3.1. Data Source and Data Labelling.** We prepared our dataset in the fashion of distant supervision. Using BeautifulSoup (<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>) we scraped sentences from the website [www.yourdictionary.com](http://www.yourdictionary.com) (YD.com). The scraping is made for the top 3K English words listed by Oxford Learner's Dictionaries (<https://www.oxfordlearnersdictionaries.com/wordlist> (accessed May 20, 2021)). On average we have got  $\approx 250$  sentences for a single word and more than 785K English sentences in total. Furthermore, the website not only provides the example sentences, but it also presents the count of thumbs-up and thumbs-down for every sentence against the very word. Hence we maintain the corpus in dictionary structure

where, for every word as a key, a list of tuples is retained. To mean it mathematically, consider equation (1) below:

$$C \leftarrow \left\{ \begin{array}{l} w_0 \longrightarrow [ \langle S_0^{w_0}, U_0, D_0 \rangle, \langle S_1^{w_0}, U_1, D_1 \rangle, \dots ], \\ w_1 \longrightarrow [ \langle S_0^{w_1}, U_0, D_0 \rangle, \langle S_1^{w_1}, U_1, D_1 \rangle, \dots ], \\ \vdots \\ w_n \longrightarrow [ \langle S_0^{w_n}, U_0, D_0 \rangle, \langle S_1^{w_n}, U_1, D_1 \rangle, \dots ], \end{array} \right\} \quad (1)$$

where  $C$  is a dictionary with key-value pairs such as word  $w$  being the key, against whom a list of tuples is retained; further, the contents of the tuple shows the example sentence  $S_i^w$  along with its thumbs-up votes ( $U_i$ ) and thumbs-down votes ( $D_i$ ); the subscript  $i$  indicates the index of sentence respectively. The target label of a sentence, i.e., *good* or *bad* (or 1 and -1 in respective order), is determined by the count of thumbs-up and thumbs-down votes. In further analysis, we notice that YD.com holds different votes for the same sentence if the very sentence is referenced as an example to the different words. Hence,  $C$  is of no use if there exist redundant sentences with different votes. To restructure the dataset we extract a set of distinct sentences  $S^*$  from  $C$  as per the following equation:

$$S^* \leftarrow \{ C_{[w][i][0]} | w \in C; \quad \forall i \in I = \{0, 1, \dots, \|C_{[w]}\| - 1\} \}. \quad (2)$$

Further, we prepared different datasets—corresponding to the pooling function  $\Psi(\cdot)$ —having sentences and their labels in the form of tuples with the manner shown in the following equation:

$$\widehat{C} \leftarrow \{ (s_0, \lambda_0), (s_1, \lambda_1), \dots, (s_n, \lambda_n) \} \quad (3)$$

where  $s_i \in S^*$  and  $\lambda_i$  is the label of respective  $i^{\text{th}}$  sentence in  $\widehat{C}$  and determined on the criteria under function  $\Phi$  given in the following equation:

$$\lambda_i \leftarrow \Phi(a_1, a_2) \leftarrow \begin{cases} \text{none,} & \text{if } a_1 = a_2 = 0, \\ \text{good,} & \text{if } a_1 \geq a_2, \\ \text{bad,} & \text{otherwise.} \end{cases} \quad (4)$$

In equation (4),  $a_j$  is a real number yielded through a pooling function  $\Psi_p(\cdot)$  (explained later in the following text). In  $\Phi(a_1, a_2)$ ; subscript  $j$  for  $a_j$  specifically indicates the incidences for  $U$  and  $D$ ; hence,  $j \in \{1, 2\}$ , to mean thumbs-up votes  $a_j = 1$  and thumbs-down votes  $a_j = 2$ . Lastly, the value of  $a_j$  is calculated as per the following equation:

$$a_j \leftarrow \Psi_f \left( \{ C_{[w][i][j]} | s \in S^* \wedge w \in C \wedge s = C_{[w][i][0]}; \forall i \in I \} \right), \quad (5)$$

where  $p$  is a final score calculating function,  $f \in \{\max, \text{average}, \text{sum}\}$ . The index set  $I$  in the equation above has already been defined in equation (2). Thus, we utilized these votes as the crowdsourced labelling and adjudged a sentence to be good if the total thumbs-up votes are equal to or greater than thumbs-down votes (see equation (4)).

Table 1 gives the statistical information on the labelled dataset that is employed in this experiment. The dataset for every scoring function is balanced, i.e., each class contains 20K records (which alternatively means 40K sentences, in total, are used in the experiments.) One key observation we can get from the table is the average sentence length of good examples is approximately half of its counterclass. It further asserts that the distinct supervision (or nearly crowdsourced data) appeared to have aligned with rule#1 (i.e., already stated in Subsection 2.2).

*3.2. Machine Learning-Based Classification: Feature Enhancement, Transformation, and Algorithms.* At the beginning of this section, the authors would like to maintain a summarized idea of experiments conducted for the GDEX classification based on the conventional ML algorithms; in the same context, the following itemized text provides a brief commentary on the components depicted in Figure 1.

- (i) We experimented with two different approaches for the feature enhancement, such as the following:
  - (1) Bag of Word (BoW).
  - (2) Usage of Part of Speech (PoS) tags alongside the words.
- (ii) Besides the above two approaches, we set two feature transformation (or vectorization) techniques for the sentences in the dataset, such as the following:
  - (1) Word frequency-based count vectorization.
  - (2) Term Frequency-Inverse Document Frequency (TF \* IDF) features normalization-based vectorization.
- (iii) The combination of these feature enhancement approaches and feature vectorization techniques are evaluated under the five conventional ML algorithms in the 10 randomly generated training and testing subsets under the Monte Carlo method. The ML algorithms used in this paper are enumerated below:
  - (1)  $k$ -nearest neighbours ( $k$ -NN).
  - (2) Naïve Bayes (NB)/Gaussian Naïve Bayes (GNB).
  - (3) Random Forest Trees (RFT).
  - (4) Linear Support Vector Machines (linear-SVM).
  - (5) SVM with radial basis function kernel (rbf-SVM).

Thus, the total number of experiments set for ML-based GDEX classification is 60, i.e., 2 (feature enhancement approaches)  $\times$  2 (feature vectorization)  $\times$  5 (ML algorithms)  $\times$  3 (datasets yield from the three different final scoring functions) = 60. The details of each of these components are provided in the subsequent subsections.

*3.2.1. Feature Enhancement Approaches.* The BoW approach is considered a very basic approach in any task in NLP [15]. It consists of tokenization of a running text/

document and submission of tokens for further process. However, we can think that these sequences of words are of more importance and become meaningful and informative when they are analysed with the corresponding PoS tags. Thus, hundreds of papers in the domain of NLP and NLU utilized such information of words' PoS alongside words in their capacities [16, 17]. In the same regard, we can anticipate the words in addition to respective PoS tag information (BoW + PoS) will attain more robustness in the predictive ML model with two significant hypotheses:

- (i) BoW + PoS creates highly discriminative features for classifying a GDEX.
- (ii) Forbye the previous point, BoW + PoS embodies a writing pattern that exists for a comparatively longer sequence in  $n$ -grams—we surmise that it may engage better syntactic and semantic attributes.

On a technical note, we have used Natural Language ToolKit (NLTK) based word tokenizer (<https://www.nltk.org/api/nltk.tokenize.html>; there are many tokenizers provided in the module; function, which is precisely used in this paper, is, namely, `word_tokenize`) for the sentence tokenization; followed by it, the PoS tagging is also done with NLTK-based PoS tagging module (<https://www.nltk.org/api/nltk.tag.html#module-nltk.tag>). We concatenated the word and its respective PoS tag with an underscore, as it is shown for a single sentence in Table 2; however, the information on the tag-set can be accessed in the online documentation of NLTK (<https://www.nltk.org/book/ch05.html>).

*3.2.2. Feature Vectorization Techniques.* ML algorithms are not supposed to work directly on the running texts. Since there are thousands of terms in the vocabulary and a few of them are appearing in a sentence, we are required to transform every sentence through a specific mechanism that applies to all of the sentences in the dataset and is hence workable for the ML algorithms. Typically, the sentence transformation mechanism takes a sentence and projects it into a high-dimensional vector space [15]. The final structure of the dataset will be a matrix. It contains the number of rows equal to the number of sentences and the number of columns as per the size of vocabulary (or in other words, the dimensionality of a single vector is equal to the size of vocabulary). Thus, we can think of the values on the dimensions, corresponding to the words present in the sentence and carry nonzero numeric values; otherwise, they are zero (in the case of nonsparsity). The very matrix can be sparse by ignoring the indexing of words/dimensions that are not present in the sentence and retaining the records for the words that appear in the sentence.

Count vectorization, which is the first vectorization technique used in this paper, involves sentence vectorization by keeping the count of words that appear in the sentence and zero at the remaining dimensions. Figure 2 illustrates the count vectorization process, in which the first step includes generating a dictionary for word-indices, followed by utilizing very dictionary for the transformation of sentences in vector space.

TABLE 1: Class distribution and statistical insights into the labelled dataset.

Class	Final scoring function $\Psi_f$	Count of tokens	Count of distinct terms	Average words in sentence
Good (1)	$f = \text{avg}$	302,389	22,200	15.12 $\approx$ 15
Bad (-1)		536,604	40,232	26.83 $\approx$ 27
Total		838,993	47,028	20.97 $\approx$ 21
Good (1)	$f = \text{sum}$	287,794	26,228	14.39 $\approx$ 14
Bad (-1)		627,952	57,686	31.4 $\approx$ 31
Total		915,746	64,840	22.89 $\approx$ 23
Good (1)	$f = \text{max}$	279,894	25,738	13.99 $\approx$ 14
Bad (-1)		610,676	57,604	30.53 $\approx$ 31
Total		890,570	64,728	22.26 $\approx$ 22

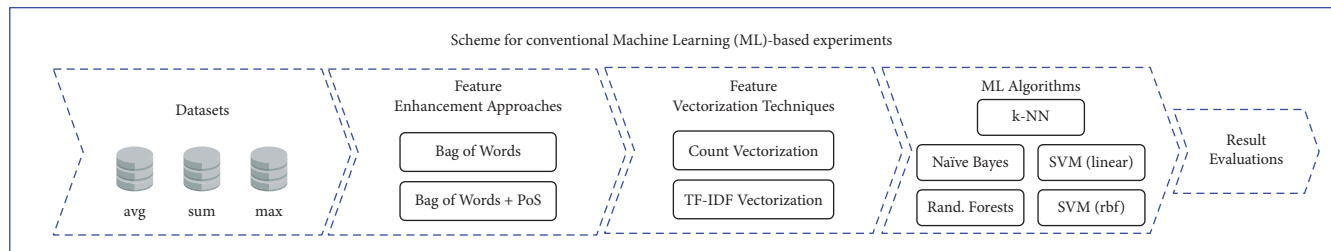


FIGURE 1: Overall scheme of experiments designed for conventional machine learning.

TABLE 2: Example for the simple word tokenization and PoS tag induction.

Example	Laura always remained an object of curious study.
BoW features	['Laura', 'always', 'remained', 'an', 'object', 'of', 'curious', 'study', '.']
BoW + PoS features	['Laura_NNP', 'always_RB', 'remained_VBD', 'an_DT', 'object_NN', 'of_IN', 'curious_JJ', 'study_NN', '.']

We may think of the cases where the most frequent words (i.e., a, an, the, of, to, *et cetera*, known as stop words) dominate in a sentence—diminishing the impact on least frequent words—hence, resulting in a larger value on their respective dimensions. In this regard, the TF \* IDF approach sets a trade-off between the high-frequent and less-frequent words [15, 18]. It works by calculating a product of term frequency relative to a document (TF) and inverse document frequency of the very term in the corpus (IDF). To mean the TF and IDF mathematically, consider the following equations; moreover, Figure 3 uses these formulae to illustrate TF \* IDF calculations.

$$\begin{aligned}
 \text{TF}(t, d) &= \frac{\text{frequency of term } t \text{ in document } d}{\text{total number of terms in the document}} \\
 &= \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \\
 \text{IDF}(t, D) &= \log \frac{\text{total number of documents in the dataset}}{\text{number of documents containing the term } t} \\
 &= \log \frac{|D|}{|\{d \in D: t' \in d\}|}.
 \end{aligned} \tag{6}$$

On a technical note, we have used the  $n$ -gram range [1, 3] in sklearn, which assumes the formation of unigrams, bigrams, and trigrams in the input string. Alongside it, we kept the same tokenization function for both of the vectorization processes, which has already been discussed in the previous subsection.

**3.2.3. Machine Learning Algorithms.** *K-Nearest Neighbours* is among the instance-based lazy learning techniques in conventional ML algorithms [19, 20]. Functionally, it computes the distance between the target document vector and all of the document vectors, followed by selecting  $k$  documents where the distance is minimum. In last, it decides the class for the target document through voting in the  $k$ -nearest neighbour vectors. The number of neighbours set for this work is five (i.e.,  $k = 5$ ). Furthermore, we like to maintain that there are many measures for computing distances between documents, and the one we have employed in this paper is cosine similarity. Since similarity is inversely proportional to the distance, with the case of similarity, the  $k$ -NN algorithm will perform voting on the  $k$  documents with the maximum similarity. The value of the cosine similarity ranges in [0, 1], where the similarity score 0 indicates no similarity whereas 1 indicates absolute similarity. The cosine similarity between two document vectors ( $A$  and  $B$ ) is calculated through the following equation [15]:

$$\text{Similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \tag{7}$$

*Naïve Bayes* is a conventional ML algorithm for classification tasks [4, 15]. It classifies the sentences by exploiting conditional probability using Bayes' theorem; however, the basic assumptions naïve Bayes holds are of the conditional independence between the features. The basic calculation



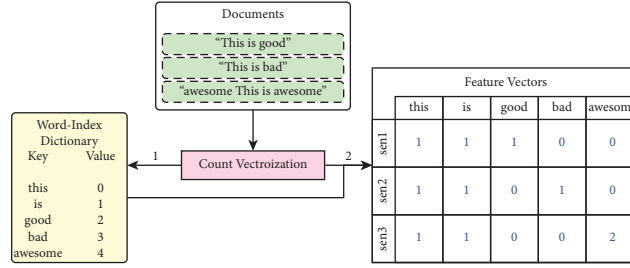


FIGURE 2: Illustration of count vectorization.

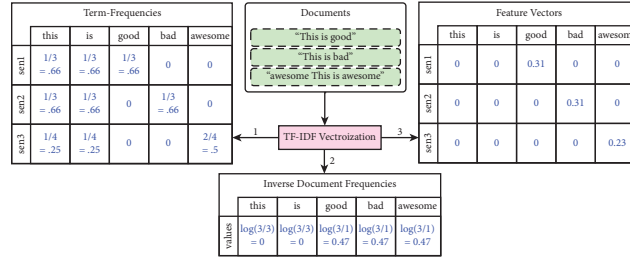


FIGURE 3: Illustration of TF \* IDF vectorization.

done by the naïve Bayes for classifying a sentence ( $X$ ) is given in the following equation:

$$f(C_k|X) = \frac{p(X|C_k)p(C_k)}{p(X)}. \quad (8)$$

Equation (7) is expanded w.r.t the individual features ( $X = \{x_0, x_1, \dots, x_n\}$ ); see equation (9) below:

$$\begin{aligned} f(C_k|x_0, x_1, \dots, x_n) &= p(C_k)p(x_0|C_k)p(x_1|C_k) \cdots p(x_n|C_k) \\ &\approx p(C_k) \prod_{i=0}^n p(x_i|C_k). \end{aligned} \quad (9)$$

However, when the documents are normalized and transformed through the TF \* IDF vectorization, the values for features are no longer discrete. Thus, for the continuous features, we cannot employ the above conventional naïve Bayes algorithm. Instead, we have to use its variant that uses Gaussian distribution (and hence, known as Gaussian naïve Bayes) [21, 22]; the substitution of  $(x_0|C_k)$  in the Gaussian naïve Bayes is defined in the following equation:

$$p(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-((v-\mu_k)^2/2\sigma_k^2)}. \quad (10)$$

Finally, the target class  $\hat{y}$  (by either of conventional naïve Bayes or Gaussian naïve Bayes) is elicited where  $(.)$  is maximum; to mean it mathematically, see equation (11), where  $K$  is the set of classes:

$$\hat{y} = \arg \max_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i|C_k). \quad (11)$$

*Random Forest* is an ensemble approach in ML classification algorithms, which is based on Decision Trees (DT) [23]. Instead of relying on a single decision tree, the basic aim is to draw multiple decision trees from the bootstrapped-random samples of training data. The testing data will be predicted on each of the DT, followed by eliciting the final label through voting [11]. Thus, we can think of RF overcoming the issue of overfitting through ensemble technique. Figure 4 shows how the RF classifier works and outputs a final class from all of the DTs. In the experiment, we have used 200 trees (or DT estimators) for building a forest.

*Support Vector Machine* is one of the widely employed classifiers in conventional ML algorithms [24]. It is well suited for the classification of complex, imbalanced ones but should be small or medium-sized datasets. The SVM aims to draw a hyperplane in an  $n$ -dimensional vector space, such that the hyperplane separates data points into two distinct partitions of data, representing the respective classes [25]. The SVM can be used for linear or nonlinear classification. However, the basic SVM, which fits a hyperplane, is conventionally known as linear-SVM [25, 26]. Equation (12) gives the mathematical semantics for understanding linear-SVM.

$$\hat{y}_i = \begin{cases} 1, & \text{if } w^T \cdot X_i - b \geq 1, \\ -1, & \text{if } w^T \cdot X_i - b < 1. \end{cases} \quad (12)$$

In this work, we have used linear-SVM and radial basis function (rbf) SVM (through kernel trick). The basic objective rbf-SVM sets are to fit a circular boundary margin for nonlinear datasets. The illustration in Figure 5(a) shows the linear-SVM, and in contrast, Figure 5(b) shows the situation where a hyperplane is not suitable for separating datasets into two distinct parts; and instead, this can be achievable

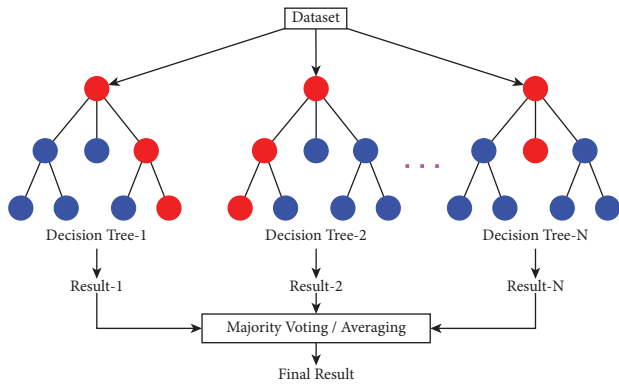


FIGURE 4: Illustration of random forest trees.

only with the rbf kernel trick. The red and blue dots are representing separate classes. The solid line in grey colour is the decision boundary; dots coinciding with the dashed line are termed as support vectors.

**3.3. Feature Engineering for Deep Learning-Based Classification.** In this section, a discussion on the DL models and input data encoding schemes are given in detail. Likewise, in an earlier summary methodology involving ML in Subsection 3.3, the authors would like to maintain a brief commentary on DL-based models; Figure 6 shows the overall scheme for these experiments.

- (i) Since we empirically found, in the suite of ML-based algorithms, the most optimal result was secured with the dataset based on a final scoring function  $\Psi_{f=sum}$ , all of the DL-based experiments are performed only on the aforesaid dataset.
- (ii) Since the NN essentially requires input data to be encoded in a numeric form, for doing the needful, we used 3 different data encoding approaches, which are as follows:
  - (1) One-hot encoding.
  - (2) Global Vectors (GloVe) based embedded encoding.
  - (3) word2vec-based embedded encoding.
- (iii) A combination of these data encoding approaches was made with the following 5 DL algorithms/networks:
  - (1) Recurrent Neural Network (RNN).
  - (2) Gated Recurrent Unit (GRU).
  - (3) Long-Short Term Memory (LSTM).
  - (4) Bidirectional GRU (Bi-GRU).
  - (5) Bidirectional LSTM (Bi-LSTM).
- (iv) We did not perform any feature engineering (for example, extraction and usage of PoS tags) in the DL-based experiments, because, congenitally, the NNs are taken as to learn and accommodate and intrinsic features present in the data.
- (v) All of the DNNs used in the methodology are programmed with the Python-based Keras (<https://>

[keras.io](https://keras.io)) library, which uses Tensor Flow 2 (<https://www.tensorflow.org>) at the backend for processing. Moreover, the experiments are run at Google Colaboratory (<https://colab.research.google.com>) on the GPU-accelerated runtime.

Thus, the total number of experiments done with DL-based methods is 15, i.e., 3 (data encoding approaches)  $\times$  5 (DNNs) = 15. The detail of these components is given in the subsequent subsections.

**3.3.1. Data Encoding Approaches.** The NNs need data to be in numeric form for which we have got many transformation or encoding approaches. One-hot encoding is one of the techniques among them, which generates a single vector against every word in a sentence, such that the index corresponding to the very word is 1 and the rest of all incidences are 0. Thus, we can see a sparse matrix-like structure (or a list of four one-hot vectors) for the sentence “This is a cat” as is illustrated in Figure 7(a). Each row of the yellow block is a vector where there exists only a single entry of 1, indicating the presence of the very word in the vector. Hence, with this technique, we can think that input data is sparse and exists in a very high-dimensional space.

In contrast, the second approach for data encoding is based on NN inspired word embeddings and statistical means, which are dense and adjustable to any of the  $n$ -dimensional spaces, provided that  $n > 0$ ; Figure 7(b) illustrates the example of word embedding where each row in blue colour is a dense representation of the word in 4-D space. The word embeddings render meanings to Firth’s philosophy “You shall know a word by the company it keeps!” [27] through realizing the capability of retaining the context of words, such that every word will exist alongside the similar words (using GloVe, the examples of the nearest words for the word “king” are “kings,” “queen,” “monarch,” *et cetera*; retrieved through online tool available at [http://bionlp-www.utu.fi/wv\\_demo](http://bionlp-www.utu.fi/wv_demo)) in the  $n$ -dimensional space of word embeddings. In this work, we have employed two different word embeddings, namely, word2vec [28] and GloVe [29], developed by Google and Stanford, respectively. In addition to this, the word2vec employs continuous BoW in NN for learning the prediction of the current word (given the input of context of words) and skip-grams for learning the similar words (given a source/input word), whereas the GloVe utilizes matrix factorization techniques such as Latent Semantic Analysis (LSA) [30] on word-word context matrix for generating word vector representations. On a technical note, the representation used in this work is based on 300 dimensions (these vectors can be accessed at <http://vectors.nlp.leu/repository>).

**3.3.2. Deep Neural Networks.** The NNs are the computational system of connected units that loosely simulate the working of biological neurons in the brain of living beings. The story of ideas and advancements made in the file of NN is historic. (The earlier NNs are devised by McCulloch and Walter [45], in 1943, for artificially simulating the working of a biological neuron

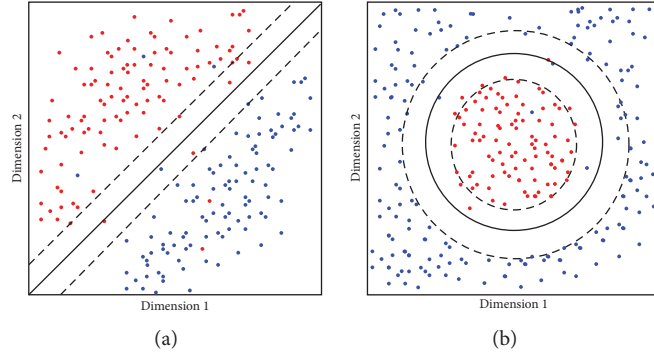


FIGURE 5: Illustration of linearity and nonlinearity in data. (a) The linear-SVM corresponds to fitting a hyperplane when the data reflects its shape as depicted in the subfigure a, (b) whereas the subfigure b reflects the situation optimal for rbf-based SVM.

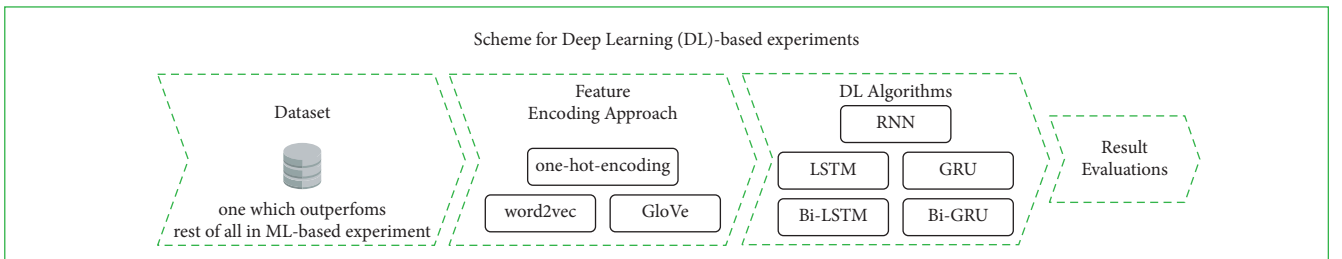


FIGURE 6: Overall scheme of experiments designed for deep learning.

[39, 46]. This earlier work is rendered with computational means known as “Calculators” and “Perceptron,” respectively, in 1954 by Farley and Clark [47] and in 1958 by Rosenblatt [48]; however these works were limited to present the working of single neuron [39, 44]. Upgrading NN with several layers (thus, called DNN) was made in 1965 by Ivakhnenko and Lapa [49]. In 1975, Werbos [50] presented that the backpropagation technique can be used for new weights learning for the training of multilayer networks [46]; the further work done by Rumelhart et al. [51] showed that the backpropagation techniques learn interesting features for text processing.) However, the authors would like to maintain a brief introduction to the basic working of these connected units or a NN (which is also illustrated in Figure 8), where inputs (or signals) received at the input layer are analysed and transmitted to further neurons to which they connected. We know the input should be a numeric value (for which we have maintained information in the previous section); thus, the input ( $X = \{x_0, x_1, \dots, x_n\}$ ) received at the units of the hidden layer and the respective weights ( $W = \{w_0, w_1, \dots, w_n\}$ ) that are correspondingly associated with the edges are taken for the dot product ( $\Sigma = X \cdot W$ )— creating a linear output. In the next step, bias ( $b$ ) is added to this linear output ( $z = X \cdot W + b$ ), and the result is converted into nonlinearity through passing it to a nonlinear activation function, that is, in our case,  $\tanh$  function, which is given in equation (13).

$$\tanh(z) = f_h(z) = \frac{e^{2z} - 1}{e^{2z} + 1}. \quad (13)$$

Similarly, the output of hidden neurons is transmitted to the final output neuron that takes a step function to compute

the class of given input data. The step function, which is used in this paper is sigmoid that returns a number in the range of  $[0, 1]$ , where we consider the prediction is relating to the positive class if the value is above 0.5; otherwise it belongs to the negative class. The sigmoid function is given in the following equation:

$$\hat{y}_i = \sigma(z) = \frac{1}{1 + e^{-z}}. \quad (14)$$

The backpropagation technique is used to update weights considering the prediction errors that occurred during the training. In this context, DNN typically divides training set into multiple batches; thus, with one batch it calculates the error followed by updating the new values for the weights. Executing the same process on each batch will mark one run, which is technically called an epoch.

In this paper, we have used three types of NN that were specifically developed for text (or generally known for sequence) processing. The RNN [31] is one of the first DNNs that attempted to involve input history in the sequential data such that the process of RNN moves onwards with subsequent inputs alongside incorporating the result (of the hidden state) of the previous input units.

W.r.t Figure 9, the RNN works for every timestamp  $t$ , and the hidden state  $a^{(t)}$  and the output  $\hat{y}^{(t)}$  are expressed as per equations (15) and (16).

$$a^{(t)} = g_1(W_{aa}a^{(t-1)} + W_{ax}x^{(t)} + b_a), \quad (15)$$

$$\hat{y}^{(t)} = g_2(W_{ya}a^{(t)} + b_y), \quad (16)$$

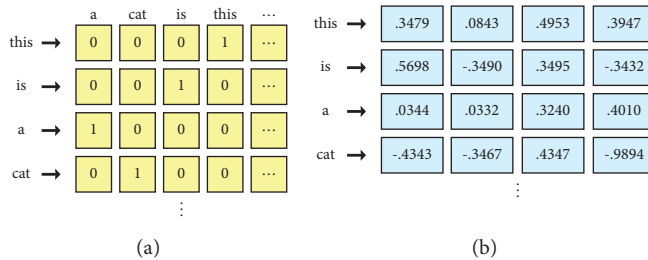


FIGURE 7: Illustration of the one-hot encoding and word embeddings. (a) One-hot encoding, and (b) word embedding.

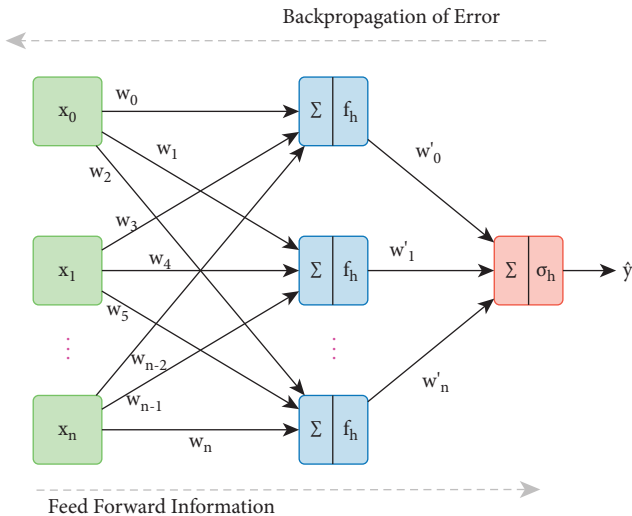


FIGURE 8: Illustration of the artificial neural network.

where  $W_{aa}, W_{ax}, W_{ya}, b_a,$  and  $b_y$  are the coefficients, and  $f$  is the activation function; comprehension of these coefficients and the internal structure of the blue box (illustrated in Figure 9) are given in Figure 10.

The RNNs though were developed to retain memory but instead, they failed on doing it for the longer sequences. Alternatively, Hochreiter and Schmidhuber [33] presented another RNN-based architecture, namely, LSTM, which served better for the problem of input retaining. The LSTM introduced the concept of the gate for remembering the inputs; however, later an upgraded form of LSTM is presented by Gers et al. [34], which added forget gate in the architecture; further, with the induction of forgetting gate LSTM became capable of resetting its state [35]. The LSTM though is the wonderful RNN architecture but it takes more memory and processing time [36, 37]. Cho et al. [38] introduced GRU, which is alike LSTM but contains fewer parameters. Traditional RNNs suffer the vanishing gradient problem, which is handled at the optimal level in LSTM and GRU [32, 33, 39]. The bidirectional LSTM and GRU are the variant of vanilla LSTM and GRU, which are capable of making the DNN process string in forward and backward directions [39]. In Table 3 the summary of gates used in LSTM and GRU is presented, in addition to which we can see their usage in the illustrations of LSTM and GRU in Table 4, where  $\odot$  shows elementwise multiplication between two vectors.

The networks we have employed in this paper have the same input and output layer.

However, the hidden layer varies w.r.t the architecture. This DNNs are programmed with Keras using the sequential model. Information on the layers hyperparameters used in this work is given in Tables 5 and 6.

### 4. Results and Discussion

In this section, we presented a thorough discussion on the evaluation and comparisons of the ML and DL models. However, before proceeding any further, it should be in the knowledge that the evaluation is done on a validation set which is extracted from the labelled corpus with Pareto principle or 80/20 rule [15, 40]. These details are maintained in separate subsequent subsections.

**4.1. Evaluation Criteria and Metrics.** The classification task in a supervised learning domain is often evaluated through the confusion matrix (CM), which statistically presents the number of correct and incorrect predictions w.r.t the actual labels in the validation set. A sample CM is given in Table 7, where TNs are the *true negatives*, which logically means the number of actually negative documents and predicted negative as well; TP (*true positives*) will mean exactly the opposite to TN (i.e., consider a positive class in place of negative). The FP is the *false positives*, which logically means the number of documents that are actually negative but misclassified as positives; FNs (*false negatives*) are the exact opposite of FP, such that they were the misclassified documents that were actually negative but falsely predicted as negative.

We can drive several evaluation statistics for assessing the quality of the Predictive System (PS) using the CM. The statistics and their derivations used for the assessment of ML and DL models in this work are defined in Table 8. Moreover, for the ideal PS, we expect to have the highest value on the left diagonal of every individual CM, whereas, at the right diagonal of the matrix, we expect the least value.

For the evaluation of performance in this paper, we consider  $R$  and  $BA$  are of more importance. The  $R$  is critical because we consider losing or misclassifying a positive document into another category is perilous—as we have got little data for the GDEX classification in comparison to the colossal dataset—thus, we will consider an ML or DL model with an optimal where the  $R$  is higher. In a similar context, this will not mean the small value of  $S$ ; hence, the  $BA$  is the

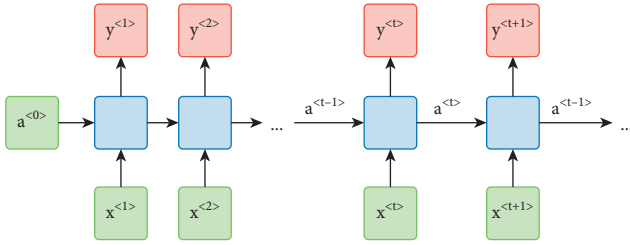


FIGURE 9: Illustration of the RNN. Image courtesy [32].

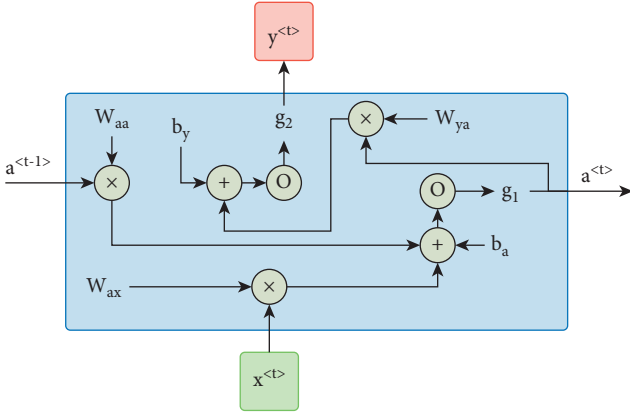


FIGURE 10: Depiction of the internal architecture of RNN. Image courtesy [32].

ultimate choice for the fair evaluation, which encompasses both of the statistics relating to TP and TN.

**4.2. Analysis of ML Models and Results.** The quantified statistics of all evaluation metrics are given (respectively, for the final scoring functions, i.e.,  $\Psi_{f=avg}$ ,  $\Psi_{f=max}$ ,  $\Psi_{f=sum}$ ) in Tables 9–11. The overall result of the ML-based models is positive. We can see an obvious insight into the better performance of all ML models (in all respective datasets corresponding to the final scoring functions) that are vectorized through the TF \* IDF approach. On the collective ground, the dataset created with  $\Psi_{f=sum}$ , parse, indicates the most optimal method for the dataset creation through distant supervision. In contrast, the results with the  $\Psi_{f=avg}$  show the least significance for making the discriminant dataset for predictive modelling; hence, we can maintain that the distant supervision cannot be used with the averaging methods for data curation in supervised learning tasks.

Since the dataset with  $\Psi_{f=sum}$  shows better results, we will consider it (considering Table 11) for the discussion in the remaining text. Coming towards the evaluation of feature enhancement technique, we see the BoW + PoS tags show better results in comparison to the only BoW approach. However, a drastic change in accuracy of  $k$ -NN (i.e., w.r.t  $\Psi_{f=sum}$ , improvement of +12% with count vectorization and +2% with the TF \* IDF vectorization) is seen when the PoS information is inducted alongside the simple words. However, in comparison to the count vectorization technique, we maintain that the improvement with the

additional PoS information is slightly more visible in the TF \* IDF vectorization technique.

The most optimal ML algorithm and combination found with maximum accuracy of 77.3% are rbf-SVM + TBP. (TBP will be the acronym for the combination of TF \* IDF vectorization + BoW + PoS tags features. Similarly, CBP will be combination of count vectorization + BoW + PoS tags features. TB will stand for the combination of TF \* IDF + BoW features; and CB will be count vectorization + BoW features.) Ignoring the trivial difference of linear-SVM with its other variant, we can consider RFT + TBP secures the second position by attaining accuracy of 76.8%. For BA,  $k$ -NN + TBP is found the best combination with a 75.5% score, followed by RFT + TB with securing a 73.9% score. Besides accuracy and balanced accuracy, the highest recall (i.e., 75.4%) is seen in a dataset with  $\Psi_{f=max}$  with RFT + CB and linear-SVM + TB. Forbye it, we see that the  $R$  is high with SVM everywhere.

Figure 11 shows the improvement of the BoW + PoS approach on the conventional BoW approach. The sub-figures in the top row indicate improvement w.r.t count vectorization, and in contrast, the bottom row carries information on the TF \* IDF vectorization. The overall observation on the improvement gives a piece of mixed information except for the TF \* IDF features on an average dataset, where the positive trend of improvement is steady. However, the least improvement, i.e.,  $\approx 0.8\%$  on an average basis, is seen for the same dataset. In the same context, on average the maximum pointer of improvement (i.e.,  $\approx 3\%$ ) is found with the dataset with  $\Psi_{f=sum}$ .

Figure 12 shows the CM of all conventional ML algorithms, separated w.r.t feature enhancement and vectorization techniques. However, instead of multiplying the figure space three times for each of the datasets with respective final scoring functions, we have presented the aggregated-normalized CM. The colour bar on the right of Figure 13 is set to serve a specific purpose such that the maximum value is 0.5 ( $\approx 50\%$ ) which corresponds to the size of data in one class.

We maintain that the SVM + TBP with its both linear and rbf variants is the most optimal algorithm among all. This is so because linear-SVM achieved  $TN + TP = 0.35 + 0.4 \approx 0.75 \equiv 75\%$  accuracy; however, the other variant, rbf-SVM, stood second. The authors would like to maintain the performance of the RFT + CBP;  $0.34 + .41 \approx .75$  also similar to the previously mentioned linear- and rbf-SVM. Forbye it, we must maintain that the competition between the SVM + CBP and RFT + CBP is near equal, but the RFT + CBP is found champion such that it has got minimum value on right diagonal (i.e.,  $FP + FN = 0.16 + 0.09 \approx 0.21$ ), and in a similar context, it has got the least FN which, per se, is an additive advantage.

**4.3. Analysis of DL Models and Results.** The NN-based DL models are used on the dataset with the scoring function  $\Psi_{f=sum}$ , as it has produced the most optimal result in comparison to the remaining two scoring functions.

TABLE 3: Summary of gates involved in LSTM and GRU architecture.

Type of gate	Description	Utilized in
Update gate ( $\Gamma_u$ )	How much of the past should be remembered now?	LSTM, GRU
Relevance gate ( $\Gamma_r$ )	Drop previous information	LSTM, GRU
Forget gate ( $\Gamma_f$ )	Erase cell or not?	LSTM
Output gate ( $\Gamma_o$ )	How much information of cell should be revealed?	LSTM

TABLE 4: Architecture and variables' information the LSTM and GRU.

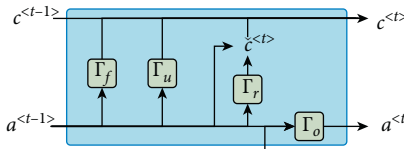
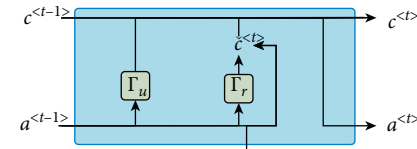
Variables	LSTM	GRU
Illustration		
$\zeta^{(t)}$	$\tanh(W_c[\Gamma_r \odot a^{(t-1)}, x^{(t)}] + b_c)$	$\tanh(W_c[\Gamma_r \odot a^{(t-1)}, x^{(t)}] + b_c)$
$c^{(t)}$	$\Gamma_u \odot \zeta^{(t)} + \Gamma_f \odot c^{(t-1)}$	$\Gamma_u \odot \zeta^{(t)} + (1 - \Gamma_u) \odot c^{(t-1)}$
$a^{(t)}$	$\Gamma_o \odot c^{(t)}$	$c^{(t)}$

TABLE 5: Summary of DNN used in this paper.

Layer	Input unit	Output units	Activation
Input	Max # of words	32	—
Hidden	32	32	tanh
Dense/output	32	1	Sigmoid

TABLE 6: Configuration of DNN.

Model settings	Values
Optimizer	Adam
Loss	Binary cross-entropy
Epochs	10
Batch size	128

TABLE 7: A sample confusion matrix.

		Predicted	
		$N$	$P$
Actual	$N$	TN	FP
	$P$	FN	TP

Table 12 shows the metrics for the validation set only. Among the three input encoding techniques, the word2vec is found for better GDEX classification. However, the unidirectional or vanilla GRU and LSTM are found biased towards the negative class. Alternatively, in other words, the aforementioned DL networks failed to discriminate between a GDEX and bad examples and hence developed a propensity towards the negative class only. (The authors would maintain that the biasedness of unidirectional NN can be overcome by introduction of dropouts but we are afraid of doing it for the reason of being unjust to the rest of NNs employed in this work.) Moreover, this behaviour is seen for both of the dense embedding techniques word2vec and GloVe. In contrast, the

bidirectional variant of these two techniques achieved approximately equal and comparatively optimal results. We maintain that word2vec with Bi-LSTM is the optimal algorithm for GDEX as it has achieved 77% accuracy (and balanced accuracy as well). Alongside it, the highest recall, i.e., 86%, is also on record for this setting. The NNs with the one-hot encodings though have shown the least but steady results.

Figure 14 shows epochwise loss and accuracy achieved in training and validation sets. We have got the typical behaviour in counting the increment in epochs; the loss, in the validation set, minimizes to an extent, and afterwards, it gets propensity to increase; in contrast, the loss continues to diminish in the training set [39, 42, 43]. We can see this behaviour in all DL models—except for the Bi-LSTM and Bi-GRU with word2vec and GloVe, which show steady performance. Furthermore, since we know that the DL is more appropriate for the largescale datasets, and currently the data employed for this experiment is comparatively smaller, we can expect a few numbers of epochs are enough for the training (or not indulging in the overfitting model on training data). In this regard, the authors maintain that the 3 epochs are enough for any of the DL-NNs used in the experiments. This is so because we see in the validation dataset that the accuracy is declining after the 3rd epoch.

Figure 13 shows the improvement in accuracy and balanced accuracy achieved by one DNN over the other networks; the quantified value of these metrics is subtracted as  $NN_x - NN_y$ , provided that  $x \neq y$ , where  $x$  is DNN (alongside the input encoding method used in it) defined on the  $x$ -axis and  $y$  is DNN in the  $y$ -axis. The cells with the shades of red colour in the figures indicate negative improvement; in contrast, the cells with grey shades indicate improvement. The intensity of shades is directly proportional to the value of the improvement. Likewise, in the observation reported in Table 8, we found that, except for the few network comparisons, the improvement in the accuracy and balanced

TABLE 8: Summary of evaluation statistics used in this paper.

Name (abbreviation)	Derivation	Definition/Notes
Precision ( $P$ )	$P = (TP/(TP + FP))$	Precision (or alternatively known as positive predictive value) reveals the ratio of TP to the documents that are predicted positive by the PS
Recall ( $R$ )	$R = (TP/(TP + FN))$	Recall (or true positive rate) shows the right potential of the PS for predicting positive documents in the subset of all positive documents in the system
Specificity ( $S$ )	$S = (TN/(TN + FN))$	Specificity (or true negative rate) is the exact opposite of $R$ . It gives the potential of the PS for negative documents
$F_1$ -measure ( $F$ )	$F = 2 \cdot ((P \cdot R)/(P + R))$	$F_1$ -measure is a harmonic mean of $P$ and $R$ . It is important to use where the dataset is imbalanced; further, it is a strict measure, which has a propensity towards the minima of $P$ and $R$ [41]
Accuracy ( $A$ )	$A = ((TP + FP)/(TP + TF + TN + FN))$	Accuracy gives the overall creditability of the PS
Balanced accuracy (BA)	$BA = (R + S)/2$	Likewise $F$ , the balanced accuracy is also a mean statistic, which gives an arithmetic mean of $R$ and $S$

TABLE 9: Results of conventional ML algorithms. Dataset with the final scoring function  $\Psi_{f=avg}$ .

Algo.	Vec.	Only BoW						BoW + PoS tags					
		$P$	$R$	$S$	$F$	$A$	BA	$P$	$R$	$S$	$F$	$A$	BA
$k$ -NN	Count	55.9	55.9	45.8	55.9	57.2	50.8	62	62	53.4	62	64.8	57.7
	TF * IDF	62.2	62.2	54.6	62.2	64.6	58.4	67.7	67.7	70.5	67.7	66.1	69.1
NB	Count	63.1	63.1	56	63.1	65.6	59.5	63.5	63.5	58.4	63.5	65.4	60.9
	TF * IDF	64.3	64.3	60.4	64.3	65.9	62.4	64.9	64.9	62.2	64.9	66.1	63.6
RFT	Count	65.2	65.2	56.7	65.2	69	61	67.2	67.2	62	67.2	70	64.6
	TF * IDF	67.2	67.2	62.9	67.2	69.5	65.1	67.4	67.4	62.8	67.4	69.9	65.1
Linear-SVM	Count	67.5	67.5	62.9	67.5	70	65.2	65.5	65.5	56.4	65.5	69.7	61
	TF * IDF	68.1	68.1	64.4	68.1	70.1	66.2	69.1	69.1	66.2	69.1	70.9	67.6
Rbf-SVM	Count	66.7	66.7	60.2	66.7	69.9	63.4	66.8	66.8	60.7	66.8	69.9	63.7
	TF * IDF	68.7	68.7	65.9	68.7	70.5	67.3	69.4	69.4	66.5	69.4	71.2	67.9

TABLE 10: Results of conventional ML algorithms. Dataset with final scoring function  $\Psi_{f=max}$ .

Algo.	Vec.	Only BoW						BoW + PoS tags					
		$P$	$R$	$S$	$F$	$A$	BA	$P$	$R$	$S$	$F$	$A$	BA
$k$ -NN	Count	57.9	57.9	44.8	57.9	60.4	51.4	67.1	67.1	57.6	67.1	72.1	62.4
	TF * IDF	66.3	66.3	57.3	66.3	70.7	61.8	74	74	75.1	74	73.1	74.5
NB	Count	68.4	68.4	62.9	68.4	71.6	65.6	67.9	67.9	64.2	67.9	70	66
	TF * IDF	69.8	69.8	66.6	69.8	71.9	68.2	69.8	69.8	68	69.8	70.9	68.9
RFT	Count	70	70	62.8	70	74.8	66.4	73.4	73.4	69.5	73.4	77	71.4
	TF * IDF	75.4	75.4	72.8	75.4	78	74.1	74.3	74.3	71.3	74.3	77.2	72.8
Linear-SVM	Count	75.4	75.4	74.9	75.4	76.1	75.1	74.2	74.2	71.7	74.2	76.5	72.9
	TF * IDF	75	75	72.4	75	77.5	73.7	75.2	75.2	72.1	75.2	78.3	73.6
Rbf-SVM	Count	73.7	73.7	69.1	73.7	77.9	71.4	73.8	73.8	69.8	73.8	77.5	71.8
	TF * IDF	74.7	74.7	71.9	74.7	77.3	73.3	75.2	75.2	72.3	75.2	78.1	73.7

TABLE 11: Results of conventional ML algorithms. Dataset with final scoring function  $\Psi_{f=sum}$ .

Algo.	Vec.	Only BoW						BoW + PoS tags					
		$P$	$R$	$S$	$F_1$	$A$	BA	$P$	$R$	$S$	$F_1$	$A$	BA
$k$ -NN	Count	58.1	58.1	42.1	58.1	61.3	50.1	67.3	67.3	57.9	67.3	72.3	62.6
	TF * IDF	66.2	66.2	57.1	66.2	70.5	61.6	74.4	74.4	76.7	74.4	72.3	75.5
NB	Count	68.6	68.6	63.6	68.6	71.5	66.1	67.9	67.9	64.5	67.9	69.8	66.2
	TF * IDF	69.6	69.6	67.3	69.6	71.0	68.4	70.3	70.3	69.5	70.3	70.9	69.9
RFT	Count	70.5	70.5	63.9	70.5	75.2	67.2	73.5	73.5	69.9	73.5	76.6	71.7
	TF * IDF	74.9	74.9	73.0	74.9	76.8	73.9	74.3	74.3	71.8	74.3	76.8	73
Linear-SVM	Count	72.3	72.3	68.7	72.3	75.3	70.5	72.7	72.7	68.7	72.7	76	70.7
	TF * IDF	74.8	74.8	72.6	74.8	76.9	73.7	74.4	74.4	71.7	74.4	77	73
Rbf-SVM	Count	73.8	73.8	70.2	73.8	77.1	72.0	74.5	74.5	71.8	74.5	77.2	73.1
	TF * IDF	74.5	74.5	72.1	74.5	76.9	73.3	74.9	74.9	72.5	74.9	77.3	73.7

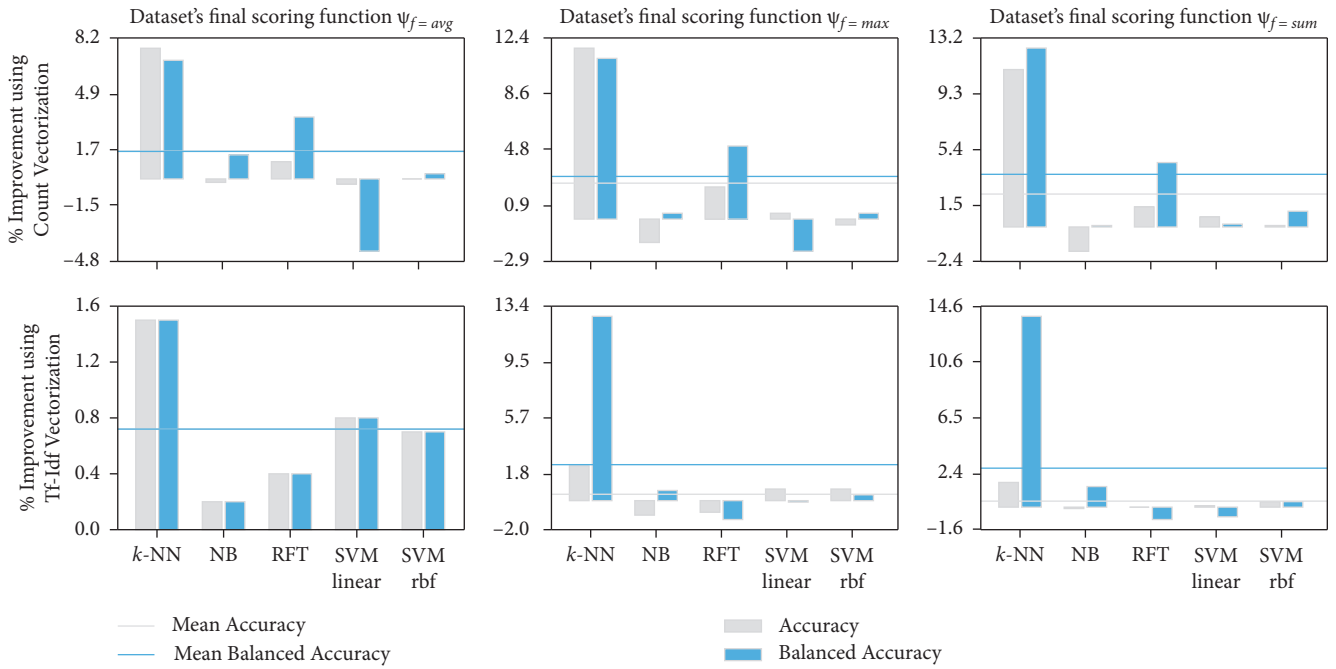


FIGURE 11: Comparative percentage improvement gained by using words + PoS tags features on only BoW features in conventional machine learning algorithms.

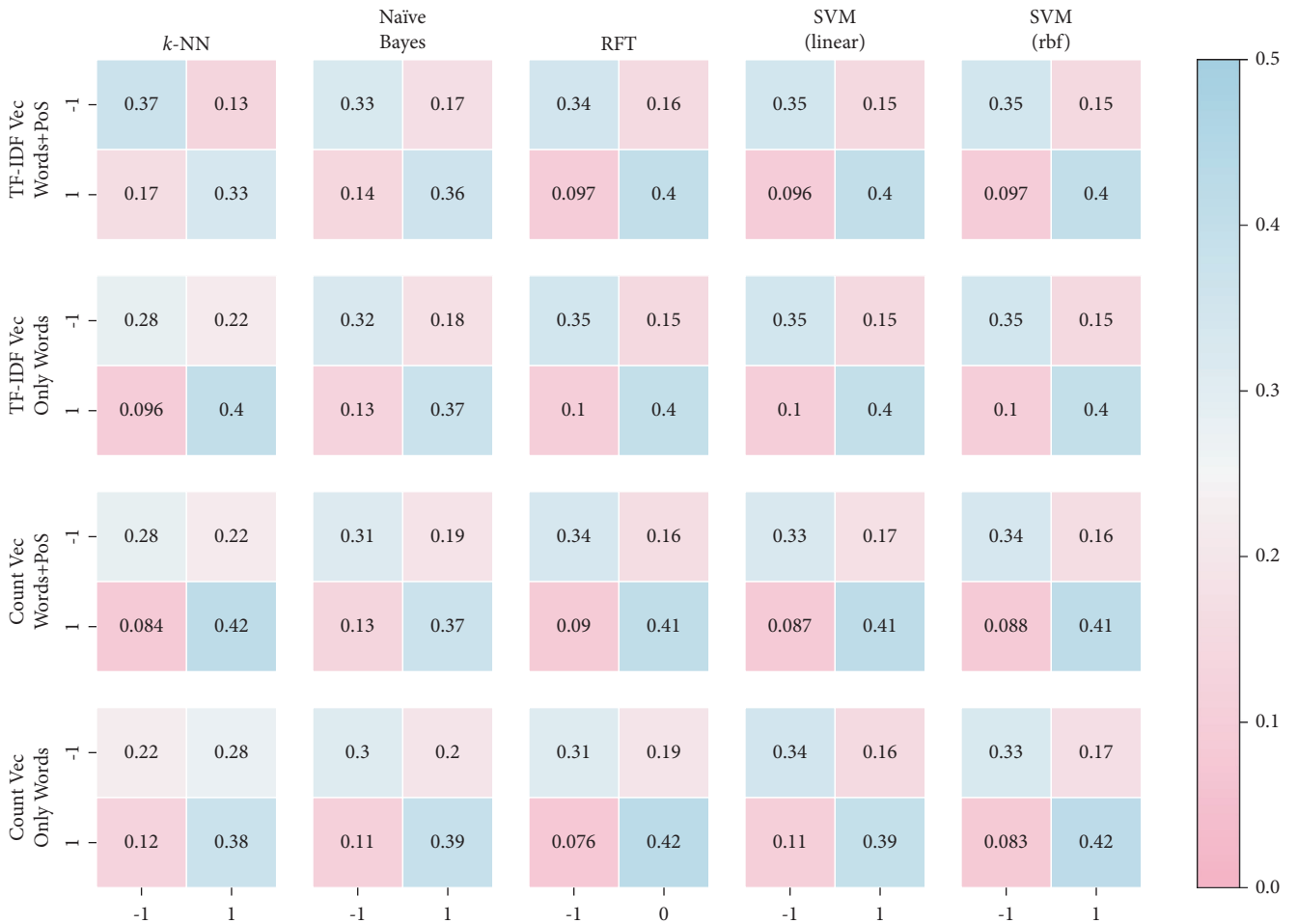


FIGURE 12: Average confusion matrices for conventional ML approaches. The confusion matrices (of datasets relating to the average, max, and sum scoring functions) are averaged, respectively, for each algorithm, vectorization, and feature set.



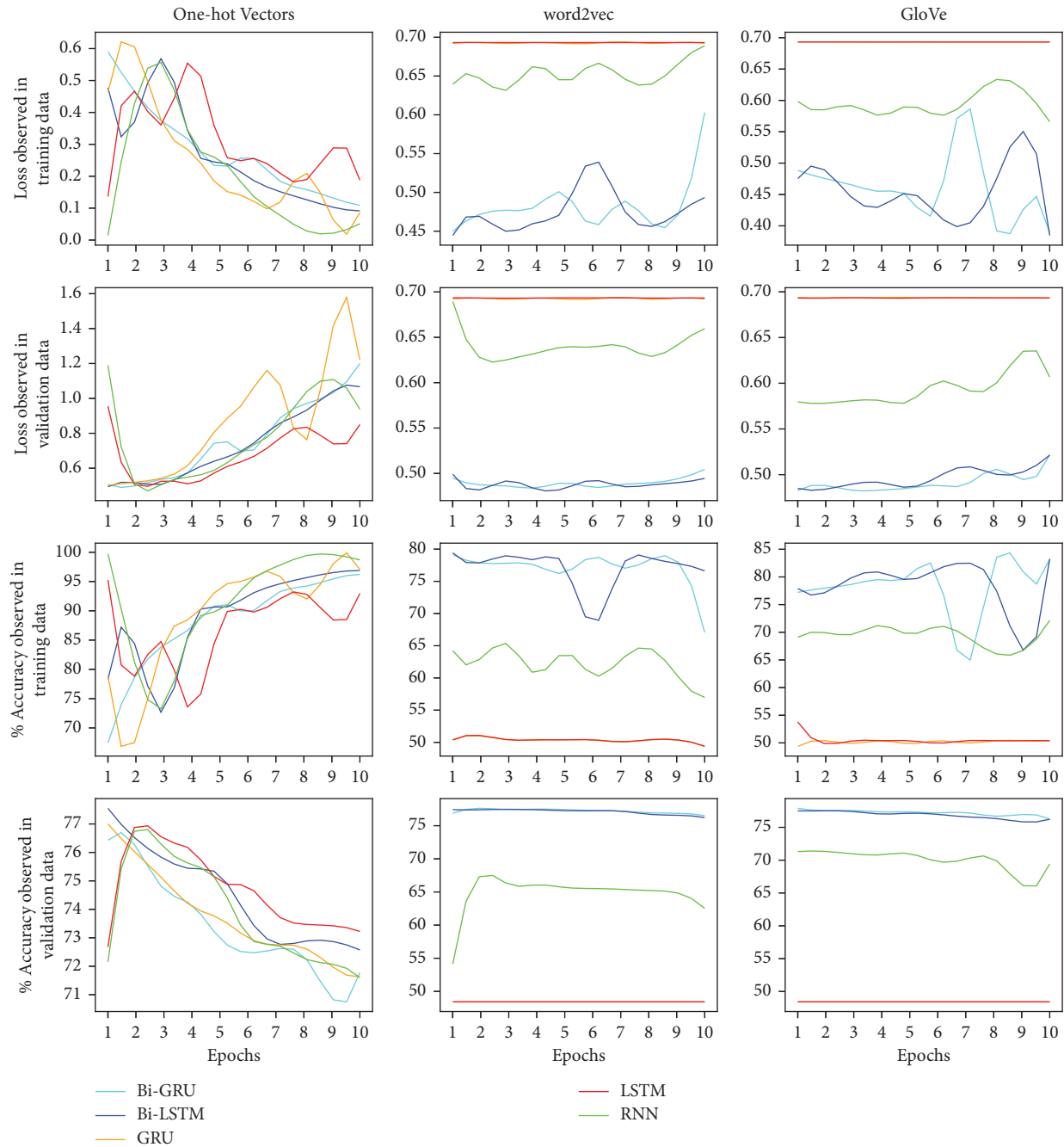


FIGURE 13: % improvement observed in accuracy and balanced accuracy. The values of the neural network and its encoding method (paired as  $NN + encoding$ ) on the  $x$ -axis are subtracted from the respective pair values on the  $y$ -axis.

accuracy yielded in DNN are equivalently identical. We find that bidirectional DNNs with word embeddings drew a major improvement on the rest of all DNNs. In a similar context, though the highest accuracy and balanced accuracy are seen over unidirectional NN we neglect this case on the ground of biased performance shown by the unidirectional DNNs (with word embeddings). Except for the previously mentioned case, the real highest gain in accuracy and balanced accuracy is seen over RNN + word2vec; i.e., Bi-GRU and Bi-LSTM have secured  $\approx 23\%$  improvement with word2vec, followed by attaining  $\approx 22\%$  improvement by the

same DNNs with GloVe. Keeping the focus on Bi-GRU and Bi-LSTM, the most optimal word embedding scheme is word2vec such that it achieved  $\approx 4\%$  and  $\approx 5\%$  improvement over vanilla one-hot encodings used for the same DNNs and  $\approx 1\%$  improvement over GloVe.

Observing CMs presented in Figure 15, we confirm that the bidirectional LSTM with word2vec is the most optimal NN and inputs data embedding pair for resolving the problem under study. We also maintain that, in comparison to the GloVe and word2vec, the one-hot encoding is the most underperforming input encoding scheme.

TABLE 12: Results of conventional DL algorithms employed for the dataset with final scoring function  $\Psi_{f=\text{sum}}$ .

Algo.	One-hot encoding						word2vec						GloVe					
	P	R	S	F1	A	BA	P	R	S	F1	A	BA	P	R	S	F1	A	BA
RNN	72.5	74.2	70	73.3	72.2	72.1	55.4	57	51	56	54	54	70.4	70.3	68.5	70.3	69.4	69.4
GRU	72.8	73	70.9	72.9	72	72	—	0	100	—	48	50	—	0	100	—	48.4	50
LSTM	74.1	72.2	73.2	73.2	72.7	72.7	—	0	100	—	48	50	—	0	100	—	48.4	50
Bi-GRU	73.8	70.2	73.5	72	71.8	71.8	77.8	77	77	78	77	77	73.1	85.5	66.5	78.8	76.3	76
Bi-LSTM	73.4	73.3	71.8	73.4	72.6	72.6	74.3	86	69	80	77	77	75	80.9	71.2	77.8	76.2	76.1

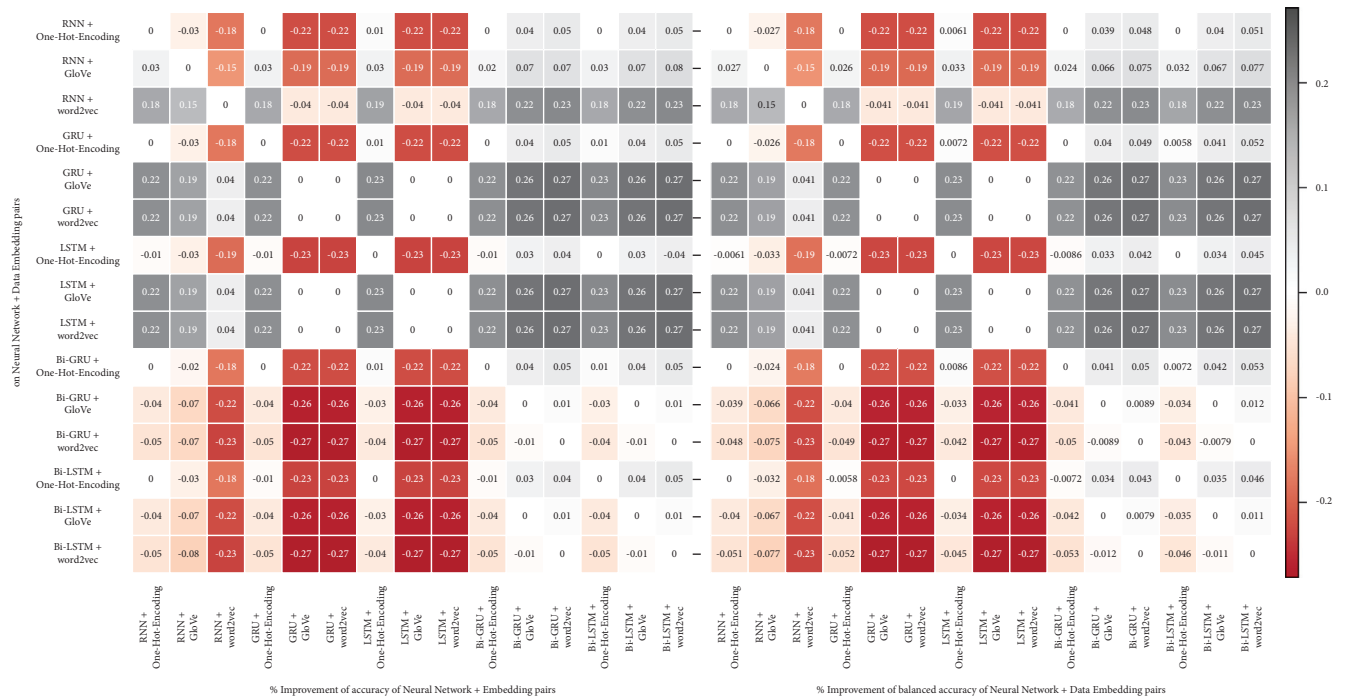


FIGURE 14: Epochwise insights into the loss and % accuracy w.r.t training and validation datasets.

**4.4. Comparative Analysis on ML vs. DL Models.** As reported in several different studies on the comparison of ML and DL models [16, 39, 43, 44], the authors of this paper reassert that the DL models outperform conventional ML models. In addition to it, we also maintain that DL-based models are revealed to attain balanced scores in accuracy and balanced accuracy. However, the DL-based unidirectional algorithms failed, which we consider specific to the problem under study; in contrast, the bidirectional DL algorithms are found the most optimal ones.

Thus, w.r.t the results compiled in Table 13, if we look at the averages of all ML models (for the dataset  $\Psi_{f=\text{sum}}$ ) and compared them with the averaged values of DL-based models (i.e., RNN, Bi-LSTM, and Bi-GRU; leaving unidirectional LSTM and GRU due to their biasedness) then, we see only an improvement of  $\approx +3.56\%$  and  $\approx +2.47\%$ , respectively, in recall and balanced accuracy for GDEX classification. However, the principal reason for such small improvement lies with the lower scores of RNN in comparison to the remaining two bidirectional NNs. In contrast, the ML-based models took very little time in preprocessing and training. In a similar context, we can see the one-hot

encoding turned training time longer, whereas the DL models with 300-dimensional dense word embeddings were trained in a small amount of time.

**4.5. Competitive Analysis on ML and DL Models with Manual GDEX Elicitation Routines.** Table 14 shows the selected examples of sentences from the test/validation set and the prediction made by the most optimal ML and DL models for them. In addition to it, we also show the GDEX rules presented in the seminal work by Kilgarrieff et al. [2]. These were actually 5 rules, which are already mentioned in the literature review (see Subsection 2.2); however, rule 3 is omitted in discussion as it deals with the penalization of a sentence containing anaphors and pronouns (though there are sentences which deal the aforesaid matters, ML/DL do not explicitly deal with such penalization). Examples 1–8 show TP and TN, wherein, specifically rule#4 is false when the actual label is *bad*. Examples 9 and 10 show FP, where rule#4 is false. Examples 11 and 12 are real mistakes, as these are the FN, and the sentence not only complies with all rules but also appears to be very succinct in structure.

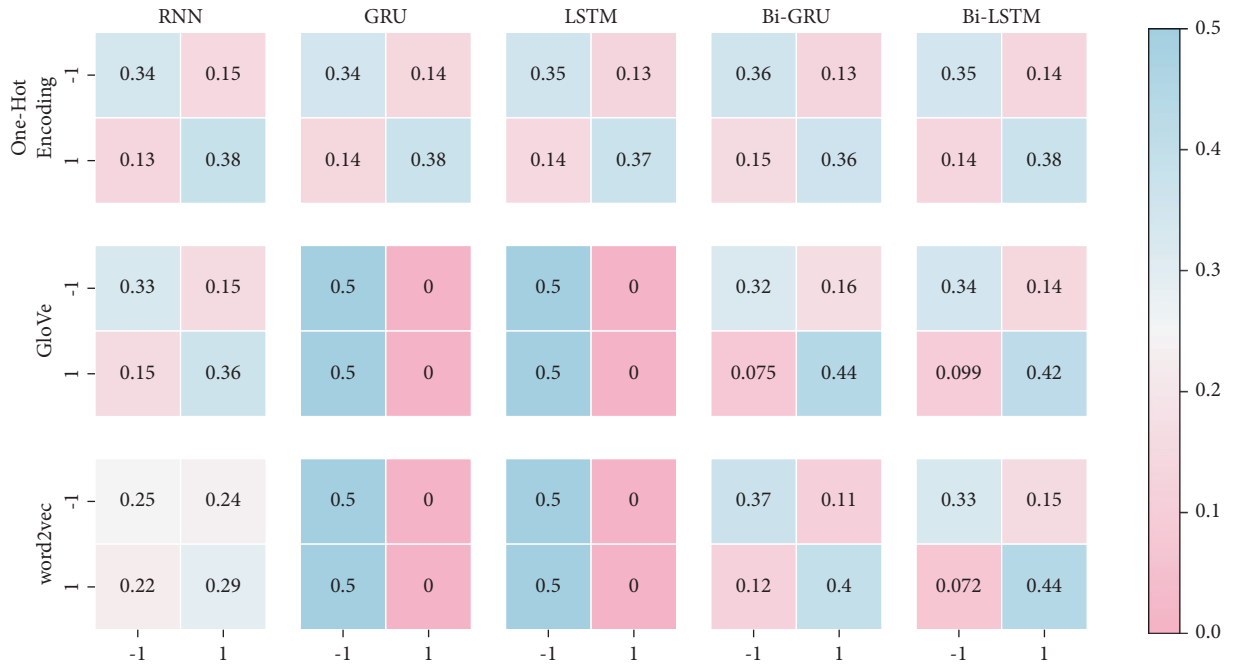


FIGURE 15: The confusion matrices (of validation set relating to the max scoring functions).

TABLE 13: Comparative averages of ML and DL performances. The \* in the last row indicates the number of seconds elapsed more than the average of ML-based model preprocessing and training.

Models	Evaluation metrics						Running time (in seconds)	
	<i>P</i>	<i>R</i>	<i>S</i>	<i>F</i>	<i>A</i>	BA	Preprocessing	Training
ML + CB	68.66	68.66	61.7	68.66	72.08	65.18	1.72	36.86
ML + TB	72	72	68.42	72	74.42	70.18	1.96	48.14
ML + CBP	71.18	71.18	66.56	71.18	74.38	68.86	9.15	33.05
ML + TBP	73.66	73.66	72.44	73.66	74.86	73.02	9.06	46.4
ML average	71.38	71.38	67.28	71.38	73.94	69.31	5.47	41.1
DL + one-hot enc.	73.23	72.57	71.77	72.9	72.7	72.17	1.23	272.33
DL + word2vec	69.17	73.33	65.67	71.33	69.33	69.33	325.68	32
DL + GloVe	72.83	78.9	68.73	75.63	73.99	73.83	32.48	31.66
DL average	71.74	74.93	68.72	73.29	72.01	71.78	119.80	112.0
Improvement in DL	0.37	3.56	1.44	1.91	-1.93	2.47	114.32 *	70.8 *

TABLE 14: Selected examples for the discussion of errors in GDEX classification.

Examples	Actual label	Prediction information		Did the example comply with rules for GDEX as defined in [2]?			
		Label	Classifier	R#1	R#2	R#4	R#5
(1) Not wanting to abandon Lori, she did nothing.	Good	Good	RFT + TBP	Yes	Yes	Yes	Yes
(2) The French have abandoned the left bank?	Bad	Bad	RFT + TBP	Yes	Yes	No	Yes
(3) She closed the cabinet door, troubled.	Good	Good	Bi-LSTM + w2v	Yes	Yes	Yes	Yes
(4) At a cabinet meeting on June 5th it is said that M.	Bad	Bad	Bi-LSTM + w2v	Yes	Yes	Yes	Yes
(5) Each dog knew its master and its call.	Good	Good	RFT + TBP	Yes	Yes	Yes	Yes
(6) The redcoats are coming, they said to each other.	Bad	Bad	RFT + TBP	Yes	Yes	No	Yes
(7) You love him very much.	Good	Good	Bi-LSTM + w2v	Yes	Yes	Yes	Yes
(8) Very. I'm so glad we have you and Jonathan	Bad	Bad	Bi-LSTM + w2v	Yes	Yes	No	No
(9) The sack and dog moved about two feet.	Bad	Good	Bi-LSTM + w2v	Yes	Yes	No	Yes
(10) For the sowing of seed see Sowing.	Bad	Good	RFT + TBP	Yes	Yes	No	Yes
(11) I see no point in telling him.	Good	Bad	Bi-LSTM + w2v	Yes	Yes	Yes	Yes
(12) He set the sack on the table.	Good	Bad	RFT + TBP	Yes	Yes	Yes	Yes

We can draw another meaningful insight into dataset curation through distant supervision. The unanimous true for rules 1 and 2 and correct assessment of rules 4 and 5 confirm the reliability of the usage of web-based data available at YD.com alongside the method for label assignment with the scoring function  $\Psi_{f=\text{sum}}$  for GDEX classification and similar other problems.

## 5. Conclusions

This paper provides the implementation of both ML and DL models for the GDEX classification. Following the results compiled in the experiments, we conclude that the proposed methodology is accomplishable for the automation of manual GDEX elicitation routine. The dataset of 50K example is extracted with the distant supervision technique, for which the summation method is found better than vote aggregation (averaging and max) methods. For the conventional ML-based methods, the distinction of TF \* IDF normalization over count vectorization is revisited during experiments. Also, we have analysed that PoS features are important and better for the easy classification and discrimination of GDEX. For the DL-based models, Bi-LSTM + word2vec is the champion among the rest of all DL-based combinations.

In the future, this work could be extended by incorporating supervised learning for the GDEX elicitation against the given target word. We would also like to evaluate the current system on the attention-based DL models. At last, we would like to apply and evaluate the current technique on oriental languages such as Arabic, Persian, and Urdu—where the GDEX is considered to have historic relevance in the poetic work.

## Data Availability

The models and data files can be accessed at <https://github.com/MuhammadYaseenKhan/english-gdex>.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was partially financed by the two educational institutes: Mohammad Ali Jinnah University, Karachi, Pakistan, and Islamic University of Madinah, Saudi Arabia.











## References

- [1] B. S. Atkins and M. Rundell, *The Oxford Guide to Practical Lexicography*, Oxford University Press, Oxford, UK, 2008.
- [2] A. Kilgariff, M. Husák, K. McAdam, M. Rundell, and P. Rychlý, “Gdex: automatically finding good dictionary examples in a corpus,” in *Proceedings of the XIII EURALEX International Congress*, Universitat Pompeu Fabra, Barcelona, Spain, July 2008.
- [3] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for the relation extraction without labeled data,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Singapore, August 2009.
- [4] W. Muhammad, M. Mushtaq, K. N. Junejo, and M. Y. Khan, “Sentiment analysis of product reviews in the absence of labelled data using supervised learning approaches,” *Malaysian Journal of Computer Science*, vol. 33, no. 2, pp. 118–132, 2020.
- [5] I. Pilán, E. Volodina, and R. Johansson, “Automatic selection of suitable sentences for language,” in *Proceedings of the 20 Years of EUROCALL: Learning from the Past, Looking to the Future: 2013 EUROCALL Conference*, Dublin, Ireland, September 2013.
- [6] I. Srdanović and I. Kosem, “GDEX for Japanese: automatic extraction of good dictionary example candidates,” in *Proceedings of the GLOBALEX 2016 Lexicographic Resources for Human Language Technology Workshop Programme*, Portorož, Slovenia, May 2016.
- [7] I. Kosem, K. Koppel, T. Zingano Kuhn, J. Michelfeit, and C. Tiberius, “Identification and automatic extraction of good dictionary examples: the case(s) of GDEX,” *International Journal of Lexicography*, vol. 32, no. 2, pp. 119–137, 2019.
- [8] A. Geyken, C. Pölitiz, and T. Bartz, “Using a maximum entropy classifier to link “good” corpus examples to dictionary senses,” in *Proceedings of the Electronic Lexicography in the 21st Century Conference*, Herstmonceux Castle, UK, August 2015.
- [9] E. T. Jaynes, “Information theory and statistical mechanics,” *Physical Review. Series II*, vol. 106, no. 4, pp. 620–630, 1958.
- [10] N. Ljubešić and M. Peronja, “Electronic lexicography in the 21st century: linking lexical data in the digital age,” in *Proceedings of the Electronic Lexicography in the 21st Century Conference (eLex)*, Herstmonceux Castle, UK, August 2015.
- [11] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [12] R. Stanković, B. Šandrih, R. Stijović, C. Krstev, D. Vitas, and A. Marković, “SASA dictionary as the gold standard for good dictionary examples for Serbian,” *Electronic lexicography in the 21st century: Smart lexicography*, 2019.
- [13] K. Koppel, *Example Sentences in Estonian Learners’ Dictionaries*, Institute of the Estonian Language, University of Tartu, Tartu, Estonia.
- [14] S. Uprety and M. Shakya, “Role of context clue sentences as dictionary examples,” *The Journal of University Grants Commission*, vol. 6, no. 1, pp. 132–140, 2017.
- [15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK, 2008.
- [16] K. Kowsari, K. J. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: a survey,” *Information*, vol. 10, no. 4, p. 150, 2019.
- [17] A. R. Martinez, “Natural language processing,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 352–357, 2010.
- [18] W. Zhang, T. Yoshida, and X. Tang, “A comparative study of TF\*IDF, LSI and multi-words for text classification,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.
- [19] E. Fix and J. L. Hodges Jr., *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*, USAF School of Aviation Medicine, Randolph Field, TX, USA, 1951.
- [20] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

- [21] G. H. Jon and P. Langley, "Estimating continuous distributions in Bayesian classifiers," 2013, <https://arxiv.org/abs/1302.4964a>.
- [22] H. Zhang, "Exploring conditions for the optimality of naïve Bayes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 2, pp. 183–198, 2005.
- [23] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [24] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.
- [25] M. Y. Khan and K. N. Junejo, "Exerting 2D-space of sentiment Lexicons with machine learning techniques: a hybrid approach for sentiment analysis," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 599–608, 2020.
- [26] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "Liblinear: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. 8, pp. 1871–1874, 2008.
- [27] J. R. Firth, "A synopsis of linguistic theory, 1930–1955," *Studies in Linguistic Analysis*, Longmans, London, UK, 1957.
- [28] M. Tomáš, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, GA, USA, June 2013.
- [29] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, October 2014.
- [30] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [31] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [32] A. Amidi and S. Amidi, "CS 230-recurrent neural networks cheatsheet," *Stanford*, <https://stanford.edu/%7Eshervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>, 2021.
- [33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [34] F. A. Gres, J. Schmidhuber, and F. Cummins, "Learning to forget: contual predicition with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
- [35] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odessey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2015.
- [36] W. Ke, D. Huang, F. Yang, and Y. Jiang, "Soft sensor development and applications based on LSTM in deep neural networks," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, USA, November-October 2017.
- [37] P. F. Moshiri, H. Navidan, R. Shahbazian, S. A. Ghorashi, and D. Windridge, "Using GAN to enhance the accuracy of indoor human activity recognition," 2020, <https://arxiv.org/abs/2004.11228>.
- [38] K. Cho, B. V. Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, <https://arxiv.org/abs/1406.1078>.
- [39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, Cambridge, UK, 2016.
- [40] V. Pareto, "Cours d'économie politique, Librairie Droz," 1964.
- [41] S. Shaikh, M. Y. Khan, and M. S. Nizami, "Using patient descriptions of 20 most common diseases in text classification for evidence-based medicine," in *Proceedings of the Mohammad Ali Jinnah University International Conference on Computing (MAJICC'21)*, Karachi, Pakistan, July 2021.
- [42] M. Y. Khan and T. Ahmed, "Pseudo transfer learning by exploiting monolingual corpus: an experiment on roman Urdu transliteration," in *Intelligent Technologies and Applications*, I. Bajwa, T. Sibalija, and D. Jawawi, Eds., pp. 422–431, Springer, Singapore, 2019.
- [43] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, San Francisco, CA, USA, 2015.
- [44] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Prentice Hall, Hoboken, NJ, USA, 3rd edition, 2020.
- [45] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [46] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [47] B. Farley and W. Clark, "Simulation of self-organizing systems by digital computer," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 76–84, 1954.
- [48] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [49] A. G. Ivakhnenko and V. G. Lapa, *Cybernetics and Forecasting Techniques*, American Elsevier Pub. Co, Princeton, NJ, USA, 1967.
- [50] P. Werbos, "Applications of advances in nonlinear sensitivity analysis," in *System Modeling and Optimization*, pp. 762–770, Springer, Berlin, Germany, 1982.
- [51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

## Research Article

# Urdu Handwritten Characters Data Visualization and Recognition Using Distributed Stochastic Neighborhood Embedding and Deep Network

Mujtaba Husnain <sup>1</sup>, Malik Muhammad Saad Missen <sup>1</sup>, Shahzad Mumtaz <sup>1</sup>,  
Dost Muhammad Khan <sup>1</sup>, Mick el Coustaty <sup>2</sup>, Muhammad Muzzamil Luqman <sup>2</sup>,  
Jean-Marc Ogier <sup>2</sup>, Hizbullah Khattak <sup>3</sup>, Sikandar Ali <sup>4</sup>, and Ali Samad <sup>1</sup>

<sup>1</sup>Department of Information Technology, Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

<sup>2</sup>L3i Lab, Universit  of La Rochelle Av. Michel Cr peau, 17000 La Rochelle, France

<sup>3</sup>Department of Information Technology, Hazara University Mansehra, 21120 Khyber Pakhtunkhwa, Pakistan

<sup>4</sup>Department of Information Technology, The University of Haripur, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Sikandar Ali; sikandar@cup.edu.cn

Received 25 May 2021; Revised 18 July 2021; Accepted 14 August 2021; Published 3 September 2021

Academic Editor: Shahzad Sarfraz

Copyright   2021 Mujtaba Husnain et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we make use of the 2-dimensional data obtained through *t*-Stochastic Neighborhood Embedding (*t*-SNE) when applied on high-dimensional data of Urdu handwritten characters and numerals. The instances of the dataset used for experimental work are classified in multiple classes depending on the shape similarity. We performed three tasks in a disciplined order; namely, (i) we generated a state-of-the-art dataset of both the Urdu handwritten characters and numerals by inviting a number of native Urdu participants from different social and academic groups, since there is no publicly available dataset of such type till date, then (ii) applied classical approaches of dimensionality reduction and data visualization like Principal Component Analysis (PCA), Autoencoders (AE) in comparison with *t*-Stochastic Neighborhood Embedding (*t*-SNE), and (iii) used the reduced dimensions obtained through PCA, AE, and *t*-SNE for recognition of Urdu handwritten characters and numerals using a deep network like Convolution Neural Network (CNN). The accuracy achieved in recognition of Urdu characters and numerals among the approaches for the same task is found to be much better. The novelty lies in the fact that the resulting reduced dimensions are used for the first time for the recognition of Urdu handwritten text at the character level instead of using the whole multidimensional data. This results in consuming less computation time with the same accuracy when compared with processing time consumed by recognition approaches applied to other datasets for the same task using the whole data.

## 1. Introduction

Data visualization deals with presenting the data in some visual context to make it trivial for the human to understand the nature of the data [1]. Furthermore, this activity helps in finding the patterns and hidden information, if they exist, in the data for further processing like data clustering and data classification. Nowadays, it is of common observation that the information related to the data science is of high dimensions, and therefore, its visualization in low-

dimensional space becomes impractical (Buja et al. [2]; Saeed et al. [1]). In almost all of the data science datasets, the researchers have had to deal with this acute and critical issue. While analyzing the high-dimensional data, almost every other researcher is interested in finding the optimal number of dimensions (or features) in order to apply any appropriate classifier for giving better performance (Nguyen and Holmes [3]; Song et al. [4]; ur Rehman et al. [5]). It is pertinent to mention that the terms “high-dimensional data visualization” and “high-dimensional visualization” are used

interchangeably in the literature; however, there is a succinct difference between these. In the first, the term *high* refers to data itself, whereas, in the second, it refers to the visualization. The interesting fact lies in visualizing the high-dimensional data on 2D or 3D plane that we have to apply some appropriate dimensionality reduction approaches on the whole data, since it is next to impossible in order to visualize the high-dimensional data on low-dimensional space (Engel et al. [6]; Song et al. [4]; ur Rehman et al. [5]).

The term “dimensionality” refers to the number of variables, characteristics, or features in which most of the datasets exist in the field of data science nowadays. Generally, these dimensions are represented as columns, and the main purpose is to reduce this number of columns. In the majority of cases, these column values are correlated and are also having redundant information that causes noise in data. This redundant information may impact negative and adverse effects in training any machine learning model thus producing error-prone results. That is the reason, the dimensionality reduction approaches have become of vital importance. Furthermore, it also helps in finding the patterns, if they exist, in the data set prior to applying any clustering or classification approach by reducing the model’s complexity thus avoiding the overfitting.

One of the key objectives of the dimensionality reduction technique is to reduce the high-dimensional data points  $D = d_1, d_2, d_3, \dots, d_n$  to a rather low dimension space, ideally, in two- (or three-) dimensional spaces  $S = s_1, s_2, s_3, \dots, s_n$  in order to get better visualization of the data, where  $S$  represents the equivalent low-dimension transformation and map of  $D$  and  $s_i$  is the corresponding data point of  $d_i$  that can be viewed on some appropriate scatter plot. The main purpose in this transformation is to preserve the characteristic features of the high-dimensional data as much as possible while transforming to the low-dimensional space. It is pertinent to mention that different dimensionality reduction approaches have a number of various capabilities of preserving the different types of properties of high-dimensional data (Engel et al. [6]; Saeed et al. [1]; Sorzano et al. [7]). Some are specific to preserve the linear dependencies, and some are experts in taking care of nonlinear features only. In order to solve these challenging issues, we engaged one of the most popular and widely used algorithms, the  $t$ -Distributed Stochastic Neighbor Embedding ( $t$ -SNE) (Maaten and Hinton [9]). The results showed that the  $t$ -SNE produced quite *faithful* clusters with clear and accurate separations while converting to low-dimensional data, thus retaining the characteristic features of the high-dimensional data. Furthermore, the reduced dimensions are then plugged into the Convolution Neural Network (CNN) to recognize and classify the Urdu handwritten numerals and characters in a separate set of experiments. The quality and efficiency of the results using the reduced dimensions obtained through  $t$ -SNE are remarkably better than those of the approaches used previously for the said purpose.

The paper is outlined as follows: Section 2 gives an overview of dimensionality reduction approaches. In Section 3, we discuss the motivation behind our work. In Section 4, the processing steps used in generating state-of-art datasets

are discussed in detail. The experimental results obtained by using reduced dimensions obtained through  $t$ -SNE and other approaches are presented in Section 5. Section 6 provides recognition results of Urdu handwritten characters and numerals using a deep CNN based model. In Section 7, we conclude the paper with some proposed future works.

## 2. Review of the Approaches Used in Dimensionality Reduction

A number of dimensionality reduction approaches for nonlinear data have been proposed in the last decade (Camastra [10]; Cunningham and Ghahramani [11]; Sorzano et al. [7]). The nonlinear techniques are more capable in comparing the standard and conventional linear techniques of dimensionality reduction; in dealing with complex nonlinear data, since most of the data sets associated with the data science and big data are likely to be strongly nonlinear in nature (Tsai [12]; Van Der Maaten et al. [8]). The related literature concludes that, among the existing number of dimensionality reduction techniques, the Principal Component Analysis (PCA) (Roweis and Saul [13]) is regarded as the most popular (unsupervised) linear technique (Maimon and Rokach [14]; Saul et al. [15]; Tsai [12]). Therefore, in this paper, we considered the PCA Roweis and Saul [13] as a benchmark. There also exist other techniques like Multi-Dimensional Scaling (MDS) Torgerson [16] that favor the data in linear form. This approach primarily focuses on the structural properties of the data points that vary in similarity. It is pertinent to mention that researchers must consider the nonlinear features of the high-dimensional data and also the very similar data points to produce clear separated clusters. This activity will also help in resolving issues associated with intracluster separations.

Some noteworthy survey articles (Camastra [10]; Cunningham and Ghahramani [11]; Sorzano et al. [7]) provide detailed information about dimensionality reduction approaches including Local Linear Embedding (LLE) (Roweis and Saul [13]); Laplacian Eigenmaps (Belkin and Niyogi [17]); Maximum Variance Unfolding (MVU) (Weinberger et al. [18]); Stochastic Neighbor Embedding (SNE) (Hinton and Roweis [19]); and Curvilinear Components Analysis (CCA) (Demartines and Hérault [20]) that specifically deal with nonlinear data by preserving the structural features of the whole data. It is also concluded from the work of Engel et al. [6] and Maaten and Hinton [9] that the approaches mentioned above did not produce the effective visualization results, since these approaches failed to manage the nonlinear characteristics of the whole data in the projected low-dimension map. Therefore, these approaches are not recommended to get the correct and perfect visualization of realistic data set having high-dimensional data points. The authors Engel et al. [6], Maaten and Hinton [9], and Song, Gretton, Borgwardt, and Smola [21] also observed that the MVU failed to visualize the English handwritten digits and provided highly overlapped clusters. To address these issues, we used the  $t$ -Distributed Stochastic Neighbor Embedding ( $t$ -SNE) (Maaten and Hinton [9]) approach for producing an efficient and effective visualization of the multidimensional

data in the form of clusters with clear and accurate separations by embedding both the pixel- and structural-based information in a principled way. It is pertinent to mention that  $t$ -SNE is a modified and extended form of SNE (Hinton and Roweis [19]). The categorization of the dimensionality reduction techniques is shown in Figure 1.

The literature related to the concept of t-SNE concludes that it is also one of the dimensionality reduction and data visualization techniques that deals with nonlinear data in an efficient way, since the math behind t-SNE is quite complex, but the idea is simple. The novelty in the efficacy of t-SNE is that it embeds the points from a higher dimension to a lower dimension trying to preserve the neighborhood of that specific point more efficiently as compared to other conventional and classical approaches like PCA, auto-encoders, High Correlation Filter, etc. Most classical dimensionality reduction approaches inherently work on preserving the global structure of the data, while t-SNE focuses tabs on both the local and global attributes of the data. This novelty of t-SNE assists in generating the clusters with high degree of compactness and intercluster separations.

### 3. Our Motivation

In this work, we performed experiments in two phases, namely, (i) the visualization of Urdu handwritten characters and numerals containing pixel-based features embedded with structural-based features using dimensionality reduction approaches and (ii) the recognition of these characters and numerals using deep network model of CNN using both pixel-based and structural based features and then using reduced dimensions of the same instances obtained through  $t$ -SNE and other approaches. In order to perform the abovementioned tasks, we prepared a novel data set of Urdu handwritten characters and numerals.

One of the issues associated with Urdu script is the shape similarity among its characters and numerals, as shown in Figure 2.

These issues may result in generating overlapping clusters during visualization in low-dimensional space, which may directly affect the accuracy rate in the recognition process. Therefore, we have to apply some suitable approach that results in precise and correct clusters with perfect separations. Moreover, intracluster separations of the data instances should be clear to depict the separations among the individual instances of the Urdu characters and numerals. The following factors and issues are the cause of our motivation to perform some experiments to resolve these issues.

To the best of our knowledge, there is a lack of noteworthy work done to date in order to transform *faithfully* the high-dimensional data of the Urdu handwritten characters to low-dimensional space.

- (i) There is no dataset associated with the Urdu handwritten characters and numerals that is available publicly to perform text recognition tasks at the character level.
- (ii) Also, there is no process of recognition adopted at the character level in Urdu handwritten text using

reduced dimensions obtained through dimensionality reduction approaches.

- (iii) In the subsequent section, we outline the procedure to generate a state-of-the-art dataset consisting of the images of the Urdu handwritten characters and numerals. Furthermore, the experimental results produced by the dimensionality reduction and recognition approaches are also given in subsequent sections.

### 4. Dataset Preparation

As mentioned in the earlier section, there is a lack of an appropriate and concise data corpus containing Urdu handwritten characters and numerals to perform text recognition tasks at the character level. There are some publicly available datasets of Urdu handwritten text like Urdu Nastaliq Handwritten Dataset (UNHD) (Ahmed et al. [22]; Das et al. [23]; Husnain et al. [24]; Sagheer et al. [25]), but unfortunately, the data sets have only the Urdu handwritten numerals. Furthermore, these instances are not enough to apply state-of-the-art machine learning algorithms to get better results. In order to bridge the gap and provide the state-of-the-art dataset of this kind, we invited about 1000 native Urdu speaking persons from different academic, administrative, and social groups of different age groups and gender. Moreover, the handicapped and physically disabled people are also involved to make the dataset more concise and comprehensive. All the authors were directed to write in a separate column of the printed sheets in his (or her) handwriting. Each of the sheets has printed images of 40 basic Urdu alphabets along with the 10 Urdu numerals in Nastaliq font.

Figure 3 depicts the sample page of our dataset. Furthermore, we have also recorded the demographic information of each author to generate the ground truth values of the whole data set. This information includes the basic information about each author, namely, age, race, gender, level of education, type of job, physical disability (if any), preference of left (or right) hand while writing, etc. This activity helped us in making the dataset more concise and comprehensive. After collection of an appropriate amount of data instances, the handwritten pages of both the Urdu characters and numerals were carefully scanned on a flatbed scanner at standard 300 dpi resolution. Furthermore, the scanned pages are then segmented manually into an image size of  $28 \times 28$  to capture Urdu handwritten character and numeral individually. As mentioned earlier, the whole dataset consists of  $1000 \times 10 = 10,000$  Urdu numerals and  $1000 \times 40 = 40,000$ .

Urdu characters images: for experimental work, we randomly selected the 6000 (600 each for the ten numerals) images of Urdu numerals and 28,000 (700 each of 40 characters) images of Urdu characters. It is pertinent to mention that we have planned to increase the number of participants to 1500 in order to include as many as possible the variations of different handwriting to create more comprehensive and multifaceted data. The complete data set, after completion, will be made publicly available for the



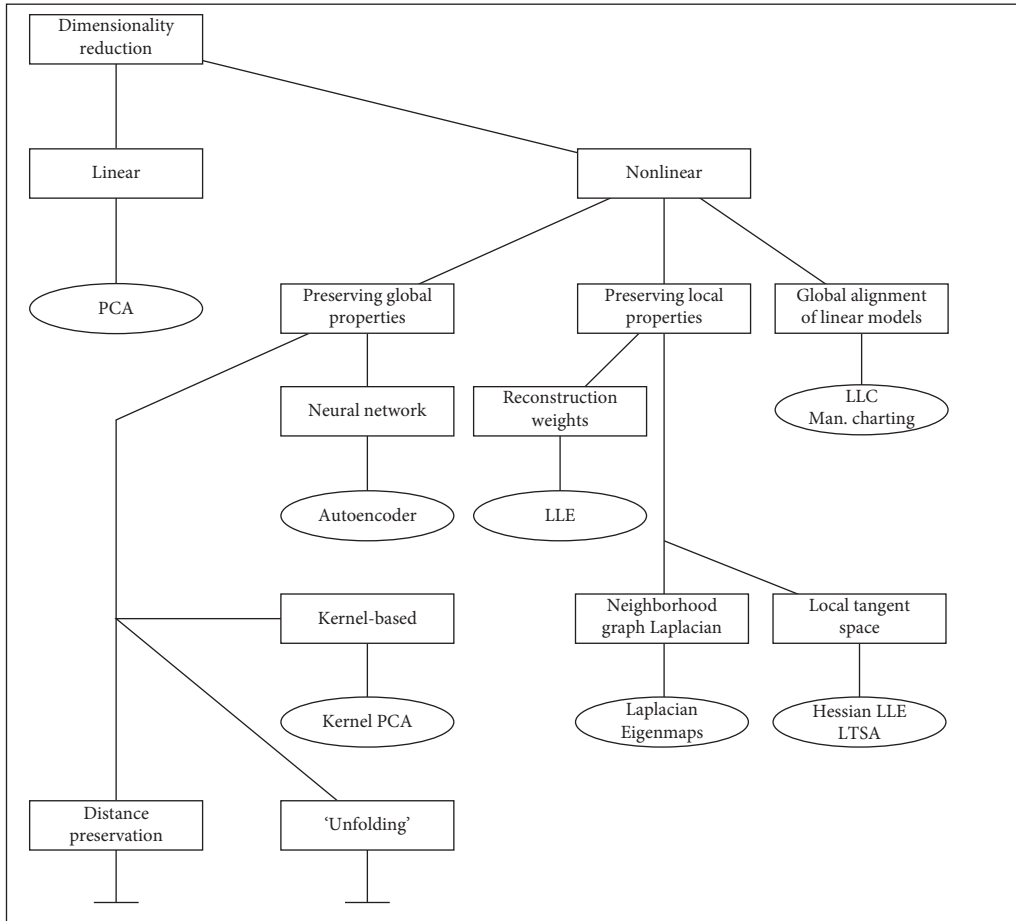


FIGURE 1: Categorization of dimensionality reduction approaches (Van Der Maaten et al. [8]).

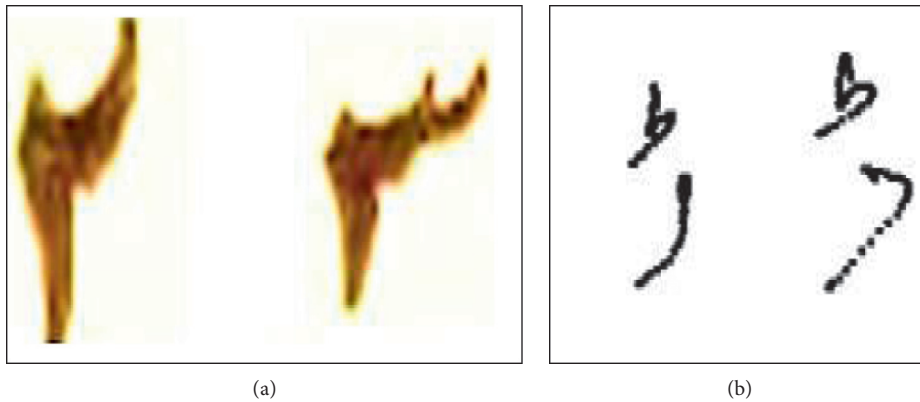


FIGURE 2: Similarity among Urdu characters and numerals.

researchers, since there is no dataset of such kind that is available till date. It is common that noise and distortion may likely occur while scanning the images. In order to remove the noise, we directed the authors to write in black ink only. This activity makes the noise removal process rather trivial task in a way that the colors other than black were considered as noise and removed easily. Furthermore, before applying any dimensionality reduction approach, we make use of some data

transformation algorithms, like gray-scale conversion, image segmentation, image resizing, extracting area of interest from the text image, normalizing the raw data, etc., to prepare our data set in some appropriate form. It is pertinent to mention that the Urdu characters as shown in Figure 4 and numerals share common characters with Arabic and Persian; therefore, our approach is equally applicable in these domains also with some minor modifications.

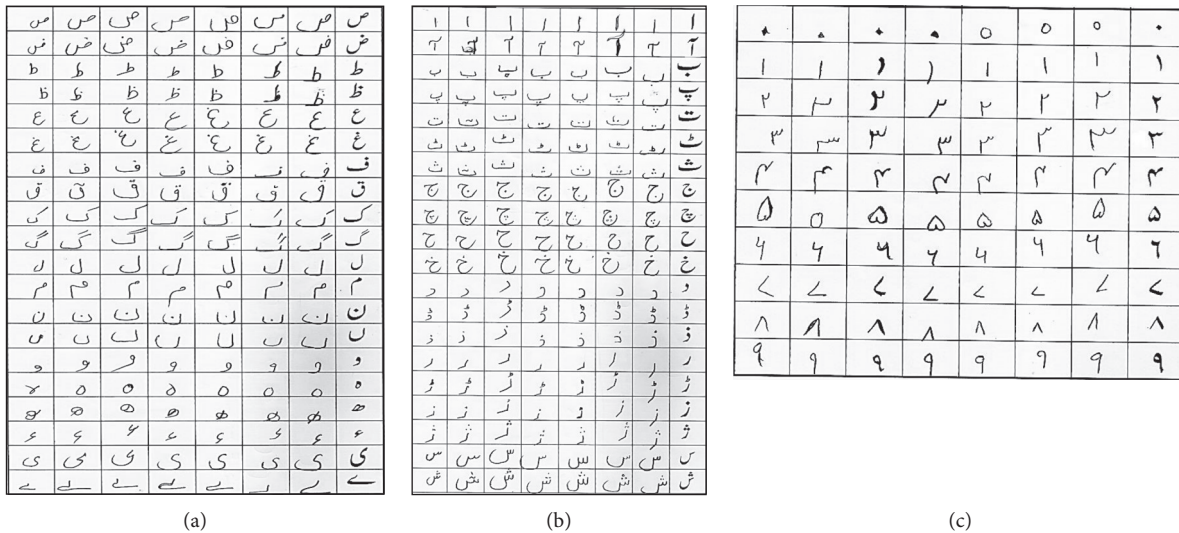


FIGURE 3: Sample pages of our dataset showing Urdu characters in (a) and (b) and Urdu handwritten numerals in (c).

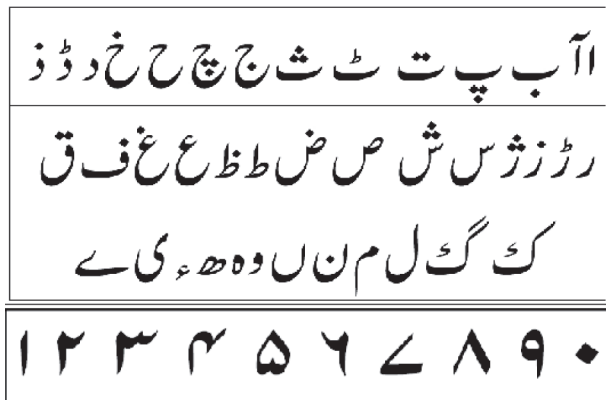


FIGURE 4: Urdu characters and numerals.

## 5. Experimental Results of Dimensionality Reduction Approaches

In this section, we present the results obtained by applying PCA (Roweis and Saul [13]), AE, and  $t$ -SNE on our two variants of the dataset. One variant contains the pixel-based features of both the Urdu numeric and characters. The second variant contains structural-based features embedded with pixel-based data of both the Urdu numeric and characters. The experimental results depict that adding structural based features presents visualization results with accurate separations among the clusters and also maintains intracluster variations.

As mentioned earlier, we represented each text image in our dataset in  $28 \times 28 = 784$  pixel values (or dimensions). For evaluation purposes, we make use of the three different dimensionality reduction approaches, namely, Principal Component Analysis (PCA) (Roweis and Saul [13]); Autoencoders (Liou et al. [26]); and  $t$ -SNE (Van Der Maaten et al. [8]) to our Urdu handwritten character and numeral dataset.

We used the following parameter setting while using  $t$ -SNE (Van Der Maaten et al. [8]) and its variant while

producing visualization results: the number of iterations,  $T$ , set to 1000 in order to achieve the optimized value of gradient descent, the fine-tuned parameter momentum term,  $\alpha(t)$ , is regulated to 0.5, where  $t$  is less than 250, while  $\alpha(t)$  is set to 0.8 for  $t$  is greater than 250. The initial value of learning rate  $\eta$  is set to 100 that may be regulated for each iteration equipped with a highly adaptive learning rate scheme. It is pertinent to mention that the experiments were executed using varying initial learning rates; however, we observed little variations in the quality of the resulting visualization results. Moreover, along with the other parameters, the perplexity is a tunable parameter that depicts how to correspond and normalize the local and global aspects of the data. In other words, we can say that perplexity helps in finding how many close neighbors each point has. It also has a complex effect on the resulting visualizations, as explained in the original  $t$ -SNE paper (Maaten and Hinton [9]). The selection of an optimal value of perplexity is of significant importance; therefore, one must have to take care, since it can be achieved only by producing multiple visualizations with varying perplexity values. Therefore, in this paper, we chose the best result based on the quality of visualization. Furthermore, it is an interesting fact that both the standard and proposed versions of  $t$ -SNE work equally and uniformly on a single assumed value of the perplexity for the whole dataset. The following subsection depicts the results generated through standard  $t$ -SNE on our proposed data set.

**5.1. Applying Standard  $t$ -SNE.** The results obtained through the standard  $t$ -SNE on the dataset having only the pixel-based information of the Urdu handwritten numerals are shown in Figure 5. It is clear from the results that there are some overlapping clusters while considering pixel-based information only. We performed a series of experiments using  $t$ -SNE on the same dataset using multiple perplexity values. It was observed from the output that, with the perplexity value of 70, the results showed some little

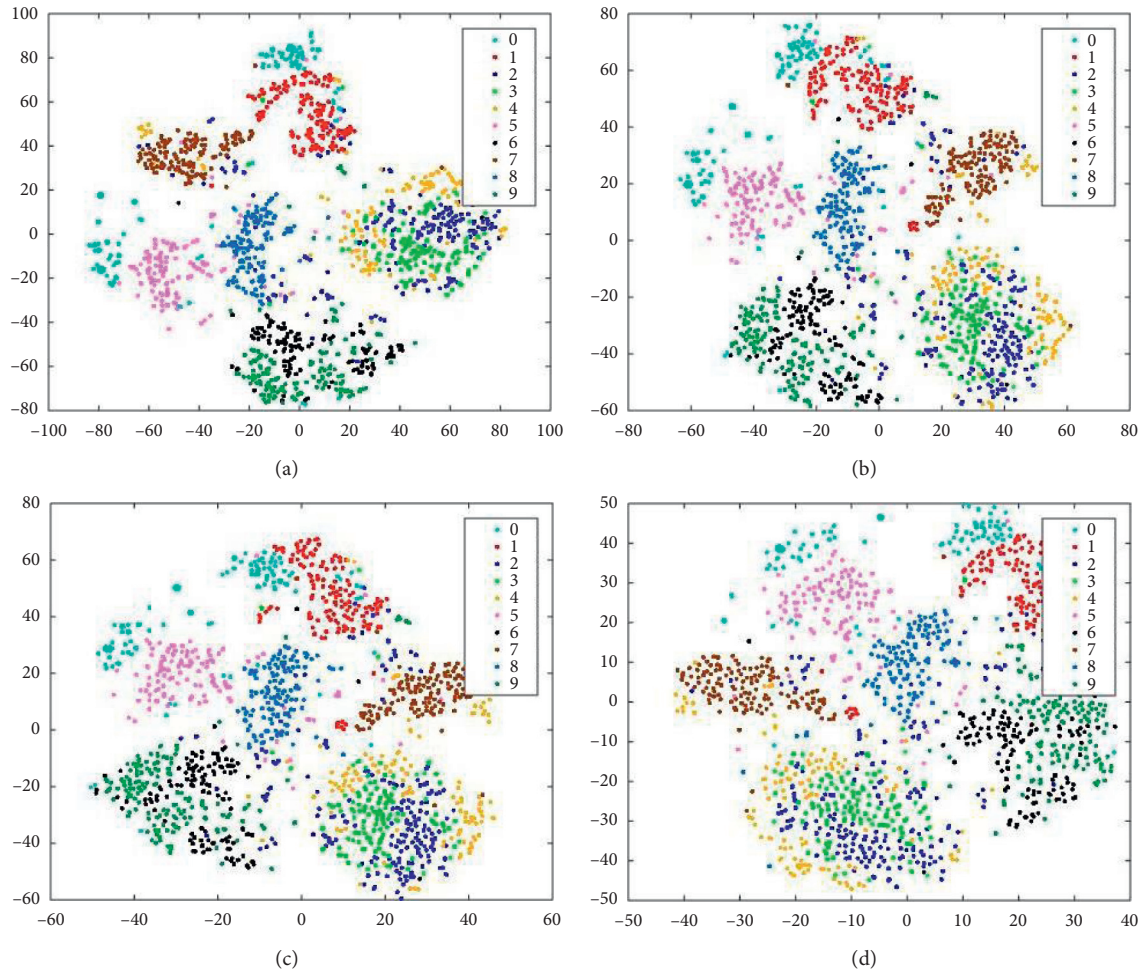


FIGURE 5: Standard  $t$ -SNE based visualization results of the data set containing pixel-based information only of the Urdu handwritten numerals. (a) Perplexity: 30 Iterations: 1,000. (b) Perplexity: 50 Iterations: 1,000. (c) Perplexity: 70 Iterations: 1, 000. (d) Perplexity: 100 Iterations: 1,000.

improvement in terms of separations among the clusters of each class of the Urdu handwritten numerals when compared with results using different perplexity values of 30, 50, and 100.

It is of general observation that, with lower perplexity value, the local structure of the data tends to show higher perseverance, i.e., the clusters having a smaller number of data points plotted very close to each other resulting in a compact visualization. On the other hand, the higher the perplexity value, the higher the perseverance in the global structure of the data; i.e., the data points will be plotted with some notable difference (intracluster difference) and also maintaining the separation between the clusters (intercluster difference).

Figure 6 depicts the detailed description of the structural features of both the Urdu numerals and characters. The results obtained by applying the standard  $t$ -SNE on the dataset having a combination of both the pixel- and structural-based features are shown in Figure 7. The results depicted much improvement by producing the clusters with clear separations. From the results presented in Figure 7, it can be observed easily that some of the Urdu numerals like 2,

3, and 4 have overlapping clusters. This overlapping is based on the fact that these Urdu numerals share much shape similarity. A similar behavior can also be witnessed in the case of Urdu numerals 0 and 1. It can be concluded that combining the pixel- (i.e., 784 features) and structural-based features (i.e., 10 features) failed to be useful when applied to the standard  $t$ -SNE algorithm.

The same approach of standard  $t$ -SNE is applied to the datasets of Urdu handwritten characters. One dataset contains the pixel-based data, and the other contains both the pixel and structural based features. It is pertinent to mention that only those Urdu characters are considered for the experiments that share much of the shape similarity. Figure 8 shows the Urdu characters grouped on the basis of shape similarity. The remaining characters are not considered to reduce the ink-noise ratio while visualizing the 40 characters individually. Therefore, it is better to visualize the characters grouped according to shape similarity.

Figure 9 shows the results of applying standard  $t$ -SNE on both the data set of Urdu handwritten characters mentioned earlier. It is pertinent to mention that the results shown are chosen among the better results produced by

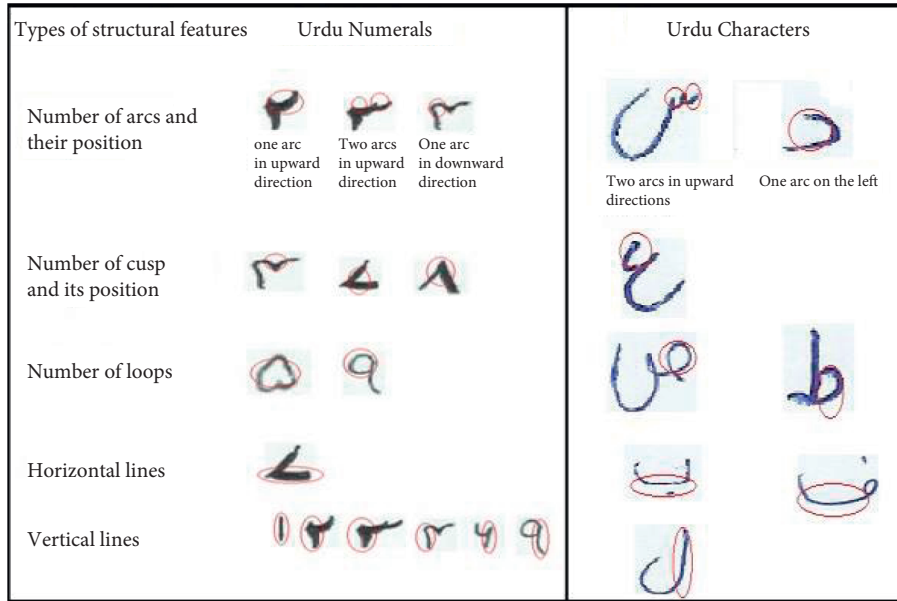


FIGURE 6: Types of structural features of Urdu numerics and characters.

fine-tuning the parameters. One of the results in Figure 9(a) shows that some of the clusters show higher overlapping than other clusters. This overlapping is due to shape similarity among the characters in Groups 2, 10, and 11. A similar behavior is observed in the characters of Groups 3 and 9; Groups 4 and 5; and Groups 6 and 7. Only Group 1, Group 8, and Group 12 characters are correctly drawn by  $t$ -SNE. This issue of overlapping is solved to some extent by embedding structural features of Urdu characters, as we have done with Urdu numerics. The results shown in Figure 9(b) depict the better results. The intracluster separation is better than the previous result. However, there is a need for modifying the standard  $t$ -SNE algorithm to make it capable of generating more precise results. To resolve these issues, a novel idea is proposed to build a fusion matrix having the pair-wise Euclidean distances of more (or multiple) independent observation spaces (i.e., pixel- and structural-based information). The standard  $t$ -SNE is then modified to assist the data in the fused matrix. The details about the fused data matrix and modified  $t$ -SNE are given in the subsequent section.

**5.2. Fused Data Matrix.** In this section, we discussed a novel way to embed the two or multiple observation spaces by calculating the pairwise Euclidean distances of the instances resulting in a fusion matrix. Furthermore, we also modified the standard  $t$ -SNE to make it able to assist the data in a fused matrix form. In our dataset, we build a single fusion matrix by calculating pairwise Euclidean distance of the data instances of the two independent spaces, i.e., pixel- and structure-based information. The resulting fusion matrix is then plugged into the modified  $t$ -SNE that makes use of both the features mentioned earlier. This modified  $t$ -SNE will give equal importance to both the features, thus generating even more clear and accurate clusters with precise separations.

Our assumption lies in the fact that since the data from both the independent spaces sources are highly conjunctive and dependent, therefore, their fusion will produce the more accurate visualization results in some low-dimensional space when the visualization results are compared with the output generated using either the pixel-based data or the structural data alone.

The pixel-based features are saved in a matrix form of size  $1 \times n$  for a single image of Urdu handwritten character, where  $n$  is the pixel-wise binary data for each image of size  $32 \times 3$ . If we consider Urdu numerals (for example), we have used 5000 images of Urdu handwritten numerals (500 each of ten numerals) for our experimental work; therefore, the dataset is of size  $500 \times 32$ . This pixel-based information is then embedded with the structural features, using the Euclidean Distance, of the Urdu handwritten numerals. Since the Urdu numerals share shape similarity, for example, digits two and three (shown in Figure 2), these structural based features are embedded to the pixel-based features to reduce the visualization issues while plotting similar shape images. Furthermore, we introduced equation (1) to balance the weighted combinations of both of the independent original spaces. It is pertinent to mention that  $t$ -SNE works on one of the tunable parameters called perplexity that can be thought of as “the number of neighboring points  $t$ -SNE must consider,” and we used different values of perplexity to encompass the whole data.  $t$ -SNE shrinks widespread data and expands densely packed data. It is, hence, suggested not to decide the size and density/spread/variance of the clusters based on the output. Furthermore, equation (1) is used for calculating the minimum value for the fused Euclidean distances that play a role in winning value for both the independent spaces. This novel activity helped us in performing the fusion process in an efficient and principled way that makes it practically possible for the independent spaces to contribute equally in order to maintain the separation of

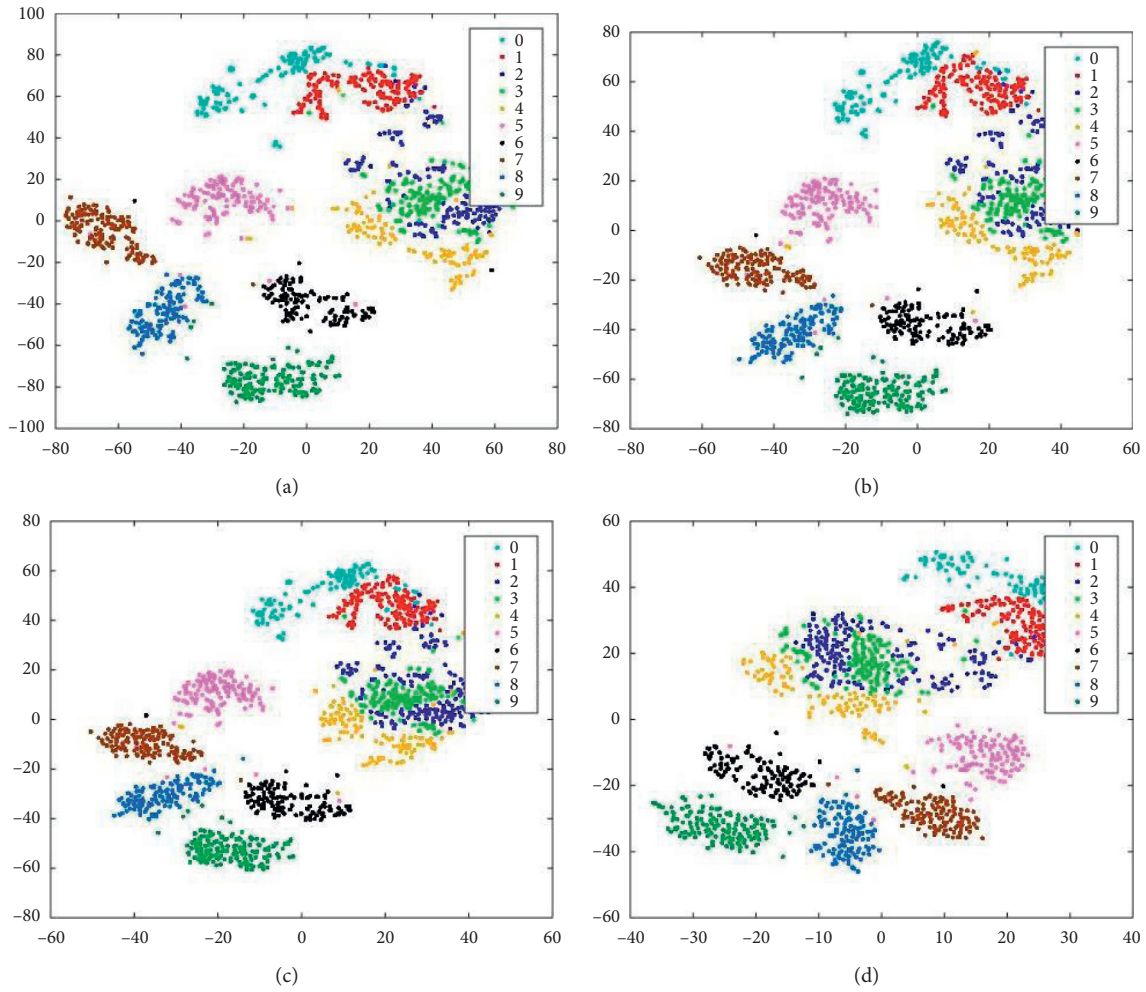


FIGURE 7: Standard (t)-SNE based visualization results of the combined data of pixel- and structure-based information of the Urdu handwritten numerals. (a) Perplexity: 30 Iterations: 1,000. (b) Perplexity: 50 Iterations: 1,000. (c) Perplexity: 70 Iterations: 1,000. (d) Perplexity: 100 Iterations: 1,000.

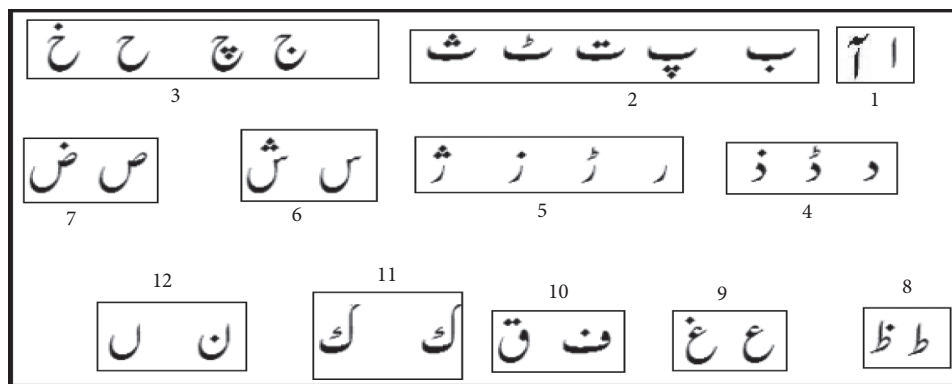


FIGURE 8: Grouping of Urdu characters having shape similarity.

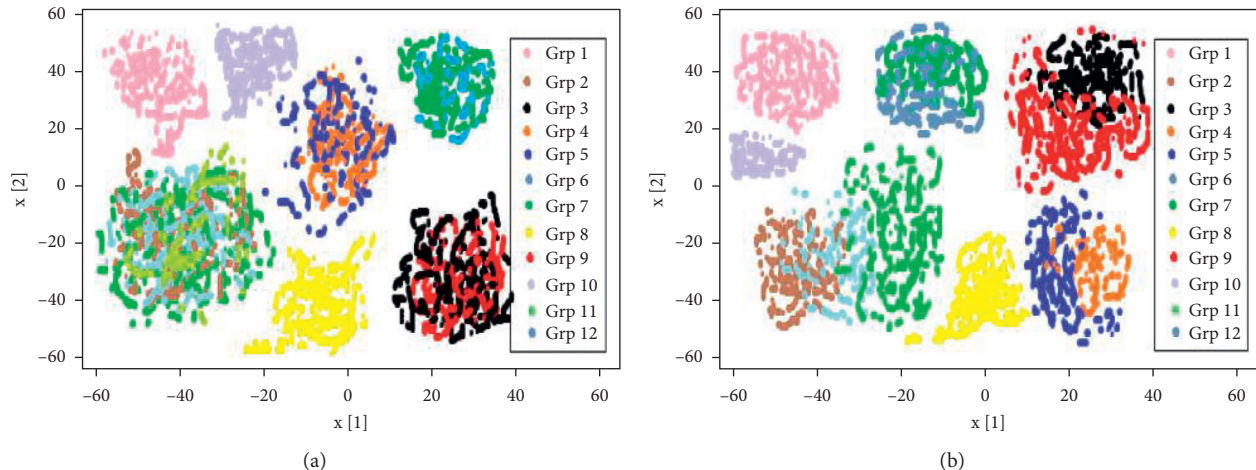


FIGURE 9: Visualization of (a) pixel-based and (b) structure-based data of our dataset of handwritten Urdu characters using standard  $t$ -SNE took 12 minutes 31 seconds. (a) Perplexity: 30 Iterations: 1,000. (b) Perplexity: 50 Iterations: 1,000.

the data instances within a cluster. In order to make the equal contribution of both the independent spaces, we assigned an equal weight ( $\alpha(t) = 0.5$ ) to both spaces.

$$\text{Euclid}_{\text{fused}}(a, b) = (t)\text{Euclid}(P_a, P_{\text{map}_b}) + (1 - \alpha(t))\text{Euclid}(S_a, S_{\text{map}_b}). \quad (1)$$

In equation (1), we computed the similarity patterns in a very disciplined way that are likely to exist in data instances of both the independent spaces where the pixel space value is represented by  $P$  and structural attribute by  $S$ . The relative weight,  $\alpha$ , depicts the relative weight among the similarities of the data instances of both the independent spaces. Whereas the  $t$  depicts the epoch number depicts the number of iterations for the dimensionality reduction process.  $\alpha$  is carefully set to 0.5 to observe the equal contribution of both of the independent spaces. This tuning helps in locating the minimum fused Euclidean distance ( $\text{Euclid}_{\text{fused}}(a, b)$ ), which in return determines the common successful unit by locating. It is pertinent to mention that in order to normalize the Euclidean distances of the two independent spaces (in equation (1)), we apply the product formula. This activity played a key role in improving the results by maintaining the intercluster separations while visualizing in low dimensions. In the next section, we discussed the results obtained by our modified  $t$ -SNE (Van Der Maaten et al. [8]); PCA (Roweis and Saul [13]); and AE (Liou et al. [26]).

The reason for reducing to two-dimensional space is to observe the behavior of the high data that assist in finding the patterns (if they exist). This activity guides the researchers to apply suitable set of classifiers. The resulting 2D features, in our case, are representing the  $(x, y)$  coordinates of each individual instance drawn by the  $t$ -SNE. These 2D features are correctly representing each instance on the map whether they are similar in shape or not. As a result, this information may be used for classification purpose using any

classifier. We used CNN again for reduced dimensional data, since it was used for classification using the pixel-based data only. It is pertinent to mention that 2D does not mean the kernel sliding window; it means that the CNN is accepting two inputs in case of reduced dimension.

**5.3. Complexity Comparison of Standard  $t$ -SNE and Our Modified  $t$ -SNE.** In the original source papers (Maaten and Hinton [9]; Van Der Maaten et al. [8]), the standard  $t$ -SNE's computational and memory cost is  $O(n^2)$ , where  $n$  is the number of data points, which constrain the application of the technique. We evolved the algorithm by reducing the computational complexity to  $O(n \log(n))$  and the memory complexity to  $O(n)$  since the data from both the independent distributions involve a normalization term that sums overall  $n \times (n - 1)$  pairs of unique objects (see Equation 1). It is also observed that the  $t$ -SNE scales quadratically in the number of objects  $n$ , and its applicability is limited to data sets with only a few thousand input objects.

**5.4. Experimental Results Obtained through PCA, AE, and Modified  $t$ -SNE.** In this section, we covered the visualization results of the fused matrix dataset of both the Urdu handwritten numerics and characters. The results showed (see Figure 10) that our modified  $t$ -SNE with fused data matrices of our dataset outperformed the classical approaches of PCA and AE.

Similarly, while visualizing the fused matrix of Urdu handwritten characters, we applied the same set of

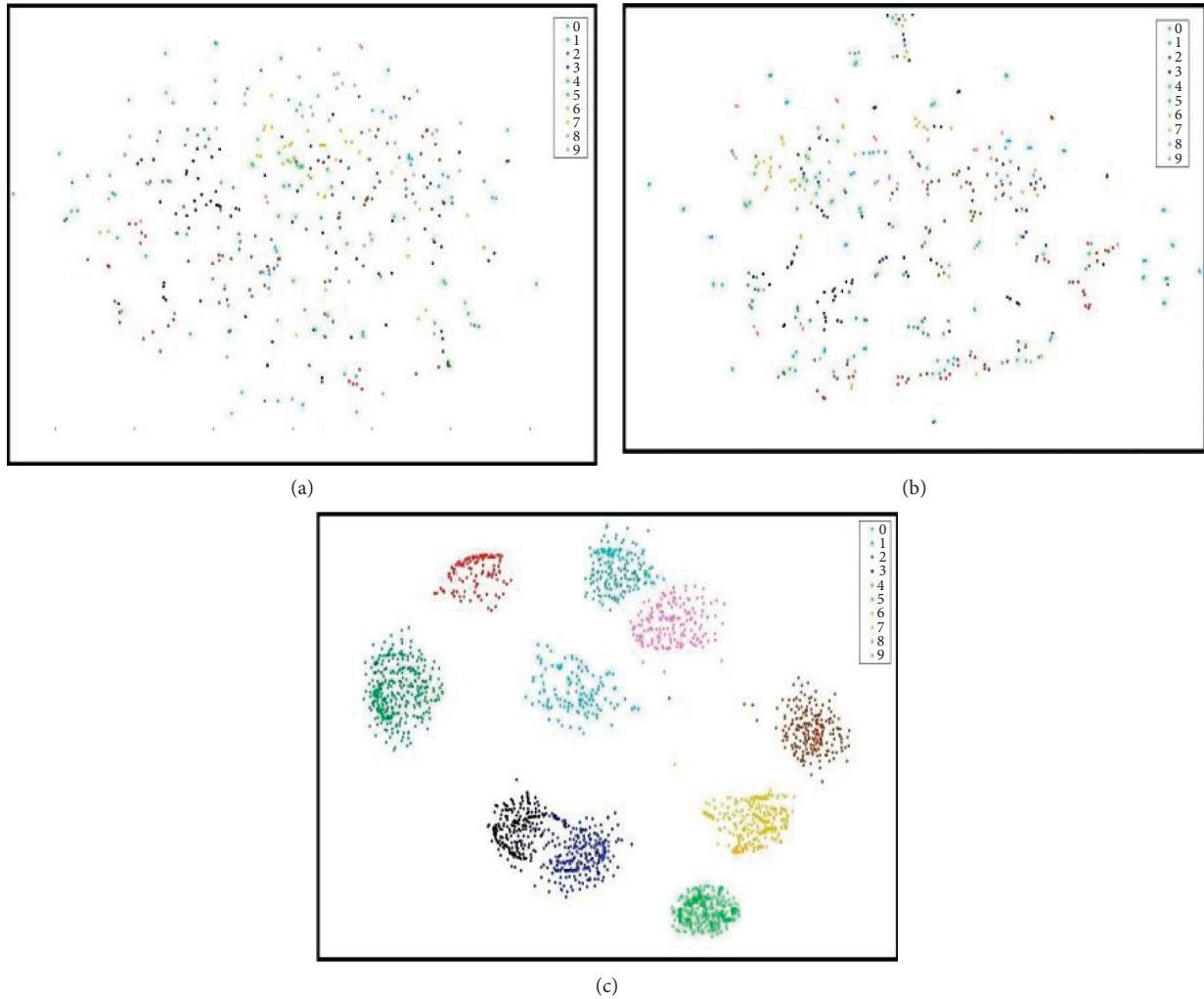


FIGURE 10: The visualization results of the high-dimensional fusion matrix data of the Urdu handwritten Urdu numerals using (a) autoencoders, (b) principal component analysis and (c) *t*-SNE.

algorithms having the same parameter settings. Figure 11 shows the visualization results produced when applied to Urdu handwritten characters.

## 6. Recognition of Urdu Handwritten Characters Using Deep Network

We make use of a deep convolutional neural network (CNN) model with an output layer generating the output on feature mapping in order to recognize the Urdu handwritten characters. CNN is one of the deep networks that are widely used in image classification problems and recognition because of its high accuracy. The CNN follows a hierarchical model, which works on building a network, like a funnel, and finally gives out a fully connected layer, where all the neurons are connected to each other, and the output is processed. Furthermore, we used 2D convolutional layers, which are ideal for processing 2D images. Compared to other image classification algorithms, CNNs actually use very little preprocessing. The key objective of our model is to classify the given input out of 10 classes of the Urdu

handwritten numerals. On the other hand, the same model will also be used in classifying the given Urdu character out of 12 classes of the Urdu handwritten characters (see Figure 8).

In research activities related to image processing, it was observed that CNN and its variants are most widely used. While dealing with two-dimensional images, we used 2 VGG16 model that is equipped with 16- and 19-layer network capable of dealing with a maximum input size of  $224 \times 224$ . It is considered to be one of the excellent vision model architectures till date. The most unique thing about VGG16 is that, instead of having a large number of hyperparameters, it focused on having convolution layers of  $3 \times 3$  filters with a stride 1 and always used the same padding and max-pool layer of  $2 \times 2$  filter of stride 2. While analyzing the high-dimensional data of our manuscript, we came along with the exploitation of both the structural- and pixel-based data in order to generate precise classification results. In order to resolve this issue, we make the standard *t*-SNE compatible with our data by implementing the pair-wise Euclidean distance formula to the data points of our dataset.

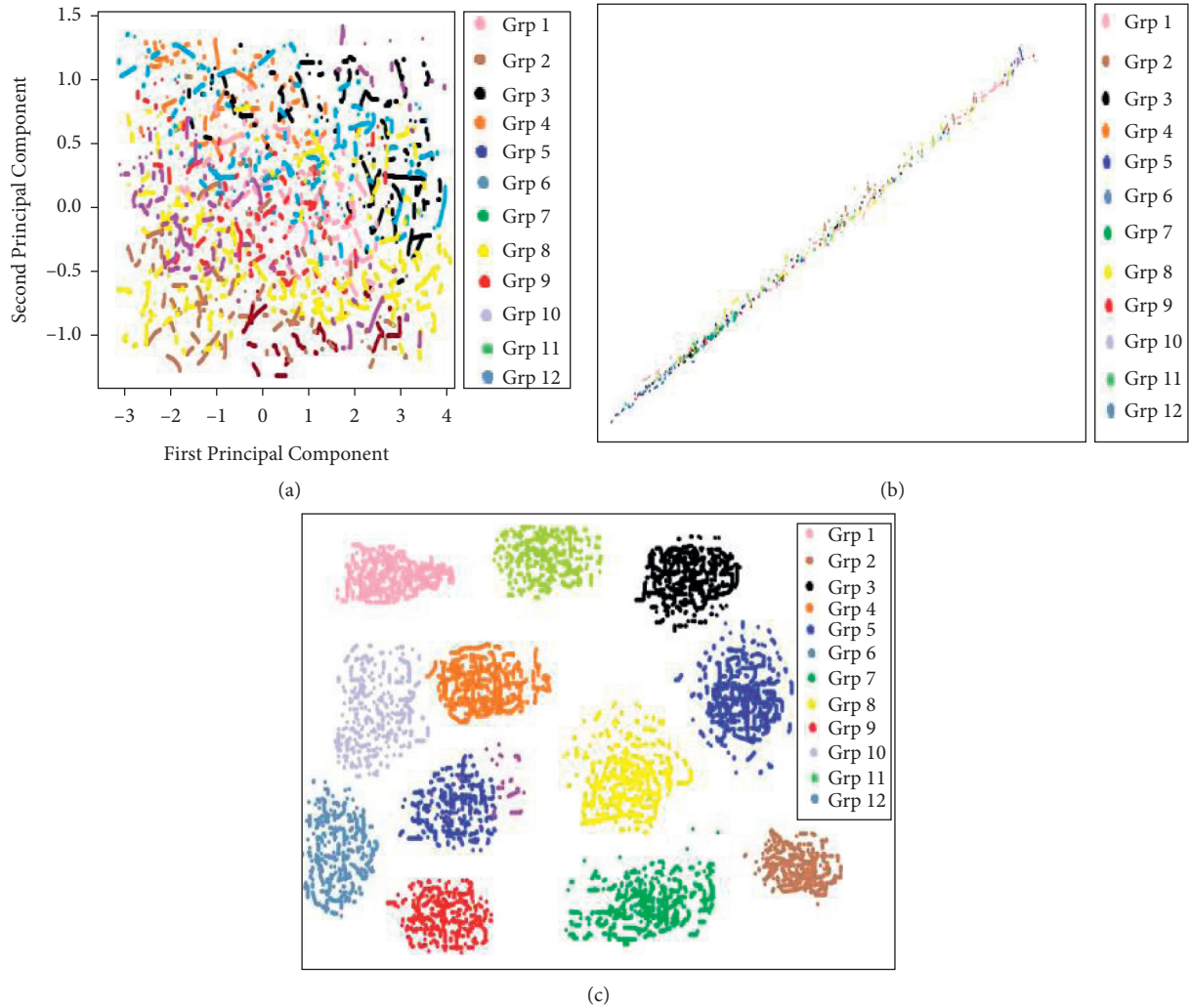


FIGURE 11: The visualization results of the high-dimensional fusion matrix data of the Urdu handwritten Urdu characters using (a) Autoencoders, (b) Principal component analysis, and (c)  $t$ -SNE.

This activity embedded the data points coming from two independent spaces in one space, thus making it compatible with the standard  $t$ -SNE.

It is noteworthy that we have not reduced the size of the image; rather, we have reduced the dimension of the feature-space, that is, the embedded version of structural and pixel-based features. These reduced dimensions are produced by the dimensionality reduction approaches discussed in detail in the sections above. This reduced-dimensional data is then plugged into the proposed model of CNN in order to recognize the numeric (or character) data. This activity takes minimum time (12 CPU seconds) and reported same accuracy rate in the classification of both the Urdu handwritten characters and numerals as compared with the same model applied on the original dimensions of the text images, reported in our work (Husnain et al. [27]) that takes 8 minutes. It is pertinent to mention that there is no need to increase the number of convolutional cores of the proposed model as we have performed in our previous work (Husnain et al. [27]) since

the dimensions of the input data are too small, and they can be handled trivially by the original model of CNN.

In order to decrease the ambiguity in the quality of results, we performed a series of experiments using different variations of the  $n$ -fold cross-validation. This activity helped in retreating the confusion among the biased results obtained through the conventional ratio of training and testing data. Tables 1 and 2 depict the confusion matrices for Urdu handwritten numerals, showing an average accuracy of 96.5% and 94.7%, respectively.

Similarly, Tables 3 and 4 show the results of Urdu handwritten characters (shown in groups in Figure 8). The results showed that our proposed model of CNN outperformed the previous approaches to perform this task, see Table 5. We also presented the comparison of the results produced by our proposed model with some state-of-the-art related approaches for the same task in Table 5. It can be observed that our approach is significantly better in terms of the number of parameters, accuracy, the number of dimensions used, and the amount of calculation.



TABLE 1: The confusion matrix showing accuracy rate of classification of Urdu handwritten numeral using 10-fold cross validation.

Eight	Five	Four	Nine	One	Seven	Six	Three	Two	Zero	Classified
8	5	4	9	1	7	6	3	2	0	As
<b>95%</b>	3%	0	0	0	0	0	0	0	2%	Eight
2%	<b>97%</b>	0	0	0	0	0	0	0	1%	Five
0	0	<b>96%</b>	0	0	0	1%	2%	1%	0	Four
0	0	0	<b>97%</b>	2%	0	1%	0	0	0	Nine
0	0	0	1%	<b>98%</b>	0	1%	0	0	0	One
0	2%	0	0	0	<b>98%</b>	0	0	0	0	Seven
0	0	0	0	3%	0%	<b>95%</b>	0	0	2%	Six
0	0	2%	0	0	0	0	<b>97%</b>	1%	0	Three
0	0	1%	0	3%	0	0	2%	<b>94%</b>	0	Two
1%	1%	0	0	0	0	0	0	0	<b>98%</b>	Zero

The values in bold (in diagonal) are representing the accuracy percentage of the classification of Urdu handwritten numerals.

TABLE 2: The confusion matrix showing accuracy rate of classification of Urdu handwritten numeral using 8-fold cross validation.

Eight	Five	Four	Nine	One	Seven	Six	Three	Two	Zero	Classified
8	5	4	9	1	7	6	3	2	0	As
<b>94%</b>	3%	0	0	0	0	0	0	0	3%	Eight
3%	<b>95%</b>	0	0	0	0	0	0	0	2%	Five
0	0	<b>94%</b>	0	0	0	2%	3%	1%	0	Four
0	0	0	<b>95%</b>	3%	0	2%	0	0	0	Nine
0	0	0	3%	<b>96%</b>	0	1%	0	0	0	One
0	4%	0	0	0	<b>96%</b>	0	0	0	0	Seven
0	0	0	2	4%	0	<b>93%</b>	0	0	3%	Six
0	0	2%	0	0	0	0	<b>96%</b>	2%	0	Three
0	0	3%	0	4%	0	0	1%	<b>92%</b>	0	Two
1%	3%	0	0	0	0	0	0	0	<b>96%</b>	Zero

The values in bold (in diagonal) are representing the accuracy percentage of the classification of Urdu handwritten numerals obtained by applying using variations of n-fold cross-validation approach.

TABLE 3: The confusion matrix showing accuracy rate of classification of Urdu handwritten characters using 10-fold cross validation.

Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Classified
1	2	3	4	5	6	7	8	9	10	11	12	As
<b>93%</b>	0	0	1%	2%	0	0	2%	1%	1%	0	0	Grp 1
0	<b>89%</b>	0	3%	0	0	0	3%	2%	1%	1%	1%	Grp 2
0	2%	<b>92%</b>	0	2%	1%	0	0	0	0	2%	1%	Grp 3
2%	0	3%	<b>91%</b>	0	2%	0	2%	0	0	0	0	Grp 4
2%	0	5%	0	<b>93%</b>	0	0	0	0	0	0	0	Grp 5
0	0	2%	3%	2%	<b>90%</b>	0	0	2%	0	1%	0	Grp 6
0	0	0	0	0	0	<b>93%</b>	4%	0	0	1%	2%	Grp 7
0	2%	3%	0	0	0	0	<b>87%</b>	4%	1%	1%	2%	Grp 8
3%	1%	0	0	0	0	0	0	<b>89%</b>	0	3%	4%	Grp 9
0	2%	0	0	2%	0	0	3%	1%	<b>90%</b>	0	2%	Grp 10
0	1%	0	0	1%	0	0	0	3%	0	<b>91%</b>	0	Grp 11
3%	2%	0	0	0	1%	1%	0	2%	0	1%	<b>90%</b>	Grp 12

Following are the reasons behind using  $k$ -fold cross validation, and its variants are that the computation time is reduced as we repeated the process only 10 times when the value of  $k$  is 10. It also reduces the biasness in the results when using the conventional 70-30 training-testing ratio, thus limiting the making the classifier to strictly select the data points from the specified training data. Furthermore, every data point gets to be tested exactly once and is used in the training process for at least  $k - 1$  times. Similarly, the variance of the resulting estimate is reduced as the value of  $k$  increases, that is, the reason we make use of 10- and 8-fold cross validation to observe the change in variance.

Our proposed model was found quite efficient (in terms of accuracy) and effective also in performing the recognition and classification tasks among the approaches used so far for the same task. The novelty of our work lies in the fact that the reduced dimensions obtained through different dimensionality reduction approaches are used for the first time for the recognition of the Urdu handwritten characters. Furthermore, our proposed approach is equally applicable for developing an efficient system for both online and offline character recognition for mobile (or handheld) devices for learning applications for children.

TABLE 4: The confusion matrix showing accuracy rate of classification of Urdu handwritten characters using 8-fold cross validation.

Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Grp	Classified
1	2	3	4	5	6	7	8	9	10	11	12	As
<b>91%</b>	3%	0	0	0	0	2%	0	2%	1%	0	1%	Group 1
0	<b>85%</b>	4%	0%	1%	3%	0	2%	0	1%	3%	1%	Group 2
0	0	<b>90%</b>	3%	0	1%	2%	0	0	0	2%	2%	Group 3
1%	1%	2%	<b>89%</b>	0	3%	0	2%	0	1%	0	1%	Group 4
0	0	1%	0	<b>95%</b>	0	1%	0	1%	0	1%	1%	Group 5
0	0	2%	0	0	<b>94%</b>	2%	0	2%	0	0	0	Group 6
0	0	0	0	0	0	<b>93%</b>	4%	0	0	1%	2%	Group 7
0	0	0	0	2%	1%	0	<b>94%</b>	0	1%	1%	1%	Group 8
0	0	1%	0	3%	0	3%	0	<b>87%</b>	0	2%	4%	Group 9
0	0	0	3%	2%	0	2%	0	0	<b>90%</b>	1%	2%	Group 10
0	0%	3%	0	0	0	0	0	4%	0	<b>90%</b>	3%	Group 11
0	0	0	3%	0	2%	1%	0	1%	0	0	<b>93%</b>	Group 12

The values in bold (in diagonal) are representing the accuracy percentage of the classification of Urdu handwritten characters obtained by applying using variations of the n-fold cross-validation approach.

TABLE 5: Comparison Analysis of our proposed approach for Urdu handwritten character classification with state-of-the-art approaches.

Reference	Approach	Features	Accuracy (%)
Accuracy rate of classification of the Urdu handwritten character			
Ali et al. [28]	Neural network	Geometrical strokes	75–80%
Haider and Khan [29]	BPNN, PNN	Geometrical strokes	66%
Shahzad et al. [30]	Linear classifier	Statistical features	66%
Ahmed et al. [31]	BLSTM	Pixel-based	92–94%
Ko and Poruran [32]	SVM	Transfer-learning features	82.30%
Ali et al. [33]	SVM	Pixel-based features	95.79%
Our approach	CNN	Pixel- and geometrical-based	96.04%
Accuracy rate of classification of Urdu handwritten numeral			
Borse and Ansari [34]	Daubechies wavelet	Pixel-based	92.05%
Razzak et al. [35]; Razzak et al. [36]	HMM, fuzzy rule	Pixel-based	97.45%, 97.09%
Sarkhel et al. [37]	Multi-column multi-scale CNN	Non-explicit feature based approach	98.90%
Takruri et al. [38]	(MMCNN) fuzzy C-Means classifier ANN	Pixel-based features	88.00%
Said et al. [39]	ANN	Pixel-based features	94.00%
Mowlaei et al. [40]	ANN	Wavelet-based features	97.34%
Our approach	CNN	Pixel- and geometrical-based	99.01%

## 7. Conclusion

In this paper, we made use of the reduced dimensions obtained through the dimensionality reduction approaches like PCA, AE, and  $t$ -SNE, in recognition of Urdu handwritten characters and numerals. Furthermore, the structural features of each handwritten character are extracted and embedded in pixel-based features to enrich the features of our dataset. In order to make it more compatible, we modified the standard  $t$ -SNE by including equations that support the pairwise Euclidean distances of the features from two independent spaces. This modification results in accelerating the efficiency of standard  $t$ -SNE by producing a quite better low-dimensional data that eventually helped in visualizing both the Urdu handwritten characters and numerals. Furthermore, this reduced dimensional data is fed to the CNN model to recognition purposes. The results produced are quite similar to our previous work, in which we used all the dimensions of the text-images. The only difference is the time efficiency shown by our approach that took about 12 CPU seconds as compared

to our previous work (Husnain et al. [27]) that consumed 12 CPU minutes in producing the confusion matrices. Hence, it can be concluded that, to make the task of classification/recognition of high-dimensional data, it is better to apply a suitable dimensionality reduction approach that will show the faithful representation of the data. Then, plug this low-dimensional data into any machine learning classifier for testing/training to perform recognition/classification tasks. The limitation of our proposed  $t$ -SNE is that the algorithms can only be used to embed or fuse data coming from two or three independent spaces. Generalizations to a higher number of independent spaces are practically not possible since the computation time increases while computing pairwise distances among the high-dimensional data instances.

To the best of our knowledge, a very limited work is observed in the field of handwritten text recognition at character level, and the dataset of this kind is not available to date. Our results are the initiatives towards classification of the handwritten text at character level in the Urdu script, and there may be some lack of quality and comprehensiveness.

Our future work will encompass the recent trends and resolve these issues observed in our current work.

Furthermore, we have also created a state-of-the-art dataset containing the Urdu handwritten characters and numerals; to the best of our knowledge, there is no publicly available dataset of such kind. The existing datasets of Urdu handwritten text mainly consist of the Urdu handwritten words and sentences. The usage of these existing datasets cannot be used efficiently for recognition of the Urdu text at the character level. We also presented a comparative analysis of the results obtained through different approaches to propose recommendations based on parameter tuning. It is also concluded that the deep network can help in performing the recognition and classification task of the handwritten text of cursive scripts in minimum time. Furthermore, our approach also helps in providing a platform to the researchers and developers to develop the applications for the children to learn how to write Urdu (and other cursive languages) characters and numerals with higher accuracy. As mentioned earlier, there is also a deficiency of some standard data repository in the Urdu domain for generating and comparing the benchmark results. In order to bridge this gap, we are working on generating and extending our dataset that will be published publicly in the near future.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no potential conflicts of interest.

## Acknowledgments

This study was supported by the China University of Petroleum-Beijing and Fundamental Research Funds for Central Universities under Grant no. 2462020YJRC001.

## References

- [1] N. Saeed, H. Nam, M. I. U. Haq, and D. B. Muhammad Saqib, "A survey on multidimensional scaling," *ACM Computing Surveys*, vol. 51, no. 3, p. 47, 2018.
- [2] A. Buja, D. Cook, and D. F. Swayne, "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 78–99, 1996.
- [3] L. H. Nguyen and S. Holmes, "Ten quick tips for effective dimensionality reduction," *PLoS Computational Biology*, vol. 15, no. 6, Article ID e1006907, 2019.
- [4] L. Song, H. Ma, M. Wu, Z. Zhou, and M. Fu, "A brief survey of dimension reduction," in *Proceedings of the International Conference on Intelligent Science and Big Data Engineering*, pp. 189–200, Nanjing, China, October 2018.
- [5] M. H. ur Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, "Big data reduction methods: a survey," *Data Science and Engineering*, vol. 1, no. 4, pp. 265–284, 2016.
- [6] D. Engel, L. Hu'ttenberger, and B. Hamann, "A survey of dimension reduction methods for high-dimensional data analysis and visualization," in *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011*, Kaiserslautern, Germany, March 2012.
- [7] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," 2014, <https://arxiv.org/abs/1403.2877>.
- [8] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *Journal of Machine Learning Research*, vol. 10, no. 66–71, p. 13, 2009.
- [9] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [10] F. Camastra, "Data dimensionality estimation methods: a survey," *Pattern Recognition*, vol. 36, no. 12, pp. 2945–2954, 2003.
- [11] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.
- [12] F. Tsai, "Comparative study of dimensionality reduction techniques for data visualization," *Journal of Artificial Intelligence*, vol. 3, no. 3, pp. 119–134, 2010.
- [13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [14] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, Springer, Berlin, Germany, 2005.
- [15] L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee, "Spectral methods for dimensionality reduction," *Semi-supervised Learning*, pp. 293–308, 2006.
- [16] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.
- [17] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, pp. 585–591, 2002.
- [18] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the Twenty-First International Conference on Machine Learning*, p. 106, Banff, Canada, July 2004.
- [19] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems*, pp. 857–864, 2003.
- [20] P. Demartines and J. Hérault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, 1997.
- [21] L. Song, A. Gretton, K. Borgwardt, and A. J. Smola, "Colored maximum variance unfolding," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1385–1392, Vancouver, Canada, December 2008.
- [22] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, "Handwritten Urdu character recognition using one-dimensional blstm classifier," *Neural Computing & Applications*, vol. 31, no. 4, pp. 1143–1151, 2019.
- [23] N. Das, J. M. Reddy, R. Sarkar et al., "A statistical-topological feature combination for recognition of handwritten numerals," *Applied Soft Computing*, vol. 12, no. 8, pp. 2486–2495, 2012.
- [24] M. Husnain, S. Mumtaz, M. Coustaty, M. Luqman, J.-M. Ogier, and S. Malik, "Urdu handwritten text recognition: a survey," *IET Image Processing*, vol. 14, no. 11, 2020.
- [25] M. W. Sagheer, C. L. He, N. Nobile, and C. Y. Suen, "A new large Urdu database for off-line handwriting recognition," in

- Proceedings of the International Conference on Image Analysis and Processing*, pp. 538–546, Vietri sul Mare, Italy, September 2009.
- [26] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, “Autoencoder for words,” *Neurocomputing*, vol. 139, pp. 84–96, 2014.
- [27] M. Husnain, M. M. Saad Missen, S. Mumtaz et al., “Recognition of Urdu handwritten characters using convolutional neural network,” *Applied Sciences*, vol. 9, no. 13, p. 2758.
- [28] A. Ali, M. Ahmad, N. Rafiq, J. Akber, U. Ahmad, and S. Akmal, “Language independent optical character recognition for hand written text,” in *Proceedings of the 8th International Multitopic Conference INMIC 2004*, pp. 79–84, Lahore, Pakistan, December 2004.
- [29] I. Haider and K. U. Khan, “Online recognition of single stroke handwritten Urdu characters,” in *Proceedings of the IEEE 13th International Multitopic Conference, 2009. INMIC 2009*, pp. 1–6, Islamabad, Pakistan, December 2009.
- [30] N. Shahzad, B. Paulson, and T. Hammond, “Urdu QAEDA: recognition system for isolated Urdu characters,” in *Proceedings of the IUI Workshop on Sketch Recognition*, Sanibel Island, Florida, February 2009.
- [31] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, “Handwritten Urdu character recognition using one-dimensional BLSTM classifier,” *Neural Computing & Applications*, pp. 1–9, 2017.
- [32] M. A. Ko and S. Poruran, “OCR-nets: variants of pre-trained CNN for Urdu handwritten character recognition via transfer learning,” *Procedia Computer Science*, vol. 171, pp. 2294–2301, 2020.
- [33] H. Ali, A. Ullah, T. Iqbal, and S. Khattak, “Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network,” *SN Applied Sciences*, vol. 2, no. 2, pp. 1–12, 2020.
- [34] R. Borse and I. Ansari, *Offline Handwritten and Printed Urdu Digits Recognition Using Daubechies Wavelet*, ER Publication, New Delhi, India, 2017.
- [35] M. I. Razzak, S. Hussain, A. Beläid, and M. Sher, “Multi-font numerals recognition for Urdu script based languages,” *International Journal of Recent Trends in Engineering*, 2009.
- [36] M. I. Razzak, S. Hussain, and M. Sher, “Numeral recognition for Urdu script in unconstrained environment,” in *Proceedings of the International Conference on Emerging Technologies 2009 ICET 2009*, pp. 44–47, Islamabad, Pakistan, October 2009.
- [37] R. Sarkhel, N. Das, A. Das, M. Kundu, and M. Nasipuri, “A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts,” *Pattern Recognition*, vol. 71, pp. 78–93, 2017.
- [38] M. Takruri, R. Al-Hmouz, A. Al-Hmouz, and M. Momani, “Fuzzy c means based hybrid classifiers for offline recognition of handwritten indian (Arabic) numerals,” *International Journal of Applied Engineering Research*, 2015.
- [39] F. N. Said, A. Yacoub, and C. Y. Suen, “Recognition of English and Arabic numerals using a dynamic number of hidden neurons,” in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR’99 (cat. no. pr00318)*, pp. 237–240, Bangalore, India, September 1999.
- [40] A. Mowlaei, K. Faez, and A. T. Haghghat, “Feature extraction with wavelet transform for recognition of isolated handwritten farsi/Arabic characters and numerals,” in *Proceedings of the 2002 14th International conference on Digital Signal Processing DSP 2002 (cat. no.02 th8628)*, vol. 2, pp. 923–926, Piscataway, NJ, USA, July 2002.

## Research Article

# Research on the Capability Maturity Evaluation of Intelligent Manufacturing Based on Firefly Algorithm, Sparrow Search Algorithm, and BP Neural Network

Li Shi <sup>1,2</sup>, Xuehong Ding <sup>2</sup>, Min Li,<sup>2</sup> and Yuan Liu<sup>3</sup>

<sup>1</sup>School of Management, Hefei University of Technology, Hefei 230009, China

<sup>2</sup>School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China

<sup>3</sup>College of Economics and Management, Huaibei Normal University, Huaibei 235000, China

Correspondence should be addressed to Xuehong Ding; shili@chnu.edu.cn

Received 1 February 2021; Revised 5 July 2021; Accepted 30 July 2021; Published 21 August 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Li Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent manufacturing capability evaluation is the key for enterprises to scientifically formulate the implementation path and continuously improve the level of intelligent manufacturing. To help manufacturing enterprises diagnose the level of intelligent manufacturing capability, this paper conducts research on intelligent manufacturing capability maturity evaluation based on maturity theory. The evaluation problem is a complex nonlinear problem, and BP neural network is particularly suitable for solving such complex mapping problems. Aiming at the problem that the BP neural network is sensitive to initial weights and thresholds, the sparrow search algorithm (SSA) is used to optimize the initial weights and thresholds of the BP neural network. In order to overcome the shortcoming of SSA that it is easy to fall into the local optimum, the firefly disturbance strategy is introduced to improve it, a new sparrow search algorithm (FASSA) is proposed, and on this basis, an intelligent manufacturing capability maturity evaluation model based on the FASSA-BP algorithm is constructed. Finally, a large battery manufacturing enterprise in China is selected for empirical research, and the comparison experiments are carried out on the FASSA-BP model, BP model, SSA-BP model, and PSO-BP model in terms of accuracy, stability, etc. The results show that the evaluation of intelligent manufacturing capability maturity through this model can effectively help companies diagnose problems in the construction of intelligent manufacturing and provide a reference for companies to accurately improve their intelligent manufacturing capabilities.

## 1. Introduction

With the deep integration of information technology, intelligent technology, and manufacturing technology, intelligent manufacturing has received great attention from governments and industries [1–3]. As the core of the new round of industrial revolution, intelligent manufacturing is leading the intelligent transformation of manufacturing enterprises globally, and the evaluation of intelligent manufacturing capability has become the key for enterprises to scientifically formulating the implementation path and continuously improving the level of intelligent manufacturing. In the implementation and promotion of

intelligent manufacturing, many enterprises do not know enough about their overall intelligent manufacturing development level and lack effective methods to diagnose intelligent manufacturing capability and identify gaps. How to carry out evaluation for enterprise intelligent manufacturing capability has become a new research area of concern for scholars at home and abroad, with the study of intelligent manufacturing capability evaluation based on maturity theory having become an important research direction.

The concept of intelligent manufacturing capability maturity (IMCM) is not yet uniformly defined. Different organizations and institutions have given their

understanding from different perspectives. De Carolis et al. evaluate the key processes of intelligent manufacturing from five different dimensions and, inspired by the integration framework of capability maturity models, propose an intelligent manufacturing capability maturity model [4]. Based on the “Made in China 2025” plan and combined with capability maturity model theory, the intelligent manufacturing maturity model was studied by Hu and Gao based on China’s national conditions from three aspects: maturity level, capability factor, and maturity requirements [5]. Zhou et al. studied the issues related to the transformation and upgrading of intelligent manufacturing in Chinese enterprises and found that Chinese manufacturing enterprises were able to develop their own intelligent manufacturing capability upgrading paths based on their own capabilities and industry characteristics [6]. From the perspective of smart factories, Ren et al. firstly proposed the capability maturity model of China’s petrochemical industry [7]. Simetinger and Zhang conducted a comparative analysis of several important maturity models to identify common features of Industry 4.0 maturity models [8].

The maturity of intelligent manufacturing capability can reflect the development level of intelligent manufacturing of enterprises, but it also needs to be evaluated with specific evaluation methods. Researchers from different countries have explored the problem of evaluating the maturity of intelligent manufacturing capabilities and the construction of evaluation models.

Wagire et al. used fuzzy hierarchical analysis to determine an Industry 4.0 maturity assessment model and implemented an evaluation of an automotive parts manufacturing company based on this, and the study showed that the model is feasible in practical application and easy to self-evaluate [9]. Ruiz et al. established a capability maturity evaluation model by fuzzy integrated evaluation method to clarify the level of enterprise intelligent manufacturing [10]. Colli et al. proposed a new 360 digit maturity assessment method based on the learning theory model, and the results of the study showed that the method can provide actionable recommendations for enterprises, making the results of Industry 4.0 capability maturity evaluation concrete [11]. Lee et al. constructed a smart factory assessment framework based on the concept of operations management, used network analysis to analyze and evaluate it, and verified its effectiveness through case studies [12]. Liu et al. explored the research issue about carbon market maturity evaluation in depth and proposed the entropy-based TOPSIS model to measure the maturity level of carbon market [13]. To determine the development level of intelligent manufacturing maturity, Xiao et al. combined hierarchical analysis with Hopfield neural network and conducted an empirical study based on it, so as to clarify the usability of the proposed method [14]. Xu and Xiao used the fuzzy comprehensive evaluation method to carry out the study of intelligent manufacturing capability [15]. Meanwhile, Zhi-qiang used the mutation level method in order to achieve the metric of intelligent manufacturing capability [16]. Shao and Wen used factor analysis method

as an evaluation method of intelligent manufacturing capability and used some Chinese provinces as a research sample to obtain the capability ranking of each province after evaluation [17].

The above-mentioned scholars have conducted research on the maturity of intelligent manufacturing capability from different perspectives, providing a certain theoretical basis and methodological reference for the evaluation of intelligent manufacturing capability maturity. Although certain research results have been achieved, the overall research is still in the initial stage of exploration; in particular, in the evaluation method, there is still room for improvement. The concrete performance is that the existing research mostly adopts traditional evaluation methods such as analytic hierarchy process (AHP) and factor analysis. Although the original information of most indicators is retained, the evaluation process is obviously subjective due to the influence of expert experience in determining the weight. In addition, these methods still have some defects, such as heavy workload, complex calculation, and being time-consuming.

In order to overcome the shortage of traditional evaluation methods of capability maturity, this paper applies intelligent algorithms in the field of computer science; on the basis of constructing evaluation indexes of enterprise intelligent manufacturing capability maturity, uses intelligent methods such as sparrow search algorithm (SSA), firefly algorithm (FA), and BP (back propagation) neural network; proposes an intelligent manufacturing capability maturity evaluation model based on FASSA-BP algorithm; and applies it to a large battery manufacturing enterprise in China for IMCM evaluation research. Through comparative experiments, the validity and applicability of the evaluation model are verified.

## 2. Constructing the IMCM Evaluation Index

Combining intelligent manufacturing capability and existing maturity theory and referring to the documents related to intelligent manufacturing capability maturity released at home and abroad [2, 3], this paper considers that intelligent manufacturing capability maturity is a comprehensive measure of the whole process of intelligent manufacturing in enterprises and describes the evolution path of intelligent manufacturing from unplanned level to leading level.

Based on the above-mentioned understanding of the maturity of intelligent manufacturing capabilities, following the construction principles of the indicator system [18, 19], and according to the industry characteristics of intelligent manufacturing enterprises, we construct the evaluation index system of intelligent manufacturing capabilities, providing methods for enterprises to carry out self-diagnosis of intelligent manufacturing capabilities. The system includes one first-level index of strategy and organization, design, production, equipment, warehouse, sales, service, network environment, network security, and architecture platform and refines 20 second level indexes to obtain the index system shown in Figure 1.

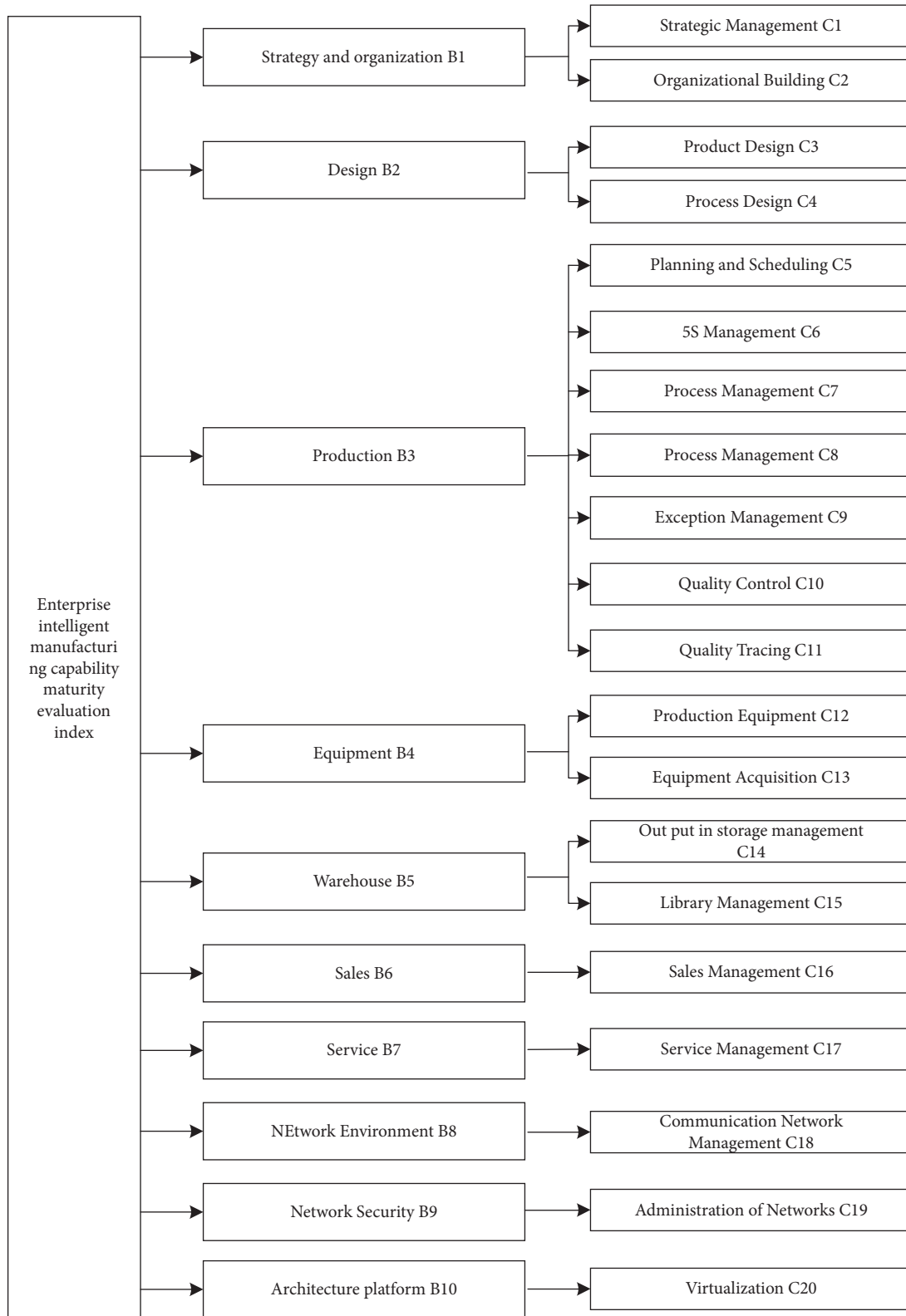


FIGURE 1: Intelligent manufacturing capability evaluation index system.

Through establishing the evaluation index system, the evaluation of the maturity of the enterprise's intelligent manufacturing capability can be based on the evidence. We

use the evaluation index system to design questionnaires to collect sample data. Firstly, the questionnaire questions are designed according to the 20 secondary indicators that have

been constructed; that is, each secondary indicator corresponds to one question. There are 20 multiple-choice questions in total; each question has six options of ABCDEF, and each option corresponds to a score of 0, 20, 40, 60, 80, 100. The collected sample data is the score of each question in the questionnaire, that is, the score of each secondary evaluation index. Therefore, after each questionnaire is filled out by each person, the selected option is converted into the corresponding score to get the sample data.

### 3. Improving the Sparrow Search Algorithm

**3.1. Basic Sparrow Search Algorithm.** The sparrow search algorithm is an intelligent optimization algorithm based on the foraging and antipredation behaviors of sparrow populations proposed by Xue and Shen in 2020 [20]. It has three behavioral strategies: producer searching for food [21], follower acquiring food through producer [22], and antipredation [23]. Among them, the producer is responsible for finding food and directing the flow of the whole population. Once an individual finds a predator, it signals an alarm, and when the alarm value is greater than the safety threshold, the producer directs all followers to a safe area. During each iteration, the producer's position is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right) \cdot (0.2), & \\ X_{i,j}^t + Q \cdot L \cdot \text{if } R_2 \geq ST. & \end{cases} \quad (1)$$

In the formula,  $t$  represents the current iteration number,  $j=1, 2, \dots, d$ .  $X_{i,j}^t$  represents the value of the  $j$ th dimension of the  $i$ th sparrow when the number of iterations is  $t$ .  $\text{iter}_{\max}$  is the constant with the maximum number of iterations.  $\alpha \in (0, 1)$  is a random number.  $R_2 (R_2 \in [0, 1])$  and  $ST (ST \in [0.5, 1.0])$  denote the alarm value and safety threshold.  $Q$  is a random number obeying normal distribution.  $L$  denotes a  $1 \times d$  matrix, where each element is 1. When  $R_2 < ST$ , this means that there are no predators around the population and the producer enters the wide-area search mode. If  $R_2 \geq ST$ , this means that some sparrows have found predators and all sparrows need to fly quickly to other safe areas.

For followers, sparrows that are in a very hungry position forage elsewhere, while some followers constantly monitor producers and compete for food to increase their own predation rate. The updated formula of follower's position is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{i,j}^t}{i^2}\right), & \text{if } i > \frac{n}{2}, \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L, & \text{otherwise,} \end{cases} \quad (2)$$

where  $X_p$  is the best position occupied by the producer.  $X_{\text{worst}}$  denotes the current global worst position.  $A$  denotes a  $1 \times d$  matrix where each element is randomly assigned as 1 or -1,  $A^+ = A^T(AA^T)^{-1}$ . When  $i > n/2$  indicates that the  $i$ th

follower with the worse fitness value is most likely to be in hunger.

Sparrows that are aware of the danger represent 10–20% of the total population. The initial positions of these sparrows are generated randomly in the population. The updated formula of these sparrows' position is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot |X_{i,j}^t - X_{\text{best}}^t| & \text{if } f_i > f_g, \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g, \end{cases} \quad (3)$$

where  $X_{\text{best}}$  is the current global optimal position. As the step control parameter,  $\beta$  is a normally distributed random number with mean 0 and variance 1.  $K \in [-1, 1]$  is a random number.  $f_i$  is the current adaptation value of the sparrow.  $f_g$  and  $f_w$  are, respectively, the current global best and worst fitness values.  $\varepsilon$  is the minimum constant. When  $f_i > f_g$ , this indicates that the sparrow is located at the edge of the population.  $X_{\text{best}}$  represents the position in the center of the population and the surroundings are safe.  $f_i = f_g$  indicates that the sparrow in the middle of the population is aware of the danger and needs to move closer to other sparrows.  $K$  indicates the direction of sparrow movement and it is also the step control coefficient.

The SSA has a strong partial search capability, but it still needs to be enhanced in terms of global search and jumping out of local optimum. Thus, it affects its convergence accuracy, but this deficiency belongs to the common problem of most optimization algorithms [24]. At present, the SSA is still a newly developed algorithm, and there are still relatively few related papers and studies, so there is still some room for improvement in the current SSA.

**3.2. Improving SSA Based on FA.** Aiming at remedying the shortcomings of the above-mentioned SSA, we introduce the firefly algorithm (FA), and the SSA search efficiency is improved mainly through the firefly disturbance strategy. The improved SSA can continue to search when the local extreme value is reached to obtain the optimal solution.

**3.2.1. Firefly Perturbation Strategy.** In the firefly algorithm [25, 26], the fireflies rely mainly on their own brightness to communicate information. Each firefly has brightness and attractiveness, and the brightness is used to determine the current position in order to choose its next flight direction, while the firefly's position update in space is influenced by attractiveness. These two are positively related, and fireflies will be attracted by ones brighter than them in order to constantly seek the best.

① The attractiveness of fireflies is

$$\beta = \beta_0 * e^{-\gamma r_{i,j}^2}, \quad (4)$$

where  $\beta_0$  is its initial attraction;  $\gamma$  is the light intensity absorption coefficient; and  $r_{i,j}$  represents the distance between the fireflies  $i$  and  $j$ .



② When the firefly  $i$  is brighter than  $j$ ,  $i$  update the position according to the following formula:

$$x_i = x_i + \beta * (x_j - x_i) + \alpha * \left(\text{rand} - \frac{1}{2}\right), \quad (5)$$

where  $x_j$ ,  $x_i$  are, respectively, the position of fireflies  $i$  and  $j$ .  $\alpha$  denotes the disturbed step size factor,  $\alpha \in [0, 1]$ . rand is a random perturbation, uniformly distributed, and takes values in the range  $[0, 1]$ .

**3.2.2. Improving SSA Using the Firefly Perturbation Strategy.** In this paper, the idea of improving the SSA is as follows: after the sparrow search, using the firefly perturbation strategy to perturb the sparrow, using the firefly disturbance method for all sparrows and the best sparrows to update their positions, improving its searchable and comparing the sparrows after the disturbance with the sparrows before the perturbation; if better, then the sparrows' position is updated; the improved SSA is called FASSA. The procedure of FASSA is as follows:

Step 1: initialize the population, the number of iterations, and the proportion of discoverers and joiners

Step 2: calculate the fitness values and rank them

Step 3: sparrow updates the finders' location according to formula (1)

Step 4: sparrow updates the followers' position according to formula (2)

Step 5: select the vigilantes randomly and update their positions according to formula (3)

Step 6: calculate the fitness value and update the sparrows' position

Step 7: use the firefly perturbation strategy; update the sparrows' positions according to formula (5)

Step 8: calculate the fitness value and update the sparrows' position

Step 9: if the end requirement is reached, output the result. If not, continuously execute Steps 2–8

**3.2.3. FASSA Validation and Comparison.** In order to verify the feasibility and effectiveness of the FASSA proposed in this paper, we select high-dimensional unimodal, high-dimensional multimodal, and low-dimensional functions from the benchmark test function set as the test functions for verification and comparison in this paper, as shown in Table 1, and compare the test results with the basic SSA.

The experimental environment is as follows: Intel® Core™ i5-4210U processor, 4 GB memory capacity, and Windows 10 64-bit operating system, with Matlab 2017b as the experimental software tool.

The number of populations and the maximum number of iterations are set to 30 and 1000, respectively. The dimension and search space settings of the test function are shown in Tables 1; within the maximum

number of iterations, the optimal value and convergence speed of the test function finally found are selected as the evaluation index for comparing the performance of the FASSA and the SSA. The experimental results of the two algorithms for the 13 test functions are shown in Figures 2–14.

By analyzing the performance of the two algorithms on the above multiple functions, it can be seen that the FASSA proposed in this paper shows good convergence and search ability compared to the basic SSA on the high-dimensional unimodal test functions. On the one hand, for the high-dimensional unimodal test functions, the performance of FASSA on  $F_1$ - $F_3$  test functions is obviously better, and for  $F_4$  and  $F_5$ , in the case of falling into local extremums, FASSA can jump out in time and continue to search to find the optimal value. On the other hand, for the high-dimensional multimodal and low-dimensional test functions, FASSA also deals with relatively better results. As can be seen from the results, on functions  $F_7$ ,  $F_8$ , and  $F_{12}$ , the FASSA gets good fitness values at the beginning of the iteration, while it has absolute advantages in  $F_6$ ,  $F_9$ ,  $F_{10}$ ,  $F_{11}$ , and  $F_{13}$  test functions, being obviously better than the basic SSA.

In summary, compared with the SSA, the FASSA proposed in this paper not only improves the convergence speed, but also has a higher accuracy in finding the best performance. Therefore, the improved FASSA is applied to optimize the BP neural network to build an intelligent manufacturing capability maturity evaluation model in this paper.

## 4. Evaluating IMCM Based on FASSA-BP Algorithm

**4.1. BP Neural Network.** Considering that the research object of this paper is the evaluation of enterprise intelligent manufacturing capability maturity, the problem studied is complex and nonlinear. However, BP neural network can realize mapping function from input to output, which is especially suitable for solving complex mapping problems with internal mechanism [27]. It can also self-adapt and learn. It can not only quantitatively analyze problems, but also expand the knowledge learned by applying network models [28]. Therefore, selecting BP neural network algorithm as the evaluation method of this paper can effectively avoid the interference of human subjective consciousness in the evaluation process.

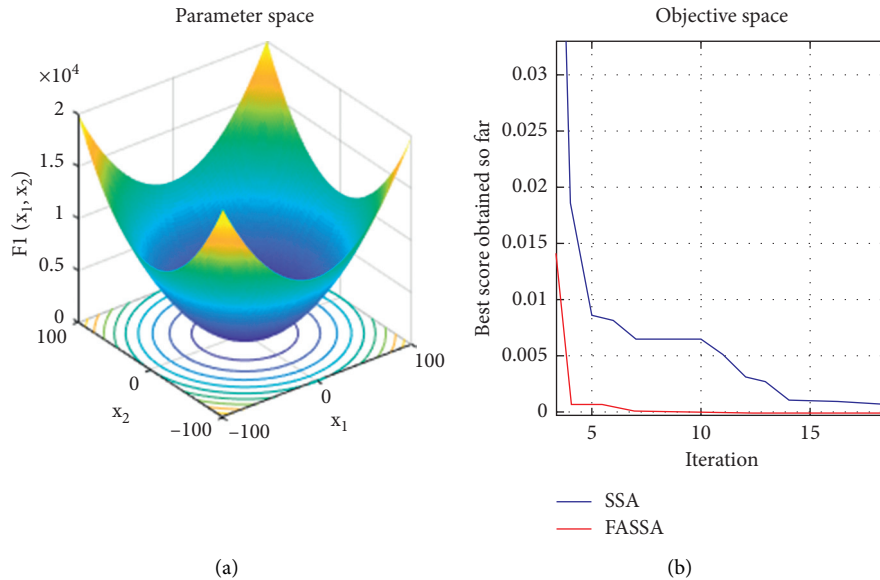
BP neural network is a multilayer feed-forward neural network with error back propagation [29]. It consists of input layer, hidden layer, and output layer. Each layer has a large number of different neurons [30]. In the learning process of BP neural network, it is mainly composed of forward propagation of signal and backward propagation of error [31]. Forward propagation means that the input data are transmitted layer by layer to each hidden layer through the input layer and finally to the output layer. The back propagation of error is that if there is an error between the training output and the expected output in the forward propagation process, the error signal is

TABLE 1: Test functions.

Type	Baseline test functions	Dimensionality	Search space	Optimum value
High-dimensional unimodal	$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
High-dimensional unimodal	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10, 10]$	0
High-dimensional unimodal	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0
High-dimensional unimodal	$F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
High-dimensional unimodal	$F_5(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	30	$[-1.28, 1.28]$	0
High-dimensional multimodal	$F_6(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-418.9829n
High-dimensional multimodal	$F_7(x) = -20 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$	0
High-dimensional multimodal	$F_8(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	30	$[-600, 600]$	0
High-dimensional multimodal	$F_9(x) = \pi/n \left\{ \begin{array}{l} 10 \sin(\pi y_1) + \\ \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \end{array} \right.$	30	$[-50, 50]$	0
Low-dimensional function	$F_{10}(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
Low-dimensional function	$F_{11}(x) = [1 -  \sin[\pi(x_1 - 2)]\sin[\pi(x_2 - 2)]/\pi^2(x_1 - 2)(x_2 - 2) ^5],$ $[2 + (x_1 - 7)^2 + 2(x_2 - 7)^2]$	2	$[0, 14]$	0
Low-dimensional function	$F_{12}(x) = \sum_{i=1}^{11} (a_i - (x_1(b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4))^2$	4	$[-5, 5]$	0.000307
Low-dimensional function	$F_{13}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	$[0, 1]$	-3.32

TABLE 2: Expected output value of the sample data.

Sample	$E$	Order of evaluation	Sample	$E$	Order of evaluation	Sample	$E$	Order of evaluation
$S_1$	68	Level three	...	...	...	$S_{491}$	55	Level three
$S_2$	49	Level two	...	...	...	$S_{492}$	69	Level three
$S_3$	41	Level two	$S_{487}$	44	Level two	$S_{493}$	53	Level three
$S_4$	49	Level two	$S_{488}$	43	Level two	$S_{494}$	64	Level three
$S_5$	57	Level three	$S_{489}$	29	Level one	$S_{495}$	62	Level three
...	...	...	$S_{490}$	54	Level three	$S_{496}$	72	Level four

FIGURE 2: Experimental results of the test function  $F_1$ .

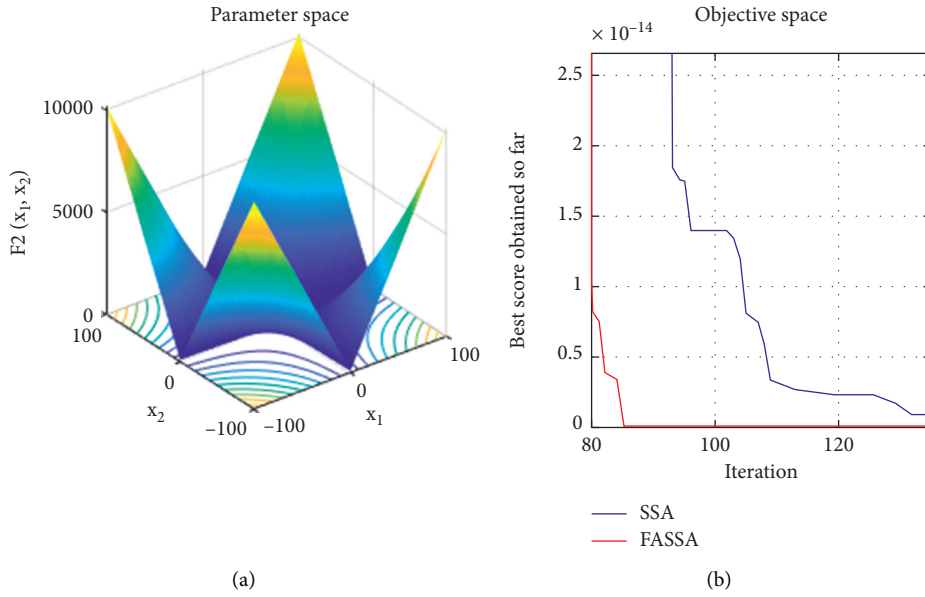


FIGURE 3: Experimental results of the test function  $F_2$ .

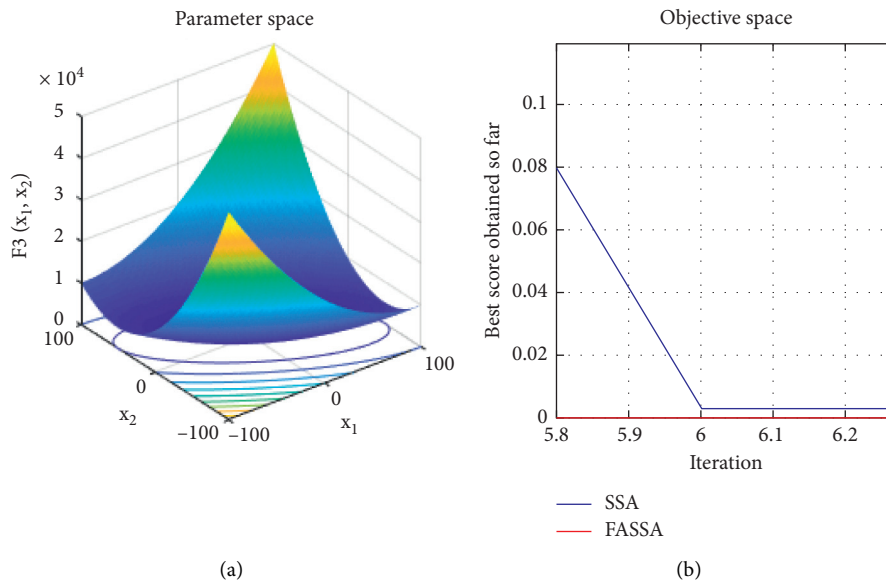


FIGURE 4: Experimental results of the test function  $F_3$ .

transmitted back to the hidden layer and the input layer by the gradient descent method, and the weights and thresholds of each neuron are continuously adjusted until the output error of BP neural network reaches the set accuracy standard or meets the number of iterations. The structure of the BP neural network is shown in Figure 15.

**4.2. Optimizing the BP Neural Network Based on FASSA.** In view of the shortcomings of BP neural network, that it is, being easy to fall into local optimum and randomly initializing weights and thresholds [32], FASSA is used to optimize the initial weights and thresholds of BP neural

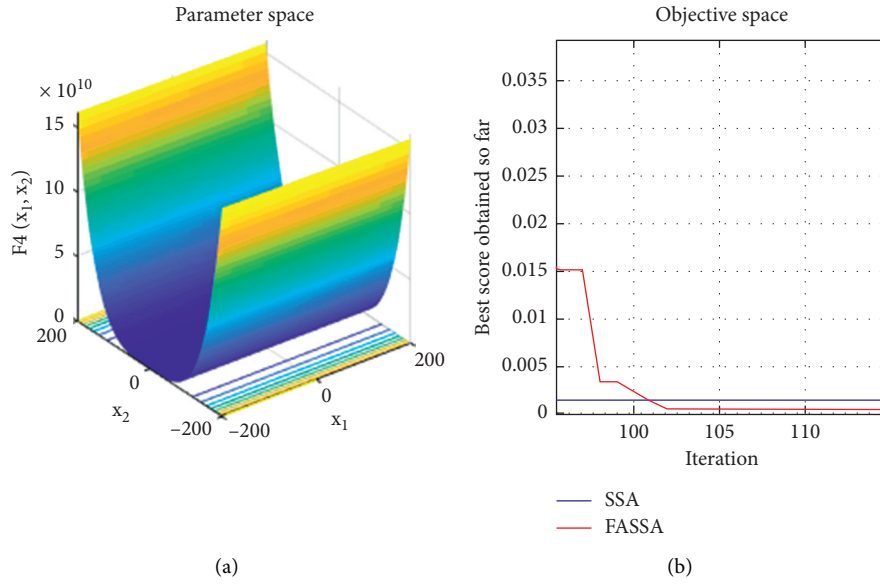
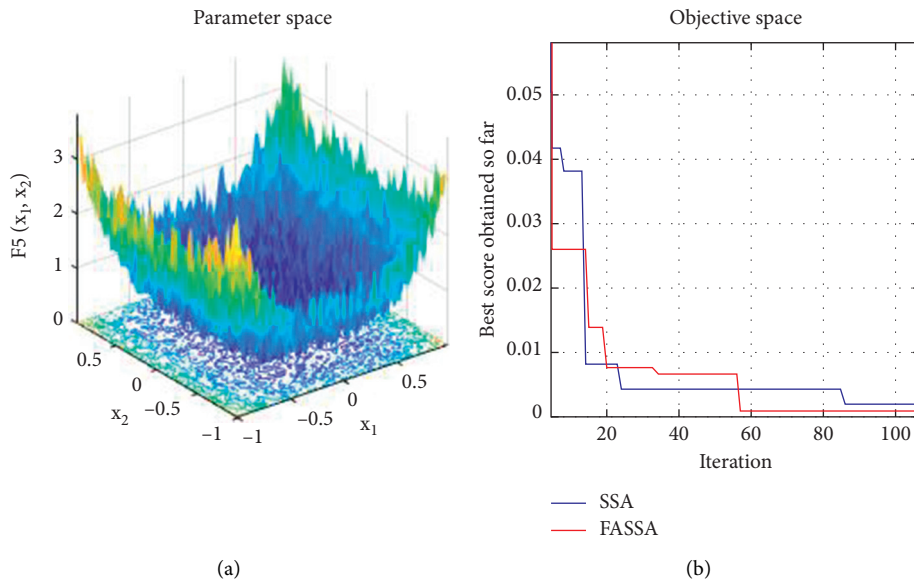
network, so as to overcome the shortcomings of BP neural network and improve the accuracy of intelligent manufacturing capability maturity evaluation model. The specific optimization process is shown in Figure 16.

The steps of optimizing BP neural network based on FASSA are as follows:

Step 1: design the structure of the BP neural network firstly.

Step 2: initialize FASSA population. Calculate the search space.

Dimension of FASSA based on the BP neural network topology structure. The calculating formula is described as follows:

FIGURE 5: Experimental results of the test function  $F_4$ .FIGURE 6: Experimental results of the test function  $F_5$ .

$$\begin{aligned} \text{Dim} &= \text{inputnum} * \text{hiddennum} \\ &+ \text{outputnum} * \text{hiddennum} \\ &+ \text{hiddennum} + \text{outputnum}. \end{aligned} \quad (6)$$

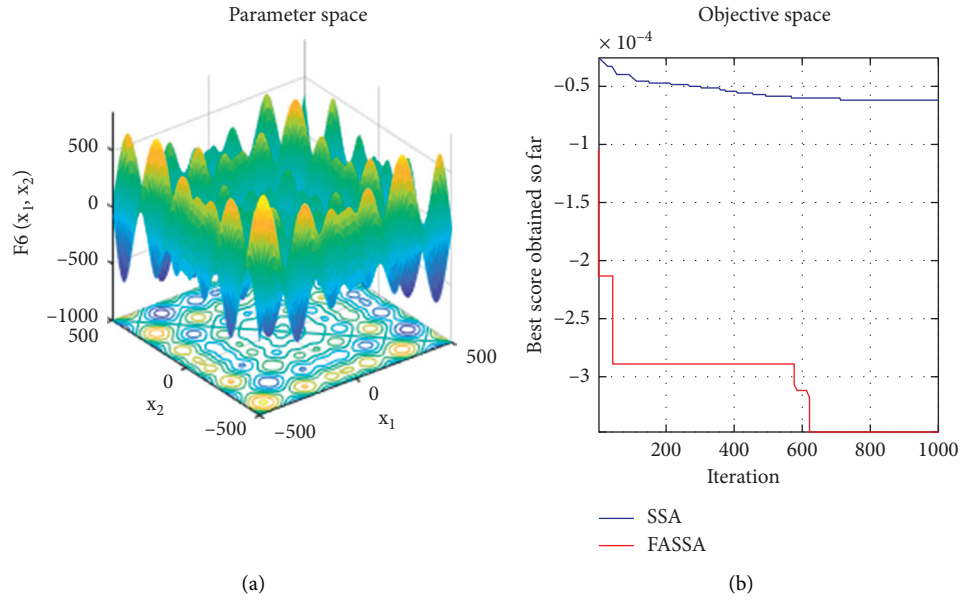
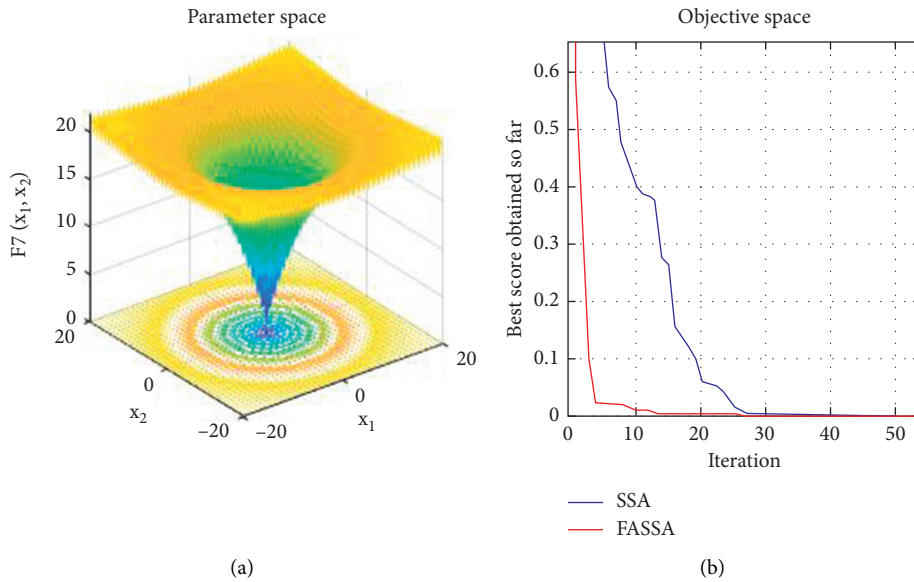
In the calculating formula, *inputnum*, *outputnum*, and *hiddennum* are the number of nodes in the input layer, output layer, and hidden layer.

Step 3: for the objective function of FASSA, the network error function is selected to design it:

$$\text{Fitness} = \frac{1}{n} \sum_i^n (\hat{y}_i - y_i)^2. \quad (7)$$

In the formula,  $n$  is the number of samples,  $y_i$  represents the expected output value of the sample, and  $\hat{y}_i$  represents the model output value of the sample.

Step 4: the position of sparrows is updated iteratively according to the formula for updating the position of sparrows with different roles in the basic SSA and calculating the fitness value.

FIGURE 7: Experimental results of the test function  $F_6$ .FIGURE 8: Experimental results of the test function  $F_7$ .

Step 5: use the firefly disturbance strategy to update and iteratively optimize the sparrows' position until the optimal solution is found or the end condition is satisfied.

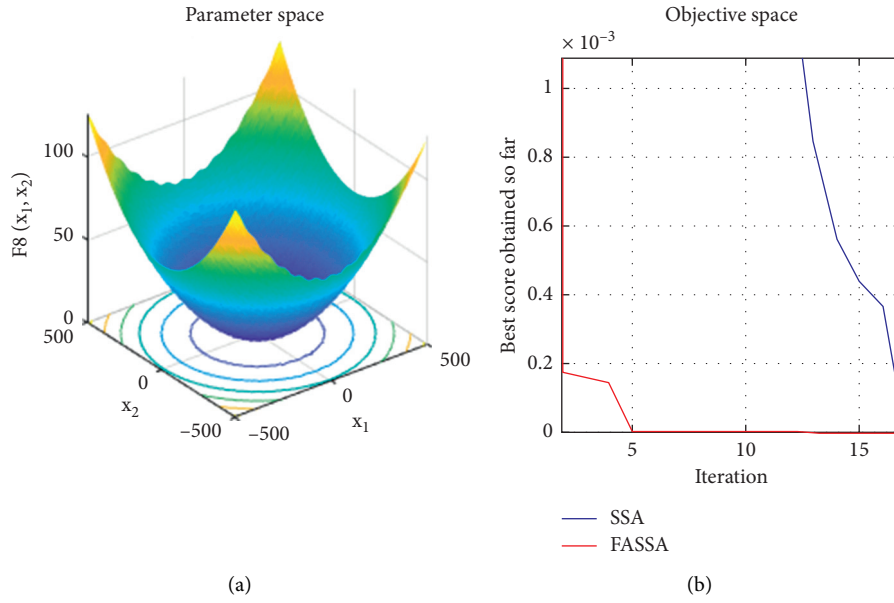
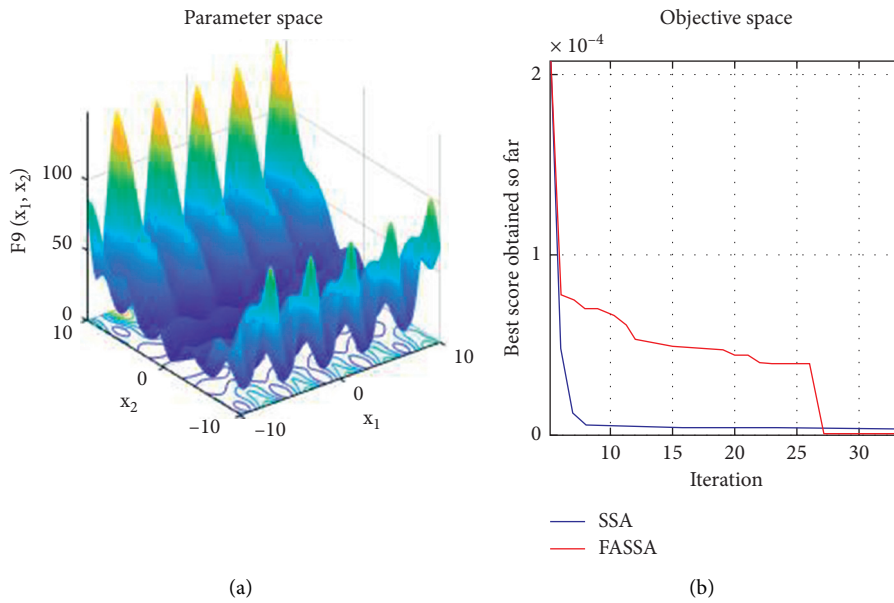
Step 6: the FASSA is stopped to search, and the final solution is divided into the initial weights and thresholds between different neurons in BP neural network; that is, the optimal solution matrix vector obtained from FASSA is assigned. Among them, the weight matrix from the initial input layer to the hidden layer is

$$W_1 = x(1: \text{inputnum} * \text{hiddennum}). \quad (8)$$

The initial hidden layer threshold matrix is

$$B_1 = x(\text{inputnum} * \text{hiddennum} + 1 : \text{inputnum} * \text{hiddennum} + \text{hiddennum}). \quad (9)$$

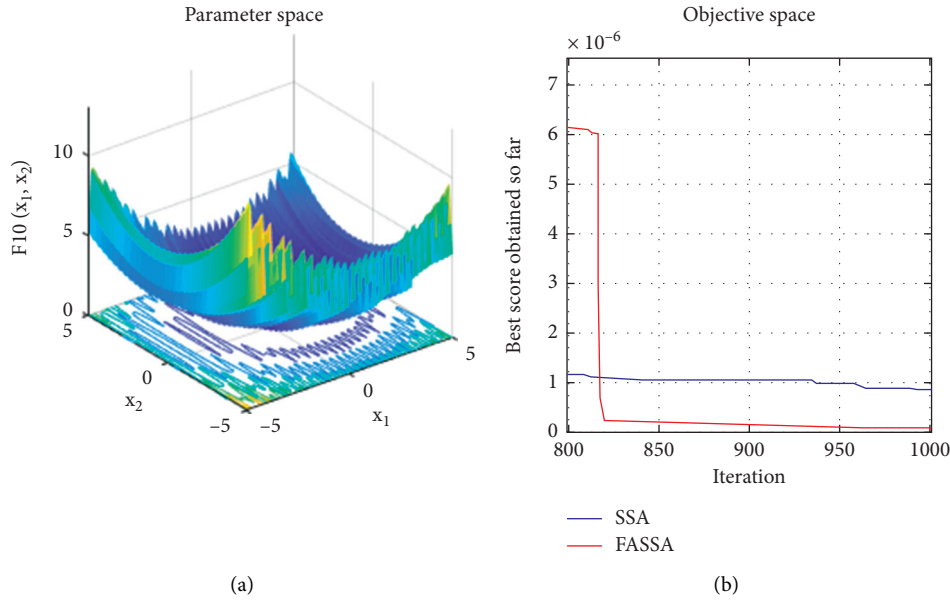
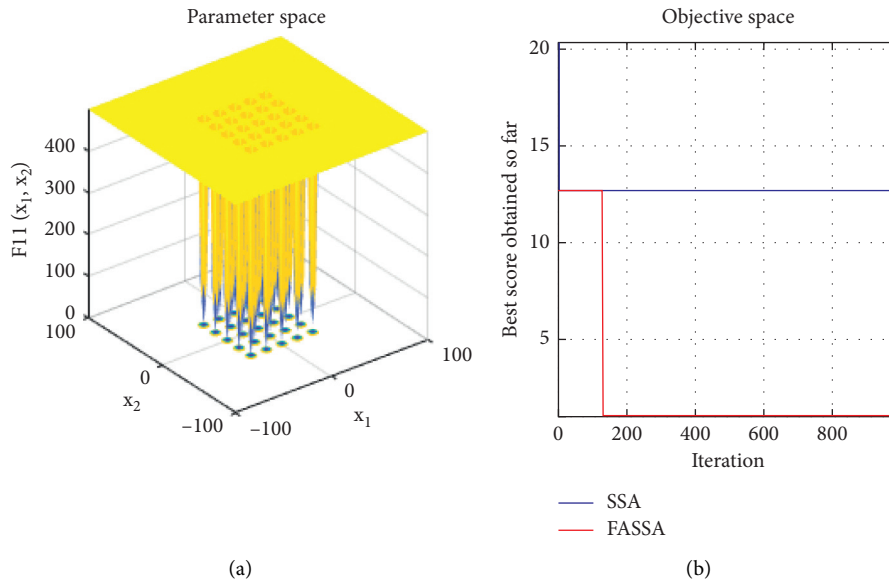
The weight matrix from the initial hidden layer to the output layer is

FIGURE 9: Experimental results of the test function  $F_8$ .FIGURE 10: Experimental results of the test function  $F_9$ .

$$W_2 = x(\text{inputnum} * \text{hiddennum} + \text{hiddennum} + 1 : \text{inputnum} * \text{hiddennum} + \text{hiddennum} + \text{hiddennum} * \text{outputnum}), \quad (10)$$

and the threshold matrix of the output layer is

$$B_2 = x(\text{inputnum} * \text{hiddennum} + \text{hiddennum} + \text{hiddennum} * \text{outputnum} + 1 : \text{inputnum} * \text{hiddennum} + \text{hiddennum} + \text{hiddennum} * \text{outputnum} + \text{outputnum}). \quad (11)$$

FIGURE 11: Experimental results of the test function  $F_{10}$ .FIGURE 12: Experimental results of the test function  $F_{11}$ .

Step 7: run the network based on the weights and thresholds that have been allocated, and the gradient descent method is used to reversely propagate the error to continuously update and correct its value. Finally, the stop condition is reached, and the evaluation results are output.

The steps of FASSA optimization of BP neural networks can be summarized as the pseudo-code shown in Algorithm 1.

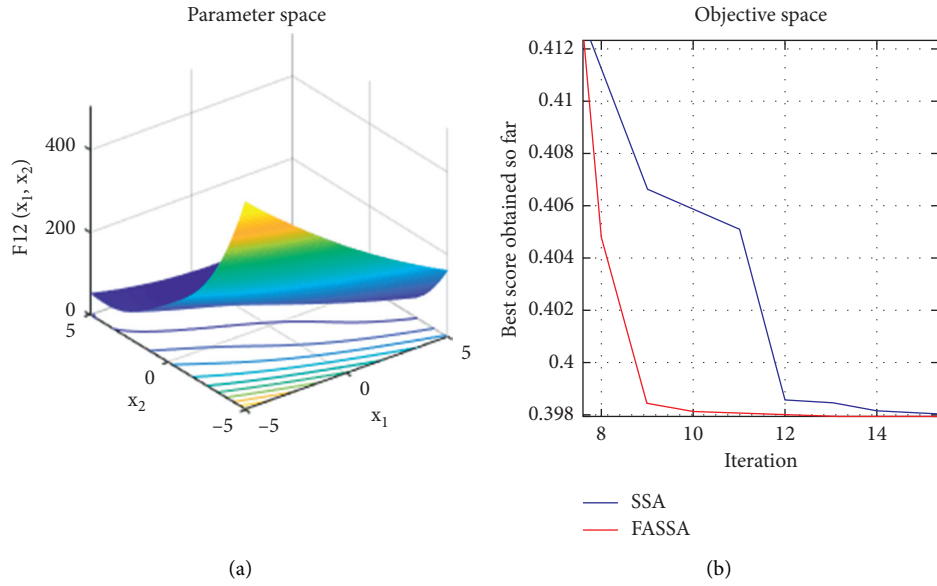
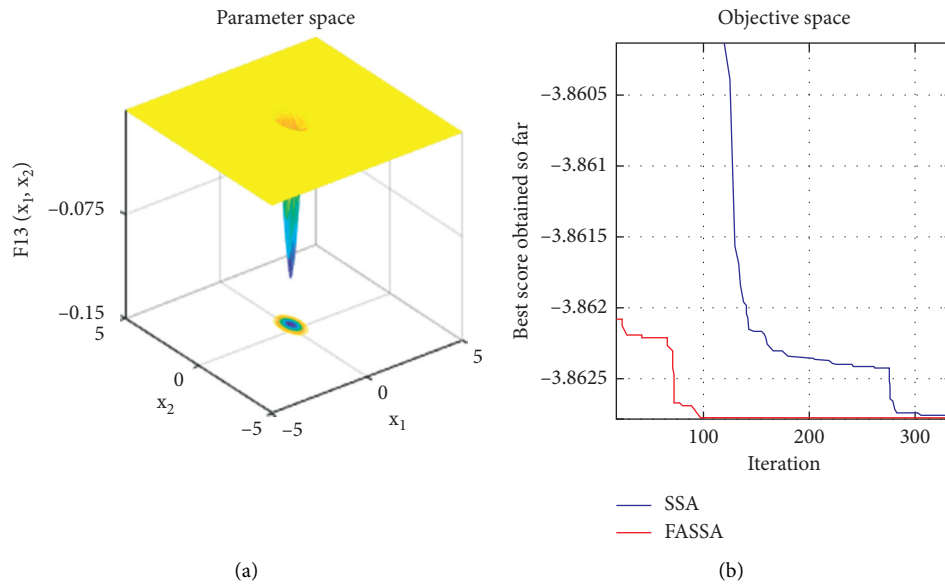
**4.3. Constructing the Evaluation Model Based on FASSA-BP Algorithm.** This section applies the improved FASSA-BP algorithm to construct the IMCM evaluation model of BP neural network based on FASSA.

**4.3.1. Construction of the Experimental Environment.** The experimental environment is as follows:

- (1) Hardware aspects: Intel® Core™ i5-4210U CPU, 4 GB memory capacity.
- (2) Software aspects: Windows 10 64-bit version, Matlab 2017b software platform.

**4.3.2. Design of the Network Structure**

**(1) Design of Input Layer Neurons.** This paper takes the constructed secondary evaluation index as the input of the network model. In this paper, 20 secondary evaluation indexes

FIGURE 13: Experimental results of the test function  $F_{12}$ .FIGURE 14: Experimental results of the test function  $F_{13}$ .

are used to measure the maturity of enterprise intelligent manufacturing capability. Therefore, the number of neurons in the input layer of the network is 20, and the vector of the input layer is expressed as  $X = (x_1, x_2, x_3, \dots, x_{20})$ .

(2) *Design of Hidden Layer Neurons.* For the setting of hidden layer nodes, this paper adopts the classical method of computation,  $S = \sqrt{(n+m) + a}$  (where  $a$  is a constant within  $[1, 10]$  and  $n$  and  $m$  are the numbers of input layer and output layer) [33]. It is proved that when the number of hidden layer nodes is 5, the model achieves the best state.

(3) *Design of Output Layer Neurons.* This paper takes the maturity score of intelligent manufacturing capability as the output layer of the network. Since there is only one output

vector, the number of nodes in the output layer of the network structure in this experiment is 1, and the output layer vector is expressed as  $Y = (y_1)$ .

Therefore, this paper builds a network with a structure of 20–5–1, as shown in Figure 17. Among them,  $W_1 = (\omega_{11}, \omega_{12}, \omega_{13}, \dots, \omega_{120})$  and  $W_2 = (\omega_{21}, \omega_{22}, \dots, \omega_{25})$  are, respectively, the weights between input layer and hidden layer, and between hidden layer and output layer; the hidden layer vector is set to  $H_1 = (h_{11}, h_{12}, \dots, h_{15})$ .

(4) *Determination and Selection of Parameters and Transfer Function.* Considering that the evaluation objects in this paper have high accuracy requirements, the basic parameters of the network model are set as follows: learning rate is 0.01, network training is 1000 times, and target error is  $10^{-7}$ .



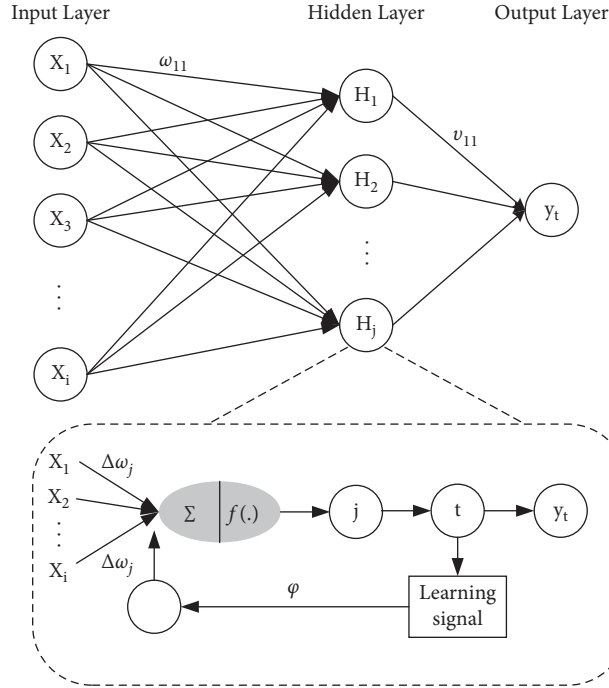


FIGURE 15: The structure of the BP neural network.

tansig and purelin act as hidden layer and output layer functions, respectively.

4.3.3. *Selection of Network Performance Indicators.* To test the performance of the evaluation model, this paper analyzes and compares the performance of FASSA-BP, SSA-BP, PSO-BP, and BP algorithms in terms of the accuracy and reliability of the model prediction results. In order to enhance the scientific nature of the evaluation of the forecast results, multiple performance metrics are used to evaluate the accuracy and stability of IMCM evaluation model results [34–38].

The performance analysis indexes used in this paper are as follows:

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|,$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|,$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2},$$

$$R^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}. \quad (12)$$

In the above analysis index formulas,  $n$  represents the number of samples,  $y_i$  represents the true value of the sample, and  $\hat{y}_i$  represents the model output value of the sample. The closer the evaluation indexes MRE, MSE, MAE, and RMSE are to 0, the smaller the evaluation error is, and the stronger the evaluation ability of the model is. The value range of  $R^2$  is  $[0, 1]$ . The closer  $R^2$  is to 1, the better the evaluation effect is.

4.3.4. *Expected Output Value of the Sample Data.* The expected output value  $E_h (h = 1, 2, \dots, n)$  represents the true value of the enterprise intelligent manufacturing capability maturity score. The calculation of the expected output value of the sample needs to calculate the weight of the evaluation index and the score of the evaluation index. Because the topic of the questionnaire is designed according to the 20 secondary evaluation indexes above, the score of the secondary evaluation index is the sample data recovered. Arithmetic average method [39] was used to calculate the

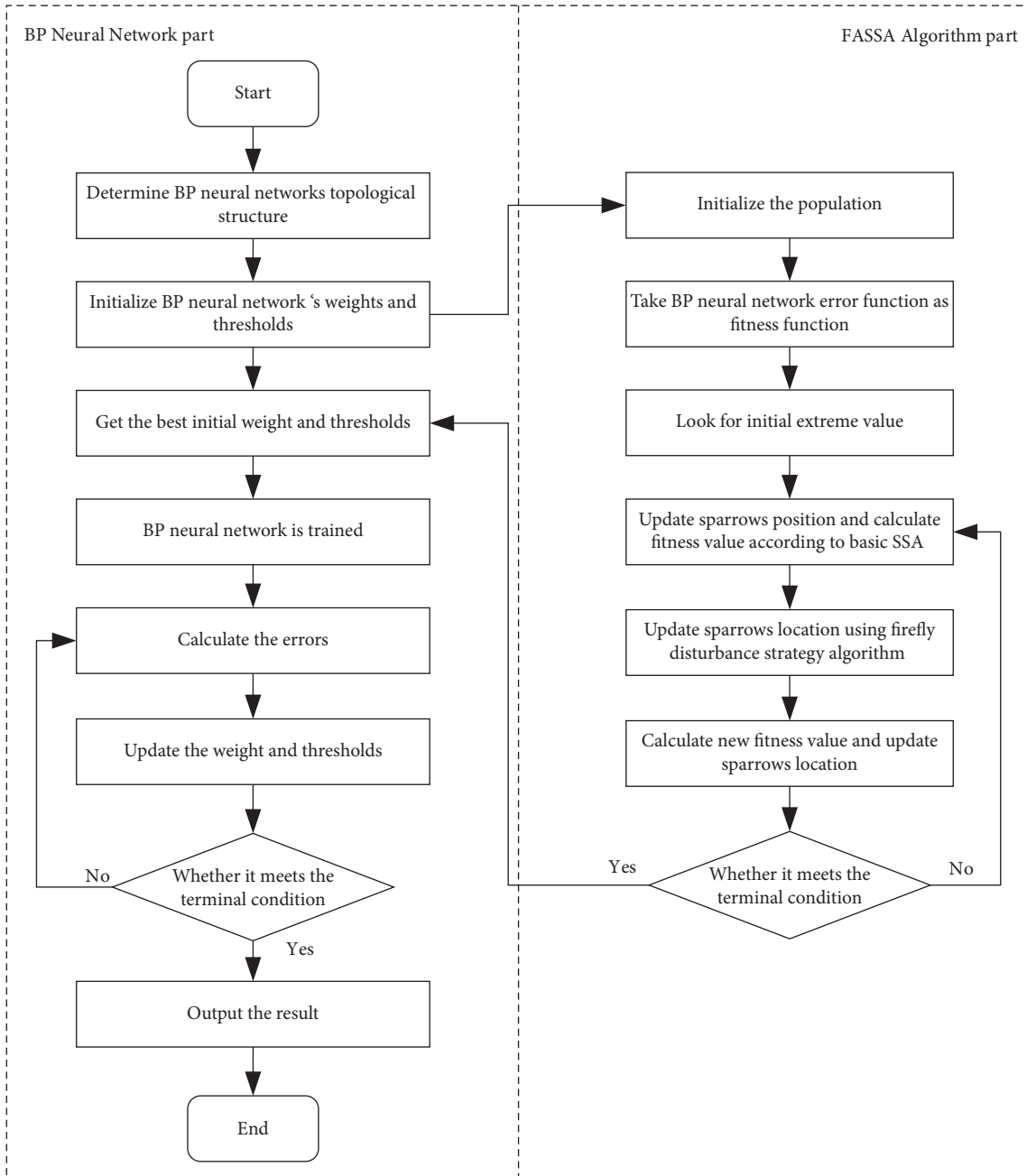


FIGURE 16: Flowchart of the BP neural network optimized by FASSA.

weight of primary index and the score of primary index, which were recorded as  $H_t$  and  $F_t$ , respectively. The primary index was recorded as  $B_t (t = 1, 2, \dots, 10)$ , and the

secondary index was recorded as  $C_k (k = 1, 2, \dots, 20)$ . The formula was as follows:

$$H_t = \frac{\text{number of secondary indicators } C \text{ corresponding to } B_t}{20} \quad (t = 1, 2, \dots, 10), \quad (13)$$

$$F_t = \frac{\text{sum of scores of secondary indicators } C \text{ corresponding to } B_t}{\text{number of secondary indicators } C \text{ corresponding to } B_t} \quad (t = 1, 2, \dots, 10). \quad (14)$$

```

Begin
(1) Determine BP network structure
(2) Initialize BP network parameter
(3) Initialize a population with FASSA
(4) Set the network error function as the objective of FASSA
(5) Calculate the initial fitness value of FASSA
(6) while( $t < G$ )
(7) Find the best and worst individuals by comparing fitness values
(8)  $R_2 = \text{rand}(1)$ 
(9) for  $i = 1 : PD$ 
(10) Update the location of the sparrow using formula (1);
(11) end for
(12) for  $i = (PD + 1) : n$ 
(13) Update the location of the sparrow using formula (2);
(14) end for
(15) for  $i = 1 : SD$ 
(16) Update the location of the sparrow using formula (3);
(17) end for
(18) Get the current new position
(19) Optimize sparrow using firefly disturbance
(20) for  $i = 1 : \text{pop}$ 
(21) Update the location of the sparrow using formula (5);
(22) Sort fitness values by size to find the best and worst individuals.
(23) Update global optimum. If the new location is better than before, update it.
(24)  $t = t + 1$ 
(25) end while
(26) return  $X_{\text{best}}, f_g$ 
(27) Bring the values in matrix  $X_{\text{best}}$  into BP neural network as initial weights and thresholds.
End

```

ALGORITHM 1: Pseudo-code of FASSA-BP algorithm.

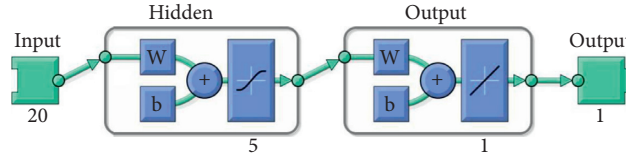


FIGURE 17: Network structure diagram of the model.

We use formula (11) to calculate the score of the first-level evaluation index and then multiply the score of each index in the sample data by the corresponding index weight with method of weighted mean. Finally, the expected output value  $E_h$  of each sample is obtained by adding. The calculation formula is as follows:

$$E_h = F_1 * H_1 + F_2 * H_2 + \dots + F_t * H_t, \quad (t = 1, 2, \dots, 10). \quad (15)$$

The expected output value of the sample data is shown in Table 2.

**4.3.5. IMCM Classification.** Some intelligent manufacturing enterprises in China are selected for field visits to understand the actual situation of intelligent manufacturing enterprises in detail. At the same time, referring to the division of intelligent manufacturing capability maturity in the reports of White Paper on Intelligent

Manufacturing Capability Maturity Model and German Industry 4.0 Maturity Index, the intelligent manufacturing capability maturity of intelligent manufacturing enterprises is divided into five grades. The interval number of (0,100) is used to divide the grades. The corresponding score intervals from low to high are (0, 30], (30, 50], (50, 70), (70, 90), (90, 100). The IMCM classification used in this paper is shown in Figure 18.

**4.4. Evaluation Steps.** The specific evaluation process of the IMCM evaluation model is shown in Figure 19.

Step 1: data acquisition and preprocessing: data are acquired and preprocessed according to the constructed evaluation index system to design the questionnaire, and then the questionnaire is distributed to the enterprise to collect data. Min-max method is used to normalize the recovered sample data.

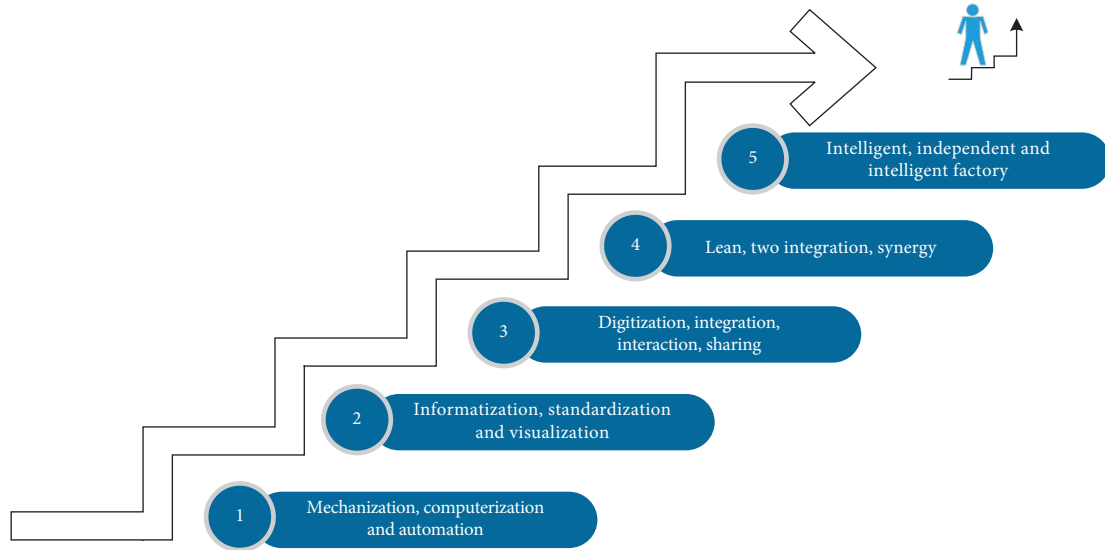


FIGURE 18: The level of enterprise intelligent manufacturing capability maturity.

Step 2: calculation of the sample expected output value: the hierarchical calculation method is adopted. First, the arithmetic average method [39] is used to calculate the first-level index weight and the first-level index score according to the recovered sample data, namely, the second level index score. Then, the weighted average method [40] is used to calculate the expected output value of the sample.

Step 3: determination of the experimental environment and model network structure: the experimental environment is introduced, and the numbers of input layer, output layer, and hidden layer of the model network structure are determined.

Step 4: comparison of neural network training and testing of different models: cross-validation experiments are conducted based on SSA-BP and PSO-BP models with the same population size and evolution times and the best population size and evolution times, respectively.

Step 5: analysis of results: the experimental results of the sample data of the test set are analyzed, and finally targeted suggestions for improvement are put forward for intelligent manufacturing enterprises.

## 5. Empirical Analysis

**5.1. Application Object Profile.** L International Technology Company Limited (hereinafter referred to as L Enterprise) is a well-known company in the manufacturing industry of China. Its main business field is battery manufacturing, and the products it produces and develops occupy an important position in the market of this field. The establishment of regional production bases at home and abroad constitutes a strong and advanced manufacturing and R&D capability of the enterprise. However, as a large battery manufacturer, L Enterprise has many subsidiaries in different regions.

Although all the companies are developing intelligent manufacturing, due to the imbalance of economic development and scientific and technological level between regions, the development level of intelligent manufacturing differs between different subsidiaries. It is urgent to clarify the level of intelligent manufacturing of subsidiaries, which is consistent with the research background selected in this paper and can accurately obtain the evaluation index data needed in this study. Therefore, the enterprise is selected as the actual case of the study.

### 5.2. Data Acquisition and Data Preprocessing

**5.2.1. Data Acquisition.** The data of this paper is collected in the form of questionnaire survey and expert interview, and the sample data is obtained according to the following four steps:

- (1) According to the quantitative method of evaluation index introduced above, a preliminary questionnaire survey is designed by using the constructed evaluation index to recover the quantitative index data. Through the questionnaire survey, these 20 secondary evaluation indexes are quantified into the determined sample data as the input layer data of the network.
- (2) The advanced manufacturing enterprises are selected, through research and interviews with the management personnel of the enterprise and the staff of the main business department; the collected opinions are summarized; the questionnaire is modified according to the refined suggestions, in order to set the appropriate questionnaire questions, options etc.; and finally the questionnaire is determined.
- (3) In order to improve the quality of the questionnaire, some questionnaires were distributed before the

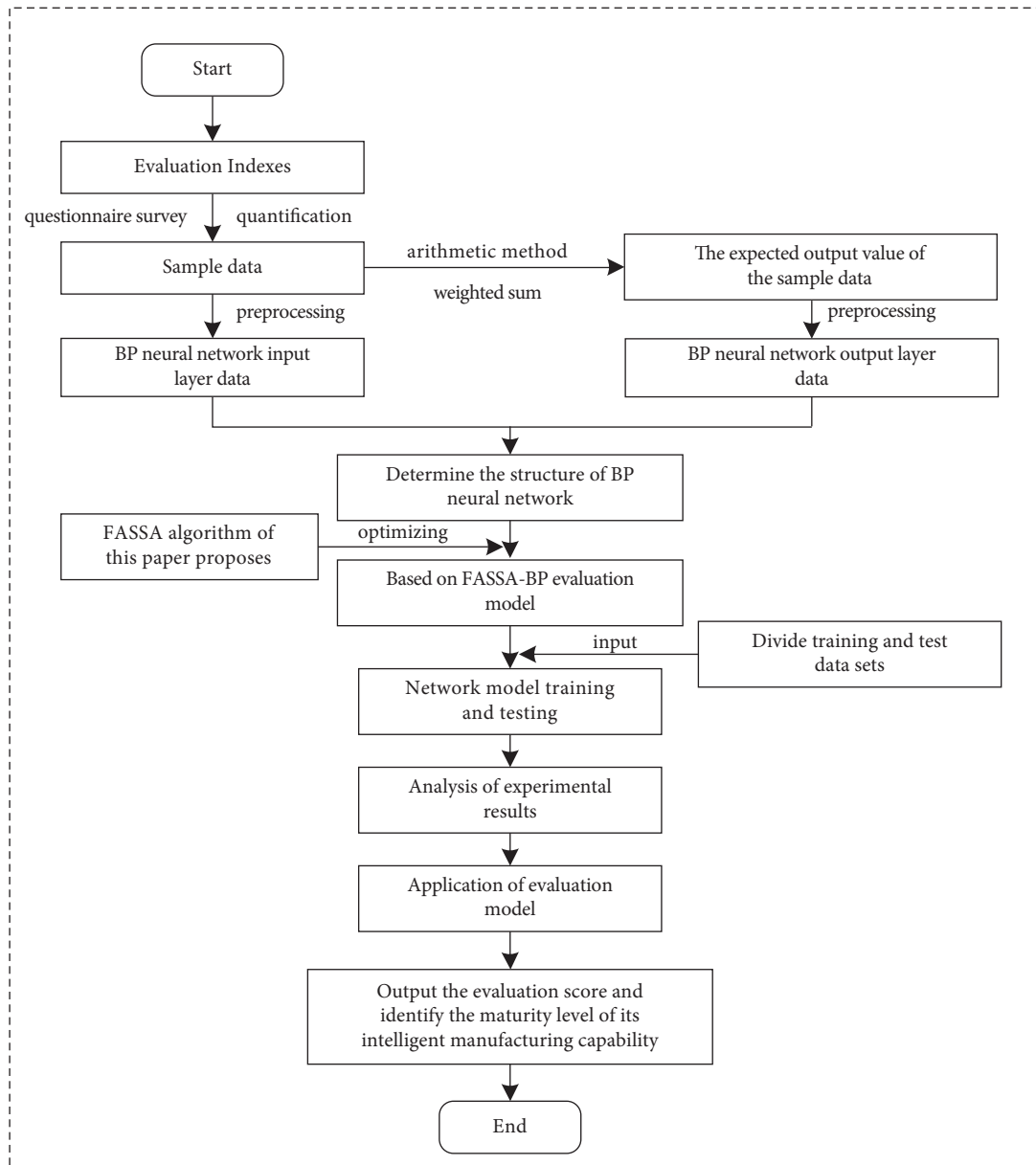


FIGURE 19: FASSA-BP model evaluation flowchart.

formal issuance of the questionnaire and according to the feedback in advance to improve the questionnaire.

- (4) Select the subsidiary of L Enterprise as the research object, and obtain data through enterprise research and questionnaire. The collected data help to support case studies in detail by understanding the development level of intelligent manufacturing enterprises in each evaluation index. A total of 536 questionnaires were returned, including 40 invalid questionnaires.

**5.2.2. Division of Datasets.** This paper collects 496 valid questionnaires and divides them into 441 data for network model training, which is the training set data. Then, samples

are selected from the remaining 55 sample data as the test set. Finally, 5 subcompany data are left, and the trained network model is used to evaluate the application.

**5.2.3. Preprocessing of the Data.** Considering that the magnitude of the original data is quite different, which has a great impact on the network training process and increases the difficulty of training, in order to better analyze the data and improve the training efficiency of the model, before the neural network training, Min – max method that belongs to the normalization method is used to sort out the sample data and its expected output value [41]. Because the evaluation indexes in this paper are positive, the normalization only considers the positive situation, and the calculation formula is shown in

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (16)$$

The original data obtained by sorting out the collected questionnaire are shown in Table 3. The original data are processed in advance by (16) to obtain the input sample data as shown in Table 4. Finally, the data processed in advance is input into the constructed evaluation model to ensure that the trained model can accurately evaluate the maturity of enterprise intelligent manufacturing capability.

**5.2.4. Calculation of the Expected Output Value of the Sample Data.** The data obtained in Section 5.2.1 is the sample data quantified by the evaluation index, that is, the input layer data of the network model. However, in order to use the network model for evaluation research, the expected output value of the sample data is also needed, that is, the ability maturity evaluation score, as the output layer data of the network model.

From Section 4.3.4, we can get the weight of the evaluation index and its score, and then the weighted sum of the two can get the expected output value of the sample data.

*(1) Calculating the Weight of Evaluation Index.* According to Section 4.3.4, the weight of evaluation index is calculated by arithmetic average method and weighted average method. According to (13), the weight of first-level evaluation index can be obtained:

$$\begin{aligned} H_1 = H_2 = H_4 = H_5 &= \frac{1}{10}, \\ H_3 &= \frac{7}{20}, \\ H_6 = H_7 = H_8 = H_9 = H_{10} &= \frac{1}{20}. \end{aligned} \quad (17)$$

*(2) Calculating Scores of Evaluation Indicators.* Through the quantification of evaluation indexes, it can be seen that, in the questionnaire survey table, each question reflects the situation of each secondary evaluation index, so the sample data collected by the questionnaire correspond to the score of the secondary evaluation index. Then, according to (14), the first-level evaluation index score is calculated.

*(3) Calculating the Expected Output of the Sample Data.* Finally, the expected output value of each sample data can be obtained by weighted summation of the first-level evaluation index score and its corresponding index weight. The expected output value of the sample data is shown in Table 5.

### 5.3. L Enterprise Intelligent Manufacturing Capability Maturity Evaluation

**5.3.1. Comparison of Evaluation Model Training and Testing.** The population size and the number of iterations have a great influence on the optimization algorithm to find the

best solution. Therefore, this paper designs comparison experiments from the two perspectives of the same population size and number of iterations, and the optimal population size and number of iterations, respectively, and selects the BP model, PSO-BP model, SSA-BP model, and FASSA-BP model proposed in this paper for performance comparison.

*Experiment 1.* Performance comparison based on the same population size and iterations.

In Experiment 1, in order to reduce the influence of initial parameters on the performance of the algorithm, the settings of the same parameters are guaranteed to be the same. Because the comparison of several algorithm models is based on BP neural network, the BP neural network parameters are set according to Section 4.3.2. The key parameters in PSO-BP model, SSA-BP model, and FASSA-BP model are set as shown in Table 6.

After setting the parameters of each model, the pre-processed sample data are input into FASSA-BP model, SSA-BP model, PSO-BP model, and BP model for training and testing. Firstly, the training samples are input into the four models to train. After the training is completed, it enters the test stage. After several iterations, the comparison results of the evaluation value and the real value of each model test set sample are obtained as shown in Figure 20, and the relative error results of the expected output value of each model test set sample data and the actual output value of the model are obtained as shown in Figure 21.

Considering the different running results of neural network algorithm and intelligent optimization algorithm, the four models are run 30 times, respectively, and the average values of the experimental results of performance evaluation indexes of different models are compared, as shown in Table 7.

Combining Figures 20 and 21 and Table 7, we can see that the evaluation accuracy of the FASSA-BP model proposed in this paper is significantly higher than that of the basic SSA-BP model, PSO-BP model, and traditional BP model, and the value of the correlation coefficient  $R_2$  is closer to 1 than that of the other three models, which verifies that the FASSA-BP model can be better applied to solve the maturity evaluation problem of intelligent manufacturing capability.

*Experiment 2.* Performance comparison based on optimal population size and iterations.

In this paper, by selecting the test function to set different population size and iteration number, the optimal population size and iteration number of PSO algorithm, SSA, and FASSA are obtained to achieve the optimal fitness value, respectively, as shown in Table 8.

In order to further test the superiority of FASSA-BP model, the parameters of FASSA-BP model, SSA-BP model, and PSO-BP model under the optimal fitness are selected as the best parameter settings, and their model performance is analyzed by experiments. In the same experiment, after the parameter setting is completed, the preprocessed sample data are input to FASSA-BP model,

TABLE 3: Original data from questionnaires.

Samples	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
S1	60	60	80	60	80	40	60	80	60	60	80	80	80	80	40	60	80	100	40	
S2	80	60	60	60	60	40	40	40	40	40	20	60	60	40	40	60	60	60	40	20
S3	60	20	20	20	40	60	40	40	0	80	0	80	60	40	100	40	60	60	0	0
S4	0	20	60	60	60	40	40	40	80	80	80	60	60	0	60	60	60	40	40	40
S5	40	40	60	20	80	80	100	100	100	100	20	20	40	40	40	80	20	20	80	60
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
S487	40	100	0	40	60	20	0	20	80	20	60	40	40	20	20	60	40	80	80	60
S488	20	80	20	60	40	40	60	40	80	20	20	40	60	80	0	60	100	40	0	0
S489	0	20	40	60	20	40	40	40	40	0	20	20	40	20	20	20	40	40	60	0
S490	100	60	40	80	80	60	60	60	40	60	40	60	0	40	20	20	80	20	100	60
S491	40	100	60	80	0	40	80	100	20	40	20	80	40	40	0	20	60	100	80	100
S492	100	80	100	60	0	100	100	100	40	100	60	100	100	60	40	60	60	20	20	80
S493	40	0	80	20	80	20	40	60	40	20	60	100	80	60	80	80	0	60	60	80
S494	80	60	40	80	60	60	20	100	80	60	80	80	80	60	60	80	60	40	60	40
S495	100	100	40	80	40	60	80	20	100	0	20	40	0	100	80	80	40	80	80	100
S496	80	100	40	60	100	100	60	80	60	80	80	80	80	60	80	80	60	100	0	0

TABLE 4: Input sample data after preprocessing.

Samples	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
S1	0.6	0.6	0.8	0.6	0.8	0.4	0.6	0.8	0.6	0.6	0.8	0.8	0.8	0.8	0.4	0.6	0.8	1.0	0.4	
S2	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.4	0.4	0.4	0.2	0.6	0.6	0.4	0.4	0.6	0.6	0.6	0.4	0.2
S3	0.6	0.2	0.2	0.2	0.4	0.6	0.4	0.4	0.0	0.8	0.0	0.8	0.6	0.4	1.0	0.4	0.6	0.6	0.0	0.0
S4	0.0	0.2	0.6	0.6	0.6	0.4	0.4	0.4	0.8	0.8	0.8	0.6	0.6	0.0	0.6	0.6	0.6	0.4	0.4	0.4
S5	0.4	0.4	0.6	0.2	0.8	0.8	1.0	1.0	1.0	1.0	0.2	0.2	0.4	0.4	0.4	0.8	0.2	0.2	0.8	0.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
S487	0.4	1.0	0.0	0.4	0.6	0.2	0.0	0.2	0.8	0.2	0.6	0.4	0.4	0.2	0.2	0.6	0.4	0.8	0.8	0.6
S488	0.2	0.8	0.2	0.6	0.4	0.4	0.6	0.4	0.8	0.2	0.2	0.4	0.6	0.8	0.6	0.0	1.0	0.4	0.0	0.0
S489	0.0	0.2	0.4	0.6	0.2	0.4	0.4	0.4	0.4	0.0	0.2	0.2	0.4	0.2	0.2	0.2	0.4	0.4	0.6	0.0
S490	1.0	0.6	0.4	0.8	0.8	0.6	0.6	0.6	0.4	0.6	0.4	0.6	0.0	0.4	0.2	0.2	0.8	0.2	1.0	0.6
S491	0.4	1.0	0.6	0.8	0.0	0.4	0.8	1.0	0.2	0.4	0.2	0.6	0.4	0.4	0.0	0.2	0.6	1.0	0.8	1.0
S492	1.0	0.8	1.0	0.6	0.0	1.0	1.0	1.0	0.2	1.0	0.6	1.0	1.0	0.6	0.4	0.6	0.6	0.2	0.2	0.8
S493	0.4	0.0	0.8	0.2	0.8	0.2	0.4	0.6	0.4	0.2	0.6	1.0	0.8	0.6	0.8	0.8	0.0	0.6	0.6	0.8
S494	0.8	0.6	0.4	0.8	0.6	0.6	0.2	1.0	0.8	0.6	0.8	0.8	0.8	0.6	0.6	0.8	0.6	0.4	0.6	0.4
S495	1.0	1.0	0.4	0.8	0.4	0.6	0.8	0.2	1.0	0.0	0.2	0.4	0.0	1.0	0.8	0.8	0.4	0.8	0.8	1.0
S496	0.8	0.8	0.4	0.6	1.0	1.0	0.6	0.8	0.6	0.8	0.8	0.8	0.8	0.8	0.6	0.8	0.8	0.6	1.0	0.0

TABLE 5: Expected output value of the sample data.

Sample	$E$	Sample	$E$	Sample	$E$
$S_1$	68	...	...	$S_{486}$	57
$S_2$	49	...	...	$S_{487}$	44
$S_3$	41	$S_{482}$	41	$S_{488}$	43
$S_4$	49	$S_{483}$	62	$S_{489}$	29
$S_5$	57	$S_{484}$	71	$S_{490}$	54
...	...	$S_{485}$	65	$S_{491}$	55

TABLE 6: Parameter settings of model algorithms in Experiment 1.

Arithmetic	Population size	Iteration times	Individual parameter
PSO	100	100	$C1 = C2 = 1.49445$
SSA	100	100	P_percent = 0.2
FASSA	100	100	P_percent = 0.2

SSA-BP model, PSO-BP model, and BP model, respectively. After several iterations, the evaluation value of each model is obtained and compared with the real value,

as shown in Figure 22. In addition, there is a relative error, as shown in Figure 23.

The FASSA-BP model, SSA-BP model, and PSO-BP model are set to the best population size and number of iterations, respectively. Because the running time of the three models is too long in the case of the best population size and number of iterations, the FASSA-BP model, SSA-BP model, and PSO-BP model are run 20 times, respectively, and the average value of the experimental data of each model in the performance evaluation index is calculated to analyze and compare the performance of these models, as shown in Table 9.

From the comparison between the evaluation value and the real value of the sample in Figure 22 test set, FASSA-BP evaluation model has better evaluation effect. It can be seen from Figure 23 that the relative error between the true value of FASSA-BP model and the actual network output value is the smallest. According to Table 10, although the SSA-BP model and the PSO-BP algorithm model are under the optimal population size and iteration number, from the perspective of

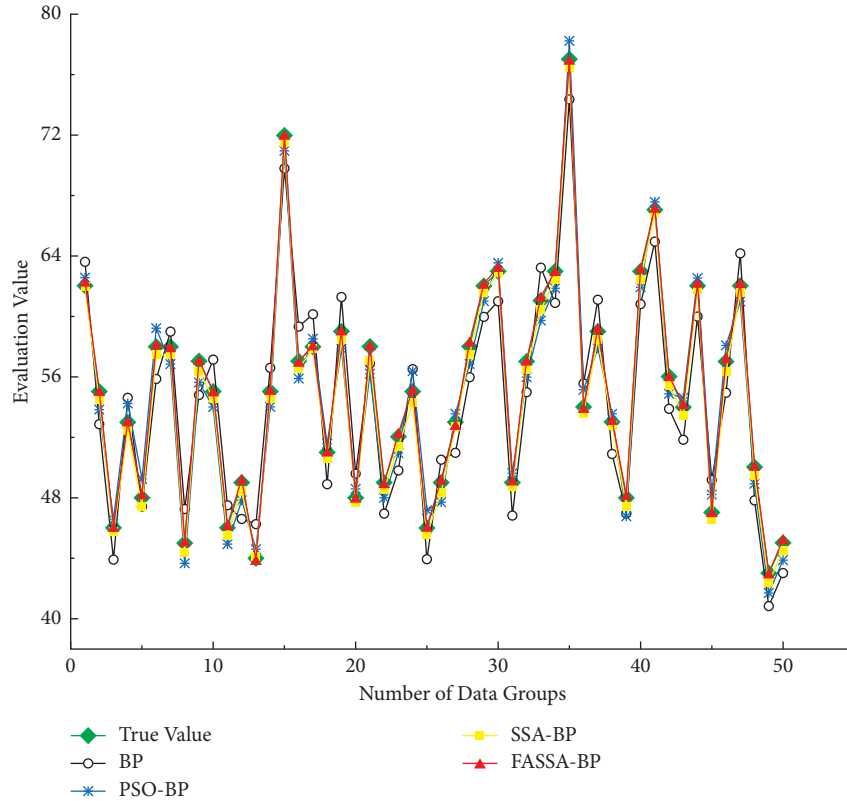


FIGURE 20: Comparison of test results of each model in Experiment 1.

evaluation accuracy, the FASSA-BP model is significantly better than the SSA-BP model, the PSO-BP model, and the traditional BP model, which further proves the superiority of the FASSA-BP model proposed in this paper.

**5.3.2. Performance Analysis of the FASSA-BP Model.** In order to analyze FASSA-BP model more accurately, regression analysis and relative error analysis of model evaluation value and expected output value of FASSA-BP model test set sample data are selected, as shown in Figure 24 and Table 10.

The regression analysis of the actual output value and the expected output value of the network (Figure 24) shows that regression coefficient  $R^2$  of FASSA-BP model is very close to 1, which indicates that the evaluation effect of FASSA-BP model is very good.

From the comparison results of the relative error between the output value of the model and the real value of the sample in Table 10 test set, it can be concluded that the error of FASSA-BP evaluation model is small, and the classification results are completely consistent, which also shows that the comprehensive evaluation score of the model for the maturity of intelligent manufacturing capability is almost the same as the actual intelligent manufacturing capability of the enterprise, and it has certain accuracy and feasibility. It is further proved that the model can be used in the evaluation of the maturity of intelligent manufacturing capability.

**5.3.3. Maturity Evaluation of Subsidiary Intelligent Manufacturing Capability.** According to the different geographical distribution, the attribute values of the secondary evaluation indexes of five subsidiaries (the numbers of subsidiaries are C1, C2, C3, C4, C5) of the enterprise are selected as the sample data. The trained FASSA-BP evaluation model is used to calculate the evaluation value of the intelligent manufacturing capability maturity of each subsidiary. According to the classification of evaluation grades, the corresponding intelligent manufacturing capability maturity levels of the five subsidiaries are obtained, as shown in Table 11.

**5.4. Empirical Result Analysis and Improvement Suggestions.** By applying the evaluation model based on FASSA-BP algorithm proposed in this paper to practical cases and comparing it with other evaluation models, it can be seen that FASSA-BP model is more accurate and reliable than other models and can better deal with the maturity evaluation of enterprise intelligent manufacturing capability.

It can be seen from Table 11 that there is a certain gap in the maturity level of intelligent manufacturing capability of these five subsidiaries. Affected by their geographical location and the level of urban economic development, these subsidiaries have different performance in different evaluation indicators. Among them, C4 is the highest rating; the ability maturity score is 73, which shows that the company's intelligent manufacturing capability is relatively mature. However, although the company's overall performance in the



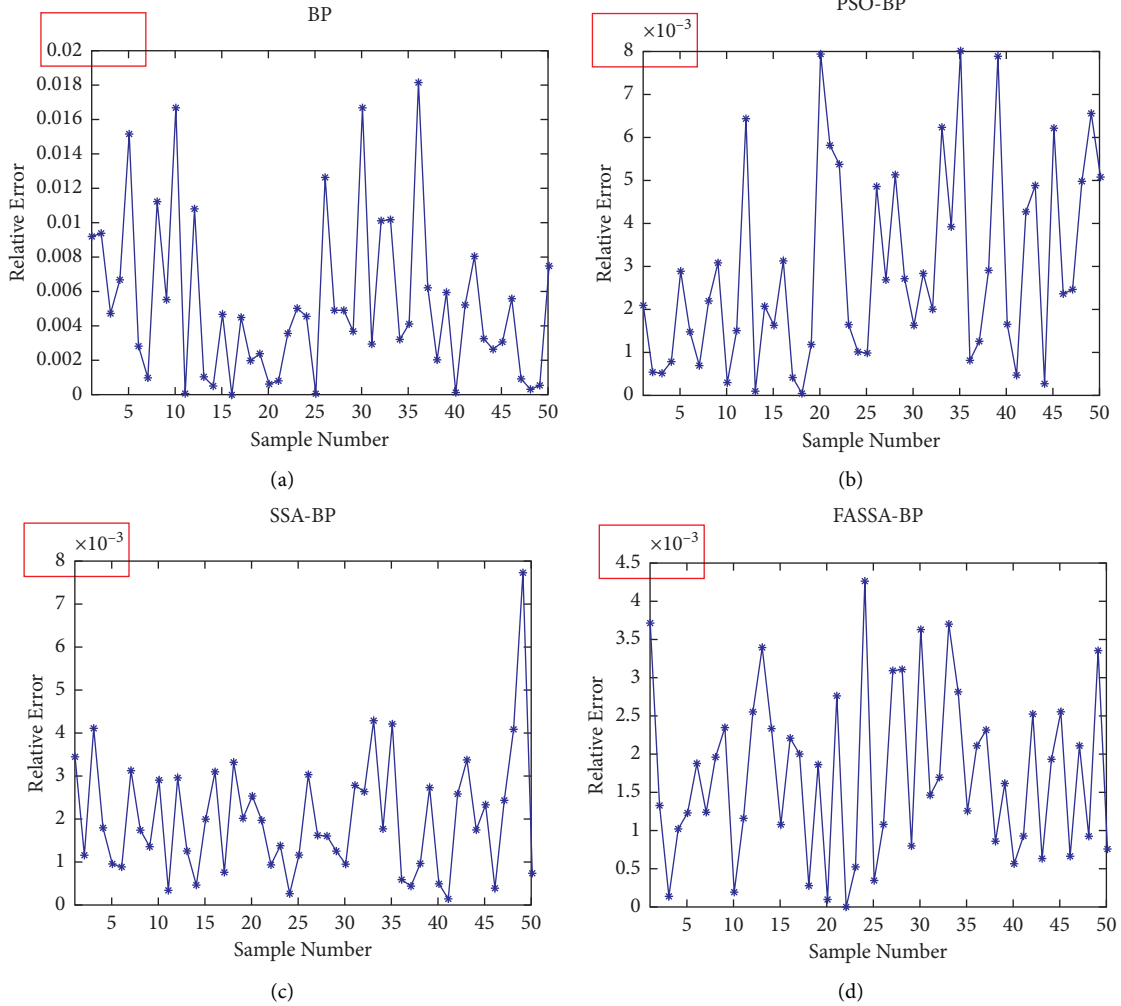


FIGURE 21: Relative error of each model test set sample in Experiment 1: (a) BP; (b) PSO-BP; (c) SSA-BP; (d) FASSA-BP.

TABLE 7: Average data of four model performance evaluation indexes in Experiment 1.

Performance evaluation index	BP	PSO-BP	SSA-BP	FASSA-BP
MRE (%)	0.4845	0.3523	0.1781	0.1483
MSE (%)	8.4504	5.8176	2.3848	1.7848
MAE (%)	25.39	18.97	11.61	9.866
RMSE	0.2957	0.2459	0.1646	0.1294
$R^2$	0.9824	0.9861	0.9995	0.9997

TABLE 8: Optimal parameter settings for model algorithms in Experiment 2.

Arithmetic	Population size	Iteration times	Individual parameter
PSO	200	1000	$C1 = C2 = 1.49445$
SSA	100	1000	$P\_percent = 0.2$
FASSA	100	1000	$P\_percent = 0.2$

production process is good, the process management is still insufficient; the company needs to focus on the next improvement. Although C1, C2, and C3 have the same rating,

they have different performance in evaluation indicators. For example, C1's advantage lies in network management, but the index to be improved is 5S management, so the company should focus on its low maturity 5S management in the future. One of the main reasons for the low level of C2 capability is quality control. Therefore, the company should establish a quality management system to help enterprises control product quality by automatically collecting quality information in the production process. C3 scored relatively low in maturity of strategic management indicators, so the company should adjust its strategy to keep the development of intelligent manufacturing consistent with the company's development strategy. C5, as the second level of evaluation, belongs to "intelligent manufacturing novice" and has a greater opportunity to enter the next maturity level. Its maturity evaluation is significant. Therefore, in the next step, the company should strengthen its organizational structure and mechanism construction. It can be seen from this that, for the maturity of intelligent manufacturing capability, different subsidiaries have differences in the evaluation index. Therefore, each company should pay more attention to the

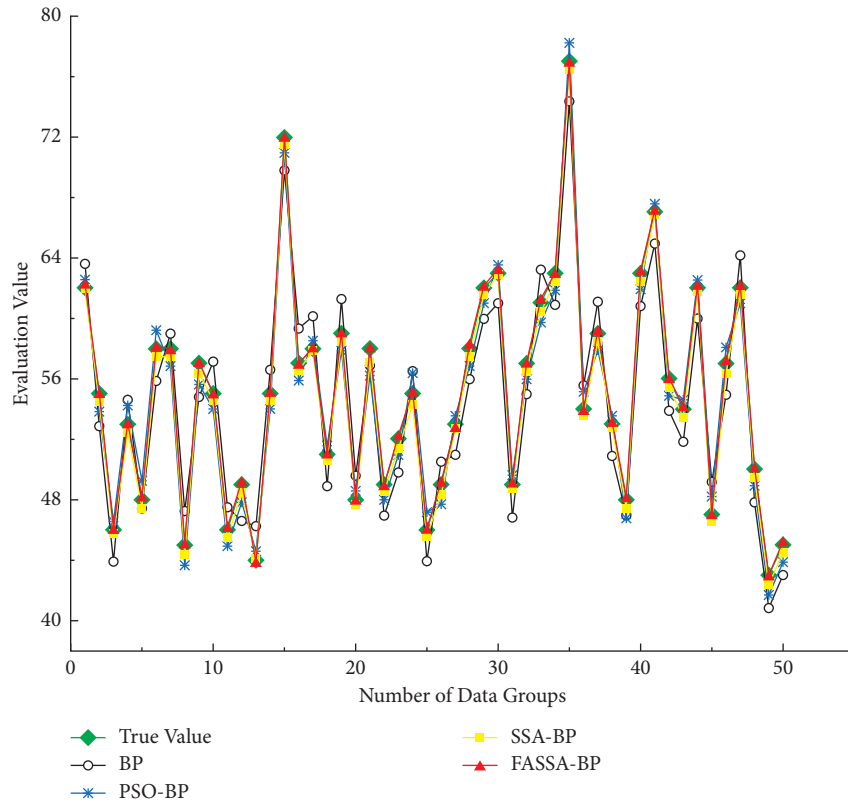


FIGURE 22: Comparison of test results of each model in Experiment 2.

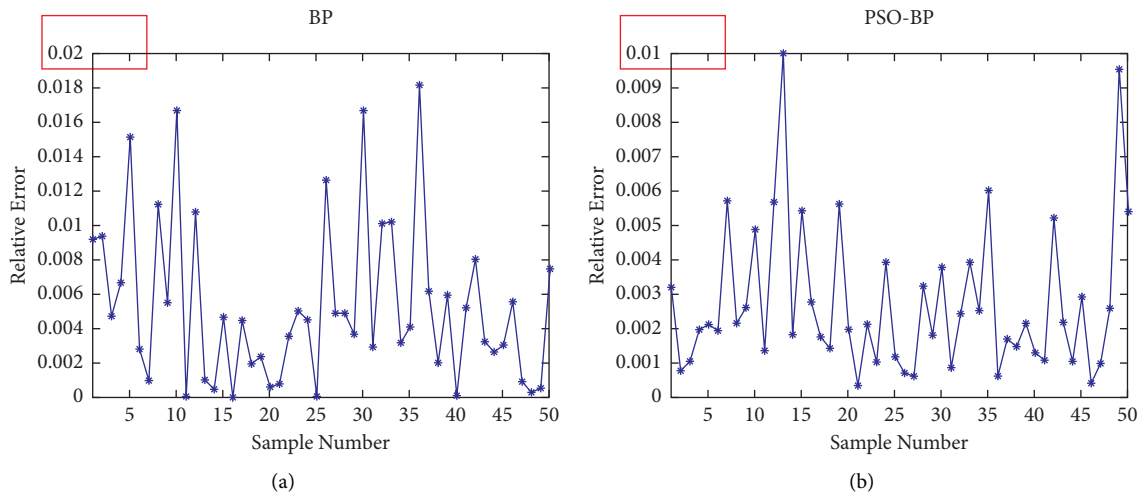


FIGURE 23: Continued.

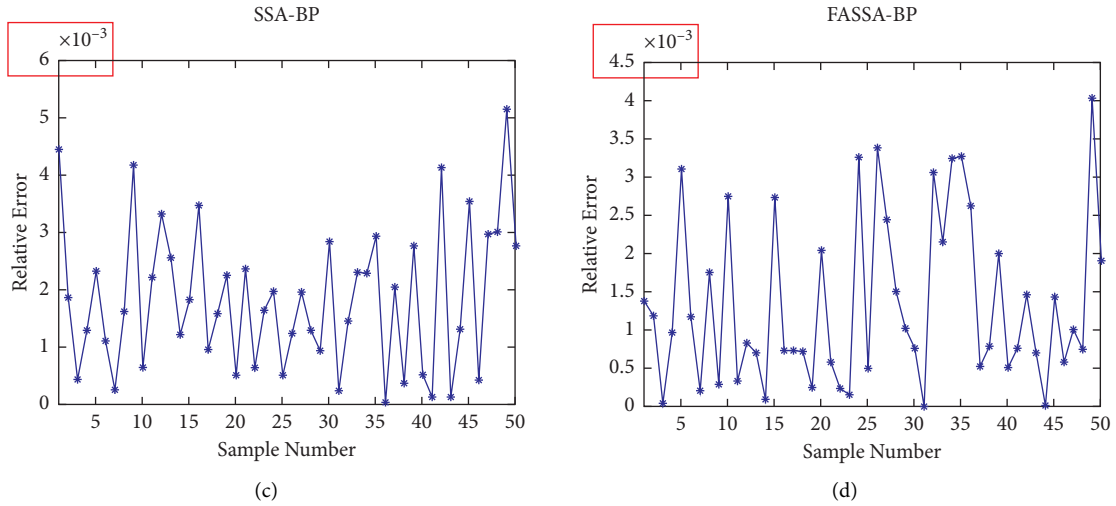


FIGURE 23: Relative error of each model test set sample in Experiment 2: (a) BP; (b) PSO-BP; (c) SSA-BP; (d) FASSA-BP.

TABLE 9: Experiment 2 data results of four model performance evaluation indexes.

Performance evaluation index	BP	PSO-BP	SSA-BP	FASSA-BP
MRE (%)	0.4845	0.2722	0.1547	0.1441
MSE (%)	8.4504	3.8626	1.9062	1.0151
MAE (%)	25.39	14.63	10.11	7.93
RMSE	0.2957	0.2049	0.1329	0.1003
R	0.9824	0.9946	0.9997	0.9998

TABLE 10: Relative error between the true value of the sample and the evaluation value of the FASSA-BP model.

Test set samples	True value	Order of evaluation	Model evaluation value	Corresponding level	Relative error	Grade accurate level
$S_1$	62	Level three	62.1728	Level three	0.0028	Yes
$S_2$	55	Level three	55.0090	Level three	0.0002	Yes
$S_3$	46	Level two	46.0456	Level two	0.0010	Yes
$S_4$	53	Level three	52.9477	Level three	0.0010	Yes
...	...	...	...	...	...	...
$S_{47}$	62	Level three	62.0847	Level three	0.0014	Yes
$S_{48}$	50	Level two	50.0713	Level two	0.0014	Yes
$S_{49}$	43	Level two	42.9258	Level two	0.0017	Yes
$S_{50}$	45	Level two	45.1327	Level two	0.0029	Yes

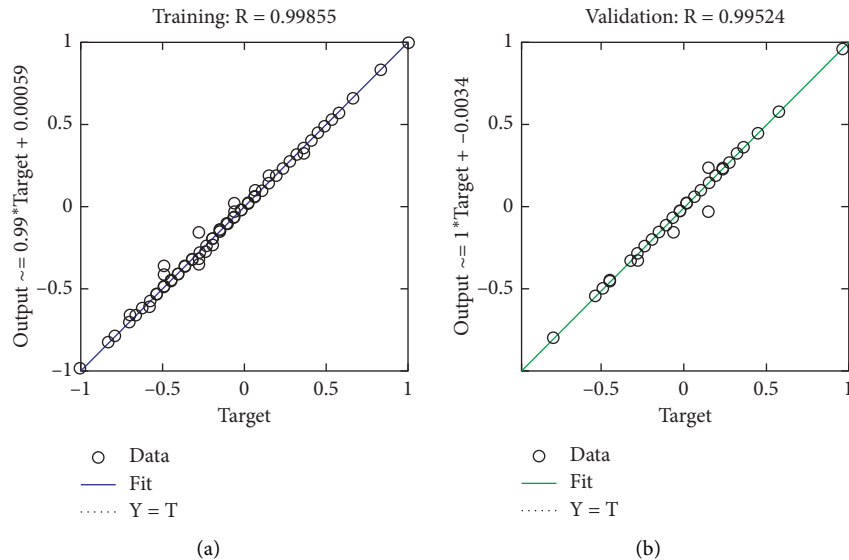


FIGURE 24: Continued.

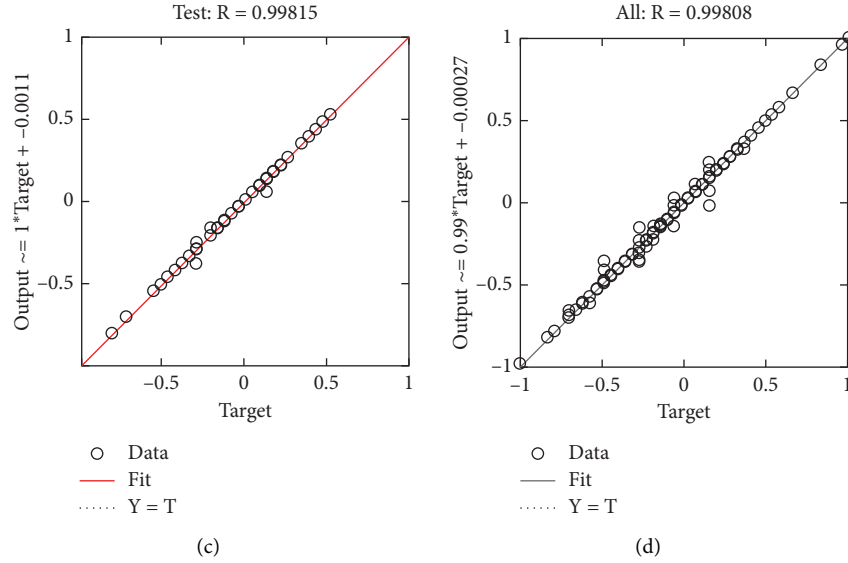


FIGURE 24: Regression analysis of the FASSA-BP model evaluation value and expected output value.

TABLE 11: Five intelligent manufacturing capability maturity levels of subsidiaries.

Subsidiary number $C$	Network output value $N$	Corresponding grade $R$
C1	59.0083	Level three, $R3$
C2	62.0154	Level three, $R3$
C3	54.9953	Level three, $R3$
C4	72.9602	Level four, $R4$
C5	43.0472	Level two, $R2$

shortcomings, formulate reasonable improvement measures, and improve the level of enterprise intelligent manufacturing.

## 6. Summary

In order to help manufacturing enterprises diagnose the level of intelligent manufacturing capability and identify the gap, this paper conducts research on the maturity evaluation of intelligent manufacturing capability based on the maturity theory. By using the knowledge of computer science such as neural network and intelligent optimization algorithm, the evaluation model based on FASSA-BP algorithm is proposed. Improving and perfecting the theories and methods in the field of maturity evaluation of intelligent manufacturing capability can not only enrich the relevant theories of intelligent algorithms, but also broaden its application fields. The specific research work is summarized as follows:

- (1) In view of the shortcoming that the sparrow search algorithm is easy to fall into local optimum, the disturbance strategy in the firefly algorithm is introduced to improve it, and an improved sparrow search algorithm (FASSA) is proposed. Low-dimensional and high-dimensional multimodal functions are selected to test the performance of the algorithm. Experiments show

that, compared with SSA, FASSA has higher convergence accuracy and speed and better global search ability.

- (2) An intelligent manufacturing capability maturity evaluation model based on FASSA-BP algorithm is established. Firstly, BP neural network is used to train and learn the relationship between the input sample data and the expected output value. Aiming at overcoming the defects of the initial weight and threshold randomization of BP neural network, we use FASSA to optimize the accuracy of network evaluation. Then, we compare and analyze the evaluation results of FASSA-BP model, BP model, SSA-BP model, and PSO-BP model from the aspects of accuracy and stability. The results show that the evaluation results of FASSA-BP model proposed in this paper are more accurate, which provides a new method for the evaluation of intelligent manufacturing capability maturity. Finally, it is applied to the actual case enterprises and combined with the evaluation results to give suggestions for improvement of intelligent manufacturing capabilities.

The research shows that the evaluation of the maturity of intelligent manufacturing capability by this model can help enterprises diagnose the problems existing in the construction of intelligent manufacturing and provide methods

for enterprises to accurately improve their intelligent manufacturing capability. Although this method has certain practical significance and promotional value for enterprises to implement intelligent manufacturing, there is also the problem that the running time of FASSA-BP algorithm needs to be improved. It is necessary to further study this problem in the future.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant nos. 71801108, 62006091, and 62006092). The authors would like to thank Editage (<http://www.editage.cn>) for their English language editing.

## References

- [1] G. Schuh, R. Anderl, and J. Gausemeier, *Industrie 4.0 Maturity Index*, National Academy of Science and Engineering, Munich, Germany, 2017.
- [2] E. Gökalp, U. Şener, and P. E. Eren, "Development of an assessment model for industry 4.0: industry 4.0-MM," *Communications in Computer and Information Science*, vol. 99, pp. 128–142, 2017.
- [3] B. Zhao, N. Guo, X. Yu et al., *White Paper on the Maturity Model of Intelligent Manufacturing Capability*, China Institute of Electronic Technology Standardization, Beijing, China, 2016.
- [4] A. De Carolis, M. Macchi, E. Negri, and S. Terzi, "A maturity model for assessing the digital readiness of manufacturing companies," *Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing*, vol. 1, pp. 13–20, 2017.
- [5] J. Hu and S. Gao, "Research and application of capability maturity model for Chinese intelligent manufacturing," *Procedia CIRP*, vol. 83, pp. 794–799, 2019.
- [6] Y. Zhou, J. Zang, Z. Miao, and T. Minshall, "Upgrading pathways of intelligent manufacturing in China: transitioning across technological paradigms," *Engineering*, vol. 5, no. 4, pp. 691–701, 2019.
- [7] W. Ren, H. Suo, and G. Zhao, "Smart factory capability maturity model in petrochemical industry," *Computer and Applied Chemistry*, vol. 36, no. 3, pp. 247–254, 2019.
- [8] F. Simetinger and Z. J. Zhang, "Deriving secondary traits of industry 4.0: a comparative analysis of significant maturity models," *Systems Research and Behavioral Science*, vol. 37, no. 4, pp. 663–678, 2020.
- [9] A. A. Wagire, R. Joshi, A. P. S. Rathore et al., "Development of maturity model for assessing the implementation of industry 4.0: learning from theory and practice," *Production Planning & Control*, vol. 30, pp. 1–20, 2020.
- [10] N. Ruiz, A. Giret, V. Botti, and V. Feria, "An intelligent simulation environment for manufacturing systems," *Computers & Industrial Engineering*, vol. 76, pp. 148–168, 2014.
- [11] M. Colli, O. Madsen, U. Berger, C. Møller, B. V. Wæhrens, and M. Bockholt, "Contextualizing the outcome of a maturity assessment for Industry 4.0," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1347–1352, 2018.
- [12] J. Lee, S. Jun, T.-W. Chang, and J. Park, "A smartness assessment framework for smart factories using analytic network process," *Sustainability*, vol. 9, no. 5, pp. 794–808, 2017.
- [13] X. Liu, X. Zhou, B. Zhu, K. He, and P. Wang, "Measuring the maturity of carbon market in China: an entropy-based TOPSIS approach," *Journal of Cleaner Production*, vol. 229, pp. 94–103, 2019.
- [14] J. Xiao, Z. Ying-qi, and J. Xu, "Research on intelligent manufacturing maturity evaluation model based on AHP and DHNN," *Journal of Systems Science*, vol. 28, no. 2, pp. 105–110, 2020.
- [15] X. Xu and C. Xiao, "Research on evaluation system of intelligent manufacturing ability," *Smart factory*, vol. 6, pp. 59–62, 2018.
- [16] Z. Zhi-qiang, "Research on evaluation of intelligent manufacturing capacity based on catastrophe progression method," *Economic Forum*, vol. 578, no. 9, pp. 27–32, 2018.
- [17] K. Shao and Y. Wen, "Research on comprehensive evaluation of intelligent manufacturing capability based on factor Analysis," *Logistics Sci-Tech*, vol. 40, no. 7, pp. 116–120, 2017.
- [18] S. Gillard, N. Banach, E. Barlow et al., "Developing and testing a principle-based fidelity index for peer support in mental health services," *Social Psychiatry and Psychiatric Epidemiology*, vol. 10, pp. 1–9, 2021.
- [19] W. Yang, K. Xu, J. Lian, C. Ma, and L. Bin, "Integrated flood vulnerability assessment approach based on TOPSIS and Shannon entropy methods," *Ecological Indicators*, vol. 89, pp. 269–280, 2018.
- [20] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [21] I. Coolen, L.-A. Giraldeau, and M. Lavoie, "Head position as an indicator of producer and scrounger tactics in a ground-feeding bird," *Animal Behaviour*, vol. 61, no. 5, pp. 895–903, 2001.
- [22] Z. Barta, A. Liker, and F. Mónus, "The effects of predation risk on the use of social foraging tactics," *Animal Behaviour*, vol. 67, no. 2, pp. 301–308, 2004.
- [23] W. D. Hamilton, "Geometry for the selfish herd," *Journal of Theoretical Biology*, vol. 31, no. 2, pp. 295–311, 1971.
- [24] D. H. Wolpert and W. G. Macready, *No Free Lunch Theorem for Search*; *Proceedings of the JPL*, Santa Fe Institute, Santa Fe, NM, USA, 1995.
- [25] X. Ding, M. Hasanipanah, H. N. Rad et al., "Predicting the blast-induced vibration velocity using a bagged support vector regression optimized with firefly algorithm," *Engineering with Computers*, vol. 10, pp. 1–12, 2020.
- [26] C. Mei-ying, N. zhi-wei, and Z. Xu-hui, "Overview on glowworm swarm optimization or firefly algorithm," *Computer Science*, vol. 42, no. 4, pp. 19–24, 2015.
- [27] X. Ding, L. Shi, M. Li et al., "Research on intelligent manufacturing capability maturity evaluation based on BP neural network," *Journal of Qingdao University (Natural Science Edition)*, vol. 32, no. 3, pp. 20–30, 2019.
- [28] X. Xue, Y. Zhang, and K. Yu, "Research on the evaluation of intelligent manufacturing capability based on BP neural

- network,” *Journal of Software*, vol. 39, no. 8, pp. 162–166, 2018.
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [30] J.-c. Li, D.-L. Zhao, B.-F. Ge, K.-W. Yang, and Y.-W. Chen, “A link prediction method for heterogeneous networks based on BP neural network,” *Physica A: Statistical Mechanics and Its Applications*, vol. 495, pp. 1–17, 2018.
- [31] B. Liu, R. Wang, G. Zhao et al., “Prediction of rock mass parameters in the TBM tunnel based on BP neural network integrated simulated annealing algorithm,” *Tunnelling and Underground Space Technology*, vol. 95, 2020.
- [32] Y. Wu, R. Gao, and J. Yang, “Prediction of coal and gas outburst: a method based on the BP neural network optimized by GASA,” *Process Safety and Environmental Protection*, vol. 133, pp. 64–72, 2020.
- [33] J. Lyu and J. Zhang, “BP neural network prediction model for suicide attempt among Chinese rural residents,” *Journal of Affective Disorders*, vol. 246, pp. 465–473, 2019.
- [34] L. Xu, T. Quan, J. Wang, T. Aaron Gulliver, and K. N. Le, “GR and BP neural network-based performance prediction of dual-antenna mobile communication networks,” *Computer Networks*, vol. 172, Article ID 107172, 2020.
- [35] L. Wu, Y. Yang, and M. Maheshwari, “Strain prediction for critical positions of FPSO under different loading of stored oil using GAIFOA-BP neural network,” *Marine Structures*, vol. 72, Article ID 102762, 2020.
- [36] B. Jia, R. Dong, and J. Du, “Ozone concentrations prediction in Lanzhou, China, using chaotic artificial neural network,” *Chemometrics and Intelligent Laboratory Systems*, vol. 204, Article ID 104098, 2020.
- [37] Y. Deng, H. Xiao, J. Xu, and H. Wang, “Prediction model of PSO-BP neural network on coliform amount in special food,” *Saudi Journal of Biological Sciences*, vol. 26, no. 6, pp. 1154–1160, 2019.
- [38] D. Ma, T. Zhou, J. Chen, S. Qi, M. Ali Shahzad, and Z. Xiao, “Supercritical water heat transfer coefficient prediction analysis based on BP neural network,” *Nuclear Engineering and Design*, vol. 320, pp. 400–408, 2017.
- [39] M. S. Muthuvalu, V. S. Asirvadam, and G. Mashadov, “Performance analysis of arithmetic mean method in determining peak junction temperature of semiconductor device,” *Ain Shams Engineering Journal*, vol. 6, no. 4, pp. 1203–1210, 2015.
- [40] K. Kraiwattanawong, N. Sano, and H. Tamon, “Investigation on porous properties of carbon/carbon composite cryogels by using weighted arithmetic mean,” *Microporous and Mesoporous Materials*, vol. 231, pp. 57–65, 2016.
- [41] L. Chen, X. Yang, C. Sun, Y. Wang, D. Xu, and C. Zhou, “Feed intake prediction model for group fish using the MEA-BP neural network in intensive aquaculture,” *Information Processing in Agriculture*, vol. 7, no. 2, pp. 261–271, 2020.

## Research Article

# Selection and Ranking of Fog Computing-Based IoT for Monitoring of Health Using the Analytic Network Approach

**Dong Xue,<sup>1</sup> Shah Nazir ,<sup>2</sup> Zhiqiang Peng ,<sup>1</sup> and Hizbullah Khattak <sup>3</sup>**

<sup>1</sup>Sichuan Tourism University, Chengdu 610100, China

<sup>2</sup>Department of Computer Science, University of Swabi, Swabi, Khyber Pakhtunkhwa, Pakistan

<sup>3</sup>Department of Information Technology, Hazara University Mansehra, Mansehra, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Shah Nazir; snshahnzr@gmail.com and Zhiqiang Peng; peszq@sina.com

Received 24 March 2021; Accepted 31 July 2021; Published 9 August 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Dong Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Numerous raised areas are established in the field of fog computing (FC), applied for various purposes, and are evaluated for running analytics on various devices including devices of internet of things and many others in a disseminated way. FC progresses the prototype of cloud computing to network edge leading various possibilities and services. FC improves processing, decision, and intervention to take place through devices of IoT and communicate essential details. The idea of FC in healthcare based on frameworks of IoT is exploited by determining dispersed delegate layer of comprehension between the cloud and sensor hubs. The clouds suggested systems improved to overcome several challenges in ubiquitous frameworks of medical services such as energy efficiency, portability, adaptableness, and quality issues by accommodating right to take care of definite weights of the distant medical services group and sensor networks. The proposed research work has considered the analytic network process (ANP) for selection and ranking of FC-based IoT for health monitoring systems. The approach works in situation when complexity arises for health monitoring. Results of the study show the success of the research for facilitating healthcare.

## 1. Introduction

Fog computing is a structure positioned anywhere between the cloud and the data source where information is computed and stored and the applications are placed to operate the requirements for some specific purpose. FC, like edge computing, takes the assistance of cloud's and power nearby to where information is formed and functioned. Fog computing and edge computing are used by various people as both involve taking understanding and computation together for the formation of information. This is generally performed for reliability improvement but can also be performed for other reason such as adherence and protection. Abundant platforms are recognized in the field of FC which are applied for various purposes and are assessed for successive analytics on different devices comprising IoT devices and several others in a disseminated means. FC improves the prototype of cloud computing to the network edge, leading various potentials and services. FC improves

processing, decision, and intervention to take place through devices of IoT and spread essential details.

FC plays an important role in healthcare, and with the help of this platform, various mechanisms are proposed. Patients suffering from chronic disease are monitored through fog computing [1]. The challenging task is the collection of context-sensitive data associated with patient health. The use of sensor-to-cloud is not only feasible, and the layer of fog computing can make a difference. By doing this, the efficiency of the entire system grows. The issues of security and deployment of fog computing layer are also analyzed. The large growth and developments in the IoT have offered excessive possibilities in the sector of healthcare. The fitness trackers, wireless technology, and body sensors have significant effect on health system reliability and living efficiency. The applications of wearable devices are increasing which measure physiological factors, enhance adherence to exercise in various populations from athletes to patients, and promote health. Fog-assisted computational

efficient wearable sensor networks have been considered for monitoring health through IoT. The applications of wearable devices for monitoring respiratory frequency, heart rate, and movement cadence through physical activity are examined. The collected data through the sensor are uploaded to the connection of IoT system Ethernet module, and the official individual access is provided to monitor the health of athletes through Internet. The wearable devices and its applications demonstrate how the resources of computing cost can reduce while maintaining health requests to access information of health stored in cloud and fog distributed setting. The effectiveness, reliability, and user-friendliness are demonstrated through results of the experiments [2].

Health clinics for sports performance services and medicine of sports team are presented with the applications of technology for supporting athletes return to play in various areas of sports. For analyzing and monitoring health operative optimization of sports person, machine learning algorithms are presented. With the help of experimental results and discussions, the efficiency of the designed system is assessed [3]. Outside the environment of hospital, wireless body sensor nodes are used to monitor patient based on real-time IoT. These devices include bio-sensor for capturing signals from body of patient and transmitting the collected signals through wireless transmitters to a server in real time. The study presented a real-time scheme of encoding performing approximation of wavelet coefficients for sparse encoding of bio-signals and iterative thresholding for reducing bandwidth consumption and reducing energy of the wireless body sensor node. The proposed system results presented energy improvement of system level 96% with an impact of 2% on signal quality [4]. The research has demonstrated a novel "Intelligent" skin monitoring device allowing patients of rural areas to remotely monitor diseases of skin. The approach includes cloud-based IoT and artificial intelligence for analyzing medical images and predictions of disease. The research also addresses the impact of the season. The proposed framework offers diagnosis and prevention for addressing the issues faced by people of remote areas with less facility of skincare [5].

The research work presented in this paper has considered and used the analytic network process approach for selection and ranking of FC based IoT for monitoring of health. The approach works in situation when complexity arises for health monitoring.

The paper is organized as follows. Section 2 depicts the literature of the proposed research inline with FC-based IoT for healthcare. Section 3 represents the literature for evaluation and quality assurance of FC-based IoT for monitoring of health. Selection and ranking of FC-based IoT for health monitoring are briefly given in Section 4. Section 5 shows the conclusion of the paper.

## 2. Related Work

Various approaches are presented for monitoring health-related activities. Various generic monitoring systems are existing. Dhingra et al. [6] have offered research on the analysis integrated fog and distributed computing

arrangement acquainted with limitations of latency, real-time analytics, and network congestion of basic cloud services for monitoring of traffic. The planned approach of cloud and fog framework is linked with Twitter for sending alerts concerning congestion of traffic. The system of IoT and fog-based healthcare for identification and regulation of Chikungunya virus (CHV) is introduced. Fuzzy-C mean clustering is used to evaluate the infected users and to provide customers from the fog layer with thoughtful and alert concerns quickly. In addition, social network analysis is used on cloud workers to speak to the CHV outbreak. With the help of the social network analysis graph, the outbreak role index is measured which signifies the possibility of any user to spread or receive the infection. It produces alerts of warning to healthcare agencies and government for controlling CHV outbreak in risk proven regions. The advantages of the research concerning cloud and fog services for obtaining efficient bandwidth with good quality of service are shown by the experimental results [7].

To find, shape, and screen customers affected with mosquito-borne diseases (MBDs), a new system of IoT sensors, fog computing, and cloud computing is planned to classify, differentiate, and monitor the users with MBD infection. The research aims to stop the outbreak of MBDs at the initial stage. The system is proposed, the similarity factor is measured for differentiating MBDs, and J48 was considered for classification of category for each user infection. The preventive procedures are formed in a split second and sent to the client from the fog layer if any anomaly should occur. Radio frequency identification is considered for identification of the closeness of customers. Temporal network analysis uses proximity information to monitor and talk to the present state of the flare-up of the MBDs. In order to find MBDs, the examining assessments of the system resulted low error rate and high accuracy of 94% [8]. A comprehensive set of topology control (TC) approach for managing and constructing a large-scale smart city network of IoT is presented [9]. The problem of TC is approached in two phases consisting of construction and maintenance phase. In the first phase, a cost effective network of IoT including fog gateways is built while in the second phase the resource utilization is optimized. For realizing the objectives, efficient algorithms are proposed and wide-ranging simulation is done based on experimental and real IoT data sets. The effectiveness of the approach is demonstrated and compared with the available algorithms. The efficient development of innovation in medical care makes it emphatically possible for clinicians to screen and go to patients with constant illness to provide patients with adequate treatment in remote areas.

A scientific examination of robotics telesurgery established on 5G, artificial intelligence, and tactile internet is presented. The paper has described the issues and challenges faced in telesurgery. A fog-based supporting interactive model is demonstrated for expediting and reducing the process latency [10]. The influence of fog computing paradigm in healthcare is studied highlighting the key assistances concerning latency, power consumption, and network usage. With foundation of these parameters, the



fog-based supporting health monitoring system is planned and its evaluation performance is carried out. The experimental results of the approach demonstrated the enhancement possibilities for minimizing data traffic with security enhancement on health information and providing optimized insights of patient health [11].

### 3. Evaluating and Quality Assurance of FC-Based IoT for Observing Health

In the literature so far, various approaches and mechanisms are available to monitor healthcare activities. Verma et al. [12] have presented a new IoT-fog cloud supporting physical system to diagnose and stage classification of ulcerative colitis (UC) through Naïve Bayes classifier and deep neural network. Real-time alert generation is the key point of the study from fog layer in the situation of emergency needed to the user with UC. Haghi Kashani et al. [13] analyzed that the quality of service confirming approaches comes under three categories including service management, application management, and communication management for the years during 2013 till October 2018. The planned study has demonstrated the tools, merits, demerits, evaluation types, and factors of quality of service. Lastly, the challenges and issues based on the reviewed studies have been suggested which need further research for the approaches in fog computing. A business process model and notation extension for enabling the IoT aware business process modeling are proposed at first. The approach considers the heterogeneous IoT and non IoT resources, quality of service constraints, resource capacities, and so forth. Secondly, the approach has considered a novel IoT-fog cloud-based architecture. Thirdly, the research has modeled the proposed BPMN 2.0 extension, smart autistic child, and coronavirus disease 2019 monitoring system. The effectiveness of the proposed system is illustrated with the help of extensive experiments [14]. Figure 1 graphically depicts the IoT-based health monitoring system [4]. The figure consists of three tiers and the details. This figure mainly discusses the IoT-based health monitoring system.

Figure 2 represents the components involved and architecture [15]. The figure contains IoT sensors, GPS module, WiFi module, collected health data, wireless local gateway, wireless remote gateway, and controller.

Karatas and Korpeoglu [16] have considered fog computing and geographically distributed cloud-based architecture of IoT and planned an approach to place IoT data into the components such as fog and cloud data centers. Data are stored in various types, and each of the data type is required by several applications. The research has modeled the problem of data placement and proposed the effective and efficient algorithm for placement of data produced and disbursed through location effectively accessed by applications which require the type of data. For evaluating the proposed research, extensive experiments are done and the effectiveness of the research is shown.

Various searching processes were done in order to obtain associated materials. The Springer library was searched for the query defined for the search process. The

materials were graphically shown for understanding. Figure 3 graphically represents the details of the papers searched in the given library.

The disciplines of the papers were also studied which are shown in Figure 4.

Alemneh et al. [17] have proposed a two-way subjective logic-based trust management enabling a service requester for verifying a service provider is offering a secure and reliable service and the service provider can check the trustworthiness of the service requester. Aslam et al. [18] have presented a solution of an automated fog node audit certification approach which certifies the secure fog layer by the proposed fog layer promising mechanism. Figure 5 shows the details of the papers searched in the IEEE library. The figure shows that more papers are published as conference papers.

Figure 6 is to represent the locations of the conferences where the papers are published.

Figure 7 depicts the topics of publications.

The details of publication titles and publications are shown in Figure 8.

Bandopadhaya et al. [19] have planned an integrated solution of healthcare monitoring for the soldiers implemented in adversative surroundings circumstances through the IoT with distributed computing. Each health parameters of individual requires to be monitored on a real-time basis and succeeding examination of the dataset for commencing suitable healthcare support on time. Bharathi et al. [20] have presented an energy efficient particle swarm optimization-based clustering approach for efficient selection of cluster head amongst different devices of IoT. The devices of IoT for healthcare sensing data are categorized into a form of cluster, and a cluster head will be elected by the use of the approach. The data are forwarded by the cluster head to the cloud server. Fog devices are used for transmitting data of the IoT devices to the cloud server. The ANN algorithm is used as the classification model for diagnosis of the healthcare data for identification of the severity of disease. Debauche et al. [21] presented the health monitoring system based on a fog IoT cloud through environmental signal and physiological permitting to generate relative information for the daily activities. The proposed system monitors behavioral changes and health state of elderly people. Recovery and monitoring rehabilitation processes of patients are provided by the proposed system. The system consists of local gateway for storing data locally, wireless sensor network, and Lambda cloud architecture for processing data.

The ScienceDirect library was considered as search library for achieving associated materials. Figure 9 shows the details of the types of articles along with number of articles.

In this library, the year wise search process was also done to show the increase/decrease in research in the area. Figure 10 depicts the year wise details of paper. The figure shows that there is increase in research work in the area.

The publication titles were searched in the same library, and it is shown in Figure 11.

The subject areas along with publications were found in the same library. Figure 12 graphically represents the subject areas with publications.

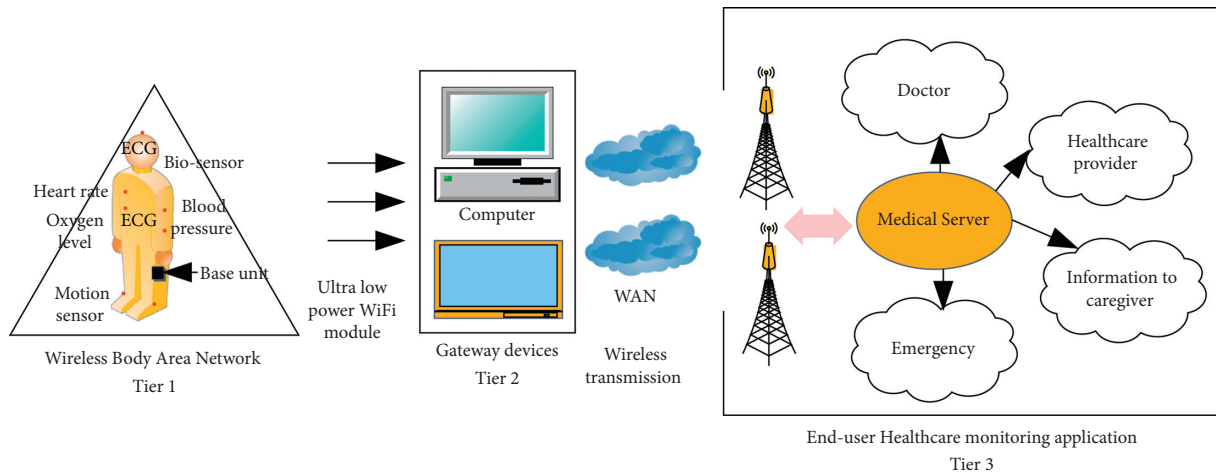


FIGURE 1: IoT-based health monitoring system.

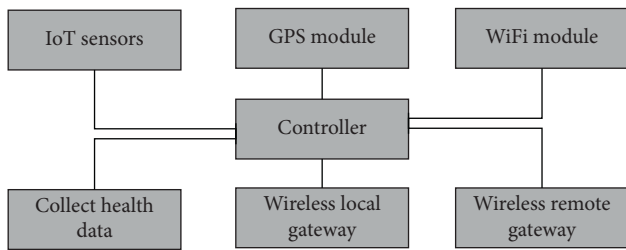


FIGURE 2: Components involved and architecture.

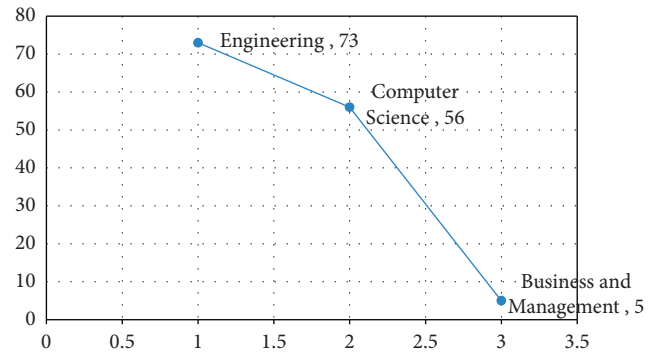


FIGURE 4: Disciplines of papers.

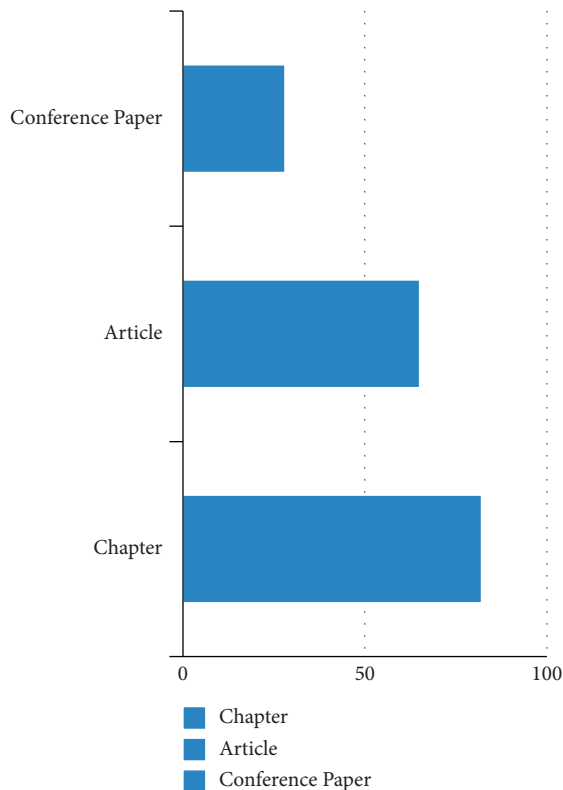


FIGURE 3: Papers details.

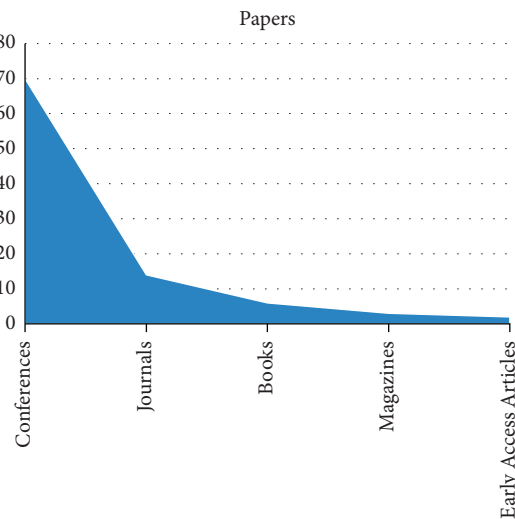


FIGURE 5: Paper types.

The Wiley online library was considered as search library for identification of relative materials. Figure 13 depicts the papers type with publications.

Figure 14 shows the subjects along with the publications.

Figure 15 is the presentation of papers published in the given library.

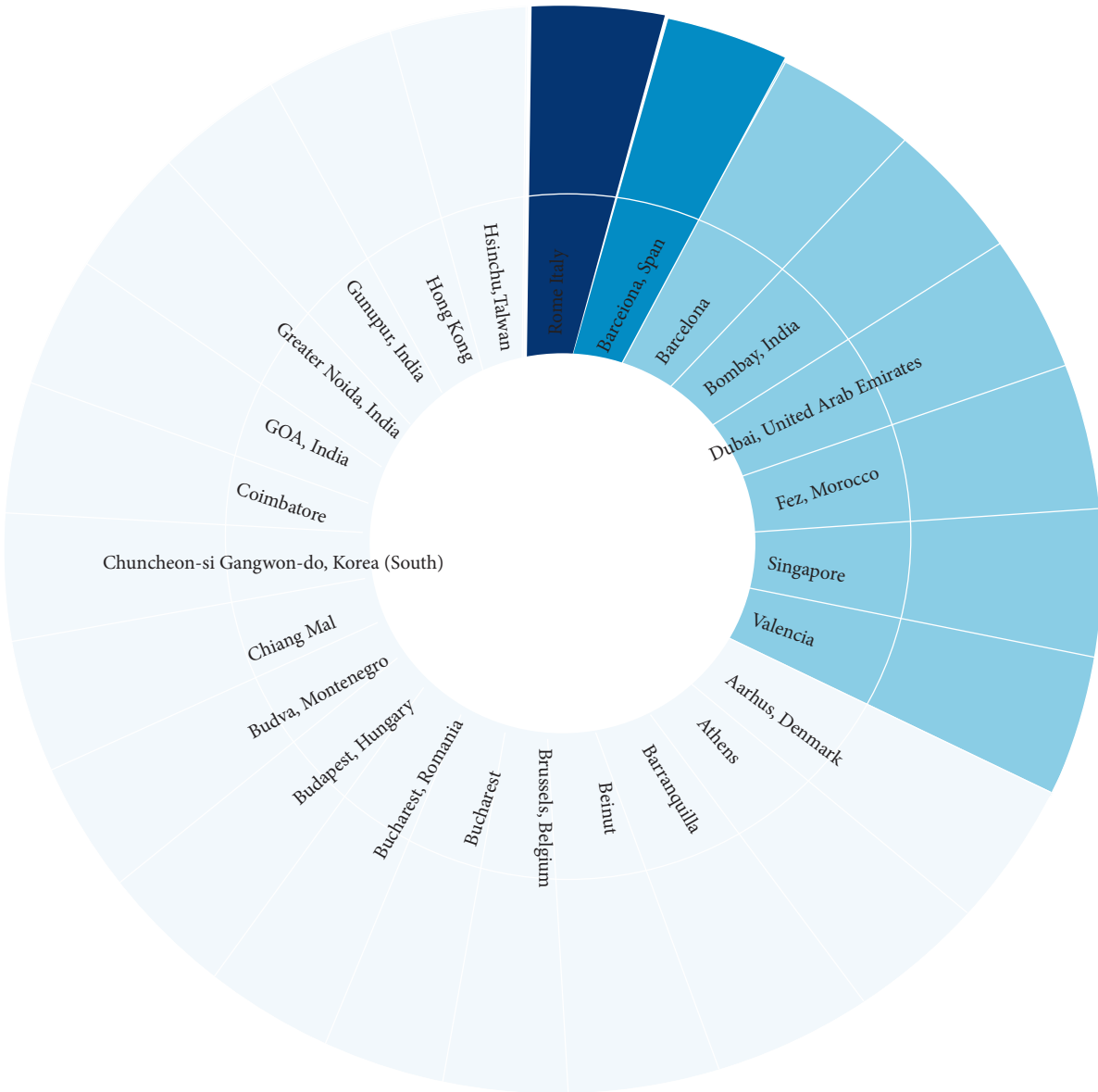


FIGURE 6: Conference locations and papers.

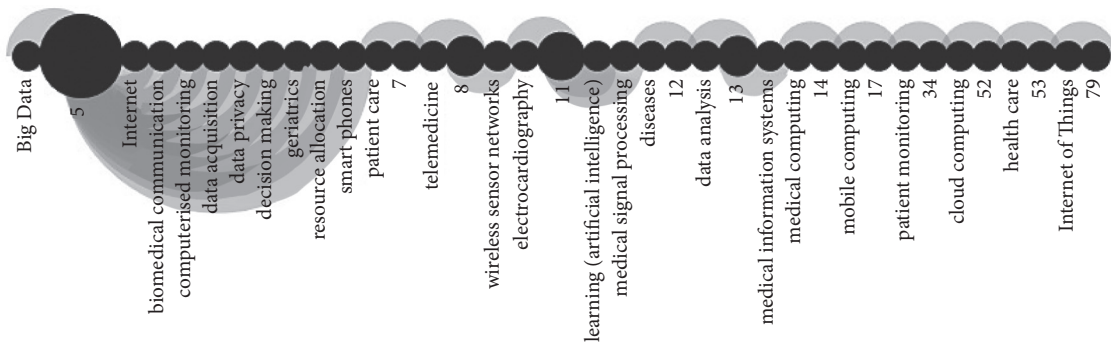


FIGURE 7: Publication topics and papers.

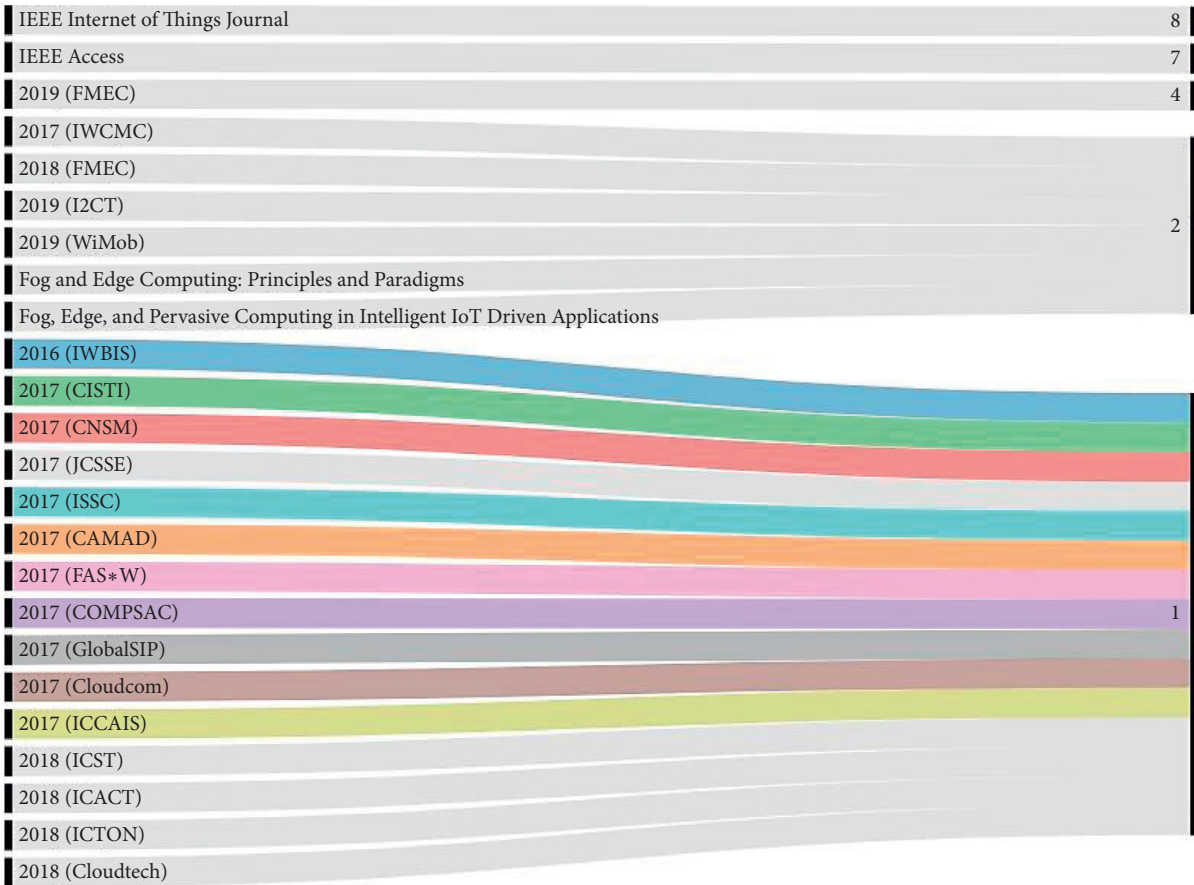


FIGURE 8: Publication titles and papers.

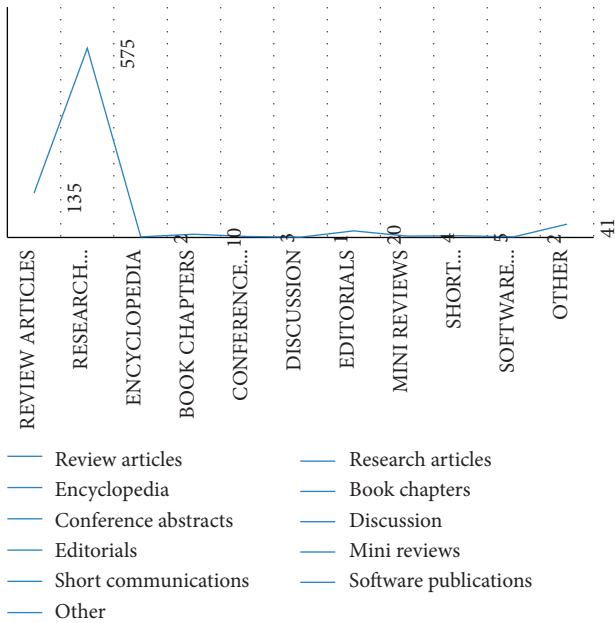


FIGURE 9: Article type and papers.

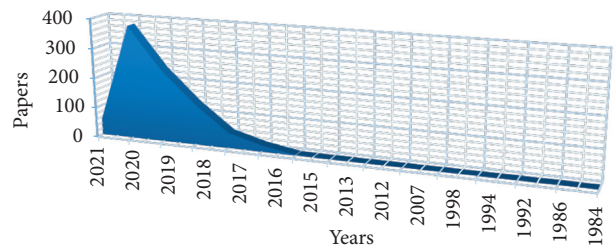


FIGURE 10: Years and papers.

#### 4. Selection and Ranking of FC-Based IoT for Monitoring of Health

Aazam et al. [22] presented a taxonomy of current offloading approaches adopted for the fields such cloud computing, FC, and the IoT. The study has discussed the technologies of middleware enabling offloading in a cloud-IoT case and the significant factors for offloading in a specific scenario. Research prospects associated with offloading in edge and

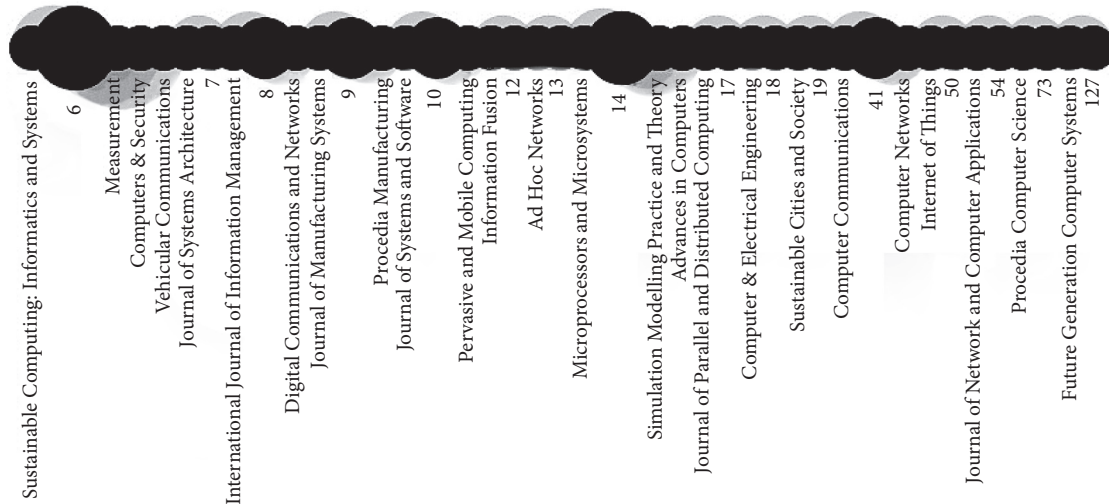


FIGURE 11: Publication titles and papers.

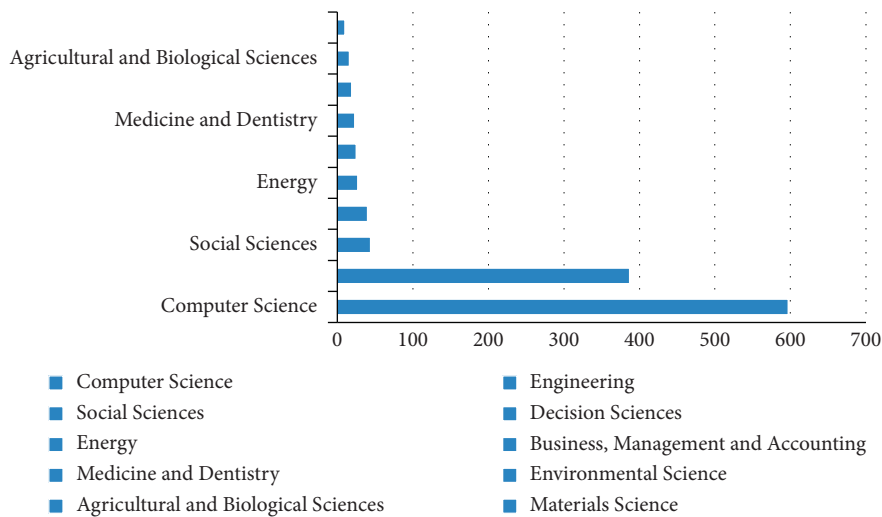


FIGURE 12: Subject areas and papers.

cloud computing are also presented. Abdelmoneem et al. [23] have proposed an interoperable cloud-fog-based architecture of IoT for healthcare. Diversity of the medical cases and mobility of the patient are supported by the proposed architecture. It demonstrates its architecture, user context, and environmental context. Individual module features are elaborated, and the association between the modules is discussed. Results of the simulations are adequate for cost, latency, and miss ratio. The process of this study was carried out step by step in this section. Table 1 shows the details of the components involved in fog computing IoT (FC-IoT).

The details of the pairwise comparison of Table 1 are shown in Table 2.

The same representation is given for another architecture of the given components. The details are given in Table 3.

The pairwise comparison of Table 3 is shown in Table 4. Table 5 shows the architecture for component 1.

The pairwise comparison process of Table 5 is given in Table 6.

The same process of values is given for architecture 2 and is shown in Table 7.

The pairwise comparison process of Table 7 is shown in Table 8.

The same process was done for rest of the architectures. After completing the process of pairwise comparisons, all the calculated values are brought into a single matrix called weighted matrix. Table 9 represents the weighted matrix.

For taking the decision regarding the components and its architectures, the weighted matrix was then converted into the limit matrix. Table 10 shows the limit matrix.

Figure 16 graphically represents the ranking and selection of FC-based IoT for monitoring of health. The figure shows that FC-IoT 3 got the highest ranking for consideration.

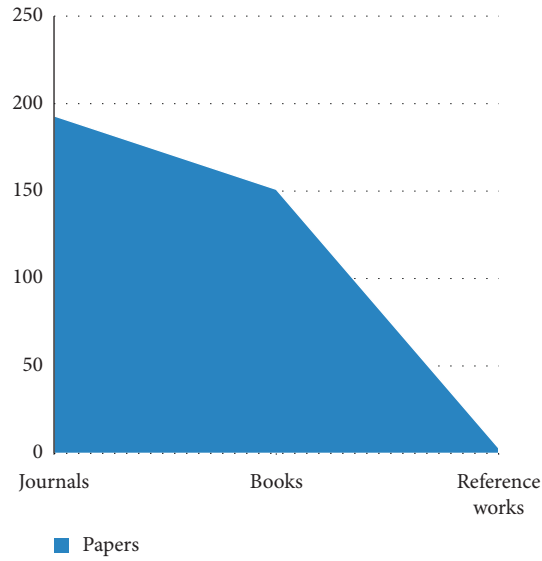


FIGURE 13: Paper type and papers.

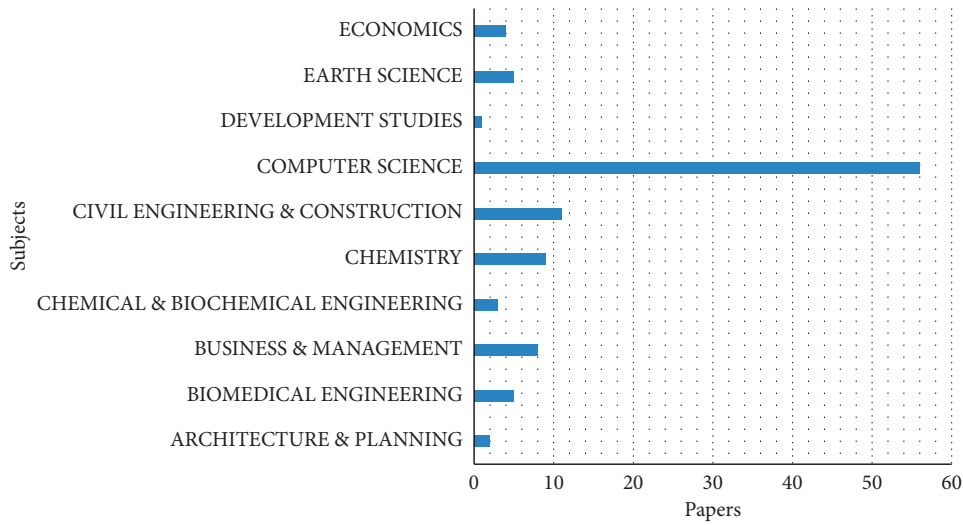


FIGURE 14: Subjects and papers.

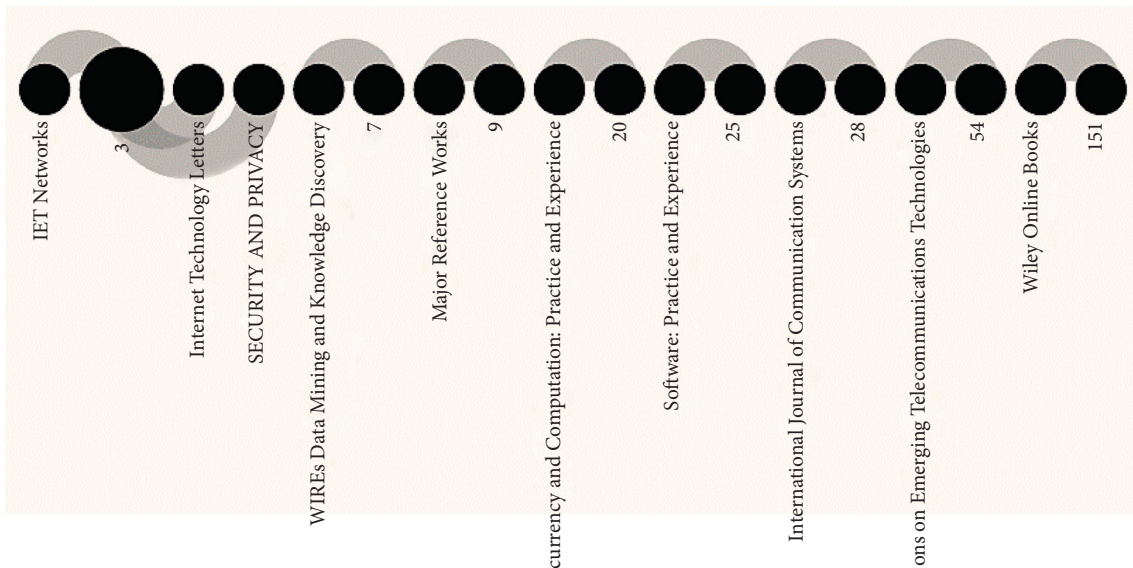


FIGURE 15: Papers published in the given library.

TABLE 1: Details of the components involved.

	IoT sensor	GPS module	WiFi module	Collect health data	Wireless local gateway	Wireless remote gateway
IoT sensor	1	1/2	1/3	1/2	1/3	1/2
GPS module	2	1	1/2	1/3	1/2	1/3
WiFi module	3	2	1	1/3	1/3	1/2
Collect health data	2	3	3	1	1/2	1/2
Wireless local gateway	3	2	3	2	1	1/2
Wireless remote gateway	2	3	2	2	2	1

TABLE 2: Details of the pairwise comparison.

	IoT sensor	GPS module	WiFi module	Collect health data	Wireless local gateway	Wireless remote gateway
IoT sensor	0.08	0.04	0.03	0.08	0.07	0.150
GPS module	0.15	0.09	0.05	0.05	0.11	0.100
WiFi module	0.23	0.17	0.10	0.05	0.07	0.150
Collect health data	0.15	0.26	0.31	0.16	0.11	0.150
Wireless local gateway	0.23	0.17	0.31	0.32	0.21	0.150
Wireless remote gateway	0.15	0.26	0.20	0.32	0.43	0.300

CR = 0.083.

TABLE 3: Details of components for architecture 2.

	IoT sensor	GPS module	WiFi module	Collect health data	Wireless local gateway	Wireless remote gateway
IoT sensor	1	2	3	2	4	3
GPS module	1/2	1	3	3	2	3
WiFi module	1/3	1/3	1	2	3	2
Collect health data	1/2	1/3	1/2	1	3	2
Wireless local gateway	1/4	1/2	1/3	1/3	1	3
Wireless remote gateway	1/3	1/3	1/2	1/2	1/3	1

TABLE 4: Pairwise comparison.

	IoT sensor	GPS module	WiFi module	Collect health data	Wireless local gateway	Wireless remote gateway	
IoT sensor	0.34	0.44	0.36	0.23	0.30	0.214	0.315
GPS module	0.17	0.22	0.36	0.34	0.15	0.214	0.243
WiFi module	0.11	0.07	0.12	0.23	0.23	0.143	0.150
Collect health data	0.17	0.07	0.06	0.11	0.23	0.143	0.131
Wireless local gateway	0.09	0.11	0.04	0.04	0.08	0.214	0.094
Wireless remote gateway	0.11	0.07	0.06	0.06	0.03	0.071	0.067

CR = 0.099.

TABLE 5: Architecture 1.

	FC-IoT 1	FC-IoT 2	FC-IoT 3
FC-IoT 1	1	1/2	2
FC-IoT 2	2	1	2
FC-IoT 3	1/2	1/2	1

TABLE 7: Architecture 2.

	FC-IoT 1	FC-IoT 2	FC-IoT 3
FC-IoT 1	1	3	2
FC-IoT 2	1/3	1	1
FC-IoT 3	1/2	1	1

TABLE 6: Pairwise comparison for architecture 1.

	FC-IoT 1	FC-IoT 2	FC-IoT 3	E.V.
FC-IoT 1	0.29	0.25	0.40	0.31
FC-IoT 2	0.57	0.50	0.40	0.49
FC-IoT 3	0.14	0.25	0.20	0.20

CR = 0.052.

TABLE 8: Pairwise comparison of architecture 2.

	FC-IoT 1	FC-IoT 2	FC-IoT 3	E.V.
FC-IoT 1	0.55	0.60	0.50	0.55
FC-IoT 2	0.18	0.20	0.25	0.21
FC-IoT 3	0.27	0.20	0.25	0.24

CR = 0.019.

TABLE 9: Weighted matrix.

	IoT sensor	GPS module	WiFi module	Collect health data	Wireless local gateway	Wireless remote gateway	FC-IoT 1	FC-IoT 2	FC-IoT 3
IoT sensor	0.000	0.000	0.000	0.000	0.000	0.000	0.076	0.315	0.338
GPS module	0.000	0.000	0.000	0.000	0.000	0.000	0.092	0.243	0.212
WiFi module	0.000	0.000	0.000	0.000	0.000	0.000	0.130	0.150	0.158
Collect health data	0.000	0.000	0.000	0.000	0.000	0.000	0.190	0.131	0.124
Wireless local gateway	0.000	0.000	0.000	0.000	0.000	0.000	0.233	0.094	0.088
Wireless remote gateway	0.000	0.000	0.000	0.000	0.000	0.000	0.279	0.067	0.079
FC-IoT 1	0.312	0.548	0.142	0.164	0.110	0.159	0.000	0.000	0.000
FC-IoT 2	0.490	0.211	0.334	0.297	0.309	0.252	0.000	0.000	0.000
FC-IoT 3	0.198	0.241	0.525	0.539	0.581	0.589	0.000	0.000	0.000

TABLE 10: Limit matrix.

0.2608	0.2608	0.2610	0.2607	0.2607	0.2607	0	0	0
0.1904	0.1904	0.1905	0.1903	0.1903	0.1903	0	0	0
0.1478	0.1478	0.1479	0.1478	0.1478	0.1478	0	0	0
0.1437	0.1437	0.1438	0.1437	0.1437	0.1437	0	0	0
0.1283	0.1283	0.1284	0.1283	0.1283	0.1283	0	0	0
0.1279	0.1279	0.1280	0.1279	0.1279	0.1279	0	0	0
0	0	0	0	0	0.2648	0.2648	0.2645	
0	0	0	0	0	0.3320	0.3320	0.3317	
0	0	0	0	0	0.4025	0.4027	0.4022	

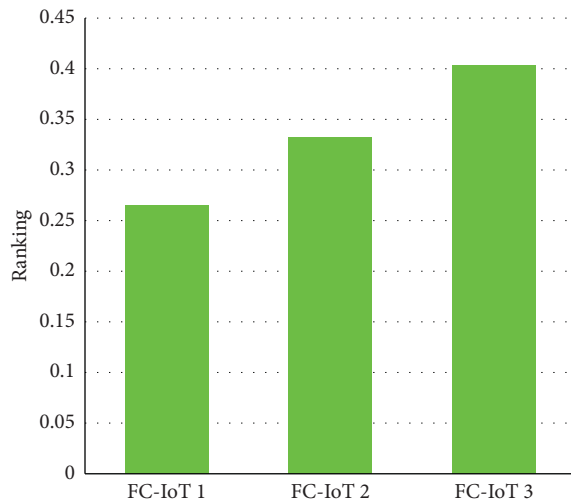


FIGURE 16: Ranking and selection of FC-based IoT for monitoring of health.

## 5. Conclusion

FC is an architecture situated anywhere amongst the cloud and data source where information is computed and stored and the applications are placed to operate the requirements for some specific purpose. Fog computing and edge computing are used by various individuals as both involve understanding and computation together to where the formation of information is done. Numerous platforms and frameworks are considered and proven in the field of FC, applied for various purposes, and are assessed for running analytics on innumerable devices comprising IoT devices and many others in a disseminated

way. FC enhances processing, decision, and intervention to take place by devices of the IoT and distributes essential details. The clouds suggested systems improved to overcome several challenges in ubiquitous medical services frameworks, such as manageability, energy efficiency, adaptability, and quality issues through accommodating right to take care of certain weights of the sensor network and a distant medical services group. The research work presented in this paper has used the analytic network process for selection and ranking of FC-based IoT for monitoring of health systems. The experimental results have shown that the proposed research work is efficient in the selection and ranking of fog computing-based IoT for monitoring of health systems. The approach works well in situation when complexity arises for health monitoring.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] A. Paul, H. Pinjari, W.-H. Hong, H. C. Seo, and S. Rho, "Fog computing-based IoT for health monitoring system," *Journal of Sensors*, vol. 2018, no. 7, Article ID 1386470, 2018.
- [2] S. Li, B. Zhang, P. Fei, P. M. Shakeel, R. D. J. Samuel, and V. Behavior, "Computational efficient wearable sensor network health monitoring system for sports athletics using IoT," *Aggression and Violent Behavior*, Article ID 101541, 2020.
- [3] W. Huifeng, S. N. Kadry, and E. D. Raj, "Continuous health monitoring of sportsperson using IoT devices based wearable technology," *Computer Communications*, vol. 160, pp. 588–595, 2020.
- [4] A. Ghosh, A. Raha, and A. Mukherjee, "Energy-efficient IoT-health monitoring system using approximate computing," *Internet of Things*, vol. 9, Article ID 100166, 2020.
- [5] S. Juyal, S. Sharma, and A. S. Shukla, "Smart skin health monitoring using AI-enabled cloud-based IoT," *Materials Today: Proceedings*, 2021.
- [6] S. Dhingra, R. B. Madda, R. Patan, P. Jiao, K. Barri, and A. H. Alavi, "Internet of things-based fog and cloud computing technology for smart traffic monitoring," *Internet of Things*, vol. 4, Article ID 100175, 2020.



- [7] S. K. Sood and I. Mahajan, "Wearable IoT sensor based healthcare system for identifying and controlling chikungunya virus," *Computers in Industry*, vol. 91, pp. 33–44, 2017.
- [8] S. K. Sood and I. Mahajan, "Fog-cloud based cyber-physical system for distinguishing, detecting and preventing mosquito borne diseases," *Future Generation Computer Systems*, vol. 88, pp. 764–775, 2018.
- [9] K. E. Srinivasa Desikan, V. J. Kotagi, and C. Siva Ram Murthy, "Topology control in fog computing enabled IoT networks for smart cities," *Computer Networks*, vol. 176, Article ID 107270, 2020.
- [10] K. Tiwari, S. Kumar, and R. K. Tiwari, "FOG assisted healthcare architecture for pre-operative support to reduce latency," *Procedia Computer Science*, vol. 167, pp. 1312–1324, 2020.
- [11] P. H. Vilela, J. J. P. C. Rodrigues, P. Solic, K. Saleem, and V. Furtado, "Performance evaluation of a fog-assisted IoT solution for e-health applications," *Future Generation Computer Systems*, vol. 97, pp. 379–386, 2019.
- [12] P. Verma, S. K. Sood, and H. Kaur, "A fog-cloud based cyber physical system for ulcerative colitis diagnosis and stage classification and management," *Microprocessors and Microsystems*, vol. 72, Article ID 102929, 2020.
- [13] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *International Journal of Communication Systems*, vol. 33, no. 8, p. e4340, 2020.
- [14] A. Kallel, M. Rekik, and M. Khemakhem, "IoT-fog-cloud based architecture for smart systems: prototypes of autism and COVID-19 monitoring systems," *Software: Practice and Experience*, vol. 51, no. 1, pp. 91–116, 2021.
- [15] G. Suseendran and D. Balaganesh, "Smart cattle health monitoring system using IoT sensors," *Materials Today: Proceedings*, 2021.
- [16] F. Karatas and I. Korpeoglu, "Fog-based data distribution service (F-DAD) for internet of things (IoT) applications," *Future Generation Computer Systems*, vol. 93, pp. 156–169, 2019.
- [17] E. Alemneh, S.-M. Senouci, P. Brunet, and T. Tegegne, "A two-way trust management system for fog computing," *Future Generation Computer Systems*, vol. 106, pp. 206–220, 2020.
- [18] M. Aslam, B. Mohsin, A. Nasir, and S. Raza, "FoNAC—an automated fog node audit and certification scheme," *Computers & Security*, vol. 93, Article ID 101759, 2020.
- [19] S. Bandopadhyaya, R. Dey, and A. Suhag, "Integrated healthcare monitoring solutions for soldier using the internet of things with distributed computing," *Sustainable Computing: Informatics and Systems*, vol. 26, Article ID 100378, 2020.
- [20] R. Bharathi, T. Abirami, S. Dhanasekaran et al., "Energy efficient clustering with disease diagnosis model for IoT based sustainable healthcare systems," *Sustainable Computing: Informatics and Systems*, vol. 28, Article ID 100453, 2020.
- [21] O. Debauche, S. Mahmoudi, P. Manneback, and A. Assila, "Fog IoT for health: a new architecture for patients and elderly monitoring," *Procedia Computer Science*, vol. 160, pp. 289–297, 2019.
- [22] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, 2018.
- [23] R. M. Abdelmoneem, A. Benslimane, E. Shaaban, S. Abdelhamid, and S. Ghoneim, "A cloud-fog based architecture for IoT applications dedicated to healthcare," in *Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

## Research Article

# An Overview of Computational Models for Industrial Internet of Things to Enhance Usability

Zhen Ying <sup>1</sup>, Iftikhar Ahmad,<sup>2</sup> Saima Mateen,<sup>2</sup> Asad Zia,<sup>2</sup> Ambreen,<sup>2</sup> Shah Nazir <sup>2</sup>,  
and Neelam Mukhtar<sup>3</sup>

<sup>1</sup>Party School of Yangzhou Municipal Committee of C.P.C, Yangzhou 225009, China

<sup>2</sup>Department of Computer Science, University of Swabi, Khyber Pakhtunkhwa, Pakistan

<sup>3</sup>College of Home Economics, University of Peshawar, Peshawar, Pakistan

Correspondence should be addressed to Zhen Ying; yingzhen1215@sina.com and Shah Nazir; snshahnzr@gmail.com

Received 12 February 2021; Revised 17 June 2021; Accepted 28 July 2021; Published 3 August 2021

Academic Editor: Shahzad Sarfraz

Copyright © 2021 Zhen Ying et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the last decade, the Internet of Things (IoT) has grown to connect a large number of smart entities, devices, and components. These connected entities provide a wide range of services to improve the current society of end customers. The Industrial Internet of Things (IIoTs) are revolutionary systems that have linked manufacturing processes with Internet access in order to precisely increase quality of services. These systems have minimized the costs of production through collaboration with electronic objects, accumulating computing, advanced analytics, and smart perception techniques. A demanding analysis of the strengths and limitations of computational models of IIoT is an essential part of the industry and before deciding which approach to use and implement for enhancing usability. Therefore, the goal of this study is to provide feedback and information to the research community and identify patterns in recommendations for future research in the context of process, development, and monitoring of additional technologies of computational models for IIoT. This paper has presented a comprehensive summary of the existing literature on IIoT for providing details about modern industrial revolutions in the context of IIoT. Associated materials were searched and filtered for identification of relevant materials to the proposed study. These materials have been collectively studied with in-depth analysis and then summarized to condense the information of computation models for the readers as well as entrepreneurs. The study will facilitate research community and practitioners to develop novel techniques, algorithms, and tools to automate and facilitate IIoT. This will develop the field of IIoT and will enhance its usability.

## 1. Introduction

The Industrial Internet was identified as linking equipment, individuals, and statistical analysis on the basis of open widespread network. The description of IIoT aims at improving the intelligence of smart services such as healthcare, manufacturing sites, decreasing energy costs, and enhancing performance through its use. The concept of the IIoT offers a wide variety of IIoT applications in operation and maintenance facilities of life services. Since the current IIoT suggests a similar approach to the system's high-level architecture, the ubiquitous use of the word IoT to contribute its use of emerging technologies in the industry is imprecise because it hinders the study of alternative process

models, along with the placement and existence of conventional data processing and related security and performance concerns. The actual world is changed into being digitized and made everything linked to the smart devices. Expansion of smart devices and advancements has permitted humankind to be in steady correspondence at anyplace and anywhere. IoT patterns have made subfragments of the IoT market known as the modern IIoT.

Various approaches have been presented to strengthen the current Industrial IoT concepts and to suggest references for understanding the work of IIoT in various situations, as the commercial internet demanding less delay and high reliability for the transmission rate requires emerging technical assistance such as cloud computing, fog, and fog

computing. Due to this, the IIoT has attracted worldwide interest, so the creation of the IIoT has become a popular option for potential possibilities and challenges for most industrial services. Sun et al. [1] provide a point of view on how hardware automation can contribute to industry. A portion of the key obstacles are included in the system of successful and adaptable operations. In addition to increasing devices' interoperability, these developments have dispatched the Industry 4.0 operation that is concerned with how digital real frameworks and the web of things can make preparation frameworks flexible, customized, efficient, and secure. In addition, the paper presents how these could be defeated by gear automation to form a real digital computerization network. A good combination of these approaches would consider the vision of Industry 4.0 in the planning of the business. Qin et al. [2] discussed the latest technologies of the Industrial Internet thoroughly, including the design of reference, key developments, relative implementations, and potential difficulties. There are summarized comparison models that have been suggested for different application circumstances and their comparative qualities. A range of applications in the Industrial Internet is supported, and the functional technologies like distributed computing, portable edge processing, and cloud figuring, which are ordered by diverse layers in the model, are implemented. In the meantime, future problems and exploration trends are also addressed to encourage further study of the industrial platform.

The proposed study has presented a comprehensive overview and summarized the recent Industrial Internet of Things literature to provide a detailed analysis of the new industrial revolution in the context of IoT. This research includes an overview of the computational and simulated processes and their applications used in solid state research, providing a shared platform for readers and experimenters.

The rest of the paper is structured in different sections. Section 2 represents the literature review associated with the proposed study. Section 3 outlines a topical guide of computational models in industry in a summarized manner. Section 4 depicts the discussions of the proposed study with details of the existing literature. Section 5 shows the conclusion of the paper, and Section 6 represents the future directions of the study.

## 2. Literature Review

The research community is actively involved in proposing new computational models and approaches for the IIoT. The study establishes an IIoT description and analyses IoT methodologies that are associated. In the current literature, it provides an analysis structure for IIoT, as there is a lack of a comprehensive and analytical review which has focused specifically on this subject of research. Next, it describes the process of applied systematic literature review. The findings and observations are then demonstrated by the study concerning the data obtained. Over the last decade, the Internet of Things has progressed to combine a monstrous number of the issues selecting and consolidating administrations that coordinate the necessary constraints.

Consequently, it has been grouped and analyzed those cutting-edge computations for administration determination under constraints in the IoT environment. Abosaif and Hamza [3] have classified and analyzed the existing algorithms for the selection of service under the constraints of quality of service in the IoT environment. Additionally, they have discussed the potential headings of exploration regarding the strategy, use, and estimation of calculations of fresh aid determination. As of late suggested and being worked on, Boncia et al. [4] explored an architecture with the aim of rendering the design and development of multiagent systems feasible for the vision and the requirement of industry in the systems of cyber-physical era. Most of the standardization research focus is on the IEC 61499 standard considered as a base for the incorporation of industrial agents. Therefore, the study has considered relational multiagent systems as a useful integration aid and a way for research in the area.

To achieve environmental maintainability, the research aims to explore the focal points and obstacles of a scenario configuration approach to sustainable manufacturing industry (SMI) achievement as there is a growing need for the manufacturing industry. Configuration of the situation is a promising approximation to the representation of various thoughts of sustainable man-making projects. The current situation and strategies related to SMIs for this function are analyzed in this paper. A case study analysis of Japan's SMI to 2050, Kishitaa et al. [5], was carried out to examine the potential of a situation plan using an approach of back-casting scenario design. The experimental results of the study depicted that computer-aided scenario design is supportive for the classification of rational and logical structure of the scenarios. In the traditional work process, the basic commissioning aspect of modern control frameworks is carried out on the basis of the actual setup and, thus, during the critical time of the undertaking. Schamp et al. [6] outlined a way to deal with virtual commissioning steps by adding specification and functional details and using formal notations inside the 3D computing system by proper documentation. Design projects within the virtual era represent the conditions, the control logic, and transition in the model. These conditions are constantly encountered during a large charge run of the computing system covering all imaginable state components and so on. The time of reduction in undoubting, dispatching, and prior discovery of value problems will be seen by a limited scope study. Menezes et al. [7] presented a methodology for the identification and design (ID) of smart activities in Industry 4.0 (I4) considering the modern industrial revolution. Industry 4.0 is the fourth industrial revolution which is also known as smart or advanced manufacturing. The elements of the paper were advanced analytics, autonomous robots and system integration, etc. The supposed approach examined four groups of I4 elements. These groups were (i) information and communication technologies, (ii) modelling and solution algorithms, (iii) mechatronics, and (iv) high-performance computing. Small-level industries with large-level systems were studied to find their technological gaps and an example of different firms are also given. At the end, the results were

tabulated on the bases of these four groups which compare the technologies that are incorporated with the industrial instances.

### 3. Computational Approaches in the Industry

Comprehensive models have been developed for the scientific analysis of industry-related problems and solutions. However, a new exciting age for solving industry-related issues arose with the invention of computers and the advancement of analytical methods, as sophisticated computational techniques were able to provide approximate but substantially precise solutions. Zhu et al. [8] have found that the IoT is highly helpful to get data from a solitary data hub about an object. That being said, in numerous applications, information may be conveyed in various hubs about a solitary object, and the accumulation of data from these hubs has become a typical activity. For this mission, an architecture distributed service-oriented is proposed. Each manufacturer provides support for its own goods in this design, and information hubs preserve the information gathered without anyone else. Semantic inventions are obtained to fix heterogeneity problems and fill in the institution to assist numerous applications. Finally, for example, utilizing this design to address the problem of the next object is illustrated. Figure 1 represents the generic overview of the IIoT. In the figure, various applications, users, servers, application entities, and IoT applications are connected.

*3.1. Industry 4.0.* Rajput and Singh [9] recommended a mixed-integer linear programming approach for the setup of Industry 4.0 for achieving cleaner production and circular economy through optimizing products-machine allocation. The approach optimizes the trade-off between consumption of energy and processing of machine, cost for gaining cleaner production, and circular economy. The suggested model is further capable of achieving ethical business through sensor deploying for capturing real-time information for establishing the facility of Industry 4.0. Industry 4.0 architecture has been concentrated in countless distributions in the areas of IIoT, enterprise architectures, and also in enterprise integration and cloud manufacturing. An enormous amount of designs had been suggested, yet no one of these had been embraced by countless explorations. Yli-Ojanperä et al. [10] have presented a study with the objectives to overview the studies on Industry 4.0 structures in a manufacturing and survey mindfulness and similarity with Architectural Model Industry 4.0 and architecture of Industrial Internet references. The second objective of the study is to adjust the formerly suggested ideas to Reference Architecture Model Industry 4.0. As for the main objective, it was found that alone a marginal of analysts knew about the said reference models and that by and large practitioners offered no conversation about the similarity of their recommendations with any universally normalized reference engineering for Industry 4.0. Concerning the second objective, it was found

that “Reference Architecture Model Industry 4.0” was adult as for correspondence and data partaking in the extent of the associated world, that further normalization empowering the interoperability of various merchants’ innovations is as yet a work in progress and that innovation normalization empowering executable business measures between organized undertakings was inadequate.

Rausch et al. [11] presented a container scheduling system enabling the platform for edge setting. The scheduler makes perceptual trade-offs between data and computation development, for example, GPU acceleration, and considers the remaining weighted unambiguous loop requirements. In addition, it has presented a technique to usually fine-tune the weights of scheduling constraints for optimizing high-level objective operations such as limiting of execution task time, availability of uplink, or cost of fog execution. The study has implemented model that targets the Kubernetes action system holder path and submits it to an edge testbed designed. The system is assessed with simulations of trace driven in various infrastructure scenarios through traces produced from processing representative workloads. The experimental results showed the scheduler enhanced the quality of task placement and fine-tuning scheduling parameters supported in meeting operational goals. The advanced IoT applications in the technological world are known as Industry 4.0 and have revitalized the idea of industrial “IoT.” The key problems are the authentication and integrity of data as remote correspondence for communication. A few computerized signature plans are suggested in writing to mitigate these difficulties. Nevertheless, such plans experience the adverse effects of inherent key escrow and mystery key appropriation problems due to personality-based or declared growth. Verma et al. [12] have proposed certificate-based proxy signature approach without pairing. The approach is secure in the random oracle model. The comparison of the performance of computational costs and the length of resulting delegation and signature has shown that the approach cost is appropriate and considerable for the environment of IIoT.

Industry 4.0, named I4.0, marks the fourth in the Industrial Revolution that centers intensely on interconnectivity, independence, AI, and continuous information. By 2020, it is assessed that more than 30 billion of the world’s devices will be associated somehow which is a 20 billion with a larger number of devices than today. The steady catching and communicating of information among machines give producing organizations numerous development openings. The IIoT is relied upon changing how we live, work, and play. The main test that looked at the Industrial IoT is security and protection. In the event that we cannot reduce huge amounts of the security and protection given that sway the Industrial IoT, we will not have the option to accomplish its maximum capacity. IoT and the pattern with more noteworthy availability implies that more information accumulated from more places, continuously, to empower ongoing choices and increment income, profitability, and proficiency [13].

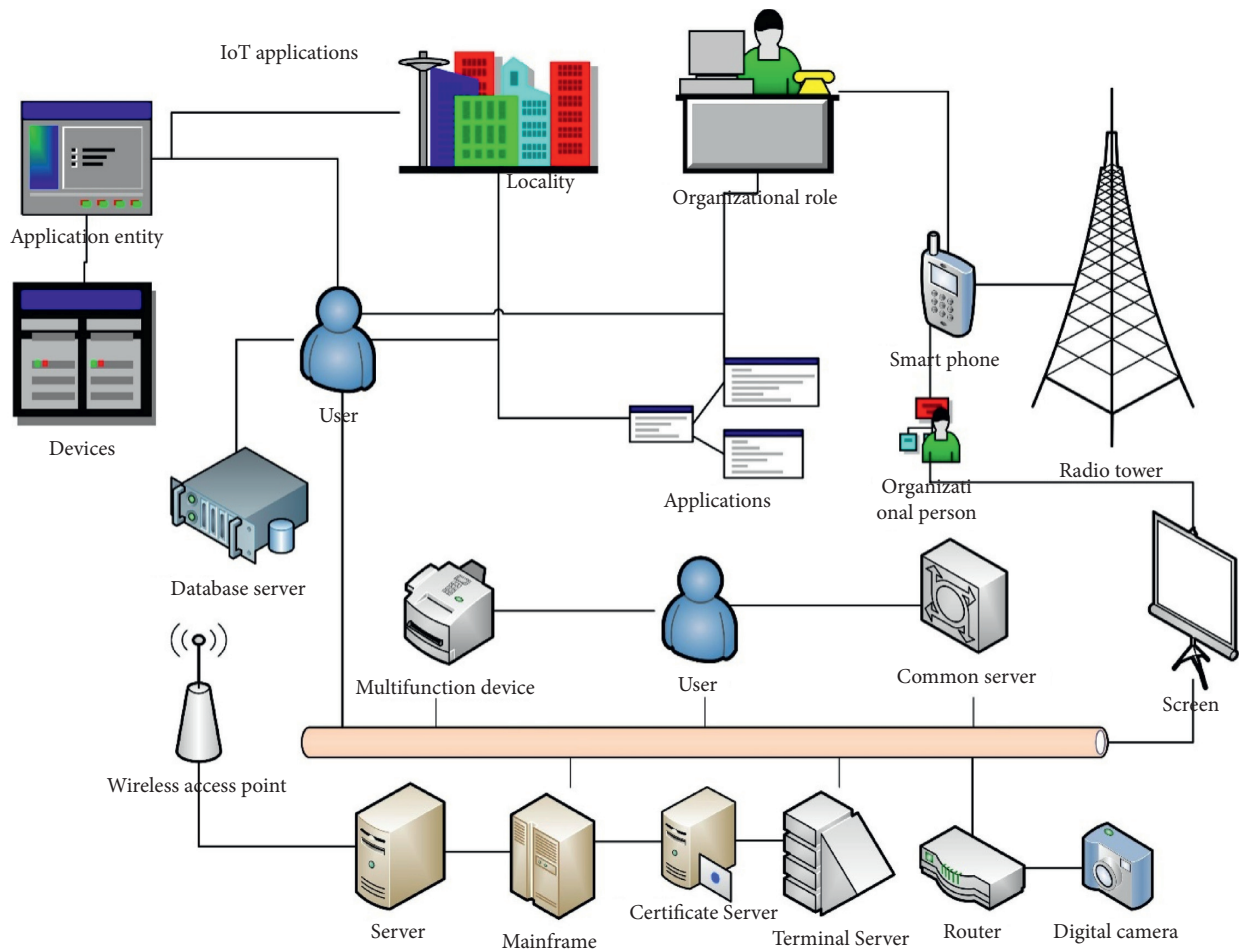


FIGURE 1: Generic overview of IIoT.

*3.2. Vulnerable Automation and Networks Exploitation.* The organizations comprise of industrial materials, services, processes, or manufacturing equipment and wireless devices implanting the activities ongoing for goods during their production period. This suggested control system was formulated by applying a familiar database in the cloud system that controls the process bringing together and manufacturing control logic. This control system has basic uses: configuration of the production system, optimization, control, and monitoring, and historical data storage. The article reports a framework for the implementation and results of the experiments for the assessment of energy. Răileanu et al. [14] have described solution of a semi-hierarchical manufacturing control based on infrastructure of cloud for collection of real-time data from smart devices linked to entities of shop-floor. Such entities consist of resources of industry and devices of mobile integrating the work in the process on products for their cycle of manufacturing. Digitalization is generally acknowledged as a primary mechanical system and is relied upon to immensely affect and change industry transcendently by a significant improvement of the whole worth chain. In any case, its execution cycle stays moderate, specifically for little- and medium-sized ventures, as they will in general have misguided judgments about the intricacies and cost of

digitalization. Kilimis et al. [15] have presented a study for analyzing factors affecting Brandenburg State, SMEs DEU in the process of their decision making for efficient implementation of digital technologies in their routine business. The response of arising automated patterns and uses of the IoT in recent framework is driven in the direction of the development of IIoT. IIoT covers as an additional vision of IoT in the modern area via computerizing perceptive items for detecting, preparing, imparting, and gathering constant occasions in mechanical frameworks. The significant goal of IIoT is to achieve high operational effectiveness, extended effectiveness, and improved administration of recent resources and cycles by item customization, canny observing applications for creation floor shops and machine wellbeing, and perceptive and precautionary maintenance of mechanical hardware. Khan et al. [16] described the novel and obvious definition of Industrial Internet of Things for supporting the readers in understanding the idea of IIoT. The existing researches in the field are defined in the study. The empowering IIoT technologies and the challenges raised by the IIoT are highlighted. Predictive models are a significant achievement factor for smart manufacturing. In a like manner, simple information-driven models just as mixture models are progressively sent inside the assembling environment for ideal control of plants. However, long-term

checking and adopting of predictive models has not been a focus point of study up till now; however, it will probably turn out to be progressively more significant as an ever increasing number of predictive models are deployed. Various proposals for successfully overseeing predictive models in smart manufacturing are presented [17].

Digital Twin Technology has the spot in the top 10 vital innovation patterns in 2017 named by Gartner Inc. Advanced Twin idea draws out the virtual or computerized representation of this present reality hardware devices or framework, although this present reality and the virtual world gets the most elevated synchronization. The advanced representation of the total life pattern of an item from its plan stage to the support stage will give the prophetic analysis of the issues in the business. Various models and prediction horizons are analyzed and contrasted to identify the strongest and effective solutions [18]. Brunelli et al. [19] have proposed Deep Learning approach for forecasting production performance in new products packaging. Gao et al. [20] have revisited the collaborative and smart robotic system role and its enabling technologies, incorporating drive system, horizontal integration of robotic network, robotic sensors, friendly and natural interaction of human-robot, and deep-learned robots. Intelligence such as computing, motion, cognition, and perception will be effortlessly incorporated for meeting diverse societal and industrial needs. Gittler et al. [21] have presented a novel method for auxiliary units online condition monitoring by considering oil mist separator linked as an example to a 5-axis machine tool. The data were analyzed through the principles of machine learning to remove sufficient condition evaluation, including environmental impacts.

*3.3. Cyber-Physical System in Industry.* “Internet of Things” related to industries targets to link industrial resources with each other and take advantages from the information that is produced from the resources. In recent years, the broad orchestration of smart devices and the development of technologies related to information and communication are redefining the importance of resources in our mechanism of industry. Social assets are belonging that work together to make the system better optimization. Cyber and physical systems are developed by inserting resources into micro-controllers or computer systems that run actual or real-time results creating algorithms upon the data created from the resources. Bakliwal et al. [22] have proposed an architecture of a multiagent system for collaborative learning and elaborated the results of operation of the architecture for a prognostics issue. IPv6 is the most encouraging convention for conveying network applications in the IoTs and Industry 4.0. While, in any case, its advance selection, specifically in brilliant assembling frameworks, has been much slower than expected. Despite of this reality that the possible main points of IPv6 for modern web of items are noteworthy, it is not efficiently usable. Feldner and Herber [23] presented a qualitative assessment of IPv6 for IIoT. Interviews were led with five experts and users of IPv6. The experimental results indicated that the key issues are tool supporting and

available libraries are insufficient, the user has to manually configure the communication of IP, and one-fits-all protocol is prevalent. Some recommendations are presented for further development. Based on trajectory data, the regular behaviour of private cars is extracted [24–26].

Adeodu et al. [27] presented an approach of simulating cavitation in industrial pumps through the technology of IoT. In centrifugal pumps, that is faulty and good, cavitation was simulated through the fast Fourier transform algorithm incorporated to an Arduino microcontroller. Artificial Neural Network was used for numerical analysis of the data. Demartini et al. [28] have built up a theoretical framework which clarifies the potential significance of utilizing computerized innovations on effectiveness, versatility, and manageability. It additionally incorporates evidence from different case analyses, which illustrate the core technologies which would potentially be able to add to a maintainable industrial future. The discoveries show some impressive outcomes concerning the supportable implications of the digitalization of the assembling process. In the event that the anticipated advantages can be accomplished through computerized advances, they could enormously affect on sustainability.

Guaranteeing an excellent set of model advancement data that precisely represent the manufacturing system is vital to the productive improvement of an information driven model. The cross-business standard cycle process for data mining structure is utilized to give a reference to what exactly stage process manufacturers will confront one unique consideration and difficulties when building up data-driven model. The study at that point investigates how data-driven models can be used to portray, measure streams, and support the implementation of round economy administrators, measure flexibility and waste valorization [29]. Guan et al. [30] have proposed BC-ETS, a safe and proficient Blockchain-based energy trading scheme. In BC-ETS, the energy exchanging model is isolated into two levels, which can secure protection as well as accomplish the harmony between power organic markets. Moreover, to adjust to the generally feeble processing intensity of the hidden IoT hardware in the energy web, a believability-based value evidence instrument is intended to enormously improve the framework accessibility. The examination shows that BC-ETS can meet the security necessities and has a superior presentation contrasting and other comparable energy exchanging plans. Hansen and Bøgh [31] have presented a comprehensive overview and investigation of how widespread AI and Internet of things are among assembled small- and medium-sized enterprise (SMEs) and talks about the current impediments and openings towards empowering prescient investigation. Right off the bat, a diagram of the empowering influences of AI and IoT is given along the four examination capacities. From now on, an exhaustive writing audit is led and its discoveries are exhibited. At long last, the arising subjects of innovative work, making AI and IoT open advances to SMEs, and the related future patterns and difficulties are summed up. In the study, end devices are considered here as specialists, which settles on its choice on whether the organization will offload the computation

undertakings to the edge devices or not. To handle the resource allotment and undertaking offloading, this paper detailed the calculation resource assignment issues as a total cost postponement of this system. An ideal paired computational offloading decision is proposed and afterward fortification learning is acquainted to taking care of the issue. Recreation results show the viability of this support learning-based plan to limit the offloading cost determined as calculation cost and postpone the cost in mechanical web of things situations [32].

The purpose of the study is to evaluate the obstacles which come in the practical accomplishment of Industry 4.0 for the maintained production. The study focuses on eight obstacles which have been shown graphically and are related to each other. The methodology describes the hurdles under supposition of two groups, having names influenced, and aims to lessen them, giving insufficient organizational resources. The two important hurdles according to our study are “technological upgradation” and “lack of policy framework.” This study is helpful for little firms and those government investigators who are finding hurdles for maintainable production in Industry 4.0. At the end, they recommend many essential directions for managers and policy makers [33]. The Internet of Things is at the core of the next industrial revolution. That being stated, peer-to-peer communication is at the core of industrial IoT, but current industries rely on centralized networking and communication schemes. These centralized networks comply with high standards of IT security. Research has analyzed the industrial communication infrastructure and communication schemes that are already in place and derives the requirements of IT security for the new cloud-based decentralized system. Then, it provides the specification and implementation of the requirements using state of the art communication schemes. Furthermore, the specification and implementations are in strict compliance with IT security standards [34].

*3.4. Cyber-Security and Industry.* Researches have been done for the identification and mapping of possible vulnerable endpoints in a generic prototype of industry where data will cross through aggregation and suggested an effective way to secure a WSN through ensuring the authenticity and integrity of parcels and identity if the user consists the WSN [35]. Wang et al. [36] have proposed an approach of missing value processing that relies on the combination of generative adversarial and denoising autoencoders aiming at missing completely at a random dataset with high missing rate and noise interference in the scenes of industry. The approach has executed the process of training on a discrete dataset with missing values for ensuring the generated dataset in whole is similar to the distribution of features of the original dataset. Experiments were conducted on various dimensional datasets for proving the feasibility and effectiveness of the approach. Industrial automation and control systems have been increasingly isolated from traditional advanced organizations. Boyes et al. [37] have reviewed the IIoT and their associations with ideas such as cyber-physical systems

and Industry 4.0. The study has developed a description of the IIoT and analyzed the associated taxonomies of IoT. An IIoT analysis framework was developed which can be considered for enumerating and characterizing IIoT devices when examining system architectures and security vulnerabilities and threats. Wu [38] introduced a novel computational framework enabling remote remote real-time sensing, scalable computing, and monitoring for detection and prognosis. The framework uses WSN, machine learning, and cloud computing. A prototype of proof-of-concept was designed for demonstrating how the framework enables manufacturers for monitoring machine health conditions and produce predictive analytics. The experimental results are given for demonstrating the ability and utilization of the framework. Machine learning algorithm was implemented on the public cloud for predicting tool wear in milling operations. The objective of the research is to evaluate IIoT business model in the machine-to-machine (M2M) environment. The study provides a supporting framework concept to classify various types of business models in the M2M environment. Such innovative designs pass by different industries and operate a system firm designs and service applications which are encouraged by smart objects. The supporting concepts can recognize different IIoT business models including (a) company-specific business model, (b) value designs, (c) systemic business model, and (d) systemic value design. It also highlights abstract levels of business models basic unit and dissimilarities among the four business models. At the end, the study advances the concept of “value design” refer to the business model, with different actors coupled together, at last bringing about complex organizations and biological systems of assorted things, cycles, and organizations [39].

Salman et al. [40] presented a study with the aim of showing a way of using augmented reality for retrofitting a production machine. With the help of Microsoft HoloLens, the augmented reality was implemented. The study has briefly described the main technologies of Industry 4.0 and ranking them based on absolute frequency existing in the literature. The research questions defined regarding the association between technologies of Industry 4.0 and productivity. Productivity is considered as the main part of comparison for industry [41]. The research has identified frameworks of open-source machine learning, aligned with the domain of industry for processing data produced from IIoT, in its usage, implementation, programming languages, and future aspects [42]. The current paper presents a first-of-a-kind reference design for creating modern computerization frameworks, dependent anxious edge computing, and blockchain innovations. It additionally presents the design of a platform that implements this reference architecture with the end goal of giving functionality in three reciprocal areas, to be specific, robotization, creation, production systems’ virtualization, and data examination. The introduced stage is bound to give some distinct performance and reliability benefits based on its edge computing and blockchain establishment [43]. Kuo et al. [44] have analyzed the relative innovation policy of China, Germany, and USA on the basis of different policy with the aim of industrial

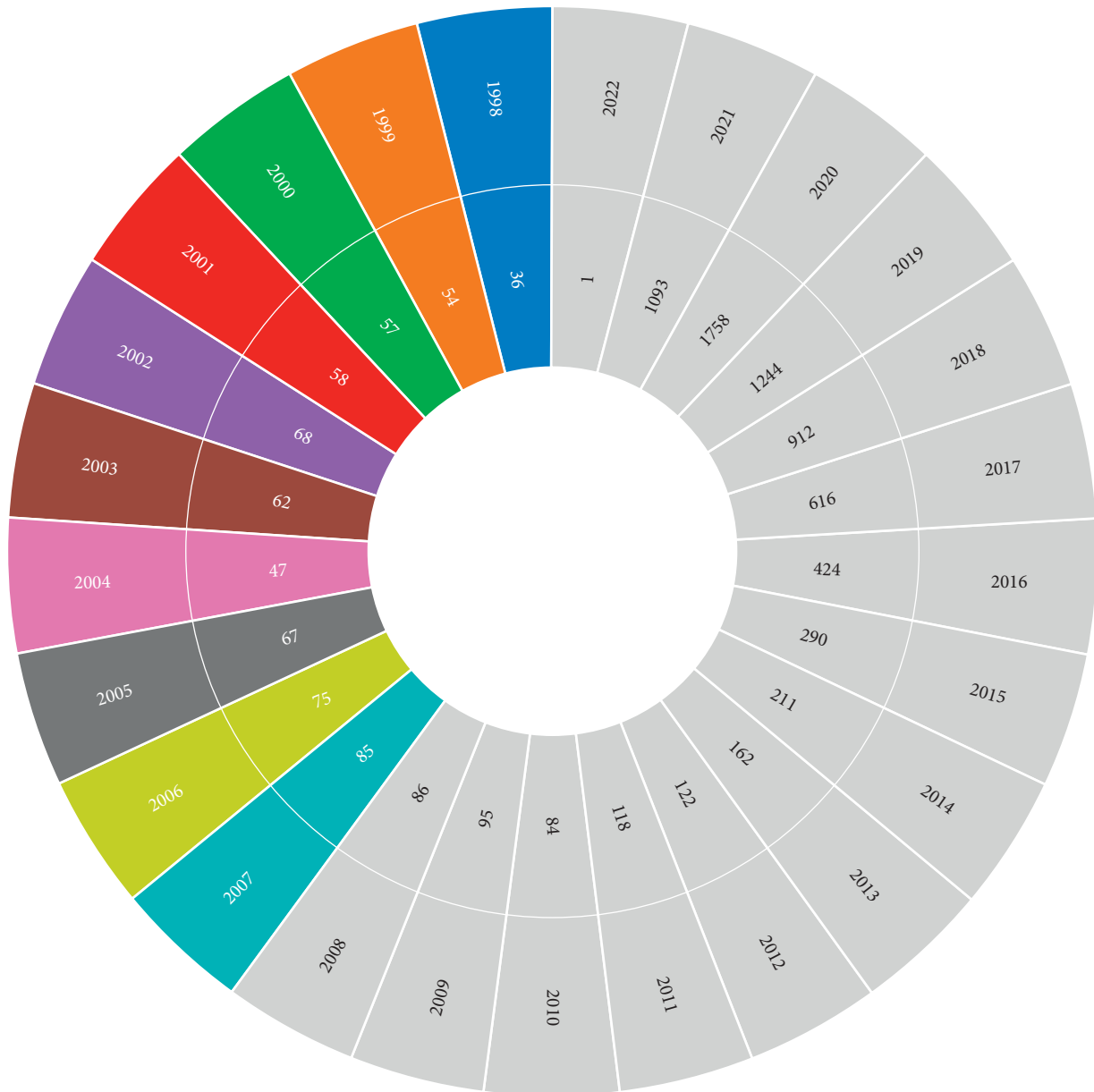


FIGURE 2: Years of publications and the number of articles published.

revitalization of Industry 4.0. Although the research is a theoretical, a practical approach of Rothwell and Zegveld framework of policy is also adopted. Results have showed that orientation policy of the USA tends to favor demand-side policy, political and education, public services, and training policies are favored, whereas policy preferences, public services, technical, scientific development and political favored by Germany and environmental-side policy, political, legal and regulatory, and public service policies by China [44]. Various other approaches have been suggested in the literature in the area of IoT [45, 46].

#### 4. Discussions

Wide-ranging models and techniques have been proposed for the scientific analysis and management of industry-

related problems to overcome their solutions. However, a new exciting age for solving industry-related issues arose with the developments in information technology and the advancement of analytical methods, as sophisticated computational approaches were able to provide imprecise but substantially precise solutions. Research have been done in the area; various approaches, techniques, and solutions were adopted to tackle the issues of modern industry. Figure 2 depicts the years of publications and the number of articles published.

In Figure 2, it is mentioned that with the passage of time, the research in the area is growing and researchers try to come across different solutions from different perspectives. A more number of articles were published in the year 2021, followed by 2020, and so on. This shows the number of researches and its growing rate in the area. The figure depicts



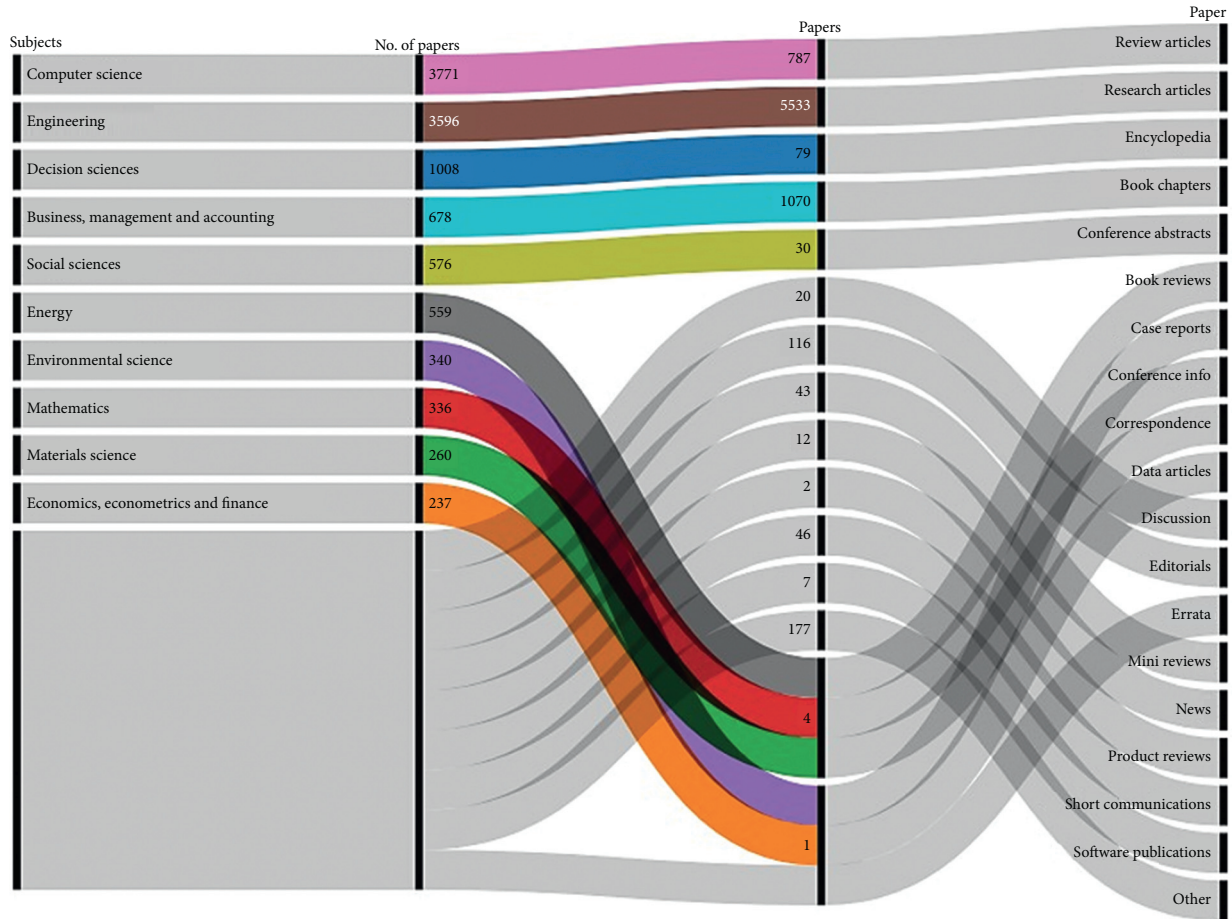


FIGURE 3: Subject, paper types, and the publications numbers.

TABLE 1: Existing approaches related to the proposed work.

Reference	Method	Year
[1]	Perspective for equipment automation	2017
[2]	Recent advances in Industrial Internet	2019
[3]	Quality of service-aware service selection for IoT	2020
[4]	RMAS architecture for industrial agents	2020
[5]	Scenario design to envisioning sustainable manufacturing industries	2016
[6]	Virtual commissioning of industrial control systems	2019
[7]	Identification and design of Industry 4.0	2019
[10]	Agile manufacturing concept to the reference architecture model Industry 4.0	2019
[11]	Optimized container scheduling for data-intensive serverless edge computing	2021
[12]	Certificate-based proxy signature scheme for IIoT	2020
[13]	Industry 4.0: IIOT	2020
[14]	Cloud-based manufacturing control system with data integration from multiple autonomous agents	2018
[15]	Digitalization for SMEs in Brandenburg, Germany	2019
[16]	IIoT: recent advances, enabling technologies, and challenges	2020
[17]	Smart manufacturing and continuous improvement and adaptation of predictive models	2020
[18]	The industry use cases for the Digital Twin idea	2020
[19]	Deep learning-based production forecasting in manufacturing	2020
[20]	From Industry 4.0 to Robotics 4.0	2020
[21]	Condition monitoring system for machine tool auxiliaries	2020
[22]	Multiagent system architecture to implement collaborative learning	2018
[23]	Qualitative evaluation of IPv6 for IoT	2018
[27]	IIoTS-based technology for prediction and control of cavitation in centrifugal pumps	2020

TABLE 1: Continued.

Reference	Method	Year
[28]	Digitalization technologies for industrial sustainability	2019
[29]	Considerations, challenges, and opportunities when developing data-driven models for process manufacturing systems	2020
[30]		2020
[31]	Secure and efficient energy trading in IIoT-enabled energy Internet	2020
[33]	AI and IoT in small- and medium-sized enterprises	2020
[34]	Modeling the Industry 4.0 adoption for sustainable production in micro, small, and medium enterprises	2021
[35]	Securing future decentralized industrial IoT infrastructures	2019
[37]	Mapping vulnerabilities in IIoT landscape	2019
[38]	IIoT: an analysis framework	2018
[39]	Fog computing-based framework for process monitoring and prognosis in cyber-manufacturing	2017
[41]	IIoT business models in the machine-to-machine context	2020
[42]	Relationship between Industry 4.0 and productivity	2019
[44]	Open-source machine learning for IIoT	2020
[46]	Industrial revitalization via Industry 4.0	2019
	Parallel computing for efficient and intelligent IIoHT	2021

that year-wise the publications are rising. Figure 3 briefly represents the subjects, paper type, and the materials published. The figure contains mainly two different categories including the paper type and its subjects of publications. The figure depicts that more number of articles were published as “research article” followed by “book chapter”, and so on. Also, more number of articles were published in the discipline of “computer science.”

The existing approaches used in the area of research are given in Table 1. This table represents mainly the reference of the articles published, along with the method and year of publication.

## 5. Conclusion

The IIoT aims at enlightening the intelligence of smart services such as healthcare, smart cities, and other manufacturing sites, decreasing energy costs, and enhancing performance through its use. The idea of the IIoT offers a wide variety of IIoT smart applications in operation and maintenance facilities. The IIoTs are revolutionary systems that have connected manufacturing processes with Internet access in order to affectedly increase quality and minimizes costs of production through collaboration with electronic objects accumulates computing, advanced analytics, and smart perception techniques. A demanding analysis of the strengths and weaknesses of computational models of IIoT is an essential part of the industry and before deciding which ones to implement for enhancing usability of industry is a challenging issue. The goal of this analysis is to facilitate response and information to the research community and recognize patterns of propositions for future research in context of the process, development and monitoring of other technologies. This paper has summarized the existing related work on IIoT for providing a comprehensive review about modern industrial revolutions in the context of IIoT with the support of computational models. Related materials were searched and filtered for identifications of the most important materials to the proposed study. These materials have been studied with in-depth investigation and then summarized to condense the information of computation models for the readers as well as entrepreneurs.

## 6. Future Direction

The contribution of the study is to present a compressive summary of the existing literature on IIoT to provide details about modern industrial revolutions in the context of IIoT. Related researches and study materials were searched and filtered for identification of associated materials. These materials have been collectively studied and analyzed in-depth, and then summary of the approaches has been presented to condense the information of computation models. This study will facilitate the research community and practitioners to develop and devise novel techniques, algorithms, solutions, and tools to automate and facilitate modern industrial Internet of Things. This will enhance the developments in the field of IIoT and will improve its usability and effectiveness.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was supported by study on the present situation and control of water ecology in Yangzhou Section of Grand Canal (16XZB023).

## References

- [1] B. Sun, S.-L. Jämsä-Jounela, Y. Todorov, L. E. Olivier, and I. K. Craig, “Perspective for equipment automation in process industries,” *IFAC-PapersOnLine*, vol. 502, pp. 65–70, 70.
- [2] W. Qin, S. Chen, and M. Peng, “Recent advances in industrial internet: insights and challenges,” *Digital Communications and Networks*, vol. 61, pp. 1–13, 8648.
- [3] A. N. Abosaif and H. S. Hamza, “Quality of service-aware service selection algorithms for the internet of things environment: a review paper,” *Array, Journal*, vol. 8, p. 2590, 2020.

- [4] A. Boncia, S. Longhia, E. Lorenzonia, and M. Pirania, "RMAS architecture for industrial agents in IEC 61499," *Procedia Manufacturing*, vol. 90, no. 42, <http://creativecommons.org/licenses/by-nc-nd/4.0/>, 2020.
- [5] Y. Kishita, Y. Mizuno, and Y. Umeda, "Scenario design approach to envisioning sustainable manufacturing industries to 2050," *Procedia CIRP*, vol. 48, pp. 407–412, 2016, <http://creativecommons.org/licenses/by-nc-nd/4.0/>.
- [6] M. Schamp, L. V. D. Ginste, S. Hoedt, A. Claeys, E.-H. Aghezaf, and J. Cottyn, "Virtual commissioning of industrial control systems - a 3D digital model approach," *Procedia Manufacturing*, vol. 39, pp. 66–73, 2019.
- [7] B. C. Menezes, J. D. Kelly, and A. G. Leal, "Identification and design of industry 4.0 opportunities in manufacturing: examples from mature industries to laboratory level systems," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2494–2500, 2019.
- [8] T. Zhu, S. Dhelim, Z. Zhou, S. Yang, and H. Ning, "An architecture for aggregating information from distributed data nodes for industrial internet of things," *Computers & Electrical Engineering*, vol. 58pp. 337–349, 349.
- [9] S. Rajput and S. P. Singh, "Industry 4.0 Model for circular economy and cleaner production," *Journal of Cleaner Production*, vol. 277, Article ID 123853, 2020.
- [10] M. Yli-Ojanperä, S. Sierla, N. Papakonstantinou, and V. Vyatkin, "Adapting an agile manufacturing concept to the reference architecture model industry 4.0: a survey and case study," *Journal of Industrial Information Integration*, vol. 15, pp. 147–160, 2019.
- [11] T. Rausch, A. Rashed, and S. Dustdar, "Optimized container scheduling for data-intensive serverless edge computing," *Future Generation Computer Systems*, vol. 114, pp. 259–271, 2021.
- [12] G. K. Verma, B. B. Singh, N. Kumar, M. S. Obaidat, D. He, and H. Singh, "An efficient and provable certificate-based proxy signature scheme for IIoT environment," *Information Sciences*, vol. 518, pp. 142–156, 2020.
- [13] S. Munirathinam, "Industry 4.0: industrial internet of things (IIOT)," in *Advances in Computers*, P. Raj and P. Evangeline, Eds., vol. 117, , pp. 129–164, Elsevier, 2020.
- [14] S. Răileanu, F. Anton, T. Borangiu, S. Anton, and M. Nicolae, "A cloud-based manufacturing control system with data integration from multiple autonomous agents," *Computers in Industry*, vol. 102, pp. 50–61, 2018.
- [15] P. Killimis, W. Zou, M. Lehmann, and U. Berger, "A survey on digitalization for SMEs in Brandenburg, Germany," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2140–2145, 2019.
- [16] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial internet of things: recent advances, enabling technologies and open challenges," *Computers & Electrical Engineering*, vol. 81, p. 106522, Article ID 106522, 2020.
- [17] G. Kronberger, F. Bachinger, and M. Affenzeller, "Smart manufacturing and continuous improvement and adaptation of predictive models," *Procedia Manufacturing*, vol. 42, pp. 528–531, 2020.
- [18] P. Augustine, "The industry use cases for the digital twin idea," in *Advances in Computers*, P. Raj and P. Evangeline, Eds., vol. 117, pp. 79–105, Elsevier, Amsterdam, Netherlands, 2020.
- [19] L. Brunelli, C. Masiero, D. Tosato, A. Beghi, and G. A. Susto, "Deep learning-based production forecasting in manufacturing: a packaging equipment case study," *Procedia Manufacturing*, vol. 38, pp. 248–255, 2019.
- [20] Z. Gao, T. Wanyama, I. Singh, A. Gadhri, and R. Schmidt, "From industry 4.0 to robotics 4.0 - a conceptual framework for collaborative and intelligent robotic systems," *Procedia Manufacturing*, vol. 46, pp. 591–599, 2020.
- [21] T. Gittler, F. Stoop, D. Kryscio, L. Weiss, and K. Wegener, "Condition monitoring system for machine tool auxiliaries," *Procedia CIRP*, vol. 88, pp. 358–363, 2020.
- [22] K. Bakliwal, M. H. Dhada, A. S. Palau, A. K. Parlikad, and B. K. Lad, "A multi agent system architecture to implement collaborative learning for social industrial assets," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1237–1242, 2018.
- [23] B. Feldner and P. Herber, "A qualitative evaluation of IPv6 for the industrial internet of things," *Procedia Computer Science*, vol. 134, pp. 377–384, 2018.
- [24] Z. Xiao, S. Xu, T. Li et al., "On extracting regular travel behavior of private cars based on trajectory data analysis," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14537–14549, 2020.
- [25] Y. Huang, Z. Xiao, D. Wang, H. Jiang, and D. Wu, "Exploring individual travel patterns across private car trajectory data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5036–5050, 2020.
- [26] Y. Huang, Z. Xiao, X. Yu, D. Wang, V. Havyarimana, and J. Bai, "Road network construction with complex intersections based on sparsely-sampled private car trajectory data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 35, pp. 1–28, 2019.
- [27] A. Adeodu, I. Daniyan, O. Omitola, C. Ejimuda, E. Agbor, and O. Akinola, "An adaptive Industrial Internet of things (IIOTs) based technology for prediction and control of cavitation in centrifugal pumps," *Procedia CIRP*, vol. 91, pp. 927–934, 2020.
- [28] M. Demartini, S. Evans, and F. Tonelli, "Digitalization technologies for industrial sustainability," *Procedia Manufacturing*, vol. 33, pp. 264–271, 2019.
- [29] O. J. Fisher, N. J. Watson, J. E. Escrig et al., "Considerations, challenges and opportunities when developing data-driven models for process manufacturing systems," *Computers & Chemical Engineering*, vol. 140, Article ID 106881, 2020.
- [30] Z. Guan, X. Lu, N. Wang, J. Wu, X. Du, and M. Guizani, "Towards secure and efficient energy trading in IIoT-enabled energy internet: a blockchain approach," *Future Generation Computer Systems*, vol. 110, pp. 686–695, 2020.
- [31] E. B. Hansen and S. Bøgh, "Artificial intelligence and internet of things in small and medium-sized enterprises: a survey," *Journal of Manufacturing Systems*, vol. 58, no. 13, pp. 362–372, 2021.
- [32] M. S. Hossain, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Edge computational task offloading scheme using reinforcement learning for IIoT scenario," *ICT Express*, vol. 6, no. 4, pp. 291–299, 2020.
- [33] A. G. Khanzode, P. R. S. Sarma, S. K. Mangla, and H. Yuan, "Modeling the industry 4.0 adoption for sustainable production in micro, small & medium enterprises," *Journal of Cleaner Production*, vol. 279, Article ID 123489, 2021.
- [34] S. Plaga, N. Wiedermann, S. D. Anton, S. Tatschner, H. Schotten, and T. Newe, "Securing future decentralised industrial IoT infrastructures: challenges and free open source solutions," *Future Generation Computer Systems*, vol. 93, pp. 596–608, 2019.
- [35] D. Mourtzis, K. Angelopoulos, and V. Zogopoulos, "Mapping vulnerabilities in the industrial internet of things landscape," *Procedia CIRP*, vol. 84, pp. 265–270, 2019.
- [36] H. Wang, Z. Yuan, Y. Chen, B. Shen, and A. Wu, "An industrial missing values processing method based on generating model," *Computer Networks*, vol. 158, pp. 61–68, 2019.

- [37] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): an analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.
- [38] D. Wu, S. Liu, L. Zhang et al., "A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing," *Journal of Manufacturing Systems*, vol. 43, pp. 25–34, 2017.
- [39] S. Leminen, M. Rajahonka, R. Wendelin, and M. Westerlund, "Industrial internet of things business models in the machine-to-machine context," *Industrial Marketing Management*, vol. 84, pp. 298–311, 2020.
- [40] S. Salman, H. Al-Maena, C. Kuhnena, B. Engela, and M. Schillera, "Smart retrofitting of machine tools in the context of industry 4.0," *Procedia CIRP*, vol. 88, pp. 369–374, 2020.
- [41] S. K. Hubert Backhaus and D. Nadarajah, "Investigating the relationship between industry 4.0 and productivity: a conceptual framework for malaysian manufacturing firms," *Procedia Computer Science*, vol. 161, pp. 696–706, 2019.
- [42] A. I. Khan and A. Al-Badi, "Open source machine learning frameworks for industrial internet of things," *Procedia Computer Science*, vol. 170, pp. 571–577, 2020.
- [43] M. Isaja and J. K. Soldatos, "Distributed ledger architecture for automation, analytics and simulation in industrial environments," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 370–375, 2018.
- [44] C.-C. Kuo, J. Z. Shyu, and K. Ding, "Industrial revitalization via industry 4.0 - a comparative policy analysis among China, Germany and the USA," *Global Transitions*, vol. 1, pp. 3–14, 2019.
- [45] Q. Hao, S. Nazir, X. Gao, L. Ma, and M. Ilyas, "A review on multicriteria decision support system and industrial internet of things for source code transformation," *Scientific Programming*, vol. 2021, pp. 1–9, Article ID 6661272, 2021.
- [46] X. Yang, S. Nazir, H. U. Khan, M. Shafiq, and N. Mukhtar, "Parallel computing for efficient and intelligent industrial internet of health things: an overview," *Complexity*, vol. 2021, pp. 1–11, Article ID 6636898, 2021.

## Research Article

# Ontology-Based Smart System to Automate Higher Education Activities

**Nada Abdullah Alrehaili** <sup>1</sup>, **Muhammad Ahtisham Aslam** <sup>1</sup>, **Dimah Hussein Alahmadi**<sup>1</sup>,  
**Dina Abdullah Alrehaili** <sup>2</sup>, **Muhammad Asif**<sup>3</sup>, and **Muhammad Sheraz Arshad Malik**<sup>4</sup>

<sup>1</sup>Department of Information Systems, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>2</sup>Department of Management Information Systems, Taibah University, Madina 42353, Saudi Arabia

<sup>3</sup>Department of Computer Science, School of Science, National Textile University, Faisalabad 37610, Pakistan

<sup>4</sup>Department of Information Technology, Government College University, Faisalabad 38000, Pakistan

Correspondence should be addressed to Muhammad Ahtisham Aslam; [maaslam@kau.edu.sa](mailto:maaslam@kau.edu.sa)

Received 5 January 2021; Revised 31 January 2021; Accepted 17 February 2021; Published 3 August 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Nada Abdullah Alrehaili et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The need for smart e-learning environments is resulting in new challenges for researchers and practitioners to develop intelligent systems that can be used to automate the Higher Education (HE) activities in an intelligent way. Some common examples of such activities are “analyzing, finding, and ranking the right resource to teach a course,” “analyzing and finding the people with common research interests to start joint research projects,” and “using data analytics and machine reasoning techniques for conducting the exams with different levels of complexities.” Ontological reasoning and smart data analytics can play an important role in analyzing and automating these HE activities and processes. In this paper, we present a framework named as Higher Education Activities and Processes Automation Framework (HEAPAF). The HEAPAF framework can be used to identify, extract, process, and produce the semantically enriched data in machine understandable format from different educational resources. We also present the Higher Education Ontology (HEO) that we designed and developed to accommodate the HE data and then to perform analysis and reasoning on it. As a proof of concept, we present a case study on the topic, “analyzing, finding, and ranking the right resources to teach a course,” which can dramatically improve the learning patterns of students in the growing smart educational environment. Finally, we provide the evaluation of our framework as evidence of its competency and consistency in improving academic analytics for educational activities and processes by using machine reasoning.

## 1. Introduction

In the academic analytic domain, applying machine reasoning and data analytics techniques on semantically enriched data can help to automate the different activities and processes in HE [1]. HE systems have different activities and processes such as “analyzing, finding and ranking the right resource to teach a course,” “analyzing and finding the people with common research interests,” and “conducting the exams with different levels of complexities.” The ultimate purpose of addressing and automating these tasks and processes is to address the different challenges in HE to improve the learning curve of students without comprising the quality of education.

Keeping the importance of challenges in automating HE activities and processes, researchers and practitioners are proposing different methods and solutions to improve the performance in teaching and learning activities. For example, in [2], the authors presented an ontology-based knowledge system for e-learning. The proposed system makes use of smart question answering and data analytic techniques that are based on machine reasoning by using the SPARQL queries. The results of such smart and integrated queries can be used to analyze student’s performance in a virtual learning environment. Similarly, another ontology-based solution (i.e., Curriculum Course Syllabus Ontology (CCSO)) has been presented in [3]. In this paper, the authors presented the design and the model of an educational ontology as an acting model

for data, concepts, and entities within an academic environment. The proposed ontology is used for annotating the potentially remarkable resources that can support the reasoning-based analysis and searching of educational resources. In addition to this, the authors [4] presented an ontology model to perform a semantic-based search of the educational resources. The proposed ontology is based on real-life data taken from Rajiv Gandhi Technical University, Bhopal, to describe real-life education scenarios. In [5], the authors propose an overall process of building university datasets by using Linked Open Data (LOD). The resulting datasets cover maximum vocabularies, data items, RDF entities, and finally interlink them for query purposes. These Tsinghua University Open Data resources demonstrate the RDF-based linking with other datasets to have bigger knowledge graphs that can be used for improved accuracy in the education system.

Despite all these efforts to address different challenges in HE, no ontology-based solution has been presented to automate the different activities and processes in HE. Moreover, highlighting the significant importance of inferencing and reasoning to extract new knowledge from existing data is not yet addressed. In addition to this, investigating and making use of data characteristics in educational data that are mentioned implicitly or/and explicitly is not yet investigated.

In this paper, we aim at addressing the abovementioned limitation and challenges in the current state of the art and to provide the solution to automate the HE activities and processes by using machine reasoning and academic analytic techniques. This paper has the following contributions:

- (i) A Higher Education Activities and Processes Automation Framework (HEAPAF) that can be used to identify, extract, process, and produce the semantically enriched data from different educational resources such as Word files, Excel sheets, CSV files, and traditional databases
- (ii) A Higher Education Ontology (HEO) that can be used to document the knowledge in semantically enriched format about different HE activities and processes based on the LOD principles [6]
- (iii) A case study on the topic, “finding and ranking the right resource to teach a course.” The case study also describes our approach to find:
  - (a) Which faculty member is suitable to teach which course?
  - (b) The criteria for ranking the faculty members, by applying machine reasoning on semantically enriched data
- (iv) Finally, we also provide an evaluation of our methodology

The rest of the paper is organized as follows: Section 2 describes the motivation and the background of this work. The related work is discussed in Section 3. In Section 4, we describe our methodology for automating HE activities by using machine reasoning. Section 5 describes the case study of “finding and ranking the right resource to teach a course.” The evaluation of our methodology is described in Section 6.

Finally, the conclusion and the discussion of the future work is given in Section 7.

## 2. Motivation and Background

Over the preceding two years, there was evidence of a rising number of establishments seeking to disseminate information to their members [7]. Such desires were further strengthened by the transformation of unprocessed primary data into the semantically enriched form, thereby confirming the feasibility of employing and sharing such data for various purposes. Educational establishments possess a vast amount of data that can be accessed by users to automate different activities and procedures in university programs.

Semantically enriched data can also be deployed by a number of other providers including governmental bodies, schools, colleges, and universities. On an individual basis, it can be deployed by students, parents, and teachers, and can be used for various forms of learning resources. Accordingly, it is clear that the learning arena is a good means of employing semantically enriched data to enable the automation of related activities.

The educational processes can be assisted by adding semantics to education data. It can also reduce the obstacles faced since it provides the transparency that will help avoid accountability in the decision-making process. As a matter of transparency, the educational data used must be Findable, Accessible, Interoperable, and Reusable (FAIR) [8] and should be associated with the target audience and comprehended for its purposes.

The openness and availability of the education material is not the only benefit of such a system, another benefit is derived from the associations formed. Moreover, there is an added benefit from the interlinking of data processes as it allows the data to be searched and browsed. The educational data can be put into a context to create new information and services by using Linked Open Data (LOD). The scope of this research is to develop a framework that can support the automation of different activities and processes in HE by making use of machine reasoning and ontologies. As a case study and proof of concept, we implemented the framework at the Faculty of Computing and Information Technology (FCIT) at King Abdulaziz University (KAU).

## 3. Related Work

Lately, there has been a greater emphasis on the use of semantically enriched linked data in varying fields, and this also includes that of the learning environment. One study by Dietze et al. [9] indicated the rise in reliance on LOD by many universities. They concluded that the primary difficulties that will be faced by educational establishments will be their great level of dependency on technology and extensive use of linked data. This involves (1) the combining of a varied and vast number of heterogeneous resources, (2) the capability of acclimatizing to many modifications through services that are provided, e.g., Application Programming Interfaces (APIs) on the web, (3) planning and facilitating metadata via varied learning sources, and (4) the forming of

primary data that is, for instance, derived from textual reports that are deeper and more reliant. That said, the time of many researchers has been employed in combating such difficulties and it is about time that learning establishments benefitted from the provision of linked data. Various platforms have come to be in the learning environment, and these consequently supply the linked data that is to be directly used or reused.

Numerous researchers have made and developed ontology for universities; here, we review some of them. A “knowledge-based university examination ontology” [10] was created and developed to provide a huge degree of help for the examination system in the university by the utilization of the protégé tool. In this work, the authors presented the organization of the examination system by dividing it into several sections (classes, subclasses, objects, and data properties) and by making the use of ontological reasoning. Also, in [11], the authors presented an ontology for Massive Open Online Courses (MOOCs) domain in the Coursera platform to retrieve the educational data quickly according to the learners’ requests. They ensure the competency and consistency of their ontology by using the reasoning of Harmit and Pellet reasoner. Finally, the OWLViz Protégé plugin has been used to visualize the ontology structure and to provide the whole picture of all relations among the classes. Another ontology for the university was developed in [12]. In this work, the main focus of the authors was on building university ontology methods using protégé 4.0.

Similarly, in [13], the authors presented a framework for building an ontology to perform the process of information integration in four specific universities in Iraq. This work focuses on assisting postgraduate students in the process of searching for information relevant to postgraduate research. Also, in [14], authors presented an ontology to manage the different university-level activities [14] focusing on automating the activities and minimizing the administrative burden.

In [15], the authors presented an ontology in the domain of HE (focusing on the engineering domain). The authors also describe the consistency and evaluation of the proposed ontology by using different reasoners such as Fact++ and Pallet. Finally, they used the SPARQL endpoint to query and to perform reasoning on the data as individuals. The main limitation of this work is that it is specific to the engineering field and cannot be generalized for HE activities. A university-level ontology is described in [4]. The authors also described different methods of conducting reasoning between instances of different superclasses and subclasses. A SPARQL endpoint is also described to query the knowledge and to analyze the results. Furthermore, Ullah and Hossain [16] created ontology on the domain of university, proposed a general framework for the process of ontology searching, and explained the searching process through university ontology. Also, it provides several ways of reasoning and inferencing ontology.

In Abir et al. (2016), the authors focused on creating an ontology for the technical university of Rajiv Gandhi Bhopal, India. Their main goal was to provide the greatest help in the

process of internship assignment in the university by applying ontologies and machine reasoning. In addition, they also proposed the system of a semantic recommender for the educational processes and to enhance the results of the process of querying. They also proposed a system of semantic matching of educational processes [17]. E-campus ontology for educational purposes is presented in [18]. The proposed ontology is very specific to and focused on learning activities and a hierarchy programming language such as C-Sharp. For developing ontologies, they provide a hybrid methodology based on the approaches of software engineering.

Zeng et al. [19] provide an ontology model for the courses of the university by applying a bottom-up approach in the ontology of courses, which can explain many open courses and draw the knowledge of the field from open courses in addition to providing the greatest benefits to learners, especially learners in the process of finding their preferred courses. They also developed and built a system aimed at retrieving information-based ontology by the use of the fuzzy ontology framework of the university in the process of managing scientific research.

The ontology of Ahlia University is designed and presented in [20]. SPARQL and DL queries are used to retrieve the direct and inferred data from the ontology. In [21], the authors described a meta-model design and relevant ontology. They also explained the process of the methodology developed on the ontological improvement by applying a method of semi-supervised learning. An ontology-based e-learning system has been presented in [22]. The proposed system can be used to review the problems currently existing in Russian education, which include the poor structure of educational resources and the lack of connections between their components. It also provides a platform called Information Workbench and an ontology-based model that is used in the system. Still, the proposed system does not address the problems and issues in automating educational activities and processes to minimize human involvement and improve efficiency and accuracy. The authors in [23] presented a model for the integration and the extraction of multi-source software knowledge. Categories of entities for software were defined as well as they presented a method for keyword extraction based on K-means, TF-IDF, and TexRank methods. Most of the previous studies started from the phase of building an ontology in various fields, especially education without focusing on the activities and processes of HE. In our research, we address the phases before building and developing an ontology in terms of specification, knowledge acquisition, and the conceptualization phases. As well as how to benefit from the extracted information. For example, our ontology moved to an advanced stage, where in addition to finding certain results, by defining some criteria, we can also get more accurate results. In the next section, we address the abovementioned limitations of the existing work and present our methodology that can be used to automate the HE activities and processes by using ontologies and machine reasoning.

#### 4. Higher Education Activities and Processes Automation Framework (HEAPAF)

Higher educational bodies and institutions have a lot of data that are related to different activities, procedures, and processes in educational systems. These data can be used to automate HE activities and procedures. The main issue with these data is their availability in traditional formats such as Doc files, Excel sheets, and CSV files. Here, we describe our framework (i.e., Higher Education Activities and Processes Automation Framework (HEAPAF)) that can be used to identify, extract, process, and produce the semantically enriched data in machine-understandable format from different data sources. The resulting semantically enriched data can be used for making smart queries that ultimately can support the automation of different HE activities and procedures by using reasoning. The authors in [24] use the HEAPAF framework to automate the higher education activity. As proof of concept, the work provides an ontology-based solution to find the right resource for research collaboration. Our HEAPAF framework consists of the following phases: specification, knowledge acquisition, conceptualization, HEO development, and SPARQL Endpoint (as shown in Figure 1). In the following subsections, we describe each of these phases in detail.

**4.1. Specification.** The purpose of this phase was to identify and specify the data entities for semantic annotation and ontology development. The ultimate goal is to automate the HE activities, improve the educational process, and lower the cost. Moreover, specifying the data items helps to improve interoperability which leads to sharing and raising the opportunities for accessibility and portability. It also makes unstructured data that come, for example, from text documents more interdependent and richer as LOD. The purpose of the ontology is to define a schema that can be used to accommodate HE activities and processes in a machine-understandable format. Also, the ontology can be used by the knowledge engineers and developers' teams at universities.

For the purpose of our research, we specified and identified data entities from different resources and some of these resources are locally developed for accreditation purposes. The local data resources include Accreditation Information Management System (AIMS) and OUDS PLUS. These resources contain all the course data including the course topics, weekly plan, and prerequisites. AIMS also contains the faculty CVs in a well-structured format that provides information about faculty member research interest, expertise, courses taught in the past, and publications.

We collected, identified, extracted, and transformed the data from these resources into a machine-understandable format (i.e., RDF). The data at a later stage are used to perform inferencing and machine reasoning to conduct different tasks automatically. For example, finding and ranking the right resource to teach a course, finding the people with common research interests, and conducting the exams. Figure 1 shows the proposed methodology.

To guarantee the scope of the ontology, we define a list of competency questions [25], which provides some scenarios for the proposed ontology in terms of its applications. These questions play two important roles. First, they determine the expectations that the ontology must fulfill. Secondly, it is used to evaluate applications of ontology by examining the retrieved answers.

Hence, the following competency questions are identified:

- (i) Which course does an academic staff teach based on his/her research interests?
- (ii) Which course does an academic staff teach based on his/her publications?
- (iii) Which course does an academic staff teach based on his/her academic and professional experiences?
- (iv) Which course does an academic staff teach based on his/her certifications and trainings?
- (v) Which academic staff has the most priority to teach a course based on his/her experiences in some research areas/interests?
- (vi) Which academic staff has the most priority to teach a course based on his/her experiences in some publications?
- (vii) Which academic staff has the most priority to teach a course based on his/her experiences in some academic and professional experiences?
- (viii) Which academic staff has the most priority to teach a course based on his/her experiences in some certifications and trainings?

**4.2. Knowledge Acquisition.** The necessary data to create HEO and individuals are revealed by using text analysis techniques such as text extraction. The resources related to the faculty member's CV and courses at the Faculty of Computing and Information Technology (FCIT) are extracted from AIMS in the format of PDF files, Excel sheets, and Doc files. These data resources are studied and analyzed to figure out which data entities can be used in the development of the HEO. These data resources can also be used to support our case study (i.e., "finding and ranking the right resource to teach a course"). The data items that are extracted from the above resources include "research interests," "certifications and trainings," "academic and professional experiences," and "publications" from the CVs of faculty members (as shown in Figure 2(a)). In addition to this, data entities such as "topics," "sub-topics," "weekly data," "course book," and "assessment plan" are extracted from the syllabus of the courses (as shown in Figure 2(b)), for example:

- (1) The academic staff  $\rightarrow$  has research interest  $\rightarrow$  keywords  
Dr. Maram  $\rightarrow$  has research interests  $\rightarrow$  Social Commerce
- (2) Keywords  $\rightarrow$  related to  $\rightarrow$  Course  
Social Commerce  $\rightarrow$  related to  $\rightarrow$  CPIS604



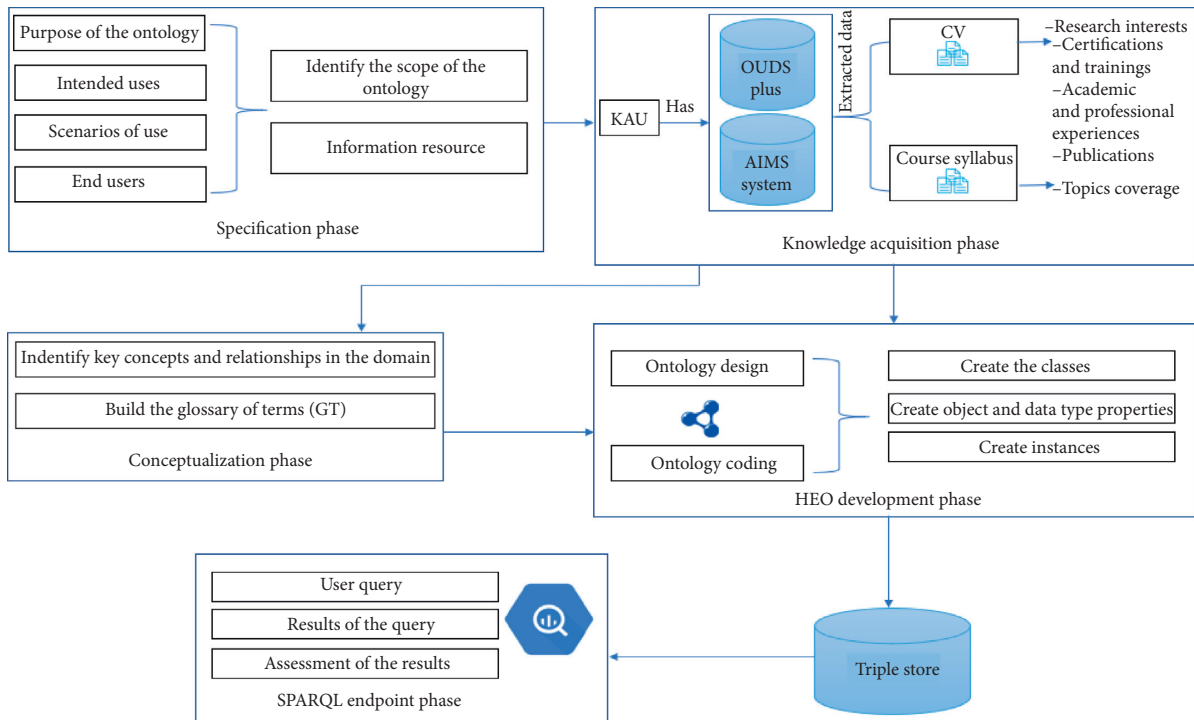


FIGURE 1: The architecture of the Higher Education Activities and Processes Automation Framework (HEAPAF).

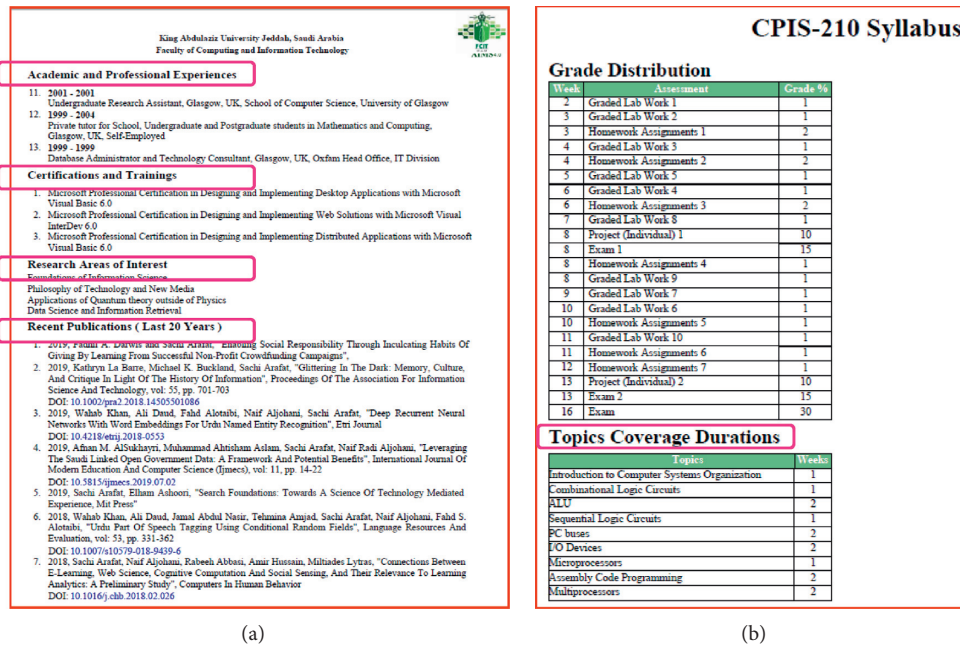


FIGURE 2: Captured criteria that support the activity.

The extracted data can be used at later stages in creating and identifying classes, subclasses, properties, and individuals.

4.3. *Conceptualization.* A conceptual model is designed by identifying the key concepts in the domain. To do that, we analyzed the existing data models and ontologies from

repositories as well as determined the missing classes and properties until we got a formal representation of the model. After that, we built a Glossary of Terms (GT), which includes the classes, object-type properties, data-type properties, and describes their uses and definitions. Here are some of the key concepts used in representing the entities in the university structure.

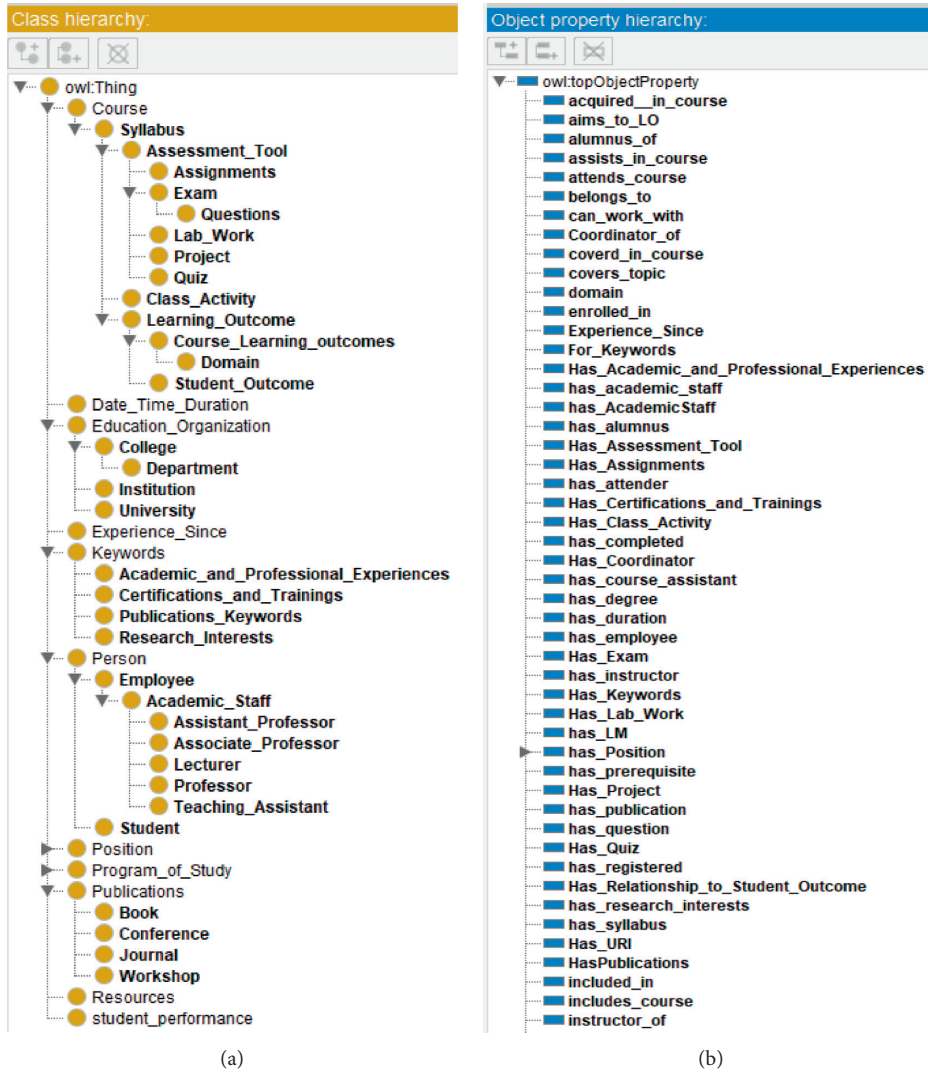
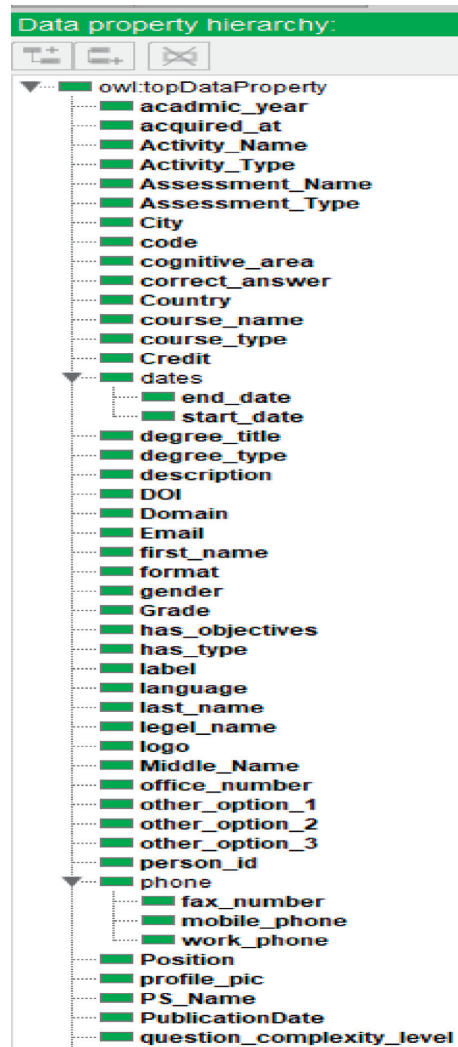


FIGURE 3: Continued.



(c)

FIGURE 3: The HEO classes, object, and data type properties hierarchy.

- (i) The class “*Person*” is a subclass of the main class “*Thing*.” It divides people at the university into two types, the first type is the *academic staff* class, which, in turn, is ranked and coded as *Assistant Professor*, *Associate Professor*, *Lecturer*, *Professor*, *Teaching assistant* classes. and the second type is the *Student* class.
- (ii) The class “*Course*” is a subclass of the main class “*Thing*.” It contains all the information about the courses. Also, it contains “*Syllabus*” as a subclass, which in turn has two subclasses “*Assessment Tool*” and “*Learning Outcome*.” The “*Assessment Tool*” has *Assignments*, *Lab Work*, *projects*, *Quiz*, and *Exams* as subclasses. Furthermore, “*Learning Outcome*” has *Course Learning Outcomes* and *Student Outcome* subclasses as shown in Figure 3(a).
- (iii) The class “*Keywords*” is a subclass of the main class “*Thing*.” It contains all the main and useful words that are extracted from the CVs of the faculty members as subclasses, which are *Academic\_and\_Professional\_Experiences*, *Certifications\_and\_Trainings*, *Publications\_Keywords*, and *Research\_Interests*.
- (iv) The class “*Publications*” is a subclass of the main class “*Thing*.” It contains the person’s work that is published and available on the web. Also, it has *Book*, *Conference*, *Journal*, *Workshop* as subclasses.
- (v) The class “*Experience\_Since*” is a subclass of the main class “*Thing*.” It contains the specific year of experience that is related to the Keywords gained by the academic staff.
- (vi) The object property “*has\_research\_interests*” is a summary of a person’s scientific trends and future directions. It has *Academic\_Staff*, *Assistant\_Professor*, *Lecturer*, *Professor*, *Teaching\_Assistant*, *Associate\_Professor* classes as a domain, and *Keywords*, *Research\_Interests* classes as a range.

- (vii) The object property “*Has\_Certifications\_and\_Trainings*” is the certification and training that was attended and gained by the academic staff. It has *Academic\_Staff*, *Assistant\_Professor*, *Lecturer*, *Professor*, *Teaching\_Assistant*, *Associate\_Professor* classes as a domain, and *Keywords*, *Certifications\_and\_Trainings* classes as a range.
- (viii) The object property “*Has\_Academic\_and\_Professional\_Experiences*” is the academic and professional experience that the academic staff obtained to satisfy one or more areas. It has *Academic\_Staff*, *Assistant\_Professor*, *Lecturer*, *Professor*, *Teaching\_Assistant*, *Associate\_Professor* classes as a domain, and *Keywords*, *Academic\_and\_Professional\_Experiences* classes as a range.
- (ix) The object property “*Has\_Keywords*” are the words that capture the essence of a specific publication. It has *Publications* class as domain, and *Keywords*, *Publications\_Keywords* classes as a range.

**4.4. Higher Education Ontology (HEO) Development.** The design and development of ontology require several stages and standards principles to follow [26]. Our HEO defines elements to describe the university and HE activities and procedures. We developed the ontology based on the map of different standard educational systems such as “Blackboard” and local systems such as “OUDS Plus” and “AIMS.” The ontology contains the creation of the classes, properties, and instances as described follows:

- (i) Create the classes and the class hierarchy: the class describes the concept of the domain and organizing them in a taxonomic hierarchy (class–subclass), as shown in Figure 3(a).
- (ii) Create the properties of the ontology: there are two types of properties, namely, object properties and datatype properties. Object properties describe the relationship between classes, as shown in Figure 3(b). Datatype properties describe the relationships between instances and data values such as string and integer, as shown in Figure 3(c).
- (iii) Create instances of the ontology: OWL allows us to define individuals and to include properties related to them. First, we select the proper class and then create its instances. Also, the instance can belong to many classes.

**4.5. SPARQL Endpoint.** We also established a SPARQL endpoint to query the semantically enriched data that we extracted from different data sources. The HEO and all the HE-related RDF data are dumped into the GraphDB database. Different properties of HEO support the reasoning and inferencing over the RDF data that are structured according to ontology schema. SPARQL queries result into RDF data in the form of RDF triples which contain either literal values or other resources as objects of resulting RDF triples. We prepared plenty of competency questions

(discussed in the next section) and also posed SPARQL-based queries to the SPARQL endpoint to get these questions answered by using machine reasoning. For the purpose of ongoing research, we did not make the datasets and the SPARQL Endpoint publicly available, but they could be provided on demand.

## 5. Case Study: Finding and Ranking the Right Resources to Teach a Course

In the above section, we described our HEAPAF framework that can be used for identifying and extracting data entities from different resources and the development of HEO that can be used for data representation in a machine-understandable format (i.e., RDF). Once the HEO is ready and filled with an individual’s data (i.e., RDF data) and links are established between data items, we can use it to ask smart queries for reasoning-based question-answer purposes. As a proof of concept to our methodology and framework, we present the case study of “finding the right resources to teach a course.” Answering the case study involves the two potential sub-questions (1) finding the right resource to teach a course and (2) defining the criteria for the ranking of faculty members. Here, we describe our methodology to answer these potential questions.

**5.1. Finding the Right Resource to Teach a Course.** Our case study topic refers to the process of determining specific faculty members who are most suitable to teach specific courses. The process of finding and matching the suitability of faculty members to teach a course is determined based on different attributes of faculty members, such as research interests, publications, certifications, and experience, as well as different course attributes such as topics and subtopics. Table 1 shows the most important factors and attributes of faculty members and courses as object and data type properties and logical description and usability of these properties.

The transitive characteristic was applied to *Related\_to* property. Table 2 explains in detail the mechanism of this property.

Here, we describe the use of machine reasoning in finding a resource to teach a course based on different contributing attributes. We also explain here the role of each attribute by describing one relevant example. A short description of these examples is as follows (Tables 3–7):

- (i) Example 1 finding the resource based on the research interests as shown in Table 3
- (ii) Example 2 finding the resource based on the certifications and trainings as shown in Table 4
- (iii) Example 3 finding the resource based on the academic and professional experiences as shown in Table 5
- (iv) Example 4 finding the resource based on the publications as shown in Table 6
- (v) Example 5 finding the resource based on the research interests, certification and trainings, academic and professional experiences, and publications as shown in Table 7

TABLE 1: The properties used with their logical description.

Property name	Logical description
<i>has_research_interests</i>	$\text{Research Interests} \in \text{Keywords}$ $\text{Keywords} \subseteq \text{Course}$ $\text{Course} \subseteq \text{Academic Staff}$ $\text{Keywords} \subseteq \text{Academic Staff}$ $\text{Research Interests} \in \text{Academic Staff}$
<i>Has_Certifications_and_Trainings</i>	$\text{Certifications and Trainings} \in \text{Keywords}$ $\text{Keywords} \subseteq \text{Course}$ $\text{Course} \subseteq \text{Academic Staff}$ $\text{Keywords} \subseteq \text{Academic Staff}$ $\text{Certifications and Trainings} \in \text{Academic Staff}$
<i>Has_Academic_and_Professional_Experiences</i>	$\text{Academic and Professional Experiences} \in \text{Keywords}$ $\text{Keywords} \subseteq \text{Course}$ $\text{Course} \subseteq \text{Academic Staff}$ $\text{Keywords} \subseteq \text{Academic Staff}$ $\text{Academic and Professional Experiences} \in \text{Academic Staff}$
<i>has_publication</i>	$\text{Publication's Keywords} \in \text{Keywords}$ $\text{Keywords} \subseteq \text{Course}$ $\text{Course} \subseteq \text{Academic Staff}$ $\text{Keywords} \subseteq \text{Academic Staff}$ $\text{Publication's Keywords} \in \text{Academic Staff}$
<i>Has_Keywords</i>	$\text{Keywords} \subseteq \text{Course}$ $\text{Course} \subseteq \text{Academic Staff}$ $\text{Keywords} \subseteq \text{Academic Staff}$

TABLE 2: Related\_to property mechanism.

Property name
<i>Related_to</i>
<i>Text description</i>
The contents of the academic and professional experiences, the certifications and trainings, research interests, topics, and the keywords of a publication has a relationship with the course.
<i>Reasoning</i>
If a property $P$ has the characteristic transitive, and the property relates individual $a$ to individual $b$ , also individual $b$ to individual $c$ , then the ontology can infer that individual $a$ is related to individual $c$ via property $P$ . Once we assign the characteristic transitive to the property “ <i>Related_to</i> ,” the ontology can figure out which academic staff can teach which course based on research interests, publication, certifications and trainings, academic and professional experiences, and topics coverage.

TABLE 3: The research interests example.

Example 1
The object property <i>Related_to</i> has the <i>Keywords</i> class as domain and <i>Course</i> class as a range. By adding the individual (Maram Abdulrahman Meccawy) of class <i>Associate_Professor</i> and the individual (Social Commerce) of class as keywords. Also, the individual (CPIS604/CIS_MSc/CIS_D/FCIT/KAU) to the class <i>Course</i> . Then link <i>Dr. Maram</i> by using the property <i>has_research_interests</i> with <i>Social Commerce</i> and link <i>Social Commerce</i> with <i>CPIS604/CIS_MSc/CIS_D/FCIT/KAU</i> by using the property <i>Related_to</i> . In other words, we can say define the following RDF triples: <i>Dr.Maram has_research_interests Social Commerce.</i> <i>Social Commerce Related_to CPIS604/CIS_MSc/CIS_D/FCIT/KAU.</i>
By applying this, no inference will be found but once we assign the characteristic transitive to the property “ <i>Related_to</i> ,” the reasoner logically found that <i>Dr. Maram</i> can teach the course <i>CPIS604/CIS_MSc/CIS_D/FCIT/KAU</i> based on her research interests. Figures 4(a) and 4(b) show the visualization of <i>Related_to</i> and research interest properties.
SPARQL query PREFIX KAU: <http://www.semanticweb.org/nada_/ontologies/2019/11/untitled-ontology-43#> SELECT DISTINCT ?Academic_Staff ?Course ?Research_Interests WHERE { ?Academic_Staff rdf:type KAU:Academic_Staff. ?Research_Interests rdf:type KAU:Research_Interests. ?Course rdf:type KAU:Course. ?Academic_Staff KAU:has_research_interests ?Research_Interests. ?Research_Interests KAU:Related_to ?Course. } Figure 4(c) shows the results of the query

TABLE 4: The certifications and trainings example.

## Example 2

The object property *Related\_to* has the *Keywords* class as domain and the *Course* class as a range. We add the individual (*Muhammad Ahtisham Aslam*) of class *Associate\_Professor*, the individual (*SW Technologies*) of class *Keywords*, and *CPIS640/CIS\_MSc/CIS\_D/FCIT/KAU* to the class *Course*. Then link *Dr. Muhammad* by using the property *Has\_Certifications\_and\_Trainings* with *SW Technologies* and link *SW Technologies* with *CPIS640/CIS\_MSc/CIS\_D/FCIT/KAU* by using the property *Related\_to*. In other words, we can say that we defined the following RDF triples:

*Dr.Muhammad Has\_Certifications\_and\_Trainings SW Technologies.*

*SW Technologies Related\_to CPIS640/CIS\_MSc/CIS\_D/FCIT/KAU.*

By applying this, no inference will be found. But once we assign the characteristic transitive to the property “*Related\_to*,” the reasoner logically found that *Dr. Muhammad* can teach the course *CPIS640/CIS\_MSc/CIS\_D/FCIT/KAU* based on his certifications and trainings. Figures 5(a) and 5(b) show the visualization of *Related\_to* and certifications and trainings’ properties.

SPARQL query

```
SELECT DISTINCT ?Academic_Staff ?Course ?Certifications_and_Trainings
```

```
WHERE {
```

```
?Academic_Staff rdf:type KAU:Academic_Staff.
```

```
?Certifications_and_Trainings rdf:type KAU:Certifications_and_Trainings.
```

```
?Course rdf:type KAU:Course.
```

```
?Academic_Staff KAU:Has_Certifications_and_Trainings ?Certifications_and_Trainings.
```

```
?Certifications_and_Trainings KAU:Related_to ?Course.
```

```
} Figure 5(c) shows the results of the query
```

TABLE 5: The academic and professional experiences example.

## Example 3

The object property *Related\_to* has the *Keywords* class as domain and *Course* class as a range. By adding the individual (*Maram Abdulrahman Meccawy*) of class *Associate\_Professor* and the individual (*Business Analyst*) of class *Keywords*. Also, the individual (*CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU*) to the class *Course*. Then link *Dr.Maram* by using the property *Has\_Academic\_and\_Professional\_Experiences* with *Business Analyst*. And link *business analyst* with *CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU* by using the property *Related\_to*. In other words, we can say define the following:

*Dr.Maram Has\_Academic\_and\_Professional\_Experiences Business Analyst.*

*Business Analyst Related\_to CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU.*

By applying this no inference will be found. But once we assign the characteristic transitive to the property “*Related\_to*,” the reasoner logically found that *Dr.Maram* can teach the course *CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU* based on her academic and professional experiences. Figures 6(a) and 6(b) show the visualization of *Related\_to* and academic and professional experience properties.

SPARQL query

```
SELECT DISTINCT ?Academic_Staff ?Course
```

```
?Academic_and_Professional_Experiences
```

```
WHERE {
```

```
?Academic_Staff rdf:type KAU:Academic_Staff.
```

```
?Academic_and_Professional_Experiences rdf:type KAU:Academic_and_Professional_Experiences.
```

```
KAU:Academic_and_Professional_Experiences.
```

```
?Academic_Staff KAU:Has_Academic_and_Professional_Experiences
```

```
?Academic_and_Professional_Experiences.
```

```
?Academic_and_Professional_Experiences KAU:Related_to ?Course.
```

```
} Figure 6(c) shows the results of the query
```

According to the results of the previous examples, the second activity is defining the criteria for the ranking of the faculty members.

5.2. *Criteria for the the Ranking of Faculty Members.* The query result of finding the right resource to teach a course shows all the academic staff without any ranking process. So, we define criteria matching the different attributes such as *Experience\_Since*, *academic and professional experiences*, *the certifications and trainings*, *research interests*, *publications*, and *topics covered*. It helps in ranking the faculty members’ suitability to teach

a course. Here, we explain the role of reasoning in ranking the faculty member in more detail by using the following potential questions:

- (i) Rank founded resources based on research interests (Table 9)
- (ii) Rank founded resources based on academic and professional experiences (Table 10)
- (iii) Rank founded resources based on certifications and trainings (Table 11)
- (iv) Rank founded resources based on publication (Table 12)

TABLE 6: The publications example.

---

Example 4

The object property *Has\_Keywords* has the *Publications* class as the domain and the *Keywords* class as a range. By adding the individual (Maram Abdulrahman Meccawy) of class *Associate\_Professor* and the individual (A Safety Tracking and Sensoring System for School Buses in Saudi Arabia) of class *Publications*. Also, the individuals (Tracking System, Sensor, Tracking, School Buses, Temperature sensor, GPS, IoT) to the class *Keywords*. Furthermore, link the individual (CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU) to the class *Course*. Then link *Dr. Maram* by using the property *has\_publication* with A Safety Tracking and Sensoring System for School Buses in Saudi Arabia. And link A Safety Tracking and Sensoring System for School Buses in Saudi Arabia with *Tracking System, Sensor, Tracking, School Buses, Temperature sensor, GPS, IoT* by using the property *Has\_Keywords*. As well as link *Tracking System, Sensor, Tracking, School Buses, Temperature sensor, GPS, IoT* with *CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU* by using the property *Related\_to*. In other words, we can say define the following:

*Dr. Maram has\_publication A Safety Tracking and Sensoring System for School Buses in Saudi Arabia.*  
*A Safety Tracking and Sensoring System for School Buses in Saudi Arabia Has\_Keywords Tracking System.*  
*Tracking System Related\_to CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU.*

By applying this, no inference will be found. But once we assign the characteristic transitive to the property “*Related\_to*,” the reasoner logically found that *Dr.Maram* can teach the course *CPIS604/CIS\_MSc/CIS\_D/FCIT/KAU* based on her publications. Figures 7(a) and 7(b) shows the visualization of *Related\_to* and publications properties.

---

SPARQL query

```
SELECT DISTINCT ?Academic_Staff ?Course ?Publications_Keywords ?Publications
WHERE {
?Academic_Staff rdf:type KAU:Academic_Staff.
?Publications rdf:type KAU:Publications.
?Academic_Staff KAU:has_publication ?Publications.
?Keywords rdf:type KAU:Keywords.
?Publications_Keywords rdf:type KAU:Publications_Keywords.
?Course rdf:type KAU:Course.
?Publications_Keywords KAU:Related_to ?Course.
?Publications KAU:Has_Keywords ?Publications_Keywords.} Figure 7(c) shows the results of the query
```

---

TABLE 7: The research interests, certification and trainings, academic and professional experiences, and publications example.

---

Example 5

The example shows the best resource to teach a course based on the research interests, certification and trainings, academic and professional experiences, and publications together in one query.

SPARQL query

```
SELECT DISTINCT ?Academic_Staff ?Course ?Research_Interests ?Certifications_and_Trainings ?Academic_and_Professional_Experiences
?Publications_Keywords
WHERE {
{?Academic_Staff rdf:type KAU:Academic_Staff.
?Research_Interests rdf:type KAU:Research_Interests.
?Course rdf:type KAU:Course.
?Academic_Staff KAU:has_research_interests ?Research_Interests.
?Research_Interests KAU:Related_to ?Course.}
UNION
{?Academic_Staff rdf:type KAU:Academic_Staff.
?Certifications_and_Trainings rdf:type KAU:Certifications_and_Trainings.
?Course rdf:type KAU:Course.
?Academic_Staff KAU:Has_Certifications_and_Trainings ?Certifications_and_Trainings.
?Certifications_and_Trainings KAU:Related_to ?Course.}
UNION
{?Academic_Staff rdf:type KAU:Academic_Staff.
?Academic_and_Professional_Experiences rdf:type KAU:Academic_and_Professional_Experiences.
?Course rdf:type KAU:Course.
?Academic_Staff KAU:Has_Academic_and_Professional_Experiences
?Academic_and_Professional_Experiences.
?Academic_and_Professional_Experiences KAU:Related_to ?Course.}
UNION
{?Academic_Staff rdf:type KAU:Academic_Staff.
?Publications rdf:type KAU:Publications.
?Academic_Staff KAU:has_publication ?Publications.}
```

---

TABLE 7: Continued.

Example 5

```
?Keywords rdf:type KAU:Keywords.
?Publications_Keywords rdf:type KAU:Publications_Keywords.
?Course rdf:type KAU:Course.
?Publications_Keywords KAU:Related_to ?Course.
?Publications KAU:Has_Keywords ?Publications_Keywords.}
} Figure 8 shows the results of the query
```

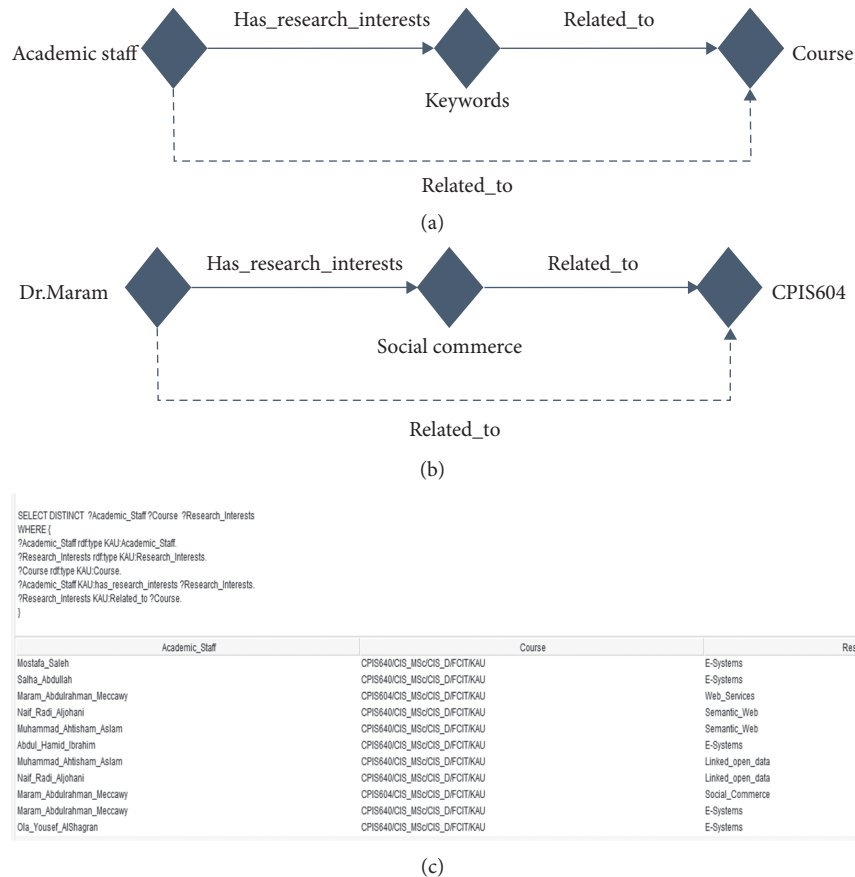


FIGURE 4: The visualization of Related\_to property with research interests and the results of the query.

Tables 8 to 12 describe the relationships of the keywords gained by the academic staff to a specific year, which is extracted from his/her CV. Accordingly, the ranking criteria were set, through which we determined who has the most priority to teach a course. For example, academic staff 1 and academic staff 2 both have Semantic Web (SW) as a research interest. However, academic staff 1 gained the SW in 2011 and academic staff 2 gained the SW in 2009, then academic staff 2 is

more likely suitable to teach the course related to SW. So, the object property *Experience\_Since* is inverse to the object property *For\_Keywords*, which can figure out the year of the research interest that is related to a person (as shown in Figure 9). And the same goes for the *academic and professional experience, certifications and trainings, and publication*. Further details of other object properties that are used in this case study are given below.



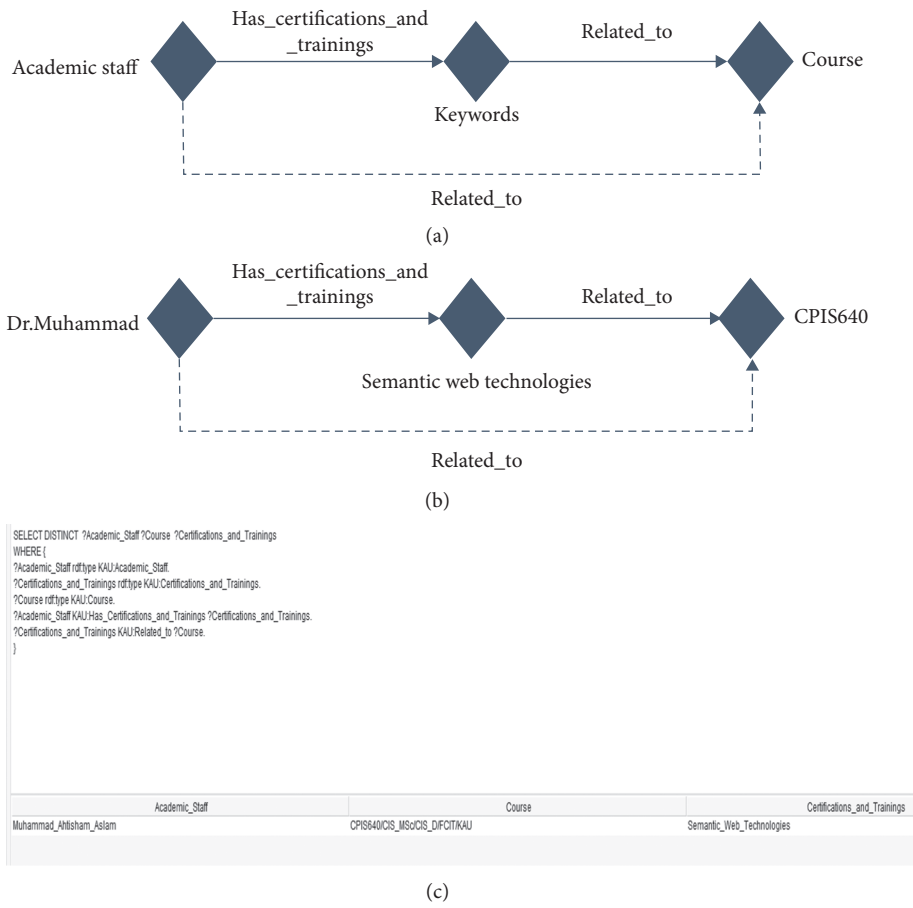


FIGURE 5: The visualization of Related\_to property with certifications and trainings and the results of the query.

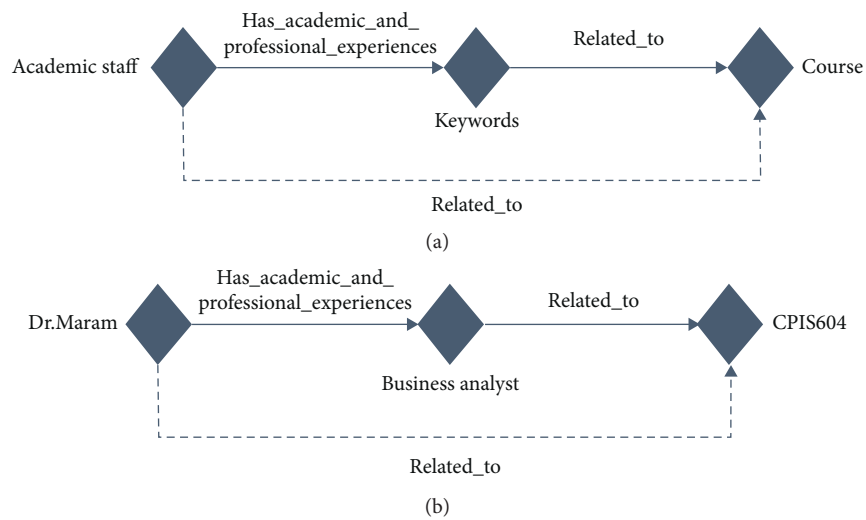
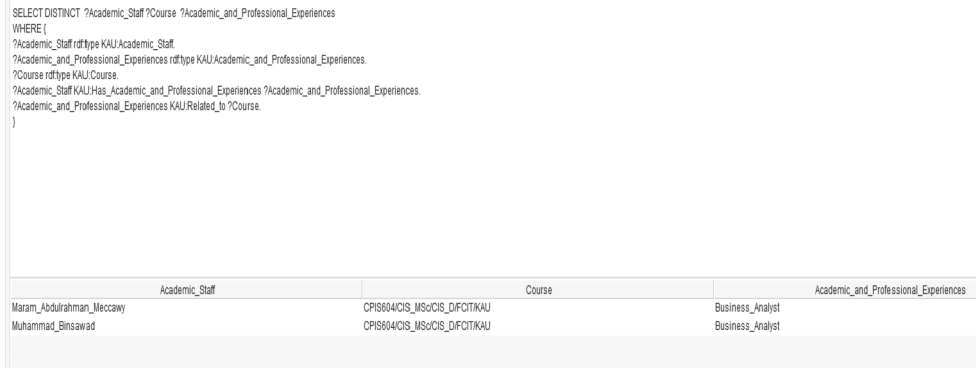
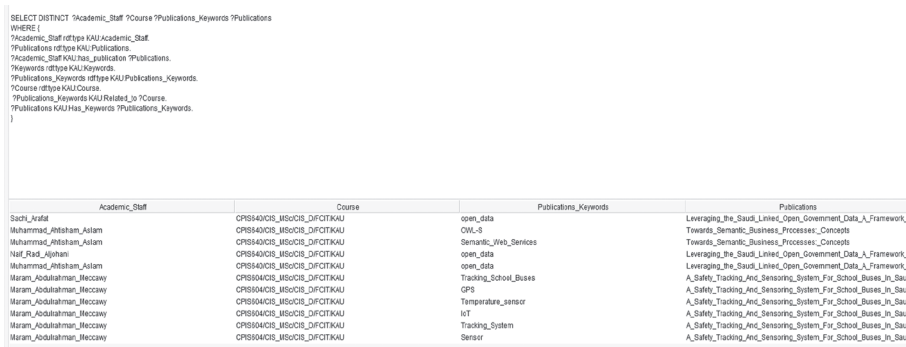
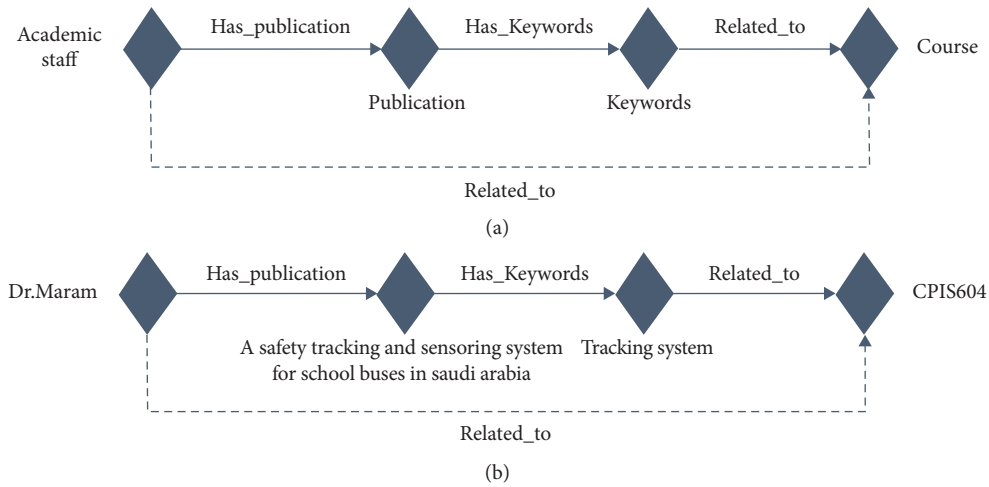


FIGURE 6: Continued.



(c)

FIGURE 6: The visualization of Related\_to property with academic and professional experiences and the results of the query.



(c)

FIGURE 7: The visualization of Related\_to property with publications and the results of the query.

```

SELECT DISTINCT ?Academic_Staff ?Course ?Research_Interests ?Certifications_and_Trainings ?Academic_and_Professional_Experiences ?Publications_Keywords
WHERE {
  {
    ?Academic_Staff rdf:type KAU:Academic_Staff.
    ?Research_Interests rdf:type KAU:Research_Interests.
    ?Course rdf:type KAU:Course.
    ?Academic_Staff KAU:has_research_interests ?Research_Interests.
    ?Research_Interests KAU:Related_to ?Course.
  }
  UNION
  {
    ?Academic_Staff rdf:type KAU:Academic_Staff.
    ?Certifications_and_Trainings rdf:type KAU:Certifications_and_Trainings.
    ?Course rdf:type KAU:Course.
    ?Academic_Staff KAU:Has_Certifications_and_Trainings ?Certifications_and_Trainings.
    ?Certifications_and_Trainings KAU:Related_to ?Course.
  }
  UNION
  {
    ?Academic_Staff rdf:type KAU:Academic_Staff.
    ?Academic_and_Professional_Experiences rdf:type KAU:Academic_and_Professional_Experiences.
    ?Course rdf:type KAU:Course.
    ?Academic_Staff KAU:Has_Academic_and_Professional_Experiences
    ?Academic_and_Professional_Experiences.
  }
}

```

Academic_Staff	Course	Research_Interests	Certifications_and_Trainings	Academic_and_Professional_Experiences	Publications_Keywords
Mostafa_Saleh	CPIS640/CIS_MSoCIS_DIFCITKAU	E-Systems			
Salha_Abdullah	CPIS640/CIS_MSoCIS_DIFCITKAU	E-Systems			
Maram_Abdurrahman_Meccawy	CPIS640/CIS_MSoCIS_DIFCITKAU	Web_Services			
Nalif_Radi_Aljohani	CPIS640/CIS_MSoCIS_DIFCITKAU	Semantic_Web			
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU	Semantic_Web			
Abdul_Hamid_Ibrahim	CPIS640/CIS_MSoCIS_DIFCITKAU	E-Systems			
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU	Linked_open_data			
Nalif_Radi_Aljohani	CPIS640/CIS_MSoCIS_DIFCITKAU	Linked_open_data			
Maram_Abdurrahman_Meccawy	CPIS640/CIS_MSoCIS_DIFCITKAU	Social_Commerce			
Maram_Abdurrahman_Meccawy	CPIS640/CIS_MSoCIS_DIFCITKAU	E-Systems			
Ola_Yousef_AlShagran	CPIS640/CIS_MSoCIS_DIFCITKAU	E-Systems			
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU		Semantic_Web_Technologies		
Maram_Abdurrahman_Meccawy	CPIS640/CIS_MSoCIS_DIFCITKAU			Business_Analyst	
Muhammad_Binsawad	CPIS640/CIS_MSoCIS_DIFCITKAU			Business_Analyst	
Sachi_Arafat	CPIS640/CIS_MSoCIS_DIFCITKAU				open_data
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU				OWL-S
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU				Semantic_Web_Services
Nalif_Radi_Aljohani	CPIS640/CIS_MSoCIS_DIFCITKAU				open_data
Muhammad_Ahltsham_Aslam	CPIS640/CIS_MSoCIS_DIFCITKAU				open_data
Maram_Abdurrahman_Meccawy	CPIS640/CIS_MSoCIS_DIFCITKAU				Tracking_School_Buses

FIGURE 8: The result of the research interests, certification and trainings, academic and professional experiences, and publications query.

TABLE 8: The properties used for the ranking of the faculty members.

Object property name	Domain	Range	Characteristic
<i>Experience_Since</i>	<i>Keywords</i>	<i>Experience_Since</i>	Inverse of
<i>For_Keywords</i>	<i>Experience_Since</i>	<i>Keywords</i>	Inverse of
<i>related_to_person</i>	<i>Experience_Since</i>	<i>Academic_Staff</i>	Symmetric

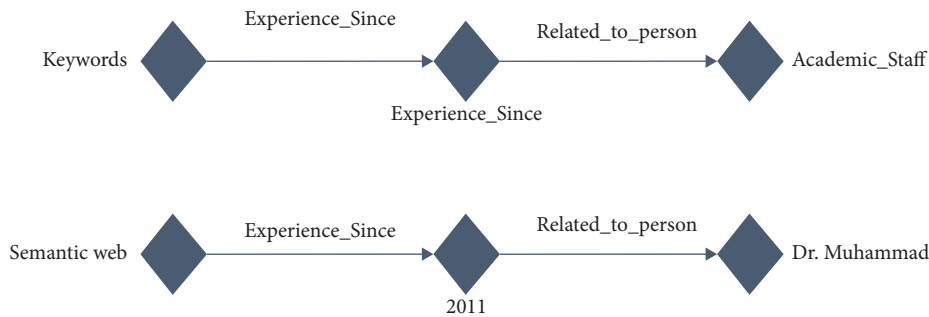


FIGURE 9: The visualization of Experience\_Since and related\_to\_person properties.

TABLE 9: The research interests' example after criteria.

```

SPARQL query related to the research interests
SELECT DISTINCT ?Academic_Staff ?Course
?Experience_Since ?Research_Interests
WHERE {
  ?Academic_Staff rdf:type KAU:Academic_Staff.
  ?Research_Interests rdf:type KAU:Research_Interests.
  ?Course rdf:type KAU:Course.
  ?Academic_Staff KAU:has_research_interests ?Research_Interests.
  ?Research_Interests KAU:Experience_Since ?Experience_Since.
  ?Experience_Since KAU:related_to_person ?Academic_Staff.
  ?Research_Interests KAU:Related_to ?Course.
}
ORDER BY ASC (?Experience_Since)

```

Figure 10(a) shows the results of the query.

TABLE 10: The academic and professional experiences example after criteria.

---

 SPARQL query related to the academic and professional experiences
 

---

```

SELECT DISTINCT ?Academic_Staff ?Course ?Academic_and_Professional_Experiences ?Experience_Since
WHERE {
?Academic_Staff rdf:type KAU:Academic_Staff.
?Academic_and_Professional_Experiences rdf:type KAU:Academic_and_Professional_Experiences.
?Course rdf:type KAU:Course.
?Academic_Staff KAU:Has_Academic_and_Professional_Experiences ?Academic_and_Professional_Experiences.
?Academic_and_Professional_Experiences KAU:Related_to ?Course.
?Academic_and_Professional_Experiences KAU:Experience_Since ?Experience_Since.
?Experience_Since KAU:related_to_person ?Academic_Staff.
}
ORDER BY ASC (?Experience_Since)

```

---

 Figure 10(b) shows the results of the query.
 

---

TABLE 11: The certifications and trainings example after criteria.

---

 SPARQL query related to the certifications and trainings
 

---

```

SELECT DISTINCT ?Academic_Staff ?Course
?Certifications_and_Trainings ?Experience_Since
WHERE {
?Academic_Staff rdf:type KAU:Academic_Staff.
?Certifications_and_Trainings rdf:type KAU:Certifications_and_Trainings.
?Course rdf:type KAU:Course.
?Academic_Staff KAU:Has_Certifications_and_Trainings ?Certifications_and_Trainings.
?Certifications_and_Trainings KAU:Related_to ?Course.
?Certifications_and_Trainings KAU:Experience_Since ?Experience_Since.
?Experience_Since KAU:related_to_person ?Academic_Staff.
}
ORDER BY ASC (?Experience_Since)

```

---

 Figure 10(c) shows the results of the query.
 

---

TABLE 12: The publications example after criteria.

---

 SPARQL query related to the publications
 

---

```

SELECT DISTINCT ?Academic_Staff ?Course
?Publications_Keywords ?Publications ?Experience_Since
WHERE {
?Academic_Staff rdf:type KAU:Academic_Staff.
?Publications rdf:type KAU:Publications.
?Academic_Staff KAU:has_publication ?Publications.
?Experience_Since rdf:type KAU:Experience_Since.
?Keywords rdf:type KAU:Keywords.
?Publications_Keywords rdf:type KAU:Publications_Keywords.
?Course rdf:type KAU:Course.
?Publications_Keywords KAU:Related_to ?Course.
?Publications KAU:Has_Keywords ?Publications_Keywords.
?Publications_Keywords KAU:Experience_Since ?Experience_Since.
?Experience_Since KAU:related_to_person ?Academic_Staff.
}
ORDER BY ASC (?Experience_Since)

```

---

 Figure 10(d) shows the results of the query.
 

---

## 6. Evaluation and Analysis

The evaluation process examines technically the features, usability, and utility of the framework, HEO, reasoning mechanism that we defined in the ontology, and datasets that

we produced in this work. We used the iterative approach in our evaluation process [27] in which evaluation of activities, processes, output, datasets, and reasoning mechanism is implemented through all the phases of the framework life cycle. The goal is to discover wrong, incomplete, or missed

```

SELECT DISTINCT ?Academic_Staff ?Course ?Experience_Since ?Research_Interests
WHERE {
  ?Academic_Staff rdf:type KAUAcademic_Staff.
  ?Research_Interests rdf:type KAUResearch_Interests.
  ?Course rdf:type KAUCourse.
  ?Academic_Staff KAUHas_Research_Interests ?Research_Interests.
  ?Research_Interests KAUExperience_Since ?Experience_Since.
  ?Experience_Since KAUrelated_to_person ?Academic_Staff.
  ?Research_Interests KAURelated_to ?Course.
}
ORDER BY ASC(?Experience_Since)
    
```

Academic_Staff	Course	Experience_Since	Research_Interests
Abdul_Hamid_Ibrahim	CPIS640CIS_MSOCIS_DF0ITKAU	1985	E-Systems
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	2009	E-Systems
Mostafa_Saah	CPIS640CIS_MSOCIS_DF0ITKAU	2011	E-Systems
Saaha_Abdullah	CPIS640CIS_MSOCIS_DF0ITKAU	2011	E-Systems
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	2011	Semantic_Web
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	2011	Linked_open_data
Ola_Youssef_Ahlagrag	CPIS640CIS_MSOCIS_DF0ITKAU	2016	E-Systems
Nat_Raad_Alghani	CPIS640CIS_MSOCIS_DF0ITKAU	2016	Linked_open_data
Nat_Raad_Alghani	CPIS640CIS_MSOCIS_DF0ITKAU	2016	Semantic_Web
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	2017	Web_Services
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	2017	Social_Commerce

(a)

```

SELECT DISTINCT ?Academic_Staff ?Course ?Academic_and_Professional_Experiences ?Experience_Since
WHERE {
  ?Academic_Staff rdf:type KAUAcademic_Staff.
  ?Academic_and_Professional_Experiences rdf:type KAUAcademic_and_Professional_Experiences.
  ?Course rdf:type KAUCourse.
  ?Academic_Staff KAUHas_Academic_and_Professional_Experiences ?Academic_and_Professional_Experiences.
  ?Academic_and_Professional_Experiences KAURelated_to ?Course.
  ?Academic_and_Professional_Experiences KAUExperience_Since ?Experience_Since.
  ?Experience_Since KAUrelated_to_person ?Academic_Staff.
}
ORDER BY ASC(?Experience_Since)
    
```

Academic_Staff	Course	Academic_and_Professional_Experiences	Experience_Since
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	Business_Analyst	2009
Muhammad_Binsawad	CPIS640CIS_MSOCIS_DF0ITKAU	Business_Analyst	2013

(b)

```

SELECT DISTINCT ?Academic_Staff ?Course ?Certifications_and_Trainings ?Experience_Since
WHERE {
  ?Academic_Staff rdf:type KAUAcademic_Staff.
  ?Certifications_and_Trainings rdf:type KAUCertifications_and_Trainings.
  ?Course rdf:type KAUCourse.
  ?Academic_Staff KAUHas_Certifications_and_Trainings ?Certifications_and_Trainings.
  ?Certifications_and_Trainings KAURelated_to ?Course.
  ?Certifications_and_Trainings KAUExperience_Since ?Experience_Since.
  ?Experience_Since KAUrelated_to_person ?Academic_Staff.
}
ORDER BY ASC(?Experience_Since)
    
```

Academic_Staff	Course	Certifications_and_Trainings	Experience_Since
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	Semantic_Web_Technologies	2011

(c)

```

SELECT DISTINCT ?Academic_Staff ?Course ?Publications_Keywords ?Publications ?Experience_Since
WHERE {
  ?Academic_Staff rdf:type KAUAcademic_Staff.
  ?Publications rdf:type KAUPublications.
  ?Academic_Staff KAUHas_Publication ?Publications.
  ?Experience_Since rdf:type KAUExperience_Since.
  ?Keywords rdf:type KAUKeywords.
  ?Publications_Keywords rdf:type KAUPublications_Keywords.
  ?Course rdf:type KAUCourse.
  ?Publications_Keywords KAURelated_to ?Course.
  ?Publications KAUHas_Keywords ?Publications_Keywords.
  ?Publications_Keywords KAUExperience_Since ?Experience_Since.
  ?Experience_Since KAUrelated_to_person ?Academic_Staff.
}
ORDER BY ASC(?Experience_Since)
    
```

Academic_Staff	Course	Publications_Keywords	Publications	Experience_Since
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	OWL-D	Towards_Semantic_Business_Processes_Concepts	2008
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	Semantic_Web_Services	Towards_Semantic_Business_Processes_Concepts	2008
Saaha_Abdullah	CPIS640CIS_MSOCIS_DF0ITKAU	open_data	Leveraging_the_Semantic_Linked_Open_Government_Data_2019	2019
Nat_Raad_Alghani	CPIS640CIS_MSOCIS_DF0ITKAU	open_data	Leveraging_the_Semantic_Linked_Open_Government_Data_2019	2019
Muhammad_Ahisham_Aslam	CPIS640CIS_MSOCIS_DF0ITKAU	open_data	Leveraging_the_Semantic_Linked_Open_Government_Data_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	Tracing_School_Buses	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	GPS	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	Temperature_sensor	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	IoT	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	Tracing_System	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019
Maram_Abdurrahman_Meccawy	CPIS640CIS_MSOCIS_DF0ITKAU	Sensor	A_Safe_Tracing_And_Sensoring_System_For_School_B_2019	2019

(d)

FIGURE 10: The result of the research interests, certification and trainings, academic and professional experiences, and publications query after criteria. (a) The result of research interests query after criteria. (b) The result of academic and professional experiences query after criteria. (c) The result of certifications and trainings query after criteria. (d) The result of publications query after criteria.

definitions as soon as possible. The evaluation points include a check of the ontology structure, the syntax of definitions, and the content in the definitions, logic of reasoning, and validity of extracted RDF datasets. At the final stage, we followed two different approaches to verify and validate the ontology. These approaches are described below.

6.1. *Ontology Consistency and Validity.* The consistency and validity of ontology are verified by using the Pallet reasoner during the different stages of ontology design and development. Figure 11 shows a sample snapshot of ontology consistency when a consistency check is performed on the structure, subsumptions, and classes and properties’ design of ontology.

```

INFO 21:11:18
INFO 21:11:22 ----- Running Reasoner -----
INFO 21:11:22 Pre-computing inferences:
INFO 21:11:22   - class hierarchy
INFO 21:11:22   - object property hierarchy
INFO 21:11:22   - data property hierarchy
INFO 21:11:22   - class assertions
INFO 21:11:22   - object property assertions
INFO 21:11:22   - same individuals
INFO 21:11:23 Ontologies processed in 411 ms by Pellet
INFO 21:11:23
INFO 21:11:23 REASONER CHANGED

```

FIGURE 11: Results of consistency and validity check performed by using Pellet Reasoner.

	Academic_Staff *	Course *	Experience_Since *	Research_Interests
1	:Abdul_Hamid_Ibrahim	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:1985	:E-Systems
2	:Maram_Abdulrahman_Meccawy	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2009	:E-Systems
3	:Mostafa_Saleh	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2011	:E-Systems
4	:Muhammad_Ahtisham_Aslam	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2011	:Linked_open_data
5	:Muhammad_Ahtisham_Aslam	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2011	:Semantic_Web
6	:Salha_Abdullah	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2011	:E-Systems
7	:Naif_Radi_Aljohani	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2016	:Linked_open_data
8	:Naif_Radi_Aljohani	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2016	:Semantic_Web
9	:Ola_Yousef_Alshagran	:CPIS640/CIS_MSc/CIS_D/FCIT/KAU	:2016	:E-Systems
10	:Maram_Abdulrahman_Meccawy	:CPIS604/CIS_MSc/CIS_D/FCIT/KAU	:2017	:Social_Commerce
11	:Maram_Abdulrahman_Meccawy	:CPIS604/CIS_MSc/CIS_D/FCIT/KAU	:2017	:Web_Services

FIGURE 12: The query result of research interests after criteria.

	Academic_Staff *	Course *	Academic_and_Professional_Experiences *	Experience_Since
1	:Maram_Abdulrahman_Meccawy	:CPIS604/CIS_MSc/CIS_D/FCIT/KAU	:Business_Analyst	:2009
2	:Muhammad_Binsawad	:CPIS604/CIS_MSc/CIS_D/FCIT/KAU	:Business_Analyst	:2013

FIGURE 13: The query result of academic and professional experiences after criteria.

6.2. *Answering the Competency Questions.* After the technical validation has been completed, the ontology is evaluated against a set of competency questions that we have defined and determined in the specification phase. This test is done by using the SPARQL query [28]. The output of the query is shown as triplets such as subject, object, and predicate. Two different environments have been used for executing the queries such as the SPARQL query tab in Protégé and the SPARQL tab in GraphDB, where the ontology has been uploaded and is available as an endpoint for public queries.

The previous tables in Section 5 show the result of the queries in the protégé editor. Both activities of finding and ranking the right resource to teach a course are examined by answering the competency questions. After comparing the results that we found from the RDF datasets, we found that the results of the queries were accurate and correct when verified against the actual data.

Figures 12 and 13 describe the same queries that were executed in Tables 9 and 10, but in the platform of GraphDB.

- (1) SPARQL query for the research interests after criteria
- (2) SPARQL query for the academic and professional experiences after criteria

Since the results above are the same as the results provided by the protégé editor in Section 5, it proves that the queries can work in different environments.

## 7. Conclusion and Future Work

Challenges in smart academic environments can be better addressed by using machine reasoning and smart data analytic techniques. It can also help to automate different Higher Education (HE) activities and processes in a smart way. As an example, the course syllabus and CVs of the faculty members provide important data about the acquired skills of the academic staff and the skills required to teach a particular course. These data can be used to automatically find the best resource person to teach a course. In this paper, we presented a framework that can be used to automate different activities and processes in HE by making use of machine reasoning and data analytics techniques. We also presented the design and implementation of educational ontology. As a proof of concept, we presented a case study on “analyzing, finding and ranking the right resource to teach a course.” In our case study, we answered the two main questions, i.e., (1) which faculty member is suitable to teach which course and (2) the criteria for ranking the faculty

members, by applying machine reasoning on semantically enriched data. Finally, we also presented an evaluation of our approach by answering the potential questions in HE. As future work, we plan to enhance our framework and related ontology so that it can accommodate maximum HE activities and processes. We are also working on linking our datasets with open datasets of scientific publications so that research-related tasks and activities can be automated to the maximum extent possible.

## Data Availability

The data that support the findings of this study are available on request from the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## Acknowledgments

The authors thank the Accreditation and Information Management System (AIMS) team at King Abdulaziz University for providing access to the real-life source data to complete this research work.

## References

- [1] M. d'Aquin, "On the use of linked open data in education: current and future practices," in *Open Data for Education: Linked, Shared, and Reusable Data for Teaching and Learning*, vol. 9500, pp. 3–15, Springer, Cham, Switzerland, 2016.
- [2] D. Octavianib and M. S. Othmana, "Ontology reasoning using SPARQL query: a case study of e-learning usage," *JT Jurnal Teknologis*, vol. 78, no. 8-2, 9547 pages, 2016.
- [3] E. Katis, H. Kondylakis, G. Agathangelos, and K. Vassilakis, "Developing an ontology for curriculum and syllabus," in *The Semantic Web: ESWC 2018 Satellite Events*, vol. 11155, pp. 55–59, Springer, Cham, Switzerland, 2018.
- [4] N. Malviya, N. Mishra, and S. Sahu, "Developing university ontology using protégé OWL tool: process and reasoning," *International Journal of Scientific and Engineering Research*, vol. 2, no. 9, 2011.
- [5] Y. Ma, B. Xu, Y. Bai, and Z. Li, "Building linked open university data -Tsinghua University open data as a showcase," in *Proceedings of the Semantic Web Joint International Semantic Technology Conference (IIST)*, vol. 7185, pp. 385–393, Springer, Hangzhou, China, December 2011.
- [6] T. Heath and C. Bizer, "Linked data: evolving the web into a global data space," in *Synthesis Lectures on the Semantic Web: Theory and Technology*, Morgan and Claypool, California, CA, USA, 1st edition, 2011.
- [7] F. Zablith, "Towards a linked and reusable conceptual layer around higher education programs," in *Open Data for Education: Linked, Shared, and Reusable Data for Teaching and Learning*, vol. 9500, pp. 86–102, Springer, Cham, Switzerland, 2016.
- [8] C. I. Bucur, "Linkflows: enabling a web of linked semantic publishing workflows," in *The Semantic Web: ESWC 2018 Satellite Events*, vol. 11155, pp. 262–271, Springer, Cham, Switzerland, 2018.
- [9] S. Dietze, S. S. Alonso, H. Ebner et al., "Interlinking educational resources and the web of data: a survey of challenges and approaches," in *Program: Electronic Library and Information Systems* vol. 47, no. 1, pp. 60–91, 2013.
- [10] D. Venkataraman and K. C. Haritha, "Knowledge representation of university examination system ontology for semantic web," in *Proceedings of the 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–4, IEEE, Coimbatore, India, January 2017.
- [11] D. Abdulazeez, R. M. Salah, and R. Salah, "Developing an ontology for retrieving massive open online courses (moocs) information in coursera platform," *The Journal of The University of Duhok*, vol. 23, no. 1, pp. 103–114, 2020.
- [12] A. Ameen, K. R. Khan, and B. P. Rani, "Construction of university ontology," in *Proceedings of the World Congress on Information and Communication Technologies*, pp. 39–44, IEEE, Trivandrum, India, November 2012.
- [13] H. N. Abed, A. Y. C. Tang, and Z. C. Cob, "An ontology-based search engine for postgraduate students information at the ministry of higher education portal of Iraq," in *Proceedings of the 13th International Conference on Intelligent Systems Design and Applications*, pp. 69–73, IEEE, Salangor, Malaysia, December 2013.
- [14] S. K. Malik, N. Prakash, and S. A. M. Rizvi, "Developing an university ontology in education domain using Protégé for semantic web," *International Journal of Engineering Science and Technology*, vol. 2, no. 9, pp. 4673–4681, 2010.
- [15] C. V. S. Satyamurty, J. V. R. Murthy, and M. Raghava, "Developing higher education ontology using Protégé tool: reasoning," in *Smart Computing and Informatics*, vol. vol. 77, pp. 233–241, Springer, Singapore, 2018.
- [16] M. A. Ullah and S. A. Hossain, "Ontology-based information retrieval system for university: methods and reasoning," in *Emerging Technologies in Data Mining and Information Security*, vol. 814, pp. 119–128, Springer, Singapore, 2019.
- [17] A. M'Baya, J. Laval, N. Moalla, Y. Ouzrout, and A. Bouras, "Ontology based system to guide internship assignment process," in *Proceedings of the 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 589–596, IEEE, Naples, Italy, December 2016.
- [18] K. Jacksi, "Design and implementation of e-campus ontology with a hybrid software engineering methodology," *Science Journal of University of Zakho*, vol. 7, no. 3, pp. 95–100, 2019.
- [19] L. Zeng, T. Zhu, and X. Ding, "Study on construction of university course ontology: content, method and process," in *Proceedings of the International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, IEEE, Wuhan, China, December 2009.
- [20] K. Hadjar, "University ontology: a case study at Ahlia University," in *Semantic Web: Implications for Technologies and Business Practices*, pp. 173–183, Springer, Cham, Switzerland, 2016.
- [21] R. Gil, A. M. Borges, L. Contreras, and M. J. Martin-Bautista, "Improving ontologies through ontology learning: a university case," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, pp. 558–563, IEEE, Los Angeles, CA, USA, April 2009.
- [22] D. Mouromtsev, F. Kozlov, O. Parkhimovich, and M. Zelenina, "Development of an ontology-based e-learning system," in *Knowledge Engineering and the Semantic Web*, vol. 394, pp. 273–280, Springer, Berlin, Germany, 2013.
- [23] W. Lv, Z. Liao, S. Liu, and Y. Zhang, "MEIM: a multi-source software knowledge entity extraction integration model,"

- Computers, Materials and Continua*, vol. 66, no. 1, pp. 1027–1042, 2020.
- [24] N. A. Alrehaili, M. A. Aslam, and D. A. Alrehaili, “*Find a Research Collaborator: An Ontology-Based Solution to Find the Right Resource for Research Collaboration*”, under revision, 2021.
  - [25] M. Grüninger and M. Fox, “Methodology for the design and evaluation of ontologies,” in *Proceedings of the IJCAI’95, Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, August 1995.
  - [26] M. Fernandez-Lopez, A. Gomez-Perez, and N. Juristo, “METHONTOLOGY: from ontological art towards ontological engineering,” in *Proceedings of the AAAI97 Spring Symposium*, pp. 33–40, Palo Alto, CA, USA, March 1997.
  - [27] A. Gomez-Perez, “Some ideas and examples to evaluate ontologies,” in *Proceedings the 11th Conference on Artificial Intelligence for Applications*, pp. 299–305, IEEE, Los Angeles, CA, USA, February 1995.
  - [28] L. Yu, “*A Developer’s Guide to the Semantic Web*”, Springer, Berlin, Germany, 2 edition, 2014.



## Research Article

# Accelerating Spark-Based Applications with MPI and OpenACC

Saeed Alshahrani <sup>1</sup>, Waleed Al Shehri <sup>2</sup>, Jameel Almalki <sup>2</sup>, Ahmed M. Alghamdi <sup>3</sup>,  
and Abdullah M. Alammari <sup>4</sup>

<sup>1</sup>Department of Information Systems, College Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia

<sup>2</sup>Department of Computer Science, College of Computer in Al-Leith, Umm Al-Qura University, Makkah, Saudi Arabia

<sup>3</sup>Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

<sup>4</sup>Curriculum and Teaching Methods Department, Faculty of Education, Umm Al-Qura University, Makkah, Saudi Arabia

Correspondence should be addressed to Waleed Al Shehri; [washehri@uqu.edu.sa](mailto:washehri@uqu.edu.sa)

Received 25 March 2021; Accepted 10 July 2021; Published 21 July 2021

Academic Editor: Adil Mehmood Khan

Copyright © 2021 Saeed Alshahrani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The amount of data produced in scientific and commercial fields is growing dramatically. Correspondingly, big data technologies, such as Hadoop and Spark, have emerged to tackle the challenges of collecting, processing, and storing such large-scale data. Unfortunately, big data applications usually have performance issues and do not fully exploit a hardware infrastructure. One reason is that applications are developed using high-level programming languages that do not provide low-level system control in terms of performance of highly parallel programming models like message passing interface (MPI). Moreover, big data is considered a barrier of parallel programming models or accelerators (e.g., CUDA and OpenCL). Therefore, the aim of this study is to investigate how the performance of big data applications can be enhanced without sacrificing the power consumption of a hardware infrastructure. A Hybrid Spark MPI OpenACC (HSMO) system is proposed for integrating Spark as a big data programming model, with MPI and OpenACC as parallel programming models. Such integration brings together the advantages of each programming model and provides greater effectiveness. To enhance performance without sacrificing power consumption, the integration approach needs to exploit the hardware infrastructure in an intelligent manner. For achieving this performance enhancement, a mapping technique is proposed that is built based on the application's virtual topology as well as the physical topology of the undelaying resources. To the best of our knowledge, there is no existing method in big data applications related to utilizing graphics processing units (GPUs), which are now an essential part of high-performance computing (HPC) as a powerful resource for fast computation.

## 1. Introduction

Recently, there has been drastic growth in the volume of data generated by many different scientific fields for all big data characteristics. Big data can be characterized using different Vs: variety, volume, velocity, veracity, and value [1]. Many applications and systems in the real world depend on collecting, storing, and analyzing such large-scale data, and this trend is predicted to overgrow [2]. Big data environments, such as Hadoop, MapReduce, and Spark, have emerged for supporting parallelism in processing huge dataset volumes [3]. With a future path leading toward Exascale computing with increasing the number of high-performance CPUs and

GPUs in each compute node [4], resource and job management techniques can be considered a keystone of any scalable computing system to support a high degree of parallelism, control system processes, and impact overall performance and system efficiency [5].

Currently, Apache Spark is the largest and perhaps the most comprehensive open-source framework dealing with big data [6]. However, due to its use of the Java Virtual Machine (JVM), its performance is not as desired, and there is still much room for improvement. A high-level implementation of Spark is unaware of the underlying hardware and how to best utilize it, focusing more on the optimization of the virtual machine on which it operates.

Existing High Performance Computing (HPC) tools, such as MPI-based tools, are not designed to manage big data applications [7]. This is a missed opportunity to use the rich set of resources developed over time for HPC that are intended to deliver high performance. There is no existing method in big data applications using GPUs, which are now an essential part of HPC, as a powerful resource for fast computation. Traditionally, GPUs have been utilized to solve complex algorithms, but they can be an effective tool in big data applications and help solve the convergence challenge.

In our previous research [8], we surveyed HPC systems that should support big data for high performance and resource utilization. The studied factors include load balancing and data locality, job scheduling strategies, topology-awareness, and data decomposition. It was found that parallel programming models are dataset specific and it lacks generality. While much of the research problems regarding HPC in big data have been elaborated, complete solutions are still lacking.

In this paper, our proposed solution builds upon three systems: Spark, MPI, and OpenACC. The basic idea is to use a best-of-both-worlds approach: take the big data management offered by Spark, integrate it with the optimal computational processing offered by MPI, and further accelerate the GPUs process by using OpenACC. However, simply integrating the three systems alone is not enough. Spark is a high-level framework implemented in Java and uses the JVM for execution, whereas MPI and OpenACC require low-level programming languages and libraries, such as C, C++, or Fortran. Our proposed approach requires the establishment of a bridging process.

Such bridging alone is also not enough; however, a complete resource management mechanism needs to be established, which defines all modules within the new system and any potential optimization that can be achieved to enhance the system further. To test and verify the effectiveness of the new system, a benchmark was established that, for defined parameters, it must outperform Spark. A case study was chosen with challenging big data problems, for which both Spark and the new system and its variants are tested under strenuous circumstances. The results of the test case provide in-depth information on the effectiveness of the proposed approach.

Our paper has been structured in the following manner: Section 2 will highlight various state-of-the-art techniques used for big data applications. Our proposed hybrid system is explained in Section 3, and its software architecture and modules are discussed in Section 4. In Section 5, we present the system implementation and its evaluation based on the results of a case study along with a discussion of those results. Finally, our conclusions and future work will be discussed in Section 8.

## 2. Background and Related Work

A computing system that consists of multiple nodes of multiple processors as a part of one unit is known as high-performance computing (HPC). These machines are capable of rapidly processing large amounts of data or information

by using several clusters of processing units within a single resource unit. The core methodology used within the framework of these fast processing units is parallel computing [3]. Briefly, HPC is fabulous for its fast-processing capacity to analyze or process data and information.

In today's modern world, technologies like big data processing, artificial intelligence (AI), and the Internet of Things (IoT) are evolving and changing the spectrum of modern life. Moreover, in these times, data are considered vital for organizations, and it is expanding exceptionally. Technologies like HPC are in demand to support information processing in real time and achieve desired goals and targets [9].

To develop the HPC architecture, a network of servers is created to form clusters so that the collection of computers can work together as a single unit and process the data as fast as possible. In some cases, such as financial markets and stock exchanges, data processing needs to occur in real-time. Different algorithms and software programs are used concurrently within the architecture of the clusters. These clusters are networked together with an output unit where all the information is collected or stored [10]. There is no denying the fact that these HPC clusters seamlessly complete diverse tasks.

Within the HPC clusters, it is vital to synchronize each component's pace with other servers to get the maximum performance from the machine. For instance, the computing servers' processed data should be transferred into the storage components at the same pace as it is processed. If the storage component is not storing the data or information as fast as the clusters have processed it, then its HPC infrastructure would not be able to perform as fast as it should. Therefore, all the components within the network must perform at the same pace.

Typically, there are hundreds of computing servers in HPC clusters that are connected to form a network. Each server within the network is known as a node. These nodes within the network perform parallel computing to enhance the processing speed to deliver the output in minimum time.

In the following subsections, several state-of-the-art mechanisms and techniques used for big data applications have been highlighted. Although the topic itself is vast, it attempts to cover many techniques and aspects to support big data applications.

*2.1. Topology Awareness.* The expanding complexity of computing platforms fueled the necessity of developers and programmers understanding hardware organization and adjusting their applications accordingly [11]. As a component of the overall optimization process, there is a great need to visualize a hardware platform model. Hardware locality (HWLOC) is the most famous programming tool used for revealing a consistent perspective on central processing unit (CPU) and memory topology. The paper published in [12] demonstrates how HWLOC accesses these computing assets by joining the I/O structure and providing ways to handle different hubs.

More recently, a particular type of application has been targeted by some work, such as MapReduce-based

applications. Topology-Aware Resource Adaptation (TARA) [13] uses an application description for resource allocation purposes. However, the work is designed for a particular set of applications and is not intended to address other hardware details.

Most open-source and proprietary resource and job management systems (RJMS) consider this kind of feature by using their underlying infrastructure topology characteristics. However, they have failed to consider the importance of application behavior while allocating resources. An approach named HTCondor, formerly Condor [14], is proposed to take advantage of the matchmaking method, which allows for matching the applications' requirements with hardware resource availability. However, this matchmaking method does not consider application behavior, and HTCondor applies to both connected workstations and clusters. In [15], an open-source topology-aware hierarchical unstructured mesh partitioning and load-balancing tool is proposed and named as TreePart tool. The framework can detect and build a hierarchical MPI topology for the underlying hardware at runtime. This topology information can help in partitioning load-balancing between shared and distributed parallel algorithms intelligently.

*2.2. Load Balancing and Data Locality.* Load balancing techniques such as work stealing are used for distributed task scheduling systems, where tasks are transferred from heavily loaded schedulers and assigned to the idle ones. However, the work-stealing technique could lead to poor data locality in data-intensive applications because tasks' execution depends on extensive data processing, resulting in data-transferring overhead.

The research in [16] proposed two methods for distributing and optimizing load balancing for processing big data. The proposed method aims to reduce idle node time, which minimizes task execution time. Results show a significant decrease in processing time. One attempt to enhance the work stealing is by organizing dedicated and shared queues [17] based on their location and data size. Computing multiple tasks could be distributed via techniques such as the MATRIX task scheduler. The Distributed Key-Value Store (DKVS) organizes the metadata and other data locality and task dependency elements. Research showed that the data-aware work-stealing technique exhibits adequate performance.

A fast and light-weight task execution framework (FALKON) [18] is a centralized task scheduler for supporting many-task computing (MTC) applications. FALKON adopts a data diffusion approach for scheduling data-intensive workloads via DKVS [17]. However, FALKON suffers from uncertain task execution times resulting in poor load balancing due to its hierarchical implementation. It is also prone to the issue of scaling in Petascale systems. Data diffusion acquires computing resources and storage dynamically. The process replicates data based on demand to schedule computations close to the data. In contrast to the distributed key-value store [17], FALKON suffers from poor scalability due to deploying a centralized index server for storing metadata.

Mesos platform shares resources among many cluster computing frameworks for scheduling tasks [19]. Therefore, it allows frameworks to achieve data locality via reading the data stored on each machine. When delay scheduling systems are deployed, it waits for a limited time to get data storing nodes. Therefore, the approach generates a significant waiting time for any task that is scheduled on time, especially for larger datasets.

The flexible and dynamic framework of Quincy [20] is a distributed concurrent job scheduling that adopts fine-grain resource sharing. However, the Quincy model takes substantial time to find the best graph structure solution based on both data locality and load balancing.

MPI-based scientific applications generally prefer a computer-centric architecture that runs on several nodes using a file system. However, the unpredictable growth of scientific data threatens the efficiency and high performance of these MPI-based applications. Data locality MPI (DL-MPI) was proposed to solve this problem by acquiring data information through a data locality API and a scheduling algorithm that tasks each node based on its capacity [21]. Similarly, the data locality API assesses the amount of unprocessed local data; thereby, the scheduling algorithm allocates processing to compute nodes. However, the algorithm scales down on a large number of nodes, and its implementation requires sophistication. It would seem that the data movement overhead obstructs the scaling of the system in its baseline.

It was found that the locality-aware scheduling and data distribution on many-core processors have not been discussed adequately in the literature [8]. It was claimed [8] that leveraging the locality awareness reduces high-performance parallel-systems energy waste via optimized power-efficient techniques. The reference explains two power efficiency symbiotic techniques, which are the intralocality power-driven power optimization and the interprocess locality-driven power optimizations. The intralocality technique has the flexibility and control to assign processor frequency and manage sleep states within the same process sequence. On the other hand, the interprocess technique uses the coscheduling and coplacement of jobs in a varied set of threads from a diverse set of processes executed simultaneously on an HPC cluster. Coscheduling and coplacement group threads and processes, based on the similarity of their affinity patterns and symbiosis, reduce as much as 25% in energy consumption. However, energy reduction efficiency depends heavily on the correct identification of computer memory functions and the CPU capacity, which should be used in perfect harmony to be more beneficial. Therefore, it is prudent to analyze the results of these techniques in a systematic framework.

*2.3. Decomposition Techniques.* While software application parallelization provides efficiency, it could be error inclined and exceptionally unoptimistic, specifically in terms of the data decomposition task [22, 23]. Therefore, careful attention should be taken when choosing a parallelization methodology that provides consequent conveyance of data

or task processing running on available cores [24, 25]. While several studies were conducted to understand software usage, debugging, testing, project organization, and tuning [26, 27], data decomposition research is a significant challenge in a parallel programming environment [28]. Up to this point, there has been insufficient empirical research in the HPC field that has focused on these issues.

**2.4. Hybrid-Programming Models.** Several studies showed that MPI-based applications outperform Spark or Hadoop-based big data applications by order of magnitude or more for several applications, e.g., k-nearest neighbors and support vector machines [29], k-means [30], graph analytics [31, 32], and large-scale matrix factorizations [7]. It was found that compute load was the primary performance bottleneck in some Spark applications, precisely the duration of serialization and deserialization [33].

Big data programming models can be improved by combining them with parallel programming models like MPI. This approach can be seen in [34], which showed how to enable the Spark environment using the MPI libraries. Even though this technique signposts remarkable speedups, it must use shared memory and other overheads, potentially drawbacks. Several MPI-based communication primitives were used to resolve the performance problem of HPC over big data, for example, replacing MapReduce communicators in Hadoop (Jetty) with an MPI derivative (DataMPI) [35]. However, the approach is not a drop-in replacement for Hadoop, and existing modules need to be recoded to use DataMPI. While machine learning libraries can interoperate with Spark, such as H<sub>2</sub>O.ai [36] and deeplearning4j [37], they are limited. Such Java-based approaches do not provide the direct integration of existing native-code MPI-based libraries. One solution was spark with accelerated tasks (SWAT) [38] that creates OpenCL code from JVM code at runtime to improve Spark performance. As opposed to our work, SWAT is limited to single-node optimizations; it does not have access to the communication improvements made available through MPI. Alchemist [39] interfaces between MPI libraries and Spark and observes that such interfacing speeds up linear algebra routines. Improved performance comes from the comparative overhead of moving data over the network between Spark nodes and Alchemist, and there is still a benefit from working in the Spark environment.

### 3. The Proposed Hybrid System

Spark alone has serious deficiencies when handling computationally intensive tasks with big data, although it is excellent at managing big data itself. Alternatively, MPI-based implementations, due to their native approach, are considerably more computationally performant. However, MPI-based applications lack the efficiency to handle big data. Further, to enhance the computation process, GPU processors can be utilized if available. GPU programming standards, such as OpenACC, which support multivendor devices, are also low-level implementations supported in either C, C++, or Fortran and are compatible with MPI [40].

However, with the use of extra resources and GPUs, more energy is required. This creates a burden on the system, causing heating issues that can be a determinant in long-term computation.

Therefore, the two main factors that were considered while formulating the proposed technique are the following:

- (1) Combining the benefits of Spark data handling with the computationally intensive nature of MPI and OpenACC
- (2) Keeping power usage to a minimum by employing optimal integration techniques

**3.1. The Simple Integration.** In this section, the process of minimum integration is discussed while maintaining the two key principles explained previously. The three technologies—Spark, MPI, and OpenACC—will be integrated together to form the Hybrid Spark MPI OpenACC system (HSMO). Spark’s core is based on the resilient distributed dataset (RDD), a special data structure that has the ability to handle and compute data in parallel. RDD is the best solution for managing big data. However, RDDs are created and managed by high-level frameworks that require the use of JVM [41]. There is currently no stable interface for low-level programming languages or frameworks to manage RDDs [39]. This obstacle creates a hurdle for the transfer of data for processing to MPI- and OpenACC-based implementation. Hence, any implementation will require a bridge between Spark and low-level implementations, in which Spark will handle the data and call upon MPI-based workers to execute the desired computations of the data.

Nevertheless, any transfer of data after initial partitioning will conflict with the second principle of integration and cause a spike in power usage. Hence, the data, once partitioned, must not leave the nodes at which it resides.

To accomplish this, repartitioning is disabled, and workers are called on specific nodes. As the technique evolves, further steps are taken to ensure data movement is curtailed as much as possible to reduce power consumption. The previous discussion created a rough sketch of the components of the system that are to be created. Data are to be gathered in Spark, that is the input of the system and the point receiving user interaction. Likewise, as a high-level system, it is logical to make it the interface for the entire system. It will behave as the controller of the system, responsible for input, output, task distribution, and system management. Workers, which will do the bulk of the computations, will be implemented in MPI + OpenACC. Henceforth, they will be referred to as MPI-Workers. The job of the MPI-Worker is to process data and return a result to the controller. In Simple Integration, the goal is to combine the three subsystems to formulate the hybrid system. However, simple integration is not sufficient, and additional steps are required in certain cases and at certain points to enhance the performance of the newly formed hybrid system. Figures 1 and 2 show an abstract-level view of the hybrid system and our simple integration of Spark, MPI, and OpenACC.



FIGURE 1: Abstract-level view of the hybrid system.

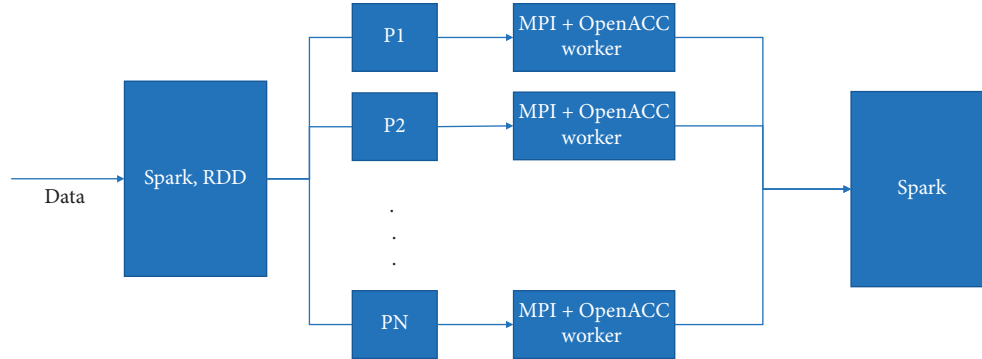


FIGURE 2: Simple integration of Spark, MPI, and OpenACC.

3.2. *The Extended Integration.* Once the basic integration of subsystems has been accomplished, it can be extended for the optimal performance of the parameters defined in the next chapter. The extension is based on three factors:

- (1) Physical topology generator.
- (2) The virtual topology generator of the application (e.g., big data application).
- (3) Mapping technique based on virtual and physical topologies.

In simple integration, the goal is to make all subsystems work together, building upon the same infrastructure. Additional components are added to the simple integration; these are as follows.

3.2.1. *Virtual Topology Generator.* A low-level component generates the topology within the MPI-worker. The grid generation can be accomplished by analyzing the application behavior to detect the number of processes/threads as well as the communication frequency between processes/threads for better mapping and task management. This can be accomplished by tracking every MPI sent and received and the messaging frequency between every two processes.

3.2.2. *Physical Topology Generator and Core Mapper.* A low-level component is responsible for two things: first, retrieve the core information for the entire system and second, bind the process/threads as per the virtual topology.

Figures 3 and 4 show the extended integration, in which C1, C2, and Cm1 are system cores, and g1, g2, and gp are the GPUs of the system. There are two forms of extended integration where the first one is extended integration with common GPU that many CPUs are sharing the same group of GPUs as shown in Figure 3. This can have less performance with less power consumption. In contrast, Figure 4 shows groups of CPUs sharing dedicated GPUs, and this

may enhance the performance with more power consumption.

3.3. *GPU Optimization.* In GPU optimization, regular cores are assigned for data handling, and partition numbers are increased, making it a function of the number of cores available across all GPUs in a combined system. While the bulk of computation is handled by GPUs, if any cores are available, they still handle processing as per the extended integration rules, because GPU integration is built upon extended integration and contains all features and characteristics of simple and extended integration. Figure 5 shows the GPU optimization variant.

## 4. Software Architecture and Modules of the Proposed Techniques

The high-level architecture of our hybrid system is shown in Figure 6; it is designed based on different software modules, resources, and interconnectivity components to create the overall system.

The data are provided by the user in Spark to the tri-model interface with the help of the metadata about the stored big data. The metadata could include useful information, such as the location, size, and type of the stored data. This can play a critical role in supporting big data computing in HPC, as the massive volume of data will be stored in many different nodes. The task here is distributed between Spark and MPI-workers, as the Spark source code is responsible for the management of the system, while the MPI-worker code processes the data. The Spark controller creates the partitions of the data based on the available HPC system resources in terms of regular CPU and GPU cores. In order to achieve this, it communicates with the HPC system and obtains the necessary system information. Based on this, partitions are created, and MPI-workers are called.

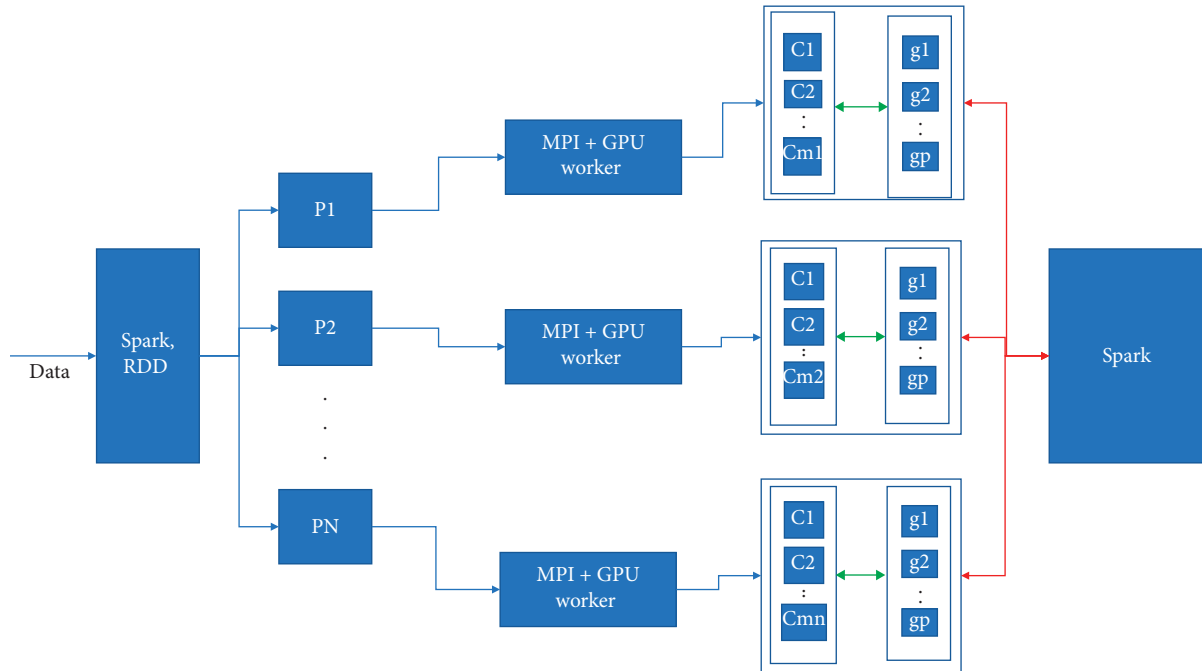


FIGURE 3: Extended integration with separate GPU grid.

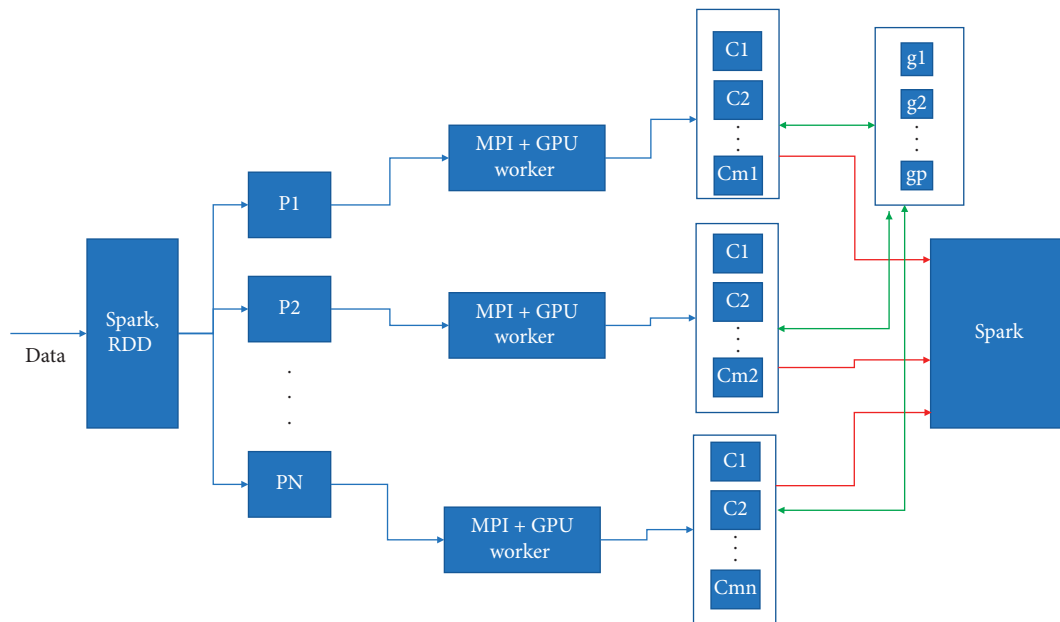


FIGURE 4: Extended integration with common GPU grid.

The MPI-worker receives the data allocated to it for processing. By receiving this and analyzing the worker behavior, the virtual topology of the worker can be generated.

Once the topology is generated, threads are created accordingly and bound to cores in the nodes in which the partitions reside. In order to do so, the physical topology of the nodes containing a bitmap and logical core tree is obtained. In the following explanation, each module, along with its algorithm, is discussed.

**4.1. Spark Controller Module.** The Spark controller module is the nerve center of the proposed technique; from here, the entire system is controlled and managed in terms of partitioning the RDD data and gathering the system resources. The first step is for the system user to input and/or choose the data. This is supported by metadata that helps the user explore and choose the required data. Then, this increases the effectiveness and quality of the system service. In addition, caches, along with system cores and GPUs, need to be fetched for partitioning purposes to maximize GPU

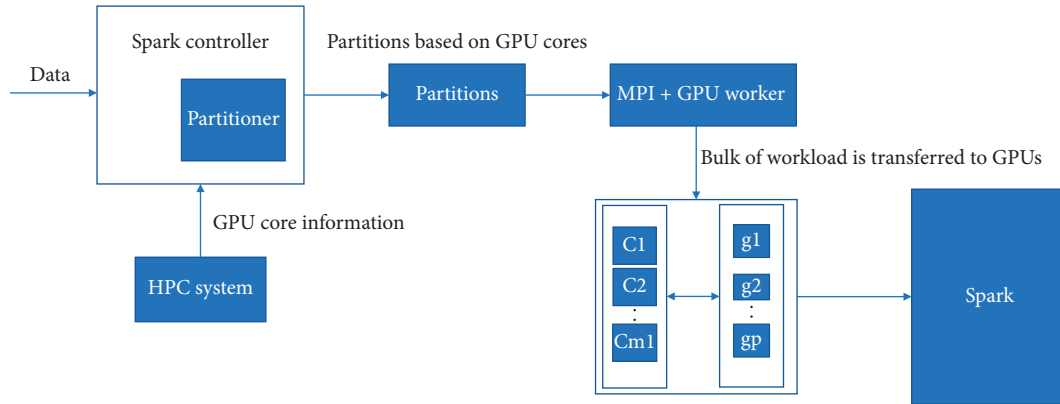


FIGURE 5: GPU optimization variant.

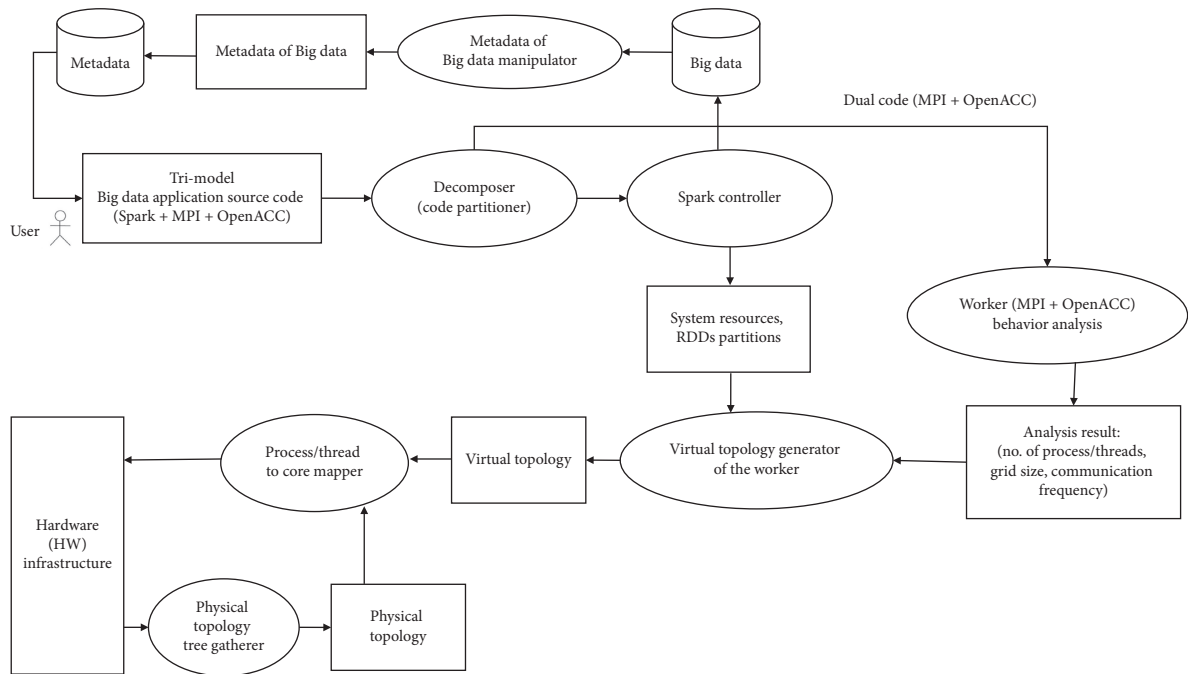


FIGURE 6: High-level architecture of the proposed system.

utilization. In order to do so, the hardware infrastructure information is fetched, and, from there, the GPU cores are obtained. The number of partitions is then made dependent on the number of GPU cores. The more GPU cores there are, the more partitions there will be.

Regular system cores are responsible for managing the partitions. If any system cores are left or idle, they may be used for processing in MPI-workers. The last step is result collection, during which MPI-workers send data back to the Spark controller.

In Algorithm 1, we show the Spark controller module mechanism.

**4.2. Virtual Topology Generator Module.** This module is responsible for generating the virtual topology of the MPI workers. The generation parameters are mainly dependent on the analysis of the MPI-worker source code to detect the number of processes/threads as well as the

communication frequency between them in order to formalize the grid's size and the ordering of the processes/threads on the grid.

A grid structure is created for threads intended to be created with possible messaging among neighbors. The reordering option allows for reordering the topology during execution. Such an option can be used based on the communication frequency between processes, so the highest communication processes should be close together and, accordingly, map the neighboring processes on physical resources close to each other. This period option allows cycling at the end of the grid. This is set to true, as it is intended to introduce new threads onto cores from the ready queue as soon as any slot becomes available.

Once options are set, the `MPI_Cart_Create` [42] method is called, which creates the new topology and returns the communicator for the new topology. The communicator holds the information regarding the structure of the

**INPUT:** Data.  
**OUTPUT:** Result.  
**START**

- (1) Let  $M$  hold  $R$ 's metadata for reference.
- (2) Let  $R$  be the RDD holding data, input by the system user.
- (3) Let  $H$  be the set of system resources containing Cores  $C$ , Memory  $r$ , Cache  $Ca$ , GPU cores  $Gc$ , and Nodes  $N$ .
- (4) Let  $P_i$  be a partition of  $R$ , where  $i = 1, 2, \dots, Gc$ .
- (5) Call worker on Node  $N_p$  for each  $P_i$ , where  $N_p$  is the node holding data for partition  $P_i$ .
  - (a) Collect partial results  $R_i$  for  $P_i$ .
  - (b) Combine  $R_i$  to form complete result  $R$  as per the application requirement.
- (6) Out Result.

**STOP**

ALGORITHM 1: Spark controller.

topology and the communicator is used in subsequent modules.

Our virtual topology generator module is presented in Algorithm 2, and the data structure example of the virtual topology is shown in Figure 7.

**4.3. MPI + OpenACC Implementer Module.** The purpose of this MPI-worker module is to implement application-specific tasks and generate partial results. This module is the core workhorse in which all the computations take place. For running on GPU, the first step is to check whether OpenACC libraries are available on the system. If so—and a compatible GPU is available—it can be initialized. OpenACC is a directive-based implementation, the benefit of which is that the absence of either OpenACC or GPU allows for normal/alternate execution to proceed smoothly.

However, within a GPU, code is parallelized even further. This process is automated and not controlled within the scope of the proposed system. This obstacle is presented as future work.

**4.4. Physical Topology Tree Gatherer Module.** An important step within the implementation is core binding. In order to bind threads to cores, the physical topology of the cores is required, and, as explained earlier, workers are called on the respective nodes of the partition. Each node may have its unique hardware topology. In order to retrieve this information, the HWLOC (Hardware locality) library is used, which provides two main items: topology and bitmap.

**4.4.1. Topology.** It is the entire system architecture of the node, core block, internal and external caches, memory, and buses. Topology is needed for referencing when binding to cores.

**4.4.2. Bitmap.** This is specific to cores. It contains the core layout, the way they are arranged in blocks, and their number.

This information can help deduce the ability of a node to handle simultaneous threads. Hardware locality provides

methods for extracting the information, which is then used for the mapping of threads.

Algorithm 3 shows our physical topology tree gatherer approach.

**4.5. Mapper Module.** The purpose of this module is to bind threads to cores. The information provided from the “Physical Topology Tree Gatherer Module” and “Virtual Topology Generator Module” is the input. The “implementer” module provides the thread in the simple and extended cases. In the GPU optimization case, core-binding is not required.

From the bitmap, a single-core reference is extracted and, again, the HWLOC provides the necessary method. An active thread is bound to a core by following the principle of virtual topology, in which neighboring threads, by following the principle of virtual topology, must be bound on neighboring cores.

In Algorithm 4, we show our core mapper mechanism.

## 5. Implementation and Evaluation

This section discusses the system implementation and its evaluation based on the results of a case study along with a discussion of those results. The details of the testing environment are also presented along with the testbed setup.

Additionally, the parameters for which the results were evaluated are discussed. Each technique is then evaluated based on those parameters. Moreover, the benefits and shortcomings of the techniques are presented. Finally, a comparative study is presented with the closest competitors of the technique.

**5.1. PageRank Algorithm as a Case Study.** PageRank Algorithm was used as a test case for the proposed system. PageRank is an iterative algorithm [16] defined by the following:

$$\mathbf{PR}^{t+1}(v) = \mathbf{r} + (1 - \mathbf{r}) \times \sum_{u | (u,v) \in E} \frac{\mathbf{PR}^t(u)}{\text{degree}(u)}, \quad (1)$$

where “ $E$ ” is the total number of nodes and “ $u$ ” is the summation variable 1, 2, 3, . . . ,  $E$ . “ $v$ ” is the function of the



**INPUT:** MPI-worker processes.  
**OUTPUT:** virtual topology for MPI-worker.  
**START**  
(1) Let  $Q$  be no. of Processes/Threads obtained using `MPI_comm_size`.  
(2) Let  $Dim$  be the dimensions of the Grid by calling the generator method.  
(3) Set options for virtual topology Reorder & Period True or False.  
(4) Call method `mpi_cart_create` with generated dimensions, and set options.  
(5) Out new communicator for virtual topology.  
**STOP**

ALGORITHM 2: Virtual topology generator.

Partition #	X	Y	Is GPU available?
0	0	9	0
0	0	10	0
0	0	1	1
0	0	2	0
0	0	7	0
0	0	3	0
0	0	8	0
0	0	5	0
0	0	4	0
0	0	6	0
0	0	0	0

FIGURE 7: Example of virtual topology data structure.

equation to be calculated, since its iterative equation “ $\mathbf{r}$ ” is a damping factor equal to 0.85, and it is based on PageRank theory, in which the surfers who click on random links will stop at the end of clicking.

The equation can be parallelized and is widely used in big data case studies. The equation is implemented using fixed iterations. The implementation algorithm is provided in Figure 8.

**5.2. Testing Environment.** Table 1 and Figure 9 show the specifications of the target machine used for developing the HSMO prototype. Figure 9 depicts the hierarchical view of the hardware specifications that are extracted by `hwloc’s lstopo` command.

**5.3. Methodology of the Experiments.** In order to test the PageRank Algorithm implementation, a large dataset is required. PageRank’s input links between nodes. Each node can represent a web page, social media account, or any similar scenario. The link between the two nodes can be represented by many means. For instance, they can be represented as a single line, in which each line has two-node entries.

For this purpose, the dataset requires millions of such links to produce enough load in order to satisfy various test cases for the system against the defined parameters.

The specifications of the dataset used for experimentation are listed in Table 2 [43]. In order to test the system, the dataset was sliced into various ranges, and readings were taken for parameters explained in a later section. The

PageRank algorithm allows for such slicing due to the nature of the algorithm.

**5.4. Experiments and Benchmarks.** There are two main parameters for which experiments are conducted: processing time and power consumption. These parameters are discussed below. The benchmark for both parameters is that they must improve on a Spark-only implementation of PageRank for all defined inputs.

**5.4.1. Processing Time.** In order to measure processing time, elapsed time “ $t_e$ ” is calculated, which is defined as the duration between an input being distributed among workers and the receipt of the overall results from all partitions. Equation (2) defines the elapsed time:

$$t_e = t_f - t_0, \quad (2)$$

where “ $t_0$ ” is the initial time recorded before the input is submitted to the workers and “ $t_f$ ” is the time recorded after the results are received.

Elapsed time “ $t_e$ ” was recorded for all techniques and is the key in determining the optimally beneficial choice of a technique for a data range. A technique is said to be more efficient for a dataset range if its elapsed time is less than that of other techniques.

**5.4.2. Power Consumption.** Power consumption is defined as the amount of power, in Watts, used over the execution time. The value is measured using the `powerstat` tool [44]. As per principle 2 of the proposed technique, power consumption is to be kept to a minimum. Hence, this parameter was essential for a complete analysis of the techniques.

`Powerstat` gives values in steps, while execution proceeds. The measurements need to be observed for idle state and spiked state when a task is submitted to the Apache Framework. Further, each step needs to be considered in the final calculation of the overall power consumption of a technique. Equation (3) gives the power consumption formula:

$$Pt = \sum_{k=1}^n (P_n - P_i), \quad (3)$$

where “ $Pt$ ” is the total power consumption,  $n$  is the number of steps in `powerstat`, “ $P_i$ ” is the idle state value, and “ $P_k$ ” is the step value of the power.

**INPUT:** HPC node hardware infrastructure.  
**OUTPUT:** Bitmap, logical core tree.  
**START**  
(1) Let NCW be the HPC node for worker W.  
(2) Let TNCW be the hardware topology for NCW obtained using `hwloc_topology_load`.  
(3) Let BNCW be the bitmap for cores obtained from TNCW.  
(4) Let LNCW be the logical core tree obtained from BNCW.  
(5) OUT BNCW, LNCW.  
**STOP**

ALGORITHM 3: Physical topology tree gatherer.

**INPUT:** bitmap, logical core tree, MPI thread,  $T$ ,  $T_d$ .  
**OUTPUT:** partial result.  
**START**  
(1) Let  $T_d$  hold the depth information based on the system topology  $T$  containing logical core information using method, `hwloc_get_type_or_below_depth`.  
(2) Let PU be the logical core obtained from  $T$  &  $T_d$  using method `hwloc_get_obj_by_depth`.  
(3) Let PU\_Cpuset hold PU core's cpuset.  
(4) Set PU\_CPuset to get only one logical processor in case of SMT/hyperthreading using method `hwloc_bitmap_singlify`.  
(5) Bind current `mpi_thread` to core using method `hwloc_set_bind`.  
(a) If bind fails, report error.  
(b) Otherwise calculate and return result  $R$ .  
(6) Out  $R$ .  
**STOP**

ALGORITHM 4: Core mapper.

```
var PR = Array.fill(n)(1.0)
val oldPR = Array.fill(n)(1.0)
for (iter <- 0 until numIter) {
  swap(oldPR, PR)
  for (i <- 0 until n) {
    PR[i] = alpha + (1 - alpha) * inNbrs[i].map(j => oldPR[j] / outDeg[j]).sum
  }
}
```

FIGURE 8: The PageRank algorithm implementation.

## 6. Discussion

Figure 10 shows the processing time comparison for the HSMO Simple and Extended variants with the Spark implementation. The first point to note is that, for the same set of resources, HSMO Simple is able to extend the range of data the system was able to handle. With HSMO Extended, the range was even further extended. It is noticeable that, as the data increased, HSMO Simple and Extended outperformed Spark. However, it is also observed that, for the smaller dataset, HSMO Simple and Extended performed almost identically, and, on some occasions, HSMO Simple performed better than HSMO Extended for smaller datasets. This can be attributed to the HSMO Extended version spending time creating a virtual topology and core

binding, whereas the simple variant used that time on computation. Only when the dataset was small could this overhead in HSMO Extended be noticed. However, these steps are necessary, since, as the dataset grows, system performance improves.

Figure 11 illustrates that power consumption is reduced in both the HSMO Simple and Extended variants in comparison with Spark, as the data size is increased. The abnormal spike in Spark's chart is due to the data movement that occurs due to the computational process being non-uniform in nature at any given stage.

The processing time comparison of HSMO Extended for various numbers of cores is shown in Figure 12. It can be seen that increasing the number of cores improves the processing time of the system, which emphasizes that the system is scalable. This is simply because the MPI-workers have more resources at their disposal.

The power consumption value comparison of HSMO Extended for various numbers of cores is displayed in Figure 13. It can be observed that, as more cores are introduced, less power is consumed. This is attributed to the fact that results are quickly calculated, resulting in lower runtime, and, hence, less power was consumed even though more resources were available for disposal. The abnormal spike in the middle of the chart can be attributed to the time required to create the virtual topology, so more time leads to more power usage.

Figure 14 shows the time and power statistics for HSMO GPU optimized. It can be observed that HSMO GPU optimized is able to outperform the Extended variant quite significantly in

TABLE 1: System specification.

OS	Ubuntu 18.04.1
Memory	Description: system memory Physical id: 3 b Slot: system board or motherboard Size: 16 GiB
Cache	*-cache:0 Description: L1 cache-size: 384 KiB *-cache:1 Description: L2 cache-size: 1536 KiB *-cache:2 Description: L3 cache-size: 9 MiB
CPU	Description: CPU-12 cores-product: Intel(R) Core (TM) i7-8750H CPU @ 2.20 GHz
GPU	NVIDIA GeForce GTX 1060

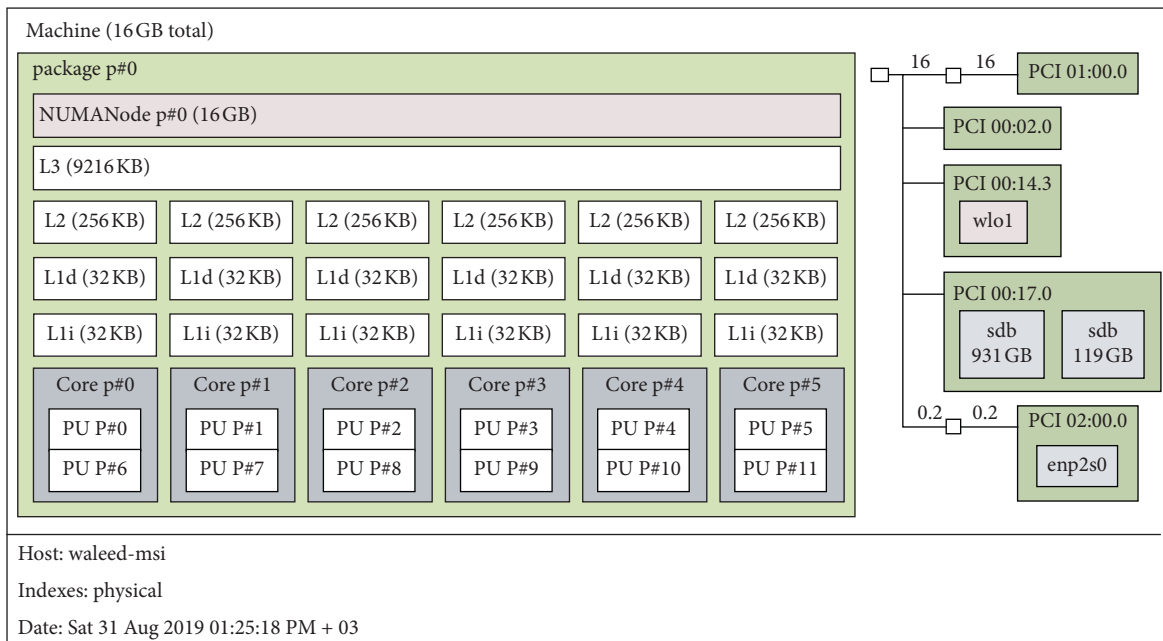


FIGURE 9: Hardware information of the target machine.

TABLE 2: Dataset specification.

Dataset property	Characteristics
Dataset nature	Social circles: Twitter
Nodes	81,306
Edges	1,768,149

terms of both time and power. This can be attributed to the fact that the capabilities of GPUS outperform CPUs performance significantly, and this clarifies that our system exploits such capabilities to achieve research objectives successfully.

It can also be realized from Figures 15 and 16 that HSMO GPU optimized is able to handle large datasets, with significant improvements in both time and power consumption. From Figure 15, we noticed that even after increasing the data size by

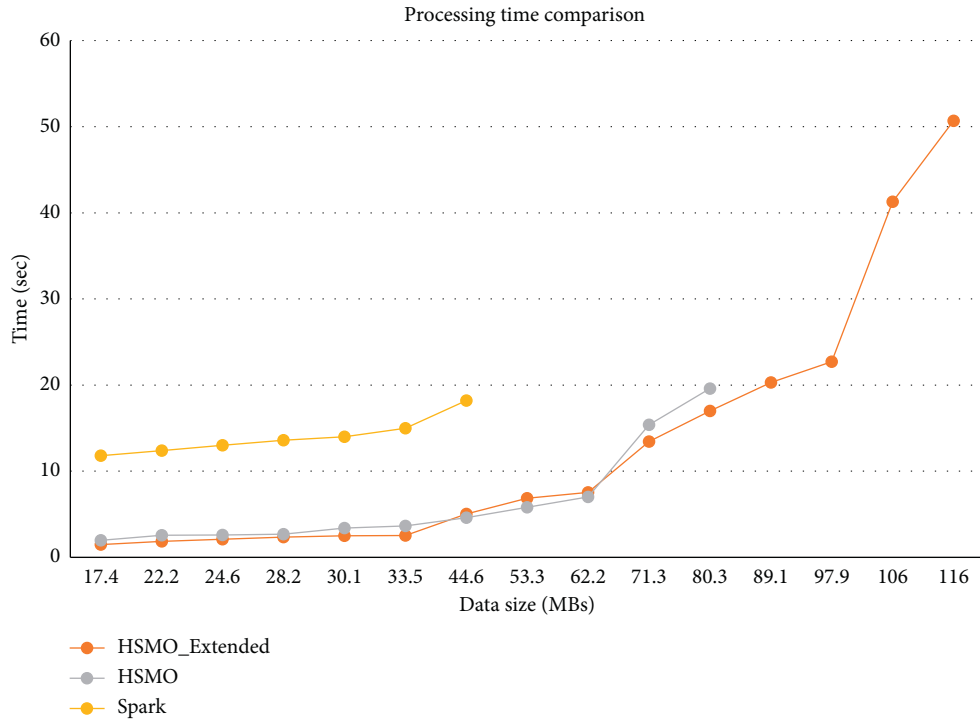


FIGURE 10: Processing time comparison for different approaches.

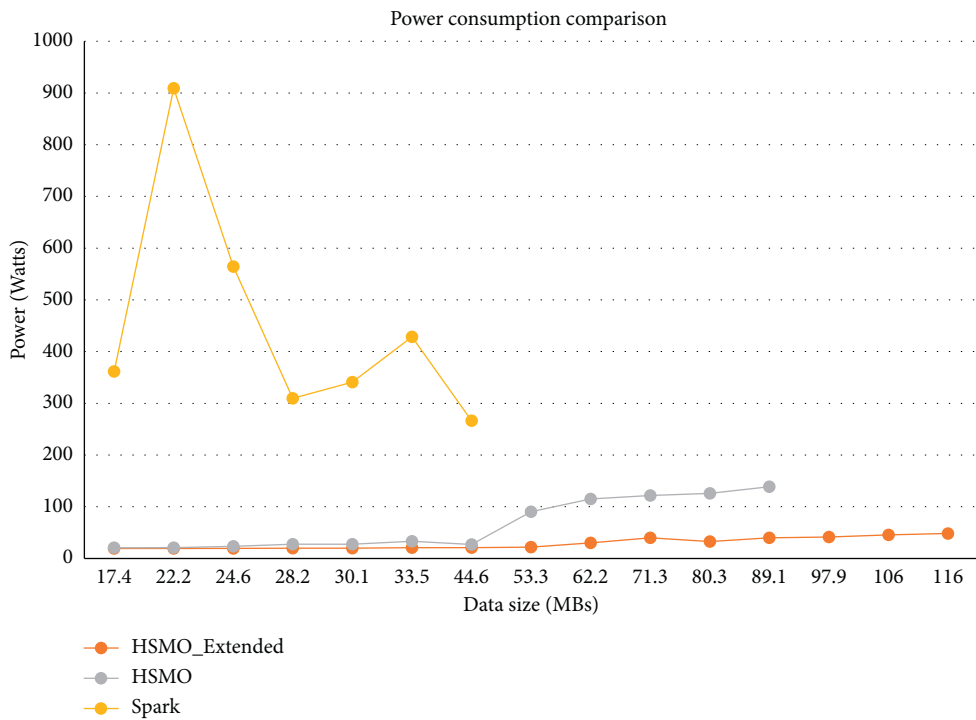


FIGURE 11: Power consumption comparison for different approaches.

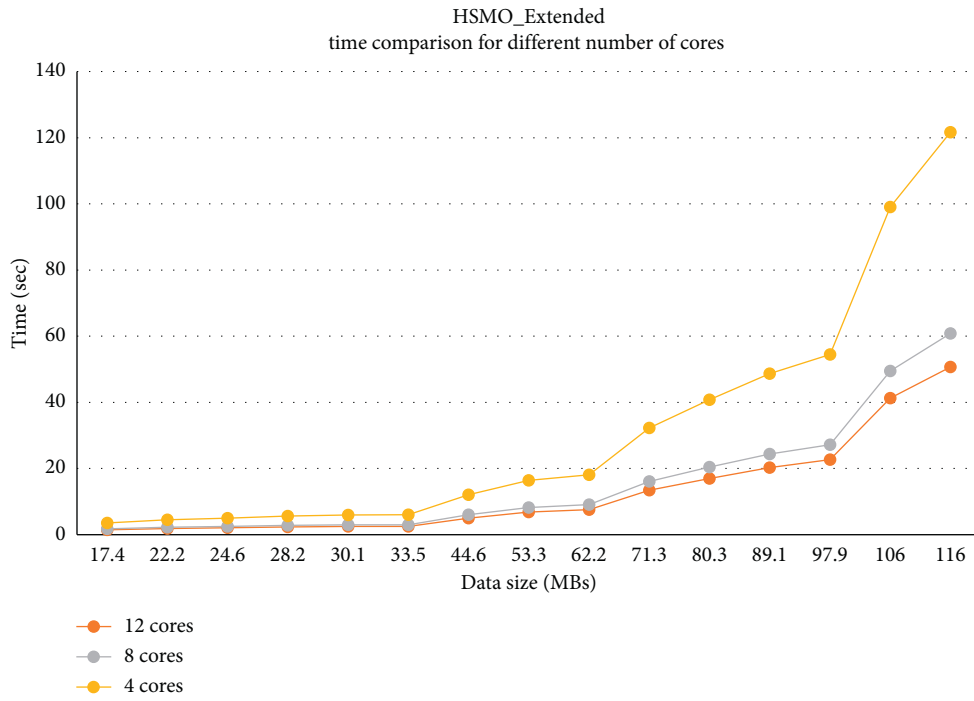


FIGURE 12: HSMO\_Extended/time comparison for different number of cores.

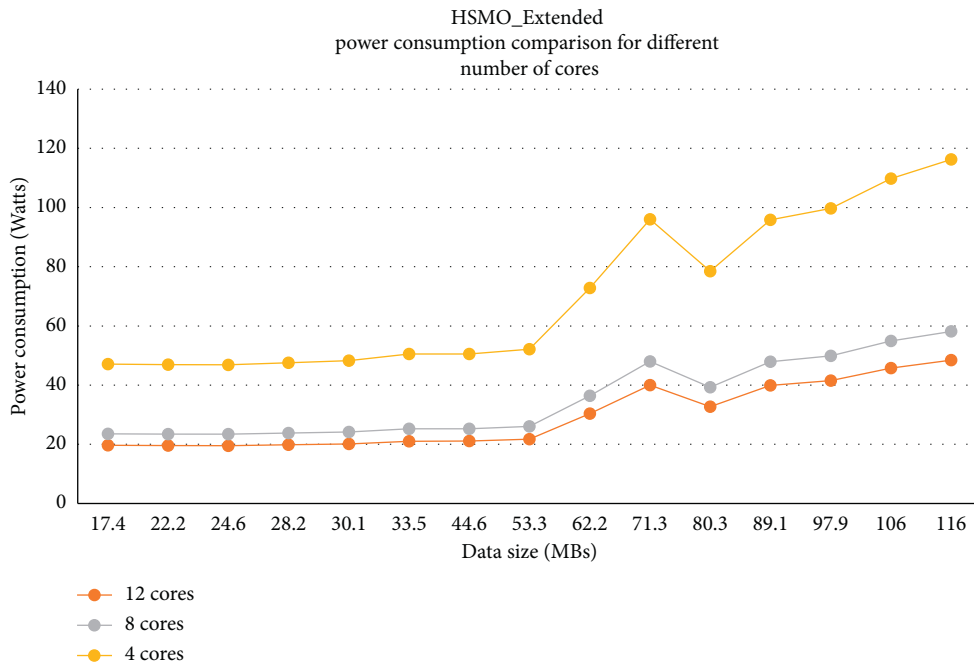


FIGURE 13: HSMO\_Extended/power consumption comparison for different number of cores.

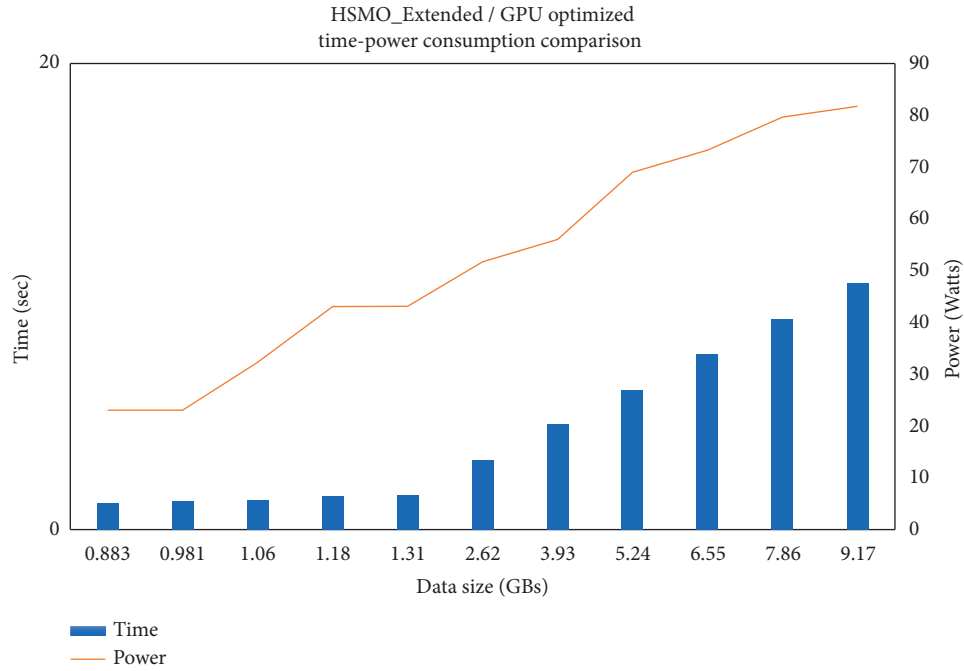


FIGURE 14: HSMO\_Extended/GPU optimized—time-power consumption comparison.

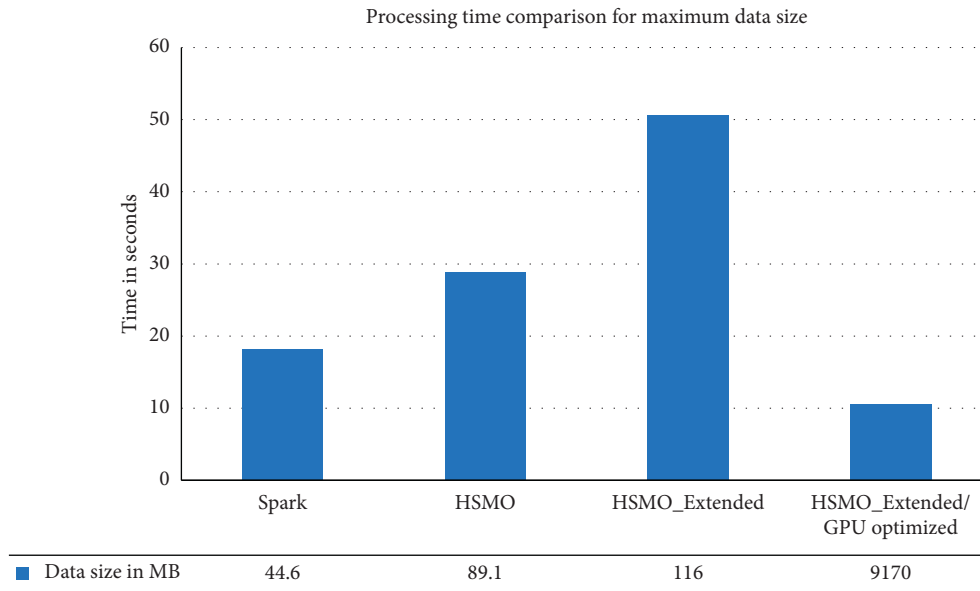


FIGURE 15: Processing time comparison for maximum data size.

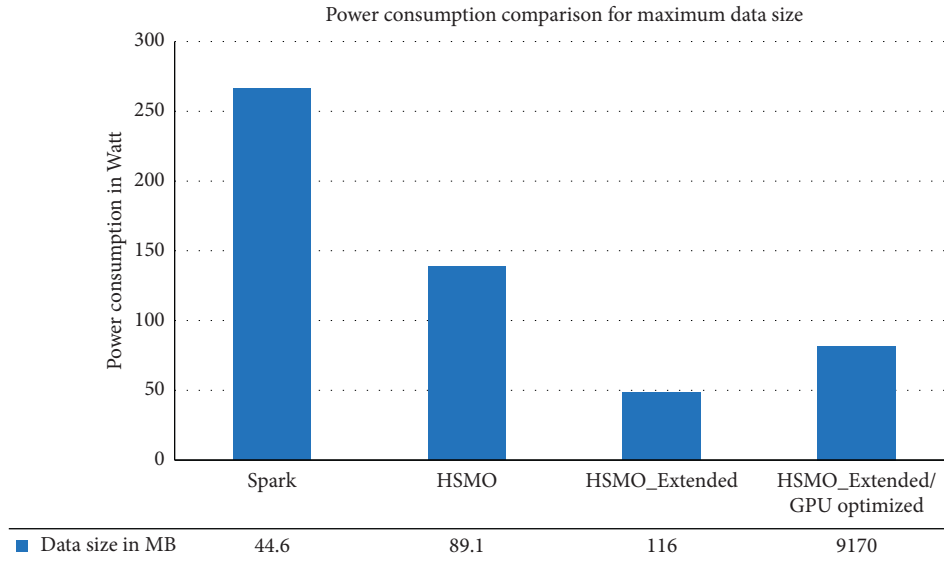


FIGURE 16: Power consumption comparison for maximum data size.

more than 20.000% (exactly 20.461%), the processing time has decreased by 42%, which improved the system performance. As we see in Figure 16, the power consumption has decreased by 69% even with the huge increase in the data size as we mentioned previously by more than 20,000%.

## 7. Comparative Study

The novelty of the proposed technique in this research can be seen in the tri-model that involves Spark, MPI, and OpenACC. In addition, and as far as we know, there is no work yet that employs this tri-model and considers both big data application behavior and the resource management of the hardware infrastructure. Furthermore, most of the related work focuses on enhancing performance—whether explicitly or implicitly. However, such enhancement is usually accompanied by an increase in power consumption. Targeting performance enhancement while controlling power consumption is a key challenge. This challenge arises clearly in the field of big data and HPC environments as the volume of data is enormous with high velocity and various varieties, and the underlying infrastructure has hundreds of different resources. Accordingly, it is important to consider parameters in both domains. Particularly the convergence of these two domains is going to be more necessary in many different scientific fields.

The two competing approaches, “bridging the gap between HPC and big data frameworks” [34] and Alchemist [39] integrate Spark with MPI. The first approach focuses on shared memory and optimizing Spark. Alchemist also fails to address the data locality question. Apart from not including OpenACC, it solely focuses on integration and not optimizing the integration. There is no mention of how MPI handles computations, a fault shared with [34].

The approach presented in “bridging the gap between HPC and big data frameworks” is more of an effort to optimize existing techniques rather than providing a new technique. It also fails to present how the actual integration module works. Alchemist

also suffers from data transfer overhead when data are on different nodes, indicating a lack of utilization of the Spark framework, as Spark is able to manage initial data transfers.

In comparison, HSMO placed an additional accelerant in the form of a GPU and presented variants for optimizing its use and also addresses the question of data locality. Furthermore, HSMO focuses on optimizing the hybrid system instead of the high-level system. This shows an understanding of the proven fact that high-level systems cannot compete in computational performance when compared with low-level systems. Moreover, HSMO has well-defined roles for each subsystem. User interactivity is not compromised, as the high-level module controls all user interactions. Moreover, performance is not compromised, as the low-level module controls all computational processes.

Comparing HSMO with Alchemist [34, 39] yields that it is a more dynamic technique that addresses practical questions of integration, specifically for MPI, which, by itself, lacks the ability to handle big data.

## 8. Conclusion and Future Work

This research demonstrated the performance gap challenge for big data applications on HPC clusters. The big data revolution is still moving at a high rate of growth in terms of the volume, velocity, and variety of data, which makes dealing with such data increasingly difficult. Accordingly, big data technologies keep emerging to face this drastic growth and overcome the difficulty of storing, transferring, and analyzing such enormous data to extract value and desired benefits.

In the same context, HPC, with its hardware infrastructure combining hundreds of thousands of computer nodes, packages, and parallel programming models, is a very attractive environment and a fertile field for supporting big data computing as well as many other types of research and experiments.

The convergence of HPC and big data is inevitable, which makes its progress subject to many accompanying

challenges, especially in the field of resource and job management and their effect on performance and power consumption.

Usually, there is a performance issue when deploying big data platforms on HPC clusters resulting from the underutilization of the capabilities afforded by HPC. This lack of performance is due to the difference between the two environments in terms of architectural design and the level of programming models that each support. The architectural design of big data platforms differs from that of HPC, which makes resource management and job scheduling very difficult. Furthermore, big data technologies are written in high-level programming languages that do not effectively support parallelism when compared with the lower-level programming languages supported by HPC environments. The large volume of big data may also hinder the application of a high level of parallelism when employing parallel programming models. As a result, big data needs to be tamed and optimized to maximize the potential from using HPC.

This was the focus of this paper, and it is achieved by developing the Hybrid Spark MPI OpenACC (HSMO) System that employs a novel resource management and mapping technique to stimulate high parallelism. HSMO takes into account decomposition and data locality strategies to schedule tasks according to the capabilities and availabilities of hardware infrastructure to enhance the performance while also controlling power consumption. Our future work can be summarized as the following:

- (1) GPU internally manages its resources—mainly cores. A mechanism should be developed, allowing access to those resources for an overall HSMO system.
- (2) When collecting results, data movement is required; this creates power consumption, and an alternate method may be investigated either to minimize the power consumption or to obtain alternative results.
- (3) Direct access to RDD partitions from MPI workers is required; this will entail the following:
 

The integrity of the original data for future reference. A referencing mechanism within MPI-workers for threads. If achieved, this will further enhance both power consumption and processing time measures.
- (4) Currently, when data need to be transferred to workers, a copy is made in the system even, as elaborated in the discussion of the proposed technique, if that data do not leave the node of the partition. This is due to the limitation of a stable interface between the HDFS File system, which is the backbone of RDDs, and low-level programming languages such as C, C++, or Fortran.
- (5) A copy creates a burden on the system, eats up resources that may be utilized for computation, or extends the data range of the available HPC system.
- (6) The metadata extraction in the Spark controller can be extensive and linked with MPI-workers for better user interactivity.
- (7) HSMO needs to be tested on HPC clusters; this includes clusters of various configurations.

## Data Availability

The Social circles: Twitter data used to support the findings of this study have been deposited in the Stanford Network Analysis Project repository (<https://doi.org/10.17616/R3XP7Q>).

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] S. Bergamaschi, C. Cavazzoni, A. Curioni, and G. Fox, “New opportunities in high performance data analytics (HPDA) and high performance computing (HPC),” in *Proceedings of 2014 International Conference on High Performance Computing & Simulation (HPCS)*, Bologna, Italy, July 2014.
- [2] K. M. Tolle, D. S. W. Tansley, and A. J. G. Hey, “The fourth paradigm: data-intensive scientific discovery [point of view],” *Proceedings of the IEEE*, vol. 99, no. 8, pp. 1334–1337, 2011.
- [3] S. Fiore, M. Bakhouya, and W. W. Smari, “On the road to exascale: advances in high performance computing and simulations—an overview and editorial,” *Future Generation Computer Systems*, vol. 82, pp. 450–458, 2018.
- [4] L. DeRose, “The path to delivering programable exascale systems,” in *Proceedings of 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, p. 727, Rio de Janeiro, Brazil, May 2019.
- [5] W. A. Shehri, M. Khemakhem, A. Basuhail, and F. E. Eassa, “A proposed architecture for parallel HPC-based resource management system for big data applications,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 1, pp. 40–44, 2019.
- [6] A. Uta, A. L. Varbanescu, A. Musaaifir, C. Lemaire, and A. Iosup, “Exploring HPC and big data convergence: a graph processing study on intel knights landing,” in *Proceedings of 2018 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 66–77, Belfast, UK, 2018–September.
- [7] A. Gittens, A. Devarakonda, E. Racah et al., “Matrix factorizations at scale: a comparison of scientific data analytics in spark and C+MPI using three case studies,” in *Proceedings of 2016 IEEE International Conference on Big Data (Big Data)*, pp. 204–213, Washington, DC, USA, December 2016.
- [8] W. Al Shehri, M. Khemakhem, F. Eassa, A. Basuhail, and F. E. Eassa, “Evaluation of high-performance computing techniques for big data applications,” *Science International*, vol. 31, no. 1, pp. 149–163, 2019.
- [9] S. Usman, R. Mehmood, and I. Katib, *Big Data and HPC Convergence for Smart Infrastructures: A Review and Proposed Architecture*, Springer Science and Business Media Deutschland GmbH, Berlin, Germany, pp. 561–586, 2020.
- [10] D. A. Reed and J. Dongarra, “Exascale computing and big data,” *Communications of the ACM*, vol. 58, no. 7, pp. 56–68, 2015.
- [11] W. Al Shehri, M. Khemakhem, A. Basuhail, and F. E. Eassa, “A hybrid spark MPI OpenACC system,” *International Journal of Computer Science and Network Society*, vol. 19, no. 5, pp. 81–86, 2019.
- [12] D. Unat, A. Dubey, T. Hoefler et al., “Trends in data locality abstractions for HPC systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 3007–3020, 2017.



- [13] G. Lee, N. Tolia, P. Ranganathan, and R. H. Katz, "Topology-aware resource allocation for data-intensive workloads," in *Proceedings of the First ACM Asia-Pacific Workshop on Workshop on Systems-APSys '10*, Hangzhou, China, August 2010.
- [14] R. Raman, M. Livny, and M. Solomon, "Matchmaking: distributed resource management for high throughput computing," in *Proceedings of the Seventh International Symposium on High Performance Distributed Computing*, pp. 140–146, Chicago, IL, USA, July 1998.
- [15] P. Mohanamurthy and G. Staffelbach, "Hardware locality-aware partitioning and dynamic load-balancing of unstructured meshes for large-scale scientific applications," in *Proceedings of the Platform for Advanced Scientific Computing Conference*, Geneva, Switzerland, June 2020.
- [16] S. L. Mirtaheri and L. Grandinetti, "Optimized load balancing in high-performance computing for big data analytics," *Concurrency and Computation: Practice and Experience*, p. e6265, 2021, Inpress.
- [17] K. Wang, X. Zhou, T. Li, D. Zhao, M. Lang, and I. Raicu, "Optimizing load balancing and data-locality with data-aware scheduling," in *Proceedings of 2014 IEEE International Conference on Big Data (Big Data)*, pp. 119–128, Washington, DC, USA, October 2014.
- [18] C. N. Nguyen, J. Lee, S. Hwang, and J.-S. Kim, "On the role of message broker middleware for many-task computing on a big-data platform," *Cluster Computing*, vol. 22, no. S1, pp. 2527–2540, 2019.
- [19] A. Di Stefano, A. Di Stefano, G. Morana, and D. Zito, "Coope4M: a deployment framework for communication-intensive applications on mesos," in *Proceedings of 2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 36–41, Paris, France, June 2018.
- [20] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles-SOSP'09*, p. 261, Big Sky, MT, USA, October 2009.
- [21] J. Yin, A. Foran, and J. Wang, "DL-MPI: enabling data locality computation for MPI-based data-intensive applications," in *Proceedings of 2013 IEEE International Conference on Big Data*, pp. 506–511, Silicon, Valley, CA, USA, October 2013.
- [22] H. Vandierendonck and T. Mens, "Averting the next software crisis," *Computer*, vol. 44, no. 4, pp. 88–90, 2011.
- [23] J. D. Garcia, D. del Rio, M. Aldinucci et al., "Challenging the abstraction penalty in parallel patterns libraries," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 5139–5159, 2020.
- [24] A. Johanson and W. Hasselbring, "Software engineering for computational science: past, present, future," *Computing in Science & Engineering*, vol. 20, no. 2, pp. 90–109, 2018.
- [25] N. Giacaman and O. Sinnen, "Preparing the software engineer for a modern multi-core world," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 247–263, 2018.
- [26] T. Marrinan, G. Eisenhauer, M. Wolf, J. A. Insley, S. Rizzi, and M. E. Papka, "Parallel streaming between heterogeneous HPC resources for real-time analysis," *Journal of Computational Science*, vol. 31, pp. 163–171, 2019.
- [27] A. Mastoras and T. R. Gross, "Understanding parallelization tradeoffs for linear pipelines," in *Proceedings of the 9th International Workshop on Programming Models and Applications for Multicores and Manycores*, pp. 1–10, Vienna, Austria, February 2018.
- [28] A. Meade, J. Buckley, and J. J. Collins, "Challenges of evolving sequential to parallel code," in *Proceedings of the 12th International Workshop and the 7th Annual ERCIM Workshop on Principles on Software Evolution and Software Evolution-IWPSE-EVOL'11*, Szeged, Hungary, September 2011.
- [29] J. L. Reyes-Ortiz, L. Oneto, and D. Anguita, "Big data analytics in the cloud: spark on Hadoop vs MPI/OpenMP on beowulf," *Procedia Computer Science*, vol. 53, pp. 121–130, 2015.
- [30] S. Jha, J. Qiu, A. Luckow, P. Mantha, and G. C. Fox, "A tale of two data-intensive paradigms: applications, abstractions, and architectures," in *Proceedings of 2014 IEEE International Congress on Big Data*, pp. 645–652, Anchorage, AK, USA, June 2014.
- [31] N. Satish, N. Sundaram, M. M. A. Patwary et al., "Navigating the maze of graph analytics frameworks using massive graph datasets," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 979–990, Snowbird, UT, USA, June 2014.
- [32] G. M. Slota, S. Rajamanickam, and K. Madduri, "A case study of complex graph analysis in distributed memory: implementation and optimization," in *Proceedings of 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 293–302, Chicago, IL, USA, May 2016.
- [33] K. Ousterhout, R. Rasti, S. Ratnasamy, S. Shenker, and B.-G. Chun, "Making sense of performance in data analytics frameworks," in *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15)*, pp. 293–307, Santa Clara, CA, USA, March 2015.
- [34] M. Anderson, S. Smith, N. Sundaram et al., "Bridging the gap between HPC and big data frameworks," *Proceedings of the VLDB Endowment*, vol. 10, no. 8, pp. 901–912, 2017.
- [35] X. Lu, F. Liang, B. Wang, L. Zha, and Z. Xu, "DataMPI: extending MPI to hadoop-like big data computing," in *Proceedings of 2014 IEEE 28th International Parallel and Distributed Processing Symposium*, pp. 829–838, Phoenix, AZ, USA, May 2014.
- [36] "H<sub>2</sub>O.ai. Sparkling Water." <https://github.com/h2oai/sparkling-water> (accessed Aug. 14, 2018).
- [37] "deeplearning4j. Deep Learning for Java. Open-Source, Distributed, Deep Learning Library for the JVM." <https://deeplearning4j.org/> (accessed Aug. 14, 2018).
- [38] M. Grossman and V. Sarkar, "SWAT," in *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing*, pp. 81–92, Kyoto, Japan, June 2016.
- [39] A. Gittens, K. Rothauge, S. Wang et al., "Accelerating large-scale data analysis by offloading to high-performance computing libraries using alchemist," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 293–301, London, UK, August 2018.
- [40] L. C. Platforms, P. Strategies, and Y. Yan, "Introduction to OpenACC Directives," 2012.
- [41] Databricks, "Intro to Apache Spark," 2015. [https://stanford.edu/~rezab/sparkclass/slides/itas\\_workshop.pdf](https://stanford.edu/~rezab/sparkclass/slides/itas_workshop.pdf).
- [42] "MPI\_Cart\_create(MPI\_Comm Comm\_old, Int Ndims, Int \*dims, Int \*periods, Int Reorder, MPI\_Comm \*comm\_cart) Function." [https://mpi.deino.net/mpi\\_functions/MPI\\_Cart\\_create.html](https://mpi.deino.net/mpi_functions/MPI_Cart_create.html) (accessed Oct. 18, 2019).
- [43] "SNAP: network datasets: social circles." <https://snap.stanford.edu/data/ego-Twitter.html> (accessed Dec. 01, 2019).
- [44] "Ubuntu Manpage: Powerstat-a tool to measure power consumption." <http://manpages.ubuntu.com/manpages/xenial/man8/powerstat.8.html> (accessed Oct. 18, 2019).

## Research Article

# Solution of the Systems of Delay Integral Equations in Heterogeneous Data Communication through Haar Wavelet Collocation Approach

Hualing Wu <sup>1</sup>, Rohul Amin <sup>2</sup>, Asmatullah Khan,<sup>2</sup> Shah Nazir <sup>3</sup>, and Sultan Ahmad <sup>4</sup>

<sup>1</sup>Shanghai Lixin University of Accounting and Finance, Shanghai 200120, China

<sup>2</sup>Department of Mathematics, University of Peshawar, Peshawar 25120, Pakistan

<sup>3</sup>Department of Computer Science, University of Swabi, Ambar 23430, Pakistan

<sup>4</sup>Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia

Correspondence should be addressed to Hualing Wu; [niki\\_456@sina.com](mailto:niki_456@sina.com) and Shah Nazir; [snsahnzr@gmail.com](mailto:snsahnzr@gmail.com)

Received 25 May 2021; Accepted 6 July 2021; Published 17 July 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Hualing Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work, the Haar collocation scheme is used for the solution of the class of system of delay integral equations for heterogeneous data communication. The Haar functions are considered for the approximation of unknown function. By substituting collocation points and applying the Haar collocation technique to system of delay integral equations, we have obtained a linear system of equations. For the solution of this system, an algorithm is developed in MATLAB software. The method of Gauss elimination is utilized for the solution of this system. Finally, by using these coefficients, the solution at collocation points is obtained. The convergence of Haar technique is checked on some test problems.

## 1. Introduction

Integral equations (IEs) are equations in which the unknown functions appear under one or more integral signs [1]. Delay integral equations (DIEs) are those IEs in which the solution of the unknown function is given in the previous time interval [2]. DIEs are further classified into two main types: Fredholm DIEs and Volterra DIEs on the basis of the limits of integration. Fredholm DIEs are those IEs in which limits of the integration are constant, while in Volterra DIEs, one of the limits of the integration is a constant and the other is a variable. A Volterra-Fredholm DIEs consist of disjoint Volterra and Fredholm IEs [1]. The DIEs play an important role in mathematics [3]. These equations are used for modelling of various phenomena such as modelling of systems with memory [4], mathematical modelling, electric circuits, and mechanical systems [5].

Several researchers are trying to find out the numerical solution of delay IEs. Darania [6] used the multistep

collocation method for solving DIEs. For each subinterval, the solution is obtained through a fixed number of collocation points and of previous steps in the current and next subintervals. Avaji et al. [7] used the variational iteration method for approximate solution of nonlinear and linear Volterra DIEs. The Volterra DIEs are constructed using general Lagrange multipliers that are defined by the variational theory and the initial approximations. Zhao et al. [8] used the Sinc collocation method for solving the DIEs. This technique reduces the DIEs of Volterra type to an explicit algebraic equation. The solution of these algebraic equations gives the solution of the Volterra DIEs. Yuzbasi and Ismailov [9] solved Volterra IEs with proportion delays by the method of differential transformation. In this technique, the solutions obtained are in the series form. The solution of the series expanded to Taylor series to find the unknown coefficients. For the system of nonlinear Volterra DIEs, Sekar and Murugesan [10] used the Walsh series method. This technique reduces the Volterra IEs into a system of

equations. The solution of an algebraic equation leads to the solution of the Volterra IEs. Kurkcu [11] used generalized Mott polynomials for the approximate solution of DIES with variable bounds. Maleknejad et al. [12] developed the numerical method for the solution Volterra IEs of first, second, and singular type of equations by the use of Bernstein approximation. Raza and Khan [13] found solution of neutral delay differential equations. Ghasemia and Kajani [14] utilized Chebyshev wavelets to find the solution time delay systems. The method is based on the expansion of various time functions and truncated Chebyshev wavelets. Wang [15] used hybrid functions for the solution of system of DIES. The hybrid functions consist of Legendre polynomials and block-pulse function. Samadi and Tohidi [16] used the spectral method for solution of systems of Volterra IEs; also, they used Spectral Galerkin approach for solution of two-dimensional Volterra IEs [17]. Tohidi [18] utilized Taylor matrix technique for solution of linear two-dimensional Fredholm IEs. Demko et al. [19] presented data-type agnostic algorithm calculating a concept lattice from heterogeneous and complex data. Luo et al. [20] investigated a communication scheduling problem to address data compression and data communication together. Ding and Zheng [21] investigated the bounded consensus tracking problem of heterogeneous nonlinear multiagent systems based on asynchronous sampled-data communication. Plaz et al. [22] presented MEdit4CEP-SP, the model-driven system that integrates stream processing and complex event processing technologies for consuming, processing, and analyzing

heterogeneous data in real time. Alqarni et al. [23] proposed a semicontrolled environment system which overcomes the limitations of users' age, gender, and smartwatch wearing style. Mazzara et al. [24] proposed a surveys' Internet of things and smart and software-defined buildings' technologies and their cooperation towards the realization of smart spaces. Sohaib et al. [25] enhanced a new technology acceptance-based research with the artificial neural network method to enable more precise and in-depth research results as compared to the single-step SEM method. The Haar technique for the solution of Fredholm and Volterra IEs was used by Aziz and Islam [26]. Haar technique is used for solution of different problems in literature. Some of the recent are fractional-order delay differential equations [27], distributed order time-fractional differential equations [28], second-order linear and nonlinear integro-differential equations [29], third-order linear and nonlinear boundary value problems of integro-differential equations [30], and second-order delay differential equations [31]. Amin et al. [32] developed Haar technique for the approximate solution of delay IEs. In this work, we will extend [32] for a system of delay IEs by using Haar wavelet collocation (HWC) technique.

In this study, the HWC scheme is used for the solution of system of linear delay IEs in heterogeneous data communication. The accuracy and efficiency will be checked on some test problems. The system of delay Volterra-Fredholm IE (DVFIE)  $\xi > 0$  is [33]

$$\mathbf{W}(t) = \begin{cases} \mathbf{W}(t - \xi) + \int_0^t \mathbf{K}(t, s)\mathbf{W}(s)ds + \int_0^1 \mathbf{M}(t, s)\mathbf{W}(s - \xi)ds + \int_0^{t-\xi} \mathbf{N}(t, s)\mathbf{W}(s)ds + \mathbf{F}(t), \\ \Phi(t), \quad t \in [-\xi, 0), \end{cases} \quad (1)$$

with initial conditions  $w_1(0) = \lambda_1$  and  $w_2(0) = \lambda_2$ , where  $\mathbf{W}(t) = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix}$  is vector function of the solution of system (1),  $\Phi(t) = \begin{pmatrix} \Phi_1(t) \\ \Phi_2(t) \end{pmatrix}$  is the delay conditions,  $\mathbf{K} = [k_{i,j}]_{2 \times 2}$  and  $\mathbf{M} = [m_{i,j}(t, s)]_{2 \times 2}$  are sufficiently smooth functions known as kernels of integration, and  $\mathbf{F}(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix}$  are given functions. For explicit derivation of the HWC technique, we consider  $\xi = 1$ .

## 2. Numerical Method

In this section, the HWC scheme is developed for the solution of the system of DVFIE (1). Let

$$\mathbf{W}(t) = \mathbf{W}(t) = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} \in L_2[0, 1]; \text{ then,} \quad (2)$$

$$w_1(t) = \sum_{i=1}^N a_i h_i(t),$$

$$w_2(t) = \sum_{i=1}^N b_i h_i(t).$$

The system of DVFIE (1) can be written as

$$w_1(t) = \begin{cases} w_1(t - \xi) + \int_0^t k_{11}(t, s)w_1(s)ds + \int_0^t k_{12}(t, s)w_2(s)ds + \int_0^1 m_{11}(t, s)w_1(s - \xi)ds \\ + \int_0^1 m_{12}(t, s)w_2(s - \xi)ds + \int_0^{t-\xi} n_{11}(t, s)w_1(s)ds + \int_0^{t-\xi} n_{12}(t, s)w_2(s)ds + f_1(t), \\ \Phi_1(t), \quad t \in [-\xi, 0), \end{cases}$$

$$w_2(t) = \begin{cases} w_2(t - \xi) + \int_0^t k_{21}(t, s)w_1(s)ds + \int_0^t k_{22}(t, s)w_2(s)ds + \int_0^1 m_{21}(t, s)w_1(s - \xi)ds \\ + \int_0^1 m_{22}(t, s)w_2(s - \xi)ds + \int_0^{t-\xi} n_{21}(t, s)w_1(s)ds + \int_0^{t-\xi} n_{22}(t, s)w_2(s)ds + f_2(t), \\ \Phi_2(t), \quad t \in [-\xi, 0). \end{cases} \quad (3)$$

By applying Haar approximation to the above system, we get the expression as

$$\sum_{i=1}^N a_i h_i(t) = \begin{cases} \Phi_1(t - \xi) + \int_0^t k_{11}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^t k_{12}(t, s) \sum_{i=1}^N b_i h_i(s)ds \\ + \int_0^1 m_{11}(t, s)\Phi_1(s - \xi)ds + \int_0^1 m_{12}(t, s)\Phi_2(s - \xi)ds \\ + \int_0^{t-\xi} n_{11}(t, s)\Phi_1(s)ds + \int_0^{t-\xi} n_{12}(t, s)\Phi_2(s)ds + f_1(t), \quad \text{for } t < 0, \\ \sum_{i=1}^N a_i h_i(t - \xi) + \int_0^t k_{11}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^t k_{12}(t, s) \sum_{i=1}^N b_i h_i(s)ds \\ + \int_0^1 m_{11}(t, s) \sum_{i=1}^N a_i h_i(s - \xi)ds + \int_0^1 m_{12}(t, s) \sum_{i=1}^N b_i h_i(s - \xi)ds \\ + \int_0^{t-\xi} n_{11}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^{t-\xi} n_{12}(t, s) \sum_{i=1}^N b_i h_i(s)ds + f_1(t), \quad \text{for } t > 0, \end{cases} \quad (4)$$

$$\sum_{i=1}^N b_i h_i(t) = \begin{cases} \Phi_2(t - \xi) + \int_0^t k_{21}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^t k_{22}(t, s) \sum_{i=1}^N b_i h_i(s)ds \\ + \int_0^1 m_{21}(t, s)\Phi_1(s - \xi)ds + \int_0^1 m_{22}(t, s)\Phi_2(s - \xi)ds \\ + \int_0^{t-\xi} n_{21}(t, s)\Phi_1(s)ds + \int_0^{t-\xi} n_{22}(t, s)\Phi_2(s)ds + f_2(t), \quad \text{for } t < 0, \\ \sum_{i=1}^N a_i h_i(t - \xi) + \int_0^t k_{21}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^t k_{22}(t, s) \sum_{i=1}^N b_i h_i(s)ds \\ + \int_0^1 m_{21}(t, s) \sum_{i=1}^N a_i h_i(s - \xi)ds + \int_0^1 m_{22}(t, s) \sum_{i=1}^N b_i h_i(s - \xi)ds \\ + \int_0^{t-\xi} n_{21}(t, s) \sum_{i=1}^N a_i h_i(s)ds + \int_0^{t-\xi} n_{22}(t, s) \sum_{i=1}^N b_i h_i(s)ds + f_2(t), \quad \text{for } t > 0, \end{cases}$$

discretizing the above two equations at CPs  $t_j$ ; where  $j = 1, 2, 3, \dots, N$ , we get the following expression:

$$\begin{aligned}
\sum_{i=1}^N a_i h_i(t_j) &= \begin{cases} \Phi_1(t_j - \xi) + \int_0^{t_j} k_{11}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j} k_{12}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds \\ + \int_0^1 m_{11}(t_j, s) \Phi_1(s - \xi) ds + \int_0^1 m_{12}(t_j, s) \Phi_2(s - \xi) ds \\ + \int_0^{t_j - \xi} n_{11}(t_j, s) \Phi_1(s) ds + \int_0^{t_j - \xi} n_{12}(t_j, s) \Phi_2(s) ds + f_1(t_j), & \text{for } t_j < 0, \\ \sum_{i=1}^N a_i h_i(t_j - \xi) + \int_0^{t_j} k_{11}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j} k_{12}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds \\ + \int_0^1 m_{11}(t_j, s) \sum_{i=1}^N a_i h_i(s - \xi) ds + \int_0^1 m_{12}(t_j, s) \sum_{i=1}^N b_i h_i(s - \xi) ds \\ + \int_0^{t_j - \xi} n_{11}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j - \xi} n_{12}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds + f_1(t_j), & \text{for } t_j > 0, \end{cases} \\
\sum_{i=1}^N b_i h_i(t_j) &= \begin{cases} \Phi_2(t_j - \xi) + \int_0^{t_j} k_{21}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j} k_{22}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds \\ + \int_0^1 m_{21}(t_j, s) \Phi_1(s - \xi) ds + \int_0^1 m_{22}(t_j, s) \Phi_2(s - \xi) ds \\ + \int_0^{t_j - \xi} n_{21}(t_j, s) \Phi_1(s) ds + \int_0^{t_j - \xi} n_{22}(t_j, s) \Phi_2(s) ds + f_2(t_j), & \text{for } t_j < 0, \\ \sum_{i=1}^N a_i h_i(t_j - \xi) + \int_0^{t_j} k_{21}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j} k_{22}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds \\ + \int_0^1 m_{21}(t_j, s) \sum_{i=1}^N a_i h_i(s - \xi) ds + \int_0^1 m_{22}(t_j, s) \sum_{i=1}^N b_i h_i(s - \xi) ds \\ + \int_0^{t_j - \xi} n_{21}(t_j, s) \sum_{i=1}^N a_i h_i(s) ds + \int_0^{t_j - \xi} n_{22}(t_j, s) \sum_{i=1}^N b_i h_i(s) ds + f_2(t_j), & \text{for } t_j > 0, \end{cases} \tag{5}
\end{aligned}$$

and let

$$\begin{aligned}
L_{11i}(t_j) &= \int_0^{t_j} k_{11}(t_j, s) h_i(s) ds, & L_{11i}(t_j) &= \int_0^{t_j} k_{21}(t_j, s) h_i(s) ds, \\
L_{21i}(t_j) &= \int_0^{t_j} k_{12}(t_j, s) h_i(s) ds, & L_{12i}(t_j) &= \int_0^{t_j} k_{22}(t_j, s) h_i(s) ds, \\
L_{31i}(t_j) &= \int_0^1 m_{11}(t_j, s) \Phi_1(s - \xi) ds, & L_{13i}(t_j) &= \int_0^1 m_{21}(t_j, s) \Phi_1(s - \xi) ds, \\
L_{41i}(t_j) &= \int_0^1 m_{12}(t_j, s) \Phi_2(s - \xi) ds, & L_{14i}(t_j) &= \int_0^1 m_{22}(t_j, s) \Phi_2(s - \xi) ds, \\
L_{51i}(t_j) &= \int_0^{t_j - \xi} n_{11}(t_j, s) \Phi_1(s) ds, & L_{15i}(t_j) &= \int_0^{t_j - \xi} n_{21}(t_j, s) \Phi_1(s) ds, \\
L_{61i}(t_j) &= \int_0^{t_j - \xi} n_{12}(t_j, s) \Phi_2(s) ds, & L_{16i}(t_j) &= \int_0^{t_j - \xi} n_{22}(t_j, s) \Phi_2(s) ds, \\
L_{71i}(t_j) &= \int_0^1 m_{11}(t_j, s) h_i(s - \xi) ds, & L_{17i}(t_j) &= \int_0^1 m_{21}(t_j, s) h_i(s - \xi) ds, \\
L_{81i}(t_j) &= \int_0^1 m_{12}(t_j, s) h_i(s - \xi) ds, & L_{18i}(t_j) &= \int_0^1 m_{22}(t_j, s) h_i(s - \xi) ds, \\
L_{91i}(t_j) &= \int_0^{t_j - \xi} n_{11}(t_j, s) h_i(s) ds, & L_{19i}(t_j) &= \int_0^{t_j - \xi} n_{21}(t_j, s) h_i(s) ds, \\
L_{101i}(t_j) &= \int_0^{t_j - \xi} n_{12}(t_j, s) h_i(s) ds, & L_{20i}(t_j) &= \int_0^{t_j - \xi} n_{22}(t_j, s) h_i(s) ds, \tag{6}
\end{aligned}$$

so

$$\begin{aligned}
\sum_{i=1}^N a_i h_i(t_j) &= \begin{cases} \Phi_1(t_j - \xi) + \sum_{i=1}^N a_i L_{1i}(t_j) + \sum_{i=1}^N b_i L_{2i}(t_j) + L_{3i}(t_j) + L_{4i}(t_j) \\ + L_{5i}(t_j) + L_{6i}(t_j) + f_1(t_j), & \text{for } t_j < 0, \\ \sum_{i=1}^N a_i h_i(t_j - \xi) + \sum_{i=1}^N a_i L_{1i}(t_j) + \sum_{i=1}^N b_i L_{1i}(t_j) + \sum_{i=1}^N a_i L_{7i}(t_j) + \sum_{i=1}^N b_i L_{8i}(t_j) \\ + \sum_{i=1}^N a_i L_{9i}(t_j) + \sum_{i=1}^N b_i L_{10i}(t_j) + f_1(t_j), & \text{for } t_j > 0, \end{cases} \\
\sum_{i=1}^N b_i h_i(t_j) &= \begin{cases} \Phi_2(t_j - \xi) + \sum_{i=1}^N a_i L_{11i}(t_j) + \sum_{i=1}^N b_i L_{12i}(t_j) + L_{13i}(t_j) + L_{14i}(t_j) \\ + L_{15i}(t_j) + L_{16i}(t_j) + f_2(t_j), & \text{for } t_j < 0, \\ \sum_{i=1}^N a_i h_i(t_j - \xi) + \sum_{i=1}^N a_i + \sum_{i=1}^N b_i L_{12i}(t_j) + \sum_{i=1}^N a_i L_{17i}(t_j) + \sum_{i=1}^N b_i L_{18i}(t_j) \\ + \sum_{i=1}^N a_i L_{19i}(t_j) + \sum_{i=1}^N b_i L_{20i}(t_j) + f_2(t_j), & \text{for } t_j > 0, \end{cases} \quad (7) \\
\left\{ \begin{aligned} & \sum_{i=1}^N [a_i(h_i(t_j) - L_{1i}(t_j)) - b_i L_{2i}(t_j)] = \Phi_1(t_j - \xi) + L_{3i}(t_j) + L_{4i}(t_j) + L_{5i}(t_j) \\ & + L_{6i}(t_j) + f_1(t_j), \quad \text{for } t_j < 0, \\ & \sum_{i=1}^N [a_i(h_i(t_j) - h_i(t_j - \xi) - L_{1i}(t_j) - L_{7i}(t_j) - L_{9i}(t_j)) - b_i(L_{2i}(t_j) - L_{8i}(t_j) - L_{10i}(t_j))] \\ & = f_1(t_j), \quad \text{for } t_j > 0, \\ & \sum_{i=1}^N [a_i(L_{11i}(t_j)) - b_i(h_i(t_j) - L_{12i}(t_j))] = \Phi_2(t_j - \xi) + L_{15i}(t_j) \\ & + L_{16i}(t_j) + f_1(t_j), \quad \text{for } t_j < 0, \\ & \sum_{i=1}^N [a_i(-h_i(t_j - \xi) - L_{11i}(t_j) - L_{17i}(t_j) - L_{19i}(t_j)) + b_i(h_i(t_j) - L_{12i}(t_j) - L_{18i}(t_j) - L_{20i}(t_j))] \\ & = f_2(t_j), \quad \text{for } t_j > 0. \end{aligned} \right.
\end{aligned}$$

We get the  $2N \times 2N$  linear system of algebraic equations with unknowns  $a_i$  and  $b_i$ . Solving this system, we obtain the values of unknown constants  $a_i$  and  $b_i$ . The required solution is obtained by putting these unknowns in equation (2).

*Remark 1.* If we take  $\mathbf{K} = 0$  and  $\mathbf{N} = 0$ , then system (1) is known as system of delay Fredholm IEs (DFIEs), and if we take  $\mathbf{M} = 0$ , then system (1) is known as system of delay Volterra IEs (DVIEs). Similarly, HWC technique can be developed for DFIEs and DVIEs.

### 3. Test Problems

In this section, the HWC scheme is used for solution of some examples. To show the convergence of this scheme, the maximum absolute errors with a different choice of CPs is given in tables. If  $W_{\text{exa}}(t)$  denotes the exact solution and  $W_{\text{appr}}(t)$  denotes the approximate solution at CPs, then  $L_{\infty}$  maximum absolute error is  $L_{\infty} = \max|W_{\text{exa}}(t) - W_{\text{appr}}(t)|$ , and the  $M_{cp}$  mean square root error in CPs is defined as  $M_{cp} = \sqrt{1/N (\sum_{i=1}^N |W_{\text{exa}}(t) - W_{\text{appr}}(t)|^2)}$ .

*Problem 1.* Consider the following system of delay VIEs [33]:

$$W(t) = \begin{cases} F(t) + \int_0^t \begin{pmatrix} s & 1 \\ 1 & s \end{pmatrix} w(s) ds + \int_0^{t-1} \begin{pmatrix} s-1 & 1 \\ 1 & s \end{pmatrix} w(s) ds, & 0 \leq t < 1, \\ \begin{pmatrix} e^{-t} \\ t^3 \end{pmatrix}, & -1 \leq t < 0. \end{cases} \quad (8)$$

The function  $F(t)$  is so that the exact solution is

$$W(t) = \begin{pmatrix} e^{-t} \\ t^3 \end{pmatrix}. \quad (9)$$

*Problem 2.* Consider the following system of delay VIEs [34]:

$$\begin{cases} w_1(t) + tw_2(t) = \sin t + t \cos t - \sin(t-1) + \int_0^t (t^2 \cos sw_1(s) - t^2 \sin sw_2(s)) ds + w_1(t-1), \\ w_2(t) - 2tw_1(t) = \cos t - 2t \sin t - \cos(t-1) + \int_0^t (\sin t \cos sw_1(s) - \sin t \sin sw_2(s)) ds + w_2(t-1), \\ 0 \leq t < 1, \end{cases} \quad (10)$$

where  $w_1(0) = 0$  and  $w_2(0) = 1$  and delay condition  $w_1(t) = \sin t$  and  $w_2(t) = \cos t$ , for  $-1 < t \leq 0$ , and the exact solution is  $w_1(t) = \sin t$  and  $w_2(t) = \cos t$

*Problem 3.* Consider the following system of delay VIEs [34]:

$$\begin{cases} w_1(t) = \frac{1}{4} (\cos(t-1) - \cos(t+1)) + \cos t (2 + \sin t - t \cos t) - \frac{t}{2} \sin(t-1) - 1 \\ -\cos(t-1) + w_1(t-1) + \int_0^t (\sin((t-s)-1)w_1(s) + (1-s \cos t)w_2(s)) ds, \\ w_2(t) = w_2(t-1) + \sin t - t - \sin(t-1) + w_2(t-1) + \int_0^t \{w_1(s) + (t-s)w_2(s)\} ds, \end{cases} \quad (11)$$

where initial conditions  $w_1(0) = 1$  and  $w_2(0) = 0$  and delay conditions  $w_1(t) = \cos t$  and  $w_2(t) = \sin t$  for  $-1 < t \leq 0$ . The exact solution is  $w_1(t) = \cos t$  and  $w_2(t) = \sin t$ .

*Problem 4.* Consider the following system of delay FIEs:

$$\begin{cases} w_1(t) - tw_2(t) = e^{2t} - t \cos(2t) - \frac{1}{2}t(2 + \sin 2) - e^{-2(t-1)} + \int_0^1 (te^{-2s}w_1(s) + tw_2(s)) ds + w_1(t-1), \\ tw_1(t) + w_2(t) = te^{2t} + \cos(2t) - \frac{1}{4}(1 + 3e^t(1 + e^2) - \cos 2 + 2(-1 + t)\sin 2) - \cos(2t-2) \\ + \int_0^1 ((t-s)w_1(s) + (t+s)^2w_2(s)) ds + w_2(t-1), \end{cases} \quad (12)$$

TABLE 1:  $L_\infty$  and  $M_{cp}$  errors for Problem 1.

J	$N = 2^{J+1}$	$L_\infty$		$M_{cp}$	
		$w_1(t)$	$w_2(t)$	$w_1(t)$	$w_2(t)$
1	4	$7.227277 \times 10^{-02}$	$3.535106 \times 10^{-02}$	$3.642416 \times 10^{-02}$	$3.642416 \times 10^{-02}$
2	8	$3.651301 \times 10^{-02}$	$1.949725 \times 10^{-02}$	$1.331555 \times 10^{-02}$	$1.331555 \times 10^{-02}$
3	16	$2.021651 \times 10^{-02}$	$1.171350 \times 10^{-02}$	$5.561894 \times 10^{-03}$	$5.561894 \times 10^{-03}$
4	32	$9.590612 \times 10^{-03}$	$5.528632 \times 10^{-03}$	$1.911783 \times 10^{-03}$	$1.911783 \times 10^{-03}$
5	64	$4.865959 \times 10^{-03}$	$2.838307 \times 10^{-03}$	$7.213247 \times 10^{-04}$	$7.213247 \times 10^{-04}$
6	128	$2.360761 \times 10^{-03}$	$1.363338 \times 10^{-03}$	$2.513415 \times 10^{-04}$	$2.513415 \times 10^{-04}$
7	256	$1.174814 \times 10^{-03}$	$6.766049 \times 10^{-04}$	$8.963455 \times 10^{-05}$	$8.963455 \times 10^{-05}$

TABLE 2:  $L_\infty$  and  $M_{cp}$  errors for Problem 2.

J	$N = 2^{J+1}$	$L_\infty$		$M_{cp}$	
		$w_1(t)$	$w_2(t)$	$w_1(t)$	$w_2(t)$
0	2	$1.971049 \times 10^{-03}$	$5.864174 \times 10^{-02}$	$1.393799 \times 10^{-03}$	$4.181707 \times 10^{-02}$
1	4	$1.810205 \times 10^{-03}$	$4.057245 \times 10^{-02}$	$9.051023 \times 10^{-04}$	$2.029205 \times 10^{-02}$
2	8	$1.164381 \times 10^{-03}$	$2.319242 \times 10^{-02}$	$4.353520 \times 10^{-04}$	$9.135788 \times 10^{-03}$
3	16	$6.956396 \times 10^{-04}$	$1.312505 \times 10^{-02}$	$1.796679 \times 10^{-04}$	$3.464783 \times 10^{-03}$
4	32	$3.726557 \times 10^{-04}$	$6.850666 \times 10^{-03}$	$7.173199 \times 10^{-05}$	$1.363234 \times 10^{-03}$
5	64	$1.964274 \times 10^{-04}$	$3.565292 \times 10^{-03}$	$2.635402 \times 10^{-05}$	$4.894977 \times 10^{-04}$
6	128	$1.000512 \times 10^{-04}$	$1.804582 \times 10^{-04}$	$9.855819 \times 10^{-06}$	$1.826391 \times 10^{-04}$

TABLE 3:  $L_\infty$  and  $M_{cp}$  errors for Problem 3.

J	$N = 2^{J+1}$	$L_\infty$		$M_{cp}$	
		$w_1(t)$	$w_2(t)$	$w_1(t)$	$w_2(t)$
0	2	$7.068456 \times 10^{-02}$	$2.845448 \times 10^{-02}$	$5.381595 \times 10^{-02}$	$2.076129 \times 10^{-02}$
1	4	$4.219307 \times 10^{-02}$	$3.246160 \times 10^{-03}$	$2.141376 \times 10^{-02}$	$1.862447 \times 10^{-03}$
2	8	$2.291424 \times 10^{-02}$	$3.324980 \times 10^{-03}$	$9.682159 \times 10^{-03}$	$1.804969 \times 10^{-03}$
3	16	$1.260881 \times 10^{-02}$	$3.159590 \times 10^{-03}$	$3.464271 \times 10^{-03}$	$9.044071 \times 10^{-04}$
4	32	$6.573418 \times 10^{-03}$	$1.744969 \times 10^{-03}$	$1.377181 \times 10^{-03}$	$3.610467 \times 10^{-04}$
5	64	$3.399905 \times 10^{-03}$	$1.098218 \times 10^{-03}$	$4.897071 \times 10^{-04}$	$1.544038 \times 10^{-04}$
6	128	$1.723213 \times 10^{-03}$	$5.586336 \times 10^{-04}$	$1.832585 \times 10^{-04}$	$5.548257 \times 10^{-05}$

TABLE 4:  $L_\infty$  and  $M_{cp}$  errors for Problem 4.

J	$N = 2^{J+1}$	$L_\infty$		$M_{cp}$	
		$w_1(t)$	$w_2(t)$	$w_1(t)$	$w_2(t)$
1	4	$2.640119 \times 10^{-01}$	$1.372997 \times 10^{-01}$	$1.736979 \times 10^{-01}$	$1.230985 \times 10^{-01}$
2	8	$6.953929 \times 10^{-02}$	$3.415118 \times 10^{-02}$	$4.329946 \times 10^{-02}$	$3.057921 \times 10^{-02}$
3	16	$1.784413 \times 10^{-02}$	$8.547046 \times 10^{-03}$	$2.703951 \times 10^{-02}$	$1.907584 \times 10^{-02}$
4	32	$4.519613 \times 10^{-03}$	$2.136780 \times 10^{-03}$	$1.081759 \times 10^{-03}$	$7.633199 \times 10^{-03}$
5	64	$1.137303 \times 10^{-03}$	$5.341455 \times 10^{-04}$	$6.759601 \times 10^{-04}$	$4.768514 \times 10^{-04}$
6	128	$2.852556 \times 10^{-04}$	$1.335397 \times 10^{-04}$	$1.689883 \times 10^{-04}$	$1.192101 \times 10^{-04}$

TABLE 5:  $L_\infty$  and  $M_{cp}$  errors for Problem 5.

J	$N = 2^{J+1}$	$L_\infty$		$M_{cp}$	
		$w_1(t)$	$w_2(t)$	$w_1(t)$	$w_2(t)$
1	4	$3.510916 \times 10^{-02}$	$4.831041 \times 10^{-02}$	$2.519685 \times 10^{-02}$	$3.029705 \times 10^{-02}$
2	8	$3.115079 \times 10^{-02}$	$1.559781 \times 10^{-02}$	$1.340721 \times 10^{-02}$	$1.093659 \times 10^{-02}$
3	16	$2.035756 \times 10^{-02}$	$4.267191 \times 10^{-03}$	$5.730156 \times 10^{-03}$	$2.242606 \times 10^{-03}$
4	32	$1.126852 \times 10^{-02}$	$2.023585 \times 10^{-03}$	$2.119183 \times 10^{-03}$	$8.120494 \times 10^{-04}$
5	64	$5.900960 \times 10^{-03}$	$5.002232 \times 10^{-04}$	$7.759655 \times 10^{-04}$	$2.083728 \times 10^{-04}$
6	128	$3.019204 \times 10^{-03}$	$3.746060 \times 10^{-04}$	$2.803629 \times 10^{-04}$	$8.009401 \times 10^{-05}$



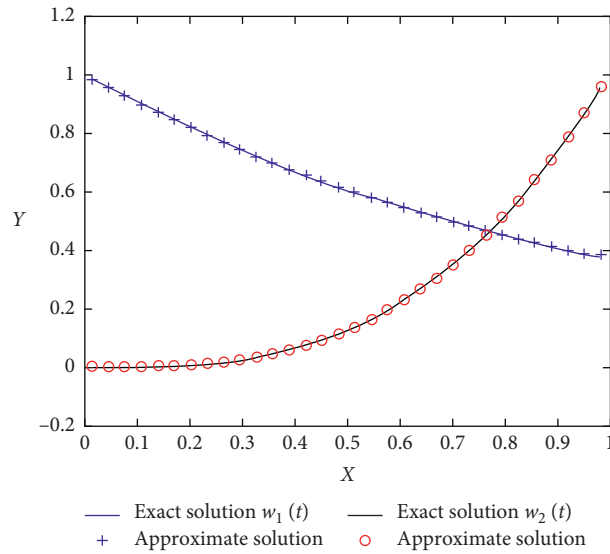


FIGURE 1: Comparison of exact and approximate solution for  $N = 32$  for Problem 1.

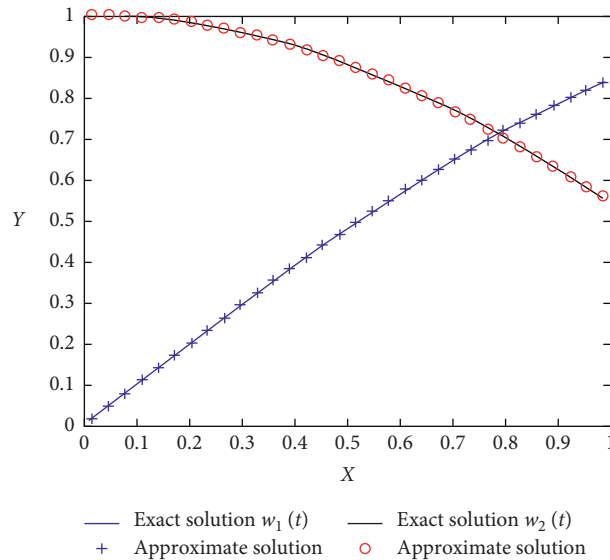


FIGURE 2: Comparison of exact and approximate solution for  $N = 32$  for Problem 2.

where initial conditions  $w_1(0) = w_2(0) = 1$  and delay conditions  $w_1(t) = e^{2t}$  and  $w_2(t) = \cos(2t)$ , for  $-1 < t \leq 0$ . The exact solution is  $w_1(t) = e^{2t}$  and  $w_2(t) = \cos(2t)$ .

*Problem 5.* Consider the following system of delay VFIEs:

$$\left( \begin{array}{l} w_1(t) = e^{-2t} - e^{-2(t-1)} - \frac{-3 - 2t + e^2(1 + t(2 + 4t - 2 \cos 2 + \sin 2))}{4e^2} + t \cos^2 t \sin^2 t \\ \quad + \int_0^1 ((t+s)w_1(s) + tsw_2(s))ds + w_1(t-1) + \int_0^t (te^{2s}w_1(s) + t \cos(2s)w_2(s))ds, \\ w_2(t) = \sin(2t-2) - \frac{1}{4}(-1 + 4(-1+e)e^{-1+t} + e^{-2t} + \cos t - \cos(3t) + t(2 + \sin^2(2))) + w_2(t-1) \\ \quad + \int_0^1 (e^{t+s}w_1(s) + t \cos(2s)w_2(s))ds + \int_0^t ((t-s)w_1(s) + t \cos(t)w_2(s))ds, \end{array} \right) \quad (13)$$

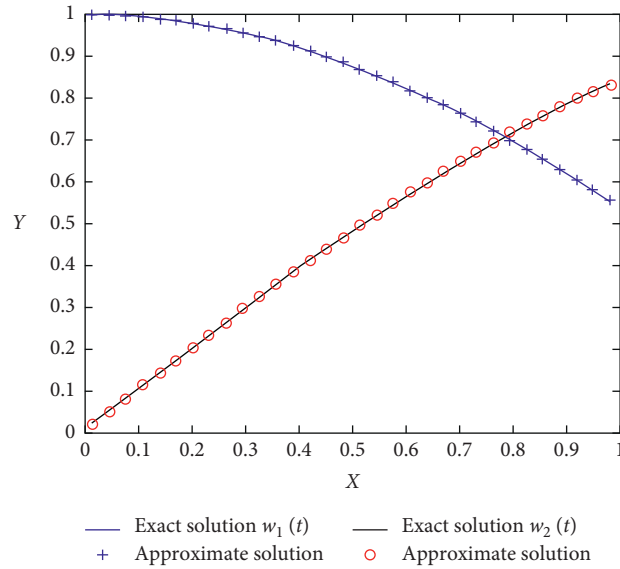


FIGURE 3: Comparison of exact and approximate solution for  $N = 32$  for Problem 3.

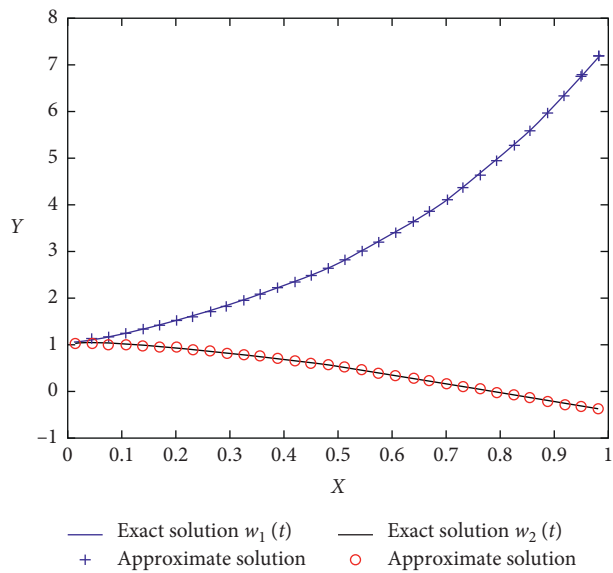


FIGURE 4: Comparison of exact and approximate solution for  $N = 32$  for Problem 4.

where initial conditions  $w_1(0) = 1$  and  $w_2(0) = 0$  and delay conditions  $w_1(t) = e^{-2t}$  and  $w_2(t) = \sin(2t)$ , for  $-1 < t \leq 0$ . The exact solution is  $w_1(t) = e^{-2t}$  and  $w_2(t) = \sin t$ .

#### 4. Results and Discussion

$L_\infty$  and  $M_{cp}$  errors are calculated for each example using different number of CPs. From results, we see that both  $L_\infty$  and  $M_{cp}$  errors are decreased by increasing number of CPs. Even better accuracy can be obtained by taking more CPs.

The results are presented in Table 1 for Problem 1, Table 2 for Problem 2, Table 3 for Problem 3, Table 4 for Problem 4, and Table 5 for Problem 5, demonstrating the proposed techniques improved accuracy and efficiency. The  $L_\infty$  errors of [33] are decreased up to  $10^{-05}$ , while the result of our method is also decreased up to  $10^{-05}$ . Figures 1–5 show a comparison of approximate and exact solutions for various numbers of CPs for Problems 1–5 respectively. We see from the figures that the approximate and exact solutions coincide.

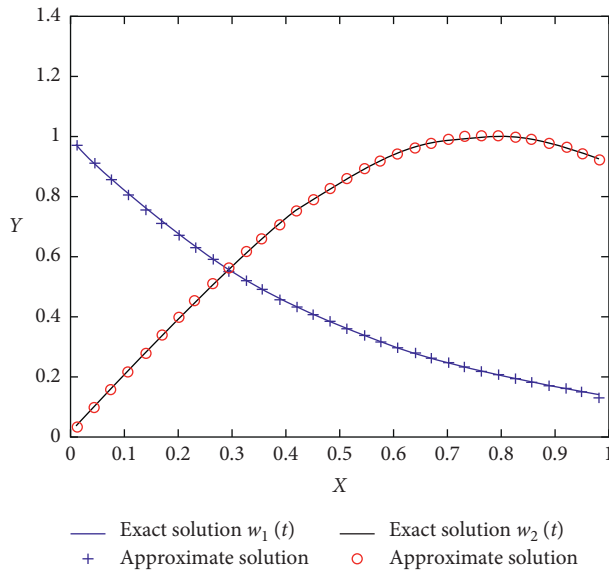


FIGURE 5: Comparison of exact and approximate solution for  $N = 32$  for Problem 5.

## 5. Conclusion

For the numerical solution of system of linear delay Volterra-Fredholm IEs using ICs in heterogeneous data communication, a HWC technique is developed. The numerical technique is used to test the accuracy and efficiency of the HWC scheme on several examples. Tables shows the  $L_\infty$  and  $M_{cp}$  errors of each example for various numbers of CPs. Comparison of exact and approximate solution is also shown in figures. The Haar technique can be applied to the system of nonlinear IEs, the system of integro-differential equations, and the system of integro-partial differential equations.

## Data Availability

No data were used to support the findings of the study.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## References

- [1] A. M. Wazwaz, *A First Course in Integral Equations*, World Scientific, London, UK, 2015.
- [2] I. Aziz and R. Amin, "Numerical solution of a class of delay differential and delay partial differential equations via Haar wavelet," *Applied Mathematical Modelling*, vol. 40, no. 23-24, pp. 10286–10299, 2016.
- [3] Z. Yang and H. Brunner, "Blow-up behavior of Hammerstein-type delay Volterra integral equations," *Frontiers of Mathematics in China*, vol. 8, no. 2, pp. 261–280, 2013.
- [4] K. Al-Khaled, "Numerical approximations for population growth models," *Applied Mathematics and Computation*, vol. 160, no. 3, pp. 865–873, 2005.
- [5] A. Bellour and M. Bousselsal, "A Taylor collocation method for solving delay integral equations," *Numerical Algorithms*, vol. 65, no. 4, pp. 843–857, 2014.
- [6] P. Darania, "Multistep collocation method for nonlinear delay integral equations," *Sahand Communications in Mathematical Analysis*, vol. 3, pp. 47–65, 2016.
- [7] M. Avaji, J. S. Hafshejani, S. S. Dehcheshme, and D. F. Ghahfarokh, "Solution of delay Volterra integral equations using the variational iteration method," *Journal of Applied Sciences*, vol. 12, no. 2, pp. 196–200, 2012.
- [8] J. Zhao, Y. Cao, and Y. Xu, "Sinc numerical solution for pantograph Volterra delay-integro-differential equation," *International Journal of Computer Mathematics*, vol. 94, no. 5, pp. 853–865, 2017.
- [9] S. Yuzbasi and N. Ismailov, "Solving systems of Volterra integral and integro-differential equations with proportional delays by differential transformation method," *Journal of Mathematics*, vol. 14, pp. 1–5, 2014.
- [10] R. C. G. Sekar and K. Murugesan, "Single term Walsh series method for the system of nonlinear delay Volterra integro-differential equations describing biological species living together," *International Journal of Applied and Computational Mathematics*, vol. 4, no. 1, pp. 42–50, 2018.
- [11] O. K. Kurkck, "A new numerical method for solving delay integral equation with variable bounds by using generalized Mott polynomials," *Eskisehir Technical University Journal of Science and Technology A-Applied Science and Engineering*, vol. 19, pp. 844–857, 2018.
- [12] K. Maleknejad, E. Hashemizadeh, and R. Ezzati, "A new approach to the numerical solution of Volterra integral equations by using Bernstein's approximation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 647–655, 2011.
- [13] A. Raza and A. Khan, "Haar wavelet series solution for solving neutral delay differential equations," *Journal of King Saud University-Science*, vol. 31, no. 4, pp. 1070–1076, 2019.
- [14] M. Ghasemia and M. Kajani, "Numerical solution of time-varying delay systems by Chebyshev wavelets," *Applied Mathematical Modelling*, vol. 35, pp. 523596–525244, 2011.
- [15] X. T. Wang, "Numerical solution of delay systems containing inverse time by hybrid functions," *Applied Mathematics and Computation*, vol. 173, no. 1, pp. 535–546, 2006.
- [16] O. Samadi and E. Tohidi, "The spectral method for solving systems of Volterra integral equations," *Journal of Applied Mathematics and Computing*, vol. 40, no. 1, pp. 477–497, 2012.
- [17] O. Samadi and E. Tohidi, "Numerical solution of two-dimensional Volterra integral equations by spectral Galerkin method," *Journal of Applied Mathematics and Bioinformatics*, vol. 1, pp. 159–174, 2011.
- [18] E. Tohidi, "Taylor matrix method for solving linear two-dimensional Fredholm integral equations with piecewise intervals," *Computational and Applied Mathematics*, vol. 34, no. 3, pp. 1117–1130, 2015.
- [19] C. Demko, K. Bertet, C. Faucher, J.-F. Viaud, and S. O. Kuznetsov, "Next priority concept: a new and generic algorithm computing concepts from complex and heterogeneous data," *Theoretical Computer Science*, vol. 845, pp. 1–20, 2020.
- [20] W. Luo, B. Gu, and G. Lin, "Communication scheduling in data gathering networks of heterogeneous sensors with data compression: algorithms and empirical experiments," *European Journal of Operational Research*, vol. 271, no. 2, pp. 462–473, 2018.

- [21] L. Ding and W. X. Zheng, "Consensus tracking in heterogeneous nonlinear multi-agent networks with asynchronous sampled-data communication," *Systems & Control Letters*, vol. 96, pp. 151–157, 2016.
- [22] D. Corral-Plaza, G. Ortiz, I. Medina-Bulo, and J. Boubeta-Puig, "MEdit4CEP-SP: a model-driven solution to improve decision-making through user-friendly management and real-time processing of heterogeneous data streams," *Knowledge-Based Systems*, vol. 213, p. 106682, 2021.
- [23] M. A. Alqarni, A. Khan, A. Khan et al., "Smartwatch-based legitimate user identification for cloud-based secure services," *Mobile Information System*, vol. 2018, Article ID 5107024, 2018.
- [24] M. Mazzara, I. Afanasyev, S. R. Sarangi, S. Distefano, V. Kumar, and M. Ahmad, "A reference architecture for smart and software-defined buildings," in *Proceedings of the IEEE International Conference on Smart Computing*, pp. 1–6, Washington, DC, USA, June 2019.
- [25] O. Sohaib, W. Hussain, M. Asif, M. Ahmad, and M. Mazzara, "A PLS-SEM neural network approach for understanding cryptocurrency adoption," *IEEE Access*, vol. 8, pp. 13138–13150, 2019.
- [26] I. Aziz and S. Siraj-Ul-Islam, "New algorithms for the numerical solution of nonlinear Fredholm and Volterra integral equations using Haar wavelets," *Journal of Computational and Applied Mathematics*, vol. 239, pp. 333–345, 2013.
- [27] R. Amin, K. Shah, M. Asif, and I. Khan, "A computational algorithm for the numerical solution of fractional order delay differential equations," *Applied Mathematics and Computation*, vol. 402, pp. 1–10, 2021.
- [28] R. Amin, B. Alshahrani, M. Mahmoud, A.-H. Abdel-Aty, K. Shah, and W. Deebani, "Haar wavelet method for solution of distributed order time-fractional differential equations," *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3295–3303, 2021.
- [29] R. Amin, I. Mahariq, K. Shah, M. Awais, and F. Elsayed, "Numerical solution of the second order linear and nonlinear integro-differential equations using Haar wavelet method," *Arab Journal of Basic and Applied Sciences*, vol. 28, no. 1, pp. 11–19, 2021.
- [30] M. M. Alqarni, R. Amin, K. Shah et al., "Solution of third order linear and nonlinear boundary value problems of integro-differential equations using Haar wavelet method," *Results in Physics*, vol. 25, p. 104176, 2021.
- [31] Y. Xuan, R. Amin, F. Zaman, Z. Khan, I. Ullah, and S. Nazir, "Second-order delay differential equations to deal the experimentation of internet of industrial things via Haar wavelet approach," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5551497, 2021.
- [32] R. Amin, K. Shah, M. Asif, and I. Khan, "Efficient numerical technique for solution of delay Volterra-Fredholm integral equations using Haar wavelet," *Heliyon*, vol. 6, pp. e05108–6, 2020.
- [33] R. C. Sekar and K. Murugesan, "Numerical solutions of delay Volterra integral equations using single-term Walsh series approach," *International Journal of Applied and Computational Mathematics*, vol. 16, pp. 1–13, 2016.
- [34] N. Sahin, S. Yuzbasi, and M. Gulsu, "A collocation approach for solving systems of linear Volterra integral equations with variable coefficients," *Computers & Mathematics with Applications*, vol. 62, pp. 755–769, 2011.

## Research Article

# An Efficient Decision Support System for the Selection of Appropriate Crowd in Crowdsourcing

**Yongjun Huang,<sup>1</sup> Shah Nazir ,<sup>2</sup> Jiyu Wu ,<sup>3</sup> Fida Hussain Khoso,<sup>4</sup> Farhad Ali ,<sup>2</sup> and Habib Ullah Khan <sup>5</sup>**

<sup>1</sup>*School of Journalism and Communication, Chongqing Normal University, Chongqing 401331, China*

<sup>2</sup>*Department of Computer Science, University of Swabi, Swabi, Pakistan*

<sup>3</sup>*School of Economics Lanzhou University, Lanzhou 730000, China*

<sup>4</sup>*Dawood University of Engineering and Technology, Karachi, Pakistan*

<sup>5</sup>*Department of Accounting & Information Systems, College of Business & Economics, Qatar University, Doha, Qatar*

Correspondence should be addressed to Shah Nazir; [snsahnzr@gmail.com](mailto:snsahnzr@gmail.com) and Jiyu Wu; [wujy318@sina.com](mailto:wujy318@sina.com)

Received 20 January 2021; Revised 5 April 2021; Accepted 28 May 2021; Published 10 June 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Yongjun Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crowdsourcing is a complex task-solving model that utilizes humans for solving organizational specific problems. For assigning a crowdsourced task to an online crowd, crowd selection is carried out to select appropriate crowd for achieving the task. The efficiency and effectiveness of crowdsourcing may fail if irrelevant crowd is selected for performing a task. Early decisions regarding selection of a crowd can ultimately lead to successful completion of tasks. To select most appropriate crowd from crowdsourcing, this paper presents a decision support system (DSS) for appropriate selection of crowd. The system has been implemented in the Superdecision tool by plotting hierarchy of goals, criteria, and alternatives. Various calculations have been done for performing the proposed research. Results of the study reveal that the proposed system is effective and efficient for selection of crowd in crowdsourcing by performing various pairwise computation of the study.

## 1. Introduction

Crowds are online people who have abilities to accomplish different types of tasks. These crowds may be newcomers who are accomplishing tasks for the first time or they may be experienced members who have completed various tasks previously. Crowdsourcing is a practice that acquires the services of huge group of people for obtaining information or completing a project [1]. It is internet-enabled collaborative activity that solves organizational problems by collecting the knowledge of online communities. The contributing editor Jeff Howe in June 2006 first used the word “crowdsourcing” in article “The Rise of Crowdsourcing” that was published in Wired magazine [2]. “Crowdsourcing is a type of participative online activity in which individual, institution, non-profit organization, or company proposes to a group of individuals of varying

knowledge, heterogeneity, and number, via a flexible open call, voluntary undertaking of a task. The undertaking of task, of variable complexity and modularity, and in which crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit [3]. The user will receive satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on type of activity undertaken” [4].

The applications of crowdsourcing are used widely for software testing [5], usability testing [6], machine learning processes [7], and decision making [8]. The productivity of large organizations has been enhanced by crowdsourcing [9]. The crowds comprise diverse-background participants who possess skills relevant to tasks and experience in the

field and have expertise in carrying out crowdsourced task or tackling complex problems [10]. Organizations are commonly using crowdsourcing to address challenges simultaneously with the large involvement of crowds. Crowdsourcing is an effective way to mitigate organizational dilemmas [11].

Crowdsourcing helps business organizations to recruit global, cheap, and skilled workers from different platforms [12, 13]. The new era of Web 3.0 is driven by innovations in ICT and social networking, and as a result organizational decision-making process has also been changed [14]. Modern corporations use the Internet to recruit a massive crowd. The Internet is a media for contact between crowds and businesses, and they work together using gadgets like iPads, mobile phones, laptops, wearable watches, etc. [15–17]. The crowds are recruited for completing different tasks from social or global societies [18]. By consuming small amount of management cost, time organization can achieve appropriate solutions with multiple crowd worker participation [19, 20]. As crowdsourcing is an online activity, it may entail certain risks, such as the announcement of tasks on websites and the selection of a suitable and qualified team [21]. The increased interest of crowdsourcing makes the selection of crowd workers a challenge. The crowded workers may be untrustworthy whose work can be followed by different errors. The choice of right and proper workers would boost the efficacy of crowdsourcing [22, 23]. Different business organization employs a suitable worker to complete task [24]. The following are the contributions of the proposed study:

- (i) A DSS is presented for the appropriate selection of crowd
- (ii) The proposed system is implemented in the Superdecision tool
- (iii) The hierarchy of goals, criteria, and alternatives is plotted with various pairwise comparisons to perform the proposed research
- (iv) Results of the study reveal that the proposed approach is effective and efficient

The organization of the paper is as follows. Section 2 presents the related work on the various aspects of the crowd and crowdsource concepts. Section 3 shows the details of the methodology with a description of the decision support system and the selection of features from literature. Results and discussion are given in Section 4. The paper is concluded in Section 5.

## 2. Literature Review

Crowdsourcing is an online process that could be linked to various challenges, such as crowd selection problem [21]. Various strategies, approaches, and models were presented in the past to address the crowd selection problem. Selection of crowd was based on the characteristics they possess which includes personal characteristics such as gender, age, qualification, education, language, and worker nationality; behavioral characteristics such as sociolects, left/right

handed, and personality traits; cognitive or perceptual characteristics that include a person's memory capacity, vision, hearing, or these may include skills, capabilities, past service, expertise, experience, and majors. Based on these characteristics, crowd are selected [25, 26]. A crowd targeting framework was implemented that automatically discovers and targets a specific crowd to improve data quality. The targeted crowd is selected by the worker characteristics such as nationality, education level, gender, and major or even personality test score and any other screening measures. Information gain that is a new characteristic measure for worker selection is also introduced. The framework selects workers using 3 main stages. The first is the probing stage in which the tasks are distributed to the whole population of crowd. Crowds are allowed to complete these tasks. The workers characteristics such as gender and location are also gathered from their profile for future use in this stage. The second is a discovery stage that is related to the discovery of the best workers, where unbiased worker samples of the entire crowd population are identified. The workers are evaluated using criteria such as good, bad, available, etc. The third is targeting stage in which the remaining and upcoming tasks are assigned to the discovered groups. The targeting stage improves the quality of data and increases budget performance [27].

Workers are selected by various organizations based on their capabilities for generating ideas or solving problems related to technology [28]. To allow a worker to participate in difficult tasks, an organization assesses the worker's ability [29]. Based on ability, a worker is selected [17]. For verifying workers ability, Borda ranking algorithm can be utilized [30]. Worker skills may also be an indicator for its selection as the skills reflect its ability to perform a task. The crowds are judged on the basis of their skills. Skills are, therefore, one of the main considerations for selection of right participants [31]. Workers possess various skills such as writing, IT, problem solving, process management, time management, communication, creative, e-skills, business thinking, and enterprise [32]. Skills' assessment or testing is used to assess various worker skills and these are helpful in the task matching processes. Organization offers certification that does certify workers posse's sufficient skills [33–35]. The certification is used for selection of workers [36]. The crowd who possess essential skills complete the task [33]. Trust is a major factor for consideration of workers for a task [23, 33].

Organization selects crowd workers for accomplishment of various task based on its trustworthiness [37]. For evaluating trust value "Trust-Based Access Control (TBAC)" model is utilized, and for decision concerning whether a worker is to be trusted or not, a discrete model was implemented [38]. Crowd trust was proposed that is a context-aware model for the evaluation of trust related to the type of task "TaTrust" and for the calculation of trust associated with task reward amount "RaTrust." For the selection of trustworthy workers with 2 context-aware trusts, "MOWS GA" that is an evolutionary algorithm and depends on NSGA-II was introduced. The dishonest workers can be identified using the crowd trust model [39]. A recruitment process was introduced in spatial mobile crowdsourcing that

automatically selects trustworthy workers by utilizing the services of IoT. A huge group of workers is reduced to potential trustworthy workers using Lovain Community detection algorithm, and the optimal set of crowd is selected by utilizing integer Linear program [40]. By utilizing approaches of machine learning, the prediction of trustworthiness was improved with the exploration of endorsement (interwork relationship) [41].

Workers may also be selected on the basis of their experience with tasks. Experience is considered as a crucial factor [31] for crowd selection. The crowd consists of huge masses of people and according to the level of experience best workers are selected [25, 42]. For selection of experienced participant for task, experience strategy is utilized [24]. Selection of crowd greatly relies on its expertise [24]. A crowd is selected based on its expertise level [43]. Only workers having requisite expertise are allowed to carry out the task [33]. For ensuring the workers expertise level filtering [28, 33] is performed. Workers are judged according to varied expertise using expertise-estimation approaches [44]. The task can considerably be performed by worker having expertise [45]. Qualified workers are judged by means of qualification tests and these assessments are superior filters for quality enhancement. The work quality can be controlled by conducting qualification tests. In these tests, a worker has to answer various questions provided by organizations. Workers must pass the qualification test before engaging with projects or tasks [46]. Workers are assessed based on their qualification level [33].

Profile based selection is also carried out for worker selection as the profile represents the personal features of the worker that can be directly observed. The profile contains worker details such as sex, age, education, and history of accomplished tasks [31, 47]. Exploiting workers' profiles would improve the assessment, assignments, and the quality of task [48]. Workers are responsible for maintaining and modifying their profiles for getting work from organizations [32]. For selection of workers based on workers profile personality based tool may be utilized [28]. Profile based approach was implemented for an effective selection of worker in crowdsourcing to reduce overhead time and budget by replacing an offline learning process with the online probing stage. This was done for the purpose of learning profile features and these features will be used by the online targeting algorithm for the selection of effective workers for different tasks [49]. The profile based selection of crowd can enhance the decision-making process of crowd selection [50].

### 3. Methodology

DSS is related to the discipline of information system area that supports and enhances the decision-making process of an organization [51]. It is difficult for decision makers to give preferences as high volumes of data regarding crowds are available. DSS is implemented for broadening the capabilities of human information and for enhancing the process of decision making when dealing with large amount of data [52]. Crowdsourcing can play a role in the organizational

decision-making process. A complex problem can easily be solved by crowd as they provide ideas, solicit opinions, give prediction, accumulate knowledge, etc. [53]. There is a lack of research which suggests a DSS for the selection of suitable crowds. Existing research studies were analyzed for the purpose of identifying the multifeatures of crowd. Table 1 represents these features. The multifeatures will be used by our DSS for the appropriate selection of crowd. Crowdsourcing activity entails three entities that are crowdsourcer/requestor which are organizations, individuals, or institutions who initiate the crowdsourcing process and seek out the ability of people for completing tasks which are shown in Figure 1 [68]; the crowd that consists of large group of people having enactive, cognitive, and perceptual abilities for solving tasks [69]; and the platform or market which is an online website or place where workers acquire and accomplish tasks [70].

The reason behind choosing the DSS for the proposed study was to consider the early decision of the crowd from the crowdsource. Various features of crowd were identified in the literature. Keeping in view the suitability of the crowd, the following key features were identified as the most suitable features from the literature. Table 1 shows the identified features of crowd based on literature.

*3.1. Experimental Setup.* The process of implementation and experimentation was done in the Superdecision software. The features were given as input to the software and then plotted as a hierarchy of goals, criteria, and alternatives. Figure 2 shows the process of making a hierarchy of the features along with the alternatives of crowds with the goal of selecting the crowd from the available options.

After plotting the features and crowd, the process of comparison was then done for each feature with respect to each crowd. For the information here only one comparison is shown. The same process is done for the comparison of all features and all crowds. Figure 3 graphically represents the process of comparison.

The values were given to each feature and then crowd. This process was done through the support of the tool. Figure 4 represents the graphical representation of the weights to each feature.

After assigning relevant weights to each feature and crowd, the process of comparison was done and the unweighted, weighted, and limit matrices were obtained for making the selection decision of crowd.

## 4. Results and Discussion

Crowdsourcing is a complex task-solving model to utilize the efforts of humans for solving organizational-specific issues. For assigning a crowdsourced task to an online crowd, the process of selecting a crowd is carried out to select a suitable crowd for attaining the given task. Making an early decision associated with the selection of the crowd can ultimately lead to successful completion of tasks. For selecting the best and right crowd from the crowdsourcing, this research presents a DSS for the appropriate selection of

TABLE 1: Features of crowd.

S. no.	Features	Citation
1	Professionals	[28]
2	Trustworthy	[23, 49, 54]
3	Skill	[10, 34, 55, 56]
4	Competent	[54, 57, 58]
5	Collaborative	[59, 60]
6	Decision maker	[61]
7	Qualified/educated	[17, 49, 62]
8	Problem solving	[34, 42, 43, 56, 63-67]
9	Experienced	[13, 18]

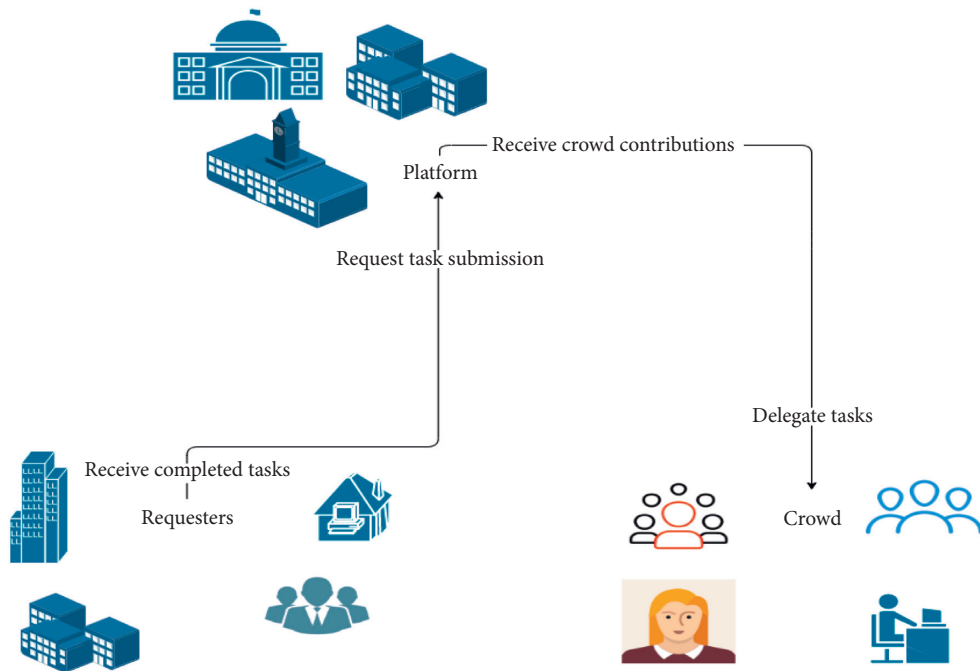


FIGURE 1: Entities of crowdsourcing.

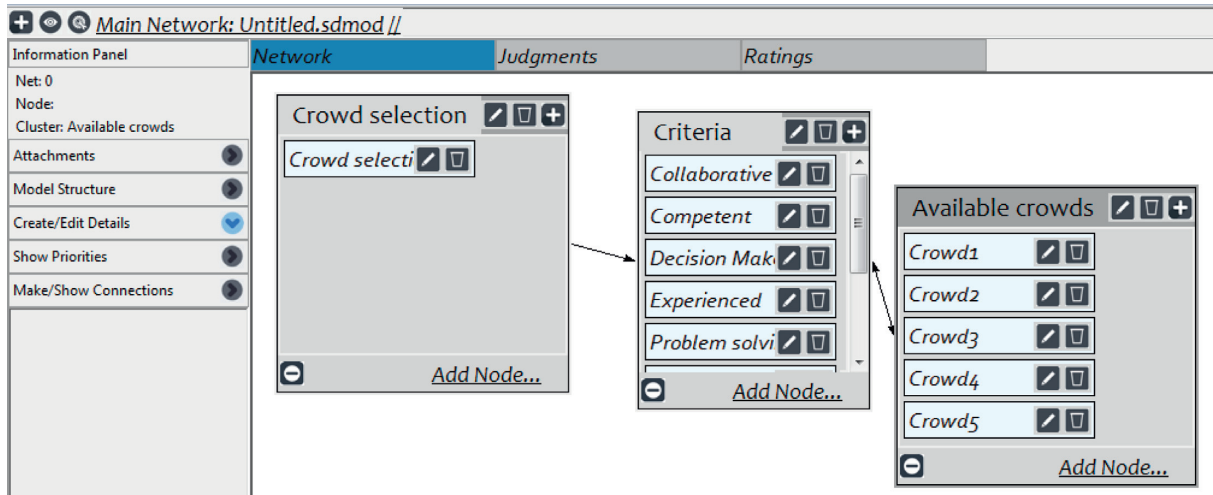


FIGURE 2: Hierarchy of features and crowds for selection of crowd.



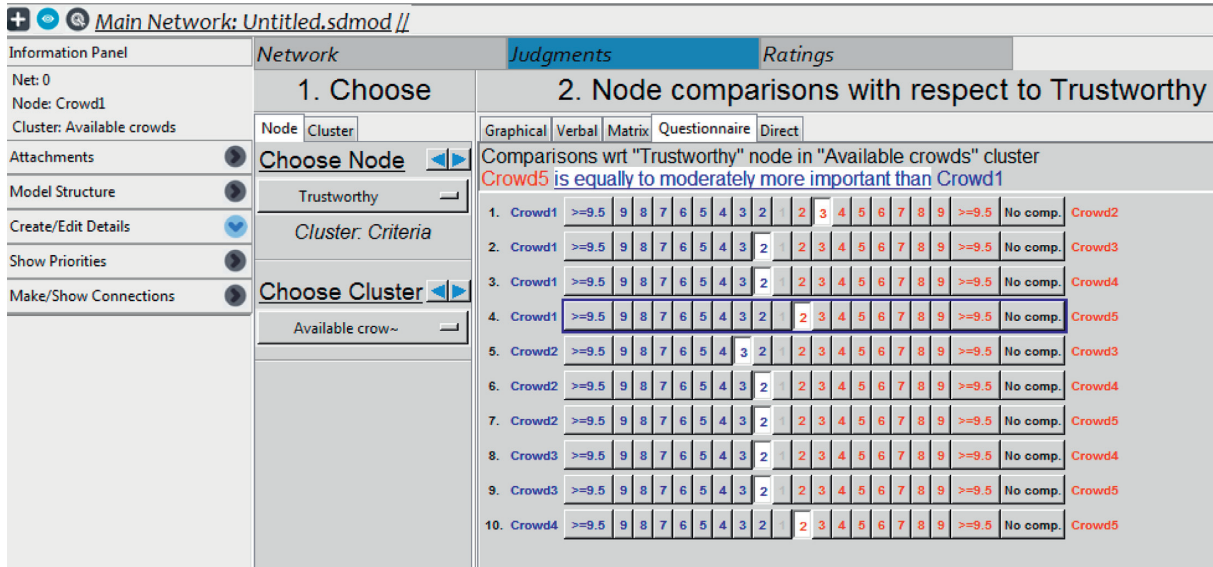


FIGURE 3: Proposed process of comparison.

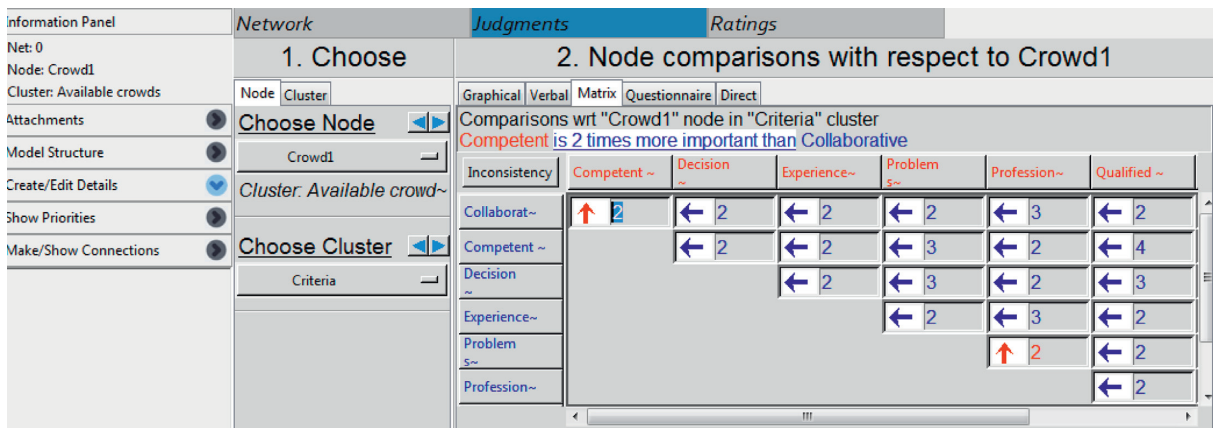


FIGURE 4: Assignment of weights to the feature.

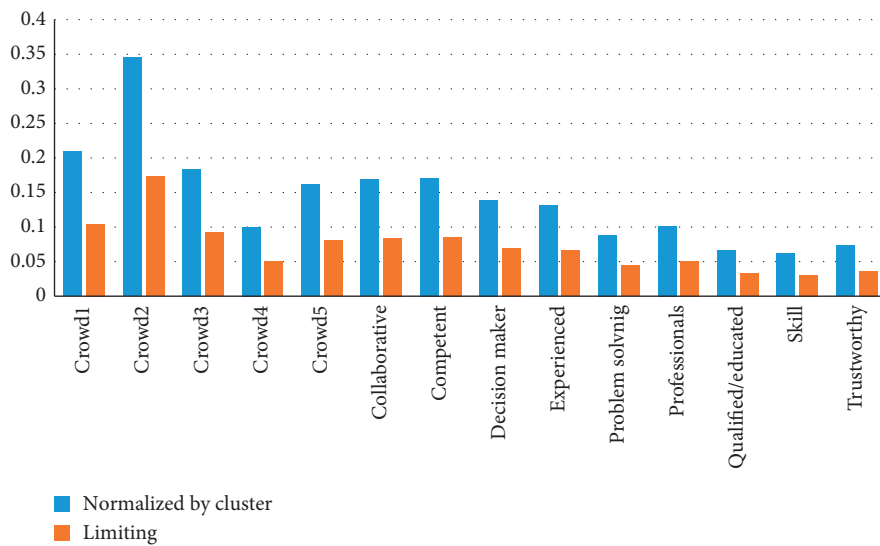


FIGURE 5: Normalization of priorities based on cluster and limiting.







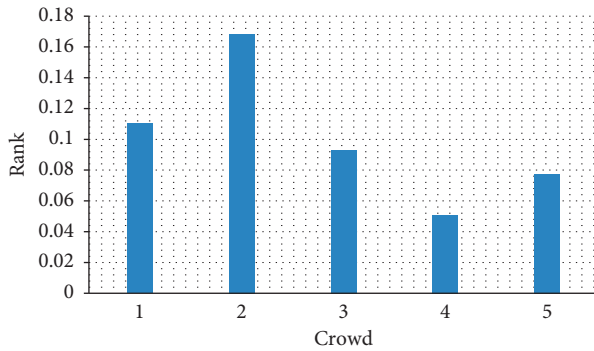


FIGURE 6: Ranking of crowds.

the crowd. The approach has been implemented in the Superdecision tool by plotting the hierarchy of goals, criteria, and alternatives. Various comparisons of the identified features with respect to crowds and crowds with respect to each feature were done. Relevant weights were given and after completion of the comparison process, different results were obtained. These results are shown in the form of tables and figures. Figure 5 graphically represents the priorities of features and available crowds based on normalization process by cluster and limiting.

As shown in Section 3, all the process of pairwise comparisons has been done in the software and for understanding only one representation is given as shown in Figures 3 and 4. The same processes have been done in the software for the rest of the attributes and alternatives. Once, all the comparisons process completed then all the normalized values of each criteria and alternatives are brought into unweighted and weighted super matrix. In unweighted matrix, the sum column values are greater than 1; then it is normalized again and then converted into weighted super matrix.

After the pairwise comparisons, all resulting comparisons for features and crowds were integrated into an unweighted matrix. The unweighted matrix is the collection of all pairwise comparisons done in the proposed research. Table 2 shows the unweighted matrix.

The unweighted matrix was then normalized for obtaining the weighted matrix. Table 3 represents the weighted matrix.

The weighted matrix was then converted to the limit matrix which is the final matrix for making the decision. The limit matrix was obtained by taking the power of the weighted matrix. Table 4 represents the limit matrix. From this matrix, the decision regarding the crowd can be made.

Figure 6 graphically shows the ranking of available crowds. Among the available alternatives of crowds, crowd2 has obtained the highest score which was considered as the highest priority, followed by crowd1, and so on. Therefore, from this figure, one can make decisions regarding the selection of the best crowd among the available alternatives.

## 5. Conclusion

Crowds are online people who have the capabilities to complete diverse types of tasks and projects. These crowds

may be new comers who are accomplishing tasks for first time or they may be experienced members who have already finished various tasks in preceding projects. Crowdsourcing is a composite task-solving approach utilizing humans for solving organizational explicit problems. For assessing the crowdsourced task with online crowd, the crowd selection is carried out for the selection of an optimal and appropriate crowd for achieving the task. Early and on-time decision associated with the selection of the crowd can eventually put forward the successful completion of tasks. To select the most appropriate crowd from the crowdsourcing, the present study endeavors to attempt and devise a DSS for the selection of crowd from the crowdsourcing. The proposed DSS has been executed in the Superdecision tool. In the given tool, the hierarchy of criteria, alternatives, and goal was defined and then a process of pairwise comparisons has been done. Each table of pairwise comparison process was normalized in order to achieve optimal results for the selection of appropriate crowd. The experimental results of the study show that the proposed DSS is efficient and effective for the appropriate selection of crowd in crowdsourcing. In the future, the applicability of the proposed DSS will be tried through various parameters against robustness of the system and its effectiveness will be checked for effective usage in the crowdsourcing projects.

## Data Availability

No data were used to support the study.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] N. Lazar, "The big picture: crowdsourcing your way to big data," *Chance*, vol. 32, 2019.
- [2] J. Howe, "The rise of crowdsourcing," 2006, <https://www.wired.com/2006/06/crowds/>.
- [3] E. Estellés-Arolas and F. González-Ladrón-de-Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information Science*, vol. 38, no. 2, pp. 189–200, 2012.
- [4] D. C. Brabham, *Crowdsourcing*, Mit Press, Cambridge, MA, USA, 2013.
- [5] M. Yan, H. Sun, and X. Liu, "iTest: testing software with mobile crowdsourcing," in *Proceedings of the 1st International Workshop on Crowd-Based Software Development Methods and Technologies*, pp. 19–24, Hong Kong, China, November 2014.
- [6] D. Liu, R. G. Bias, M. Lease, and R. Kuipers, "Crowdsourcing for usability testing," *Proceedings of the American Society for Information Science and Technology*, vol. 49, no. 1, pp. 1–10, 2012.
- [7] E. D. Simpson, S. Reece, M. Venanzi et al., "Language understanding in the wild: combining crowdsourcing and machine learning," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 992–1002, Florence, Italy, May 2015.
- [8] A. Slivkins and J. W. Vaughan, "Online decision making in crowdsourcing markets," *ACM SIGecom Exchanges*, vol. 12, no. 2, pp. 4–23, 2014.

- [9] A. Afuah and C. L. Tucci, "Crowdsourcing as a solution to distant search," *Academy of Management Review*, vol. 37, no. 3, pp. 355–375, 2012.
- [10] B. Morschheuser, J. Hamari, and A. Maedche, "Cooperation or competition-when do people contribute more? A field experiment on gamification of crowdsourcing," *International Journal of Human-Computer Studies*, vol. 127, pp. 7–24, 2019.
- [11] I. Dissanayake, J. Zhang, and B. Gu, "Virtual team performance in crowdsourcing contest: a social network perspective," in *Proceedings of the 2015 48th Hawaii International Conference on System Sciences*, vol. 5, pp. 4894–4897, Kauai, HI, USA, January 2015.
- [12] P. Shi, M. Zhao, W. Wang et al., "Best of both worlds: mitigating imbalance of crowd worker strategic choices without a budget," *Knowledge-Based Systems*, vol. 163, pp. 1020–1031, 2019.
- [13] K.-J. Stol, B. Caglayan, and B. Fitzgerald, "Competition-based crowdsourcing software development: a multi-method study from a customer perspective," *IEEE Transactions on Software Engineering*, vol. 45, no. 3, pp. 237–260, 2019.
- [14] D. R. Soriano, F. J. Garrigos-Simon, R. L. Alcamí, and T. B. Ribera, "Social networks and Web 3.0: their impact on the management and marketing of organizations," *Management Decision*, vol. 50, 2012.
- [15] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio, "Crowdsourcing: a review and suggestions for future research," *International Journal of Management Reviews*, vol. 20, no. 2, pp. 343–363, 2018.
- [16] M. Alsayyari and S. Alyahya, "Supporting coordination in crowdsourced software testing services," in *Proceedings of the 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 69–75, Bamberg, Germany, March 2018.
- [17] L. Zhai, H. Wang, and X. Li, "Optimal task partition with delay requirement in mobile crowdsourcing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5216495, 12 pages, 2019.
- [18] M. Alhamed and T. Storer, "Estimating software task effort in crowds," in *Proceedings of the 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 281–285, Cleveland, OH, USA, September 2019.
- [19] A. Dwarakanath, "CrowdBuild: a methodology for enterprise software development using crowdsourcing," in *Proceedings of the Second International Workshop on CrowdSourcing in Software Engineering*, Florence, Italy, May 2015.
- [20] C. Qiu, A. C. Squicciarini, B. Carminati, J. Caverlee, and D. R. Khare, "CrowdSelect: increasing accuracy of crowdsourcing tasks through behavior prediction and user selection," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, USA, May 2016.
- [21] E. Mourelatos and M. Tzagarakis, "An investigation of factors affecting the visits of online crowdsourcing and labor platforms," *NETNOMICS: Economic Research and Electronic Networking*, vol. 19, no. 3, pp. 95–130, 2018.
- [22] A. Dubey, G. Viridi, M. S. Kuriakose, and V. Arora, "Towards adopting alternative workforce for software engineering," in *Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pp. 16–23, August 2016, Orange County, CA, USA.
- [23] Y. Zhao, G. Liu, K. Zheng, A. Liu, Z. Li, and X. Zhou, "A context-aware approach for trustworthy worker selection in social crowd," *World Wide Web*, vol. 20, no. 6, pp. 1211–1235, 2017.
- [24] Q. Cui, J. Wang, G. Yang, M. Xie, Q. Wang, and M. Li, "Who should Be selected to perform a task in crowdsourced testing?" in *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, pp. 75–84, Turin, Italy, July 2017.
- [25] N. Leicht, I. Blohm, and J. M. Leimeister, "Leveraging the power of the crowd for software testing," *IEEE Software*, vol. 34, no. 2, pp. 62–69, 2017.
- [26] D. Archambault, H. Purchase, and T. Hoßfeld, "Evaluation in the crowd. crowdsourcing and human-centered experiments," in *Proceedings of the Dagstuhl Seminar 15481*, vol. 2017, Springer, Dagstuhl Castle, Germany, November 2015.
- [27] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 165–176, Seoul, South Korea, April 2014.
- [28] K. L. Jeffcoat, T. J. Eveleigh, and B. Tanju, "A conceptual framework for increasing innovation through improved selection of specialized professionals," *Engineering Management Journal*, vol. 31, no. 1, pp. 22–34, 2019.
- [29] F. Kamoun, D. Alhadidi, and Z. Maamar, "Weaving risk identification into crowdsourcing lifecycle," *Procedia Computer Science*, vol. 56, pp. 41–48, 2015.
- [30] B. Zhang, C. H. Liu, J. Lu et al., "Privacy-preserving QoI-aware participant coordination for mobile crowdsourcing," *Computer Networks*, vol. 101, pp. 29–41, 2016.
- [31] K. Abhinav, G. K. Bhatia, A. Dubey, S. Jain, and N. Bhardwaj, "TasRec: a framework for task recommendation in crowdsourcing," in *Proceedings of the 15th International Conference on Global Software Engineering*, pp. 86–95, Seoul, South Korea, June 2020.
- [32] S.-A. Barnes, A. Green, and M. de Hoyos, "Crowdsourcing and work: individual factors and circumstances influencing employability," *New Technology, Work and Employment*, vol. 30, no. 1, pp. 16–31, 2015.
- [33] A. Sari, A. Tosun, and G. I. Alptekin, "A systematic literature review on crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 153, pp. 200–219, 2019.
- [34] M. Modaresnezhad, L. Iyer, P. Palvia, and V. Taras, "Information technology (IT) enabled crowdsourcing: a conceptual framework," *Information Processing & Management*, vol. 57, no. 2, Article ID 102135, 2020.
- [35] Z. Peng, X. Gui, J. An, R. Gui, and Y. Ji, "TDSRC: a task-distributing system of crowdsourcing based on social relation cognition," *Mobile Information Systems*, vol. 2019, Article ID 7413460, 12 pages, 2019.
- [36] M. Christoforaki and P. G. Ipeirotis, "A system for scalable and reliable technical-skill testing in online labor markets," *Computer Networks*, vol. 90, pp. 110–120, 2015.
- [37] H. Amintoosi, S. S. Kanhere, and M. Allahbakhsh, "Trust-based privacy-aware participant selection in social participatory sensing," *Journal of Information Security and Applications*, vol. 20, pp. 11–25, 2015.
- [38] O. Folorunso and O. A. Mustapha, "A fuzzy expert system to Trust-Based Access Control in crowdsourcing environments," *Applied Computing and Informatics*, vol. 11, no. 2, pp. 116–129, 2015.
- [39] B. Ye, Y. Wang, and L. Liu, "Crowd trust: a context-aware trust model for worker selection in crowdsourcing environments," in *Proceedings of the 2015 IEEE international conference on web services*, pp. 121–128, IEEE, New York, NY, USA, July 2015.
- [40] A. Khanfor, A. Hamrouni, H. Ghazzai, Y. Yang, and Y. Massoud, "A trustworthy recruitment process for spatial

- mobile crowdsourcing in large-scale social IoT,” in *Proceedings of the 2020 IEEE Technology & Engineering Management Conference*, Novi, MI, USA, June 2020.
- [41] C. Wu, T. Luo, F. Wu, and G. Chen, “An endorsement-based reputation system for trustworthy crowdsourcing,” in *Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 89-90, IEEE, Hong Kong, China, April 2015.
- [42] H. J. Pongratz, “Of crowds and talents: discursive constructions of global online labour,” *New Technology, Work and Employment*, vol. 33, no. 1, pp. 58–73, 2018.
- [43] N. Luz, N. Silva, and P. Novais, “A survey of task-oriented crowdsourcing,” *Artificial Intelligence Review*, vol. 44, no. 2, pp. 187–213, 2015.
- [44] A. Moayedikia, W. Yeoh, K.-L. Ong, and Y. L. Boo, “Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach,” *Decision Support Systems*, vol. 122, Article ID 113065, 2019.
- [45] J. T. Bush and R. M. Balven, “Catering to the crowd: an HRM perspective on crowd worker engagement,” *Human Resource Management Review*, vol. 31, 2018.
- [46] F. R. Assis Neto and C. A. S. Santos, “Understanding crowdsourcing projects: a systematic review of tendencies, workflow, and quality management,” *Information Processing & Management*, vol. 54, no. 4, pp. 490–506, 2018.
- [47] X. Peng, “CrowdService: serving the individuals through mobile crowdsourcing and service composition,” in *Proceedings of the 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 214–219, Singapore, September 2016.
- [48] J. Mtsweni, E. K. Ngassam, and L. Burge, “A profile-aware microtasking approach for improving task assignment in crowdsourcing services,” in *Proceedings of the 2016 IST-africa Week Conference*, pp. 1–10, IEEE, Durban, South Africa, May 2016.
- [49] T. Awwad, N. Bennani, K. Ziegler, V. Sonigo, L. Brunie, and H. Kosch, “Efficient worker selection through history-based learning in crowdsourcing,” in *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, pp. 923–928, Turin, Italy, July 2017.
- [50] A. Smirnov, A. Ponomarev, and N. Shilov, “Hybrid crowd-based decision support in business processes: the approach and reference model,” *Procedia Technology*, vol. 16, pp. 376–384, 2014.
- [51] D. Arnott and G. Pervan, “A critical analysis of decision support systems research revisited: the rise of design science,” *Enacting Research Methods in Information Systems*, vol. 3, pp. 43–103, 2016.
- [52] M. Rhyne and I. Blohm, “Combining collective and artificial intelligence: towards a design theory for decision support in crowdsourcing,” in *Proceedings of the Twenty-Fifth European Conference on Information Systems (ECIS)*, Guimarães, Portugal, June 2017.
- [53] C.-M. Chiu, T.-P. Liang, and E. Turban, “What can crowdsourcing do for decision support?” *Decision Support Systems*, vol. 65, pp. 40–49, 2014.
- [54] F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab, “Modelling cognitive bias in crowdsourcing systems,” *Cognitive Systems Research*, vol. 58, pp. 1–18, 2019.
- [55] N. Mazlan, S. S. Syed Ahmad, and M. Kamalrudin, “Volunteer selection based on crowdsourcing approach,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 743–753, 2018.
- [56] I. Dissanayake, N. Mehta, P. Palvia, V. Taras, and K. Amoako-Gyampah, “Competition matters! Self-efficacy, effort, and performance in crowdsourcing teams,” *Information & Management*, vol. 56, no. 8, Article ID 103158, 2019.
- [57] O. Tokarchuk, R. Cuel, and M. Zamarian, “Analyzing crowd labor and designing incentives for humans in the loop,” *IEEE Internet Computing*, vol. 16, no. 5, pp. 45–51, 2012.
- [58] U. Gadiraju, G. Demartini, R. Kawase, and S. Dietze, “Crowd anatomy beyond the good and bad: behavioral traces for crowd worker modeling and pre-selection,” *Computer Supported Cooperative Work (CSCW)*, vol. 28, no. 5, pp. 815–841, 2019.
- [59] A. L. Zanatta, L. Machado, and I. Steinmacher, “Competence, collaboration, and time management: barriers and recommendations for crowdworkers,” in *Proceedings of the 2018 IEEE/ACM 5th International Workshop on Crowd Sourcing in Software Engineering (CSI-SE)*, pp. 9–16, Gothenburg, Sweden, May 2018.
- [60] G. Montelisciani, D. Gabelloni, G. Tazzini, and G. Fantoni, “Skills and wills: the keys to identify the right team in collaborative innovation platforms,” *Technology Analysis & Strategic Management*, vol. 26, no. 6, pp. 687–702, 2014.
- [61] T. D. LaToza, W. Ben Towne, A. van der Hoek, and J. D. Herbsleb, “Crowd development,” in *Proceedings of the 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pp. 85–88, San Francisco, CA, USA, May 2013.
- [62] L. Machado, R. Prikladnicki, F. Meneguzzi, C. R. B. d. Souza, and E. Carmel, “Task allocation for crowdsourcing using AI planning,” in *Proceedings of the 2016 IEEE/ACM 3rd International Workshop on Crowd Sourcing in Software Engineering (CSI-SE)*, pp. 36–40, Austin, TX, USA, May 2016.
- [63] X. Wang, H. J. Khasraghi, and H. Schneider, “Towards an understanding of participants’ sustained participation in crowdsourcing contests,” *Information Systems Management*, vol. 37, no. 3, pp. 213–226, 2019.
- [64] J. Lee and D. Seo, “Crowdsourcing not all sourced by the crowd: an observation on the behavior of Wikipedia participants,” *Technovation*, vol. 55-56, pp. 14–21, 2016.
- [65] E. Schenk, C. Guittard, and J. Pénin, “Open or proprietary? Choosing the right crowdsourcing platform for innovation,” *Technological Forecasting and Social Change*, vol. 144, pp. 303–310, 2019.
- [66] S. Standing and C. Standing, “The ethical use of crowdsourcing,” *Business Ethics: A European Review*, vol. 27, no. 1, pp. 72–80, 2018.
- [67] G. D. Saxton, O. Oh, and R. Kishore, “Rules of crowdsourcing: models, issues, and systems of Control,” *Information Systems Management*, vol. 30, no. 1, pp. 2–20, 2013.
- [68] M. Hosseini, K. Phalp, J. Taylor, and R. Ali, “The four pillars of crowdsourcing: a reference model,” in *Proceedings of the 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–12, IEEE, Marrakesh, Morocco, May 2014.
- [69] T. Erickson, “Some thoughts on a framework for crowdsourcing,” in *Proceedings of the Workshop on Crowdsourcing and Human Computation*, pp. 1–4, Washington, DC, USA, July 2011.
- [70] S. Faradani, B. Hartmann, and P. G. Ipeirotis, “What’s the right price? pricing tasks for finishing on time,” in *Proceedings of the Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, Citeseer, San Francisco, CA, USA, August 2011.

## Research Article

# Vibration Reliability Analysis of Drum Brake Using the Artificial Neural Network and Important Sampling Method

Zhou Yang <sup>1</sup>, Unsong Pak,<sup>1,2</sup> and Cholu Kwon<sup>1,3</sup>

<sup>1</sup>School of Mechanical Engineering & Automation, Northeastern University, Shenyang 110819, China

<sup>2</sup>Pyongyang Transportation University, Pyongyang 999093, Democratic People's Republic of Korea

<sup>3</sup>Institute of Mechanical Engineering State Academy of Sciences, Pyongyang 999093, Democratic People's Republic of Korea

Correspondence should be addressed to Zhou Yang; yangzhou@mail.neu.edu.cn

Received 22 February 2021; Revised 29 April 2021; Accepted 13 May 2021; Published 25 May 2021

Academic Editor: Adil Mehmood Khan

Copyright © 2021 Zhou Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This research aims to evaluate the calculation accuracy and efficiency of the artificial neural network-based important sampling method (ANN-IS) on reliability of structures such as drum brakes. The finite element analysis (FEA) result is used to establish the ANN sample in ANN-based reliability analysis methods. Because the process of FEA is time-consuming, the ANN sample size has a very important influence on the calculation efficiency. Two types of ANNs used in this study are the radial basis function neural network (RBF) and back propagation neural network (BP). RBF-IS and BP-IS methods are used to conduct reliability analysis on training samples of three different sizes, and the results are compared with several reliability analysis methods based on ANNs. The results show that the probability of failure of the RBF-IS method is closer to that of the Monte-Carlo simulation method (MCS) than those of other methods (including BP-IS). In addition, the RBF-IS method has better calculation efficiency than the other methods considered in this study. This research demonstrates that the RBF-IS method is well suited to structure reliability problems.

## 1. Introduction

Since brakes serve the essential purpose of slowing down and stopping vehicles, the reliability of brakes significantly impacts vehicle safety. Due to their advantages of large braking torque and low cost, drum brakes are widely used in automobiles [1–3]. Resonance occurs when the driving frequency is close to the natural frequency of the brake and can cause fatigue damage to the brakes and other components, potentially leading to failure of the braking system. Therefore, research on vibration reliability analysis of brake component is necessary.

Over the years, many scholars have contributed to significant research on structure reliability analysis. The most common methods used for reliability analysis include the first-order reliability method (FORM), second-order reliability method (SORM), and MCS. FORM [4–6] and SORM [6–8] of the approximation method are widely used for the reliability analysis due to the simplicity and efficiency

of these methods. In addition, because the MCS method is very simple and easy to program, it is widely used for the reliability analysis of the engineering structures. However, this method requires a significant computation effort for low probability of failure (POF) problem [9]. In order to address this issue, several methods have been proposed to reduce computation effort, such as the IS [10, 11], directional importance sampling simulation method [12], and subset simulation method [13, 14].

Since the limit state function (LSF) of most engineering structures is generally either nonlinear or implicit, the FEA method is required to compute structure responses, such as deformations, frequencies, and stresses. FEA is often combined with reliability analysis methods to conduct the engineering structure reliability analysis. While it is easy to combine FEA with the MCS method to conduct reliability analysis, this method is computationally expensive. In recent years, metamodeling techniques have been developed to overcome this issue, such as the model tree (MT),



evolutionary polynomial regression (EPR), multivariate adaptive regression spline (MARS), gene expression programming (GEP) [15], response surface method (RSM) [16–18], support vector machine [19, 20], kriging surrogate model [21–24], and ANN [25–32]. Metamodeling techniques are adopted to establish the approximate models, which can replace the original implicit LSF. The ANN is widely used to solve reliability problems related to practical engineering structure. In this setup, FEA is used to obtain training sample of the ANN, the random variables of the structure are used for ANN input, the structure response is used for ANN output, and explicit form of the function relationship between the structure response and random variables can be established. The BP-based MCS method is used to predict the structure reliability, and the result is compared with three reliability analysis methods including the traditional MCS, polynomial-FOSM, and BP-based advanced first-order second moment method (AFOSM) [27]. The combination of an adaptive RBF metamodeling technique and a FORM proposes a new reliability analysis method for the practical tunnel engineering problems [28]. The uniform design method is used to improve the disadvantage of the conventional ANN-based RSM method [29]. The development and use of ANN-based reliability analysis methods is surveyed to propose the improvement techniques of the ANN training set [30]. Two ANNs (including a multilayer perceptron network and RBF) based several reliability analysis methods is used to conduct a reliability analysis of the laminated composite structures [31]. In the ANN-based reliability analysis method, the most of the calculation time is spent conducting the FEA, and the ANN sample data size has a significant influence on the calculation efficiency of the ANN-based reliability analysis method.

This study evaluates the calculation accuracy and efficiency of the ANN-IS method on reliability of structures. Using two numeric verification problems and a drum brake vibration reliability problem as examples, the RBF-IS method and BP-IS method are used to conduct the reliability analysis on the training samples equal to 100, 200, and 300 units. The calculation results are compared with several methods including MCS, AFOSM, BP-MCS, BP-AFOSM, RBF-MCS, and RBF-AFOSM.

This study is structured as follows: Section 2 presents a vibration reliability analysis based on the ANN-IS method, Section 3 provides the numerical analyses and results, and Section 4 presents the discussion and conclusions.

## 2. Vibration Reliability Analysis Based on the ANN-IS Method

*2.1. Basis Theory of the Vibration Reliability Analysis.* Resonance occurs when the driving frequency is close to the natural frequency of a structure, such that the structure enters a quasifailure state [33, 34]. According to reliability theory, the state function of the structure with vibration is defined as

$$G(f, w) = |f - w|, \quad (1)$$

where  $f$  is the natural frequency of the structure, and  $w$  is the driving frequency.

Based on the relationship between the structure natural frequency  $f$  and the driving frequency  $w$ , the state function of the quasifailure can be expressed as

$$G(f, w) = |f - w| \leq \varepsilon, \quad (2)$$

where  $\varepsilon$  is the specified range, in which the value is generally between 0.1 and 0.15 times the natural frequency of structure.

The mean value  $\mu_G$  and variance  $\sigma_G^2$  of the state function can be expressed as

$$\begin{aligned} \mu_G &= E(G) \\ &= |E(f) - E(w)|, \\ \sigma_G^2 &= \text{Var}(G) \\ &= \sigma_f^2 + \sigma_w^2. \end{aligned} \quad (3)$$

The quasifailure probability of the structure is expressed as

$$P_F = P(-\varepsilon \leq f - w \leq \varepsilon). \quad (4)$$

If the natural frequency and the driving frequency are independent normally distributed, then the probability of failure can be expressed as

$$P_F = \Phi\left(\frac{\varepsilon - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{-\varepsilon - \mu_G}{\sigma_G}\right). \quad (5)$$

*2.2. Importance Sampling Method.* The importance sampling method is a commonly improved MCS method. This method is widely used because of its high sampling efficiency and small variance. The basic theory of this method is that the importance sampling density function is adopted to replace the original probability density function. The POF base on the IS method is computed as

$$\begin{aligned} P_F &= \int_{R^m} I_F(X) f(X) dX \\ &= \int_{R^m} I_F(X) \frac{f(X)}{h(X)} h(X) dX \\ &= E\left[ I_F(X) \frac{f(X)}{h(X)} \right], \end{aligned} \quad (6)$$

where  $P_F$  is the POF of the reliability performance function,  $h(X)$  is the importance sampling density function,  $f(X)$  is the joint probability density function of random variables,  $R^m$  is the  $m$ -dimension variable space, and  $I_F(X)$  is the indicator function of failure domain.

$$I_F(X) = \begin{cases} 0, & G(X) \leq 0, \\ 1, & G(X) > 0. \end{cases} \quad (7)$$

The estimated value ( $\hat{P}_F$ ) of POF is computed as

$$\hat{P}_F = \frac{1}{N} \sum_{i=1}^N \left[ I_F(X_i) \frac{f(X_i)}{h(X_i)} \right]. \quad (8)$$

The mean value, variance, and coefficient of variation (COV) of the estimated failure probability are computed as

$$\begin{aligned} E[\hat{P}_F] &= E \left\{ \frac{1}{N} \sum_{i=1}^N \left[ I_F(X_i) \frac{f(X_i)}{h(X_i)} \right] \right\} \\ &= E \left[ I_F(X) \frac{f(X)}{h(X)} \right] \\ &= P_F. \end{aligned}$$

$$\begin{aligned} \text{Var}[\hat{P}_F] &= \frac{1}{N} \text{Var} \left[ I_F(X) \frac{f(X)}{h(X)} \right] \\ &\approx \frac{1}{N-1} \left\{ \frac{1}{N} \sum_{i=1}^N \left[ I_F(X_i) \frac{f^2(X_i)}{h^2(X_i)} \right] - \hat{P}_F^2 \right\}, \\ \text{Cov}[\hat{P}_F] &= \frac{\sqrt{\text{Var}[\hat{P}_F]}}{E[\hat{P}_F]}. \end{aligned} \quad (9)$$

**2.3. Artificial Neural Network.** When the reliability analysis method is used to for a complex structure, the responses of the structure need to be calculated by sophisticated numerical methods, such as FEA. However, this method is not suitable for reliability analysis of a complex structure because of its long computation time. In order to reduce the computation time, the ANN is commonly used to establish the LSF.

**2.3.1. BP Neural Network.** BP neural network is a widely used type of the ANN. The structure of a BP neural network is shown in Figure 1.

In this study, the BP model includes a hidden layer, an input layer, and an output layer, where the design parameters (material properties and structural dimensions) of drum brake are used for input data, and the natural frequency of the drum brake is used for output data. According to references [27, 35, 36], the number of neurons in the hidden layer is calculated as

$$p = \sqrt{n+m} + a, \quad (10)$$

where  $p$  is the neuron number of the hidden layer,  $n$  is the neuron number of the input layer,  $m$  is the neuron number of the output layer, and  $a$  is an adjusting constant; according to references [27, 35, 36], the range of  $a$  is the integer between 1 and 10.

The output model of the hidden layer neuron is computed as

$$Q_j = f \left( \sum IW_{ij} \times X_i - b_j \right), \quad (11)$$

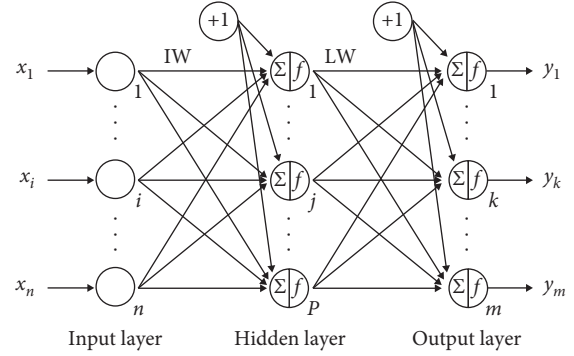


FIGURE 1: BP neural network.

where  $IW$  is the weight matrix between the input layer and the hidden layer,  $b$  is the threshold matrix,  $X$  is the vector of input variables, and  $f()$  is the neuron transfer function between the input layer and hidden layer. Logsig and tansig functions are widely used in the transfer function.

The output model of the output layer neuron is computed as

$$G_k = f \left( \sum LW_{jk} \times Q_j - b_k \right), \quad (12)$$

where  $LW$  is the weight matrix between the hidden layer and the output layer,  $b$  is the threshold matrix, and  $f$  is the neuron transfer function between the hidden layer and output layer. Purelin function is widely using in the transfer function.

If the difference between the model output value and the expected value exceeds the expected error value, gradient search technique is used to adjust the weight and threshold values, and the output value is recalculated. This process is repeated until the difference between the model output value and expected value is found to be less than the expected error value.

**2.3.2. RBF Neural Network.** RBF is a two-layer forward neural network. The RBF model includes the input layer, output layer, and one hidden layer. The RBF neural network structure model is shown in Figure 2.

RBF typically uses the Gaussian function. The activation function is computed as

$$G(X - a_j) = \exp \left( - \frac{1}{2 * \sigma^2} X - a_j^2 \right), \quad (13)$$

where  $X - a_j$  is the Euclidean norm,  $a_j$  is the Gaussian function center,  $\sigma$  is the Gaussian function variance, and  $j = 1, 2, \dots, p$  is the hidden layer neuron number.

The output function of the RBF model is computed as

$$y_k = \sum_{j=1}^p w_{jk} \exp \left( - \frac{1}{2 * \sigma^2} X - a_j^2 \right), \quad (14)$$

where  $X = (x_1, \dots, x_i, \dots, x_n)^T$  is the input parameters vector,  $k = 1, 2, \dots, m$  is the output layer neuron number, and  $w_{jk}$  is the weight matrix between the hidden layer and the output layer.

The variance of basis function is computed as

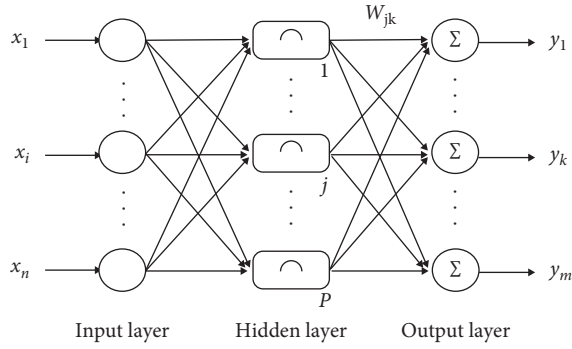


FIGURE 2: RBF neural network.

$$\sigma = \frac{1}{s} \sum_{k=1}^m d_k - y_k * a_j^2, \quad (15)$$

where  $d$  is the expected output value, and  $s$  is the sample number. RBF has a faster training speed.

**2.4. Vibration Reliability Analysis Method of the Drum Brake Based on ANN-IS.** The ANN-IS method is used to conduct the drum brake vibration reliability analysis based on the theory in Section 2.1 using ANN samples of different sizes. The steps of the method are given (for a detailed flowchart of the ANN-IS method, Figure 3.) Pseudocode of the proposed method can be viewed from the Supplementary Data 1.

- (1) Define the ANN sample data size  $N_0$ . In this study,  $N_0 = 100, 200, 300$ .
- (2) The design of the experiment method is used to generate ANN sample points, and FEA is adopted to obtain the natural frequency of the drum brake at the training sample points.
- (3) According to the relation criterion in which the difference between the natural frequency and driving frequency of the drum brake could not exceed the specific value, the ANN is used to establish the LSF. In this study, the specific value is defined 0.1 times of the natural frequency.
- (4) The optimization algorithm such as AFOSM is used to evaluate the most probable point  $X^*$  of the LSF
- (5) The important sampling density function  $h(X)$  is constructed with the most probable point as the sampling center
- (6) Define the sample size  $N_1 = 1E + 4$ . Start iteration  $k = 1$ , according to the important sampling density function, generate sample point of the input variables  $X_k = (x_1, x_2, \dots, x_n)$ .
- (7) Trained ANN is used to calculate the LSF value at the sample point  $X_k$ , and indicator function value  $I_F(X_k)$  calculates using equation (7). The Num1 and Num2 are calculated.
- (8) If the iteration number  $k = N_1$ , go to step 9. If not, iteration number  $k = k + 1$ , go to step 6.

- (9) The estimated value of the POF ( $\hat{P}_F$ ) and COV of the POF is calculated using equation (8) and equation (9).

### 3. Numerical Analyses and Results

To evaluate the calculation accuracy and efficiency of the proposed reliability analysis method, three examples are used to conduct the reliability analysis including the drum brake problem and two verification problems from the reference literature. In this study, BP and RBF neural networks are used to establish the LSF, and the same sample data are used for two neural networks. ANN sample data size is defined as  $N_0 = 100, 200, 300$ , and the ANN sample data are divided into training data and testing data of the ANN [35–37] where 90% of ANN sample data are used for training data, and 10% of ANN sample data is used for testing data. In order to evaluate the error of ANNs more accurately, the same testing sample data are used to compare the mean square errors (MSEs) of three ANN sample data ( $N_0 = 100, 200, 300$ ), in which the testing sample data have 10 sample points.

**3.1. Case 1: First Verification Problem.** The LSF with two random variables is as follows [17]:

$$G(X) = \exp(0.2x_1 + 1.4) - x_2, \quad (16)$$

where  $X = (x_1, x_2)$  is the random variables of standard normal distribution.

ANN sample data sizes are defined such as  $N_0 = 100, 200, 300$ . ANN sample points are generated to calculate the original LSF values. The trained ANNs are used to establish the LSF under three training samples of difference sizes, respectively. The same training sample is used for both RBF and BP. Figure 4 shows the comparison between the original LSF values and the predicted LSF values by ANNs. The maximum error point of the testing sample is the third sample point.

The ANN sample sizes are 100, 200, and 300, the mean square errors (MSEs) of RBF are  $4.03E-7$ ,  $9.81E-10$ , and  $1.82E-10$ ; the MSEs of BP are  $6.99E-6$ ,  $1.37E-6$ , and  $4.44E-7$ , respectively. Figure 5 shows the comparison of the MSE between RBF and BP.

The result shows the MSE of RBF is smaller than that of BP. In addition, the MSE of RBF is already very small for an ANN sample size of 200. Define the number of samples, in this study,  $N_1 = 1E + 4$ . Two ANN-IS methods and several reliability analysis methods including AFOSM, IS, BP-MCS, BP-AFOSM, RBF-MCS, and RBF-AFOSM are used to calculate the reliability indices on training sample of three difference sizes. The calculated results are compared with the MCS method. The result is given in Tables 1–3 and Figure 6.

The POF of the MCS, AFOSM, and IS is  $3.58E-04$ ,  $4.05E-04$ , and  $3.58E-04$ , respectively. The POF of the RBF-IS method is  $3.59E-04$ ,  $3.59E-04$ , and  $3.58E-04$ , and the relative errors are 0.3739%, 0.1528%, and 0.0058%, respectively. The POF of the BP-IS method is  $3.61E-04$ ,  $3.52E-04$ , and  $3.56E-04$ , and relative errors are 0.8292%,

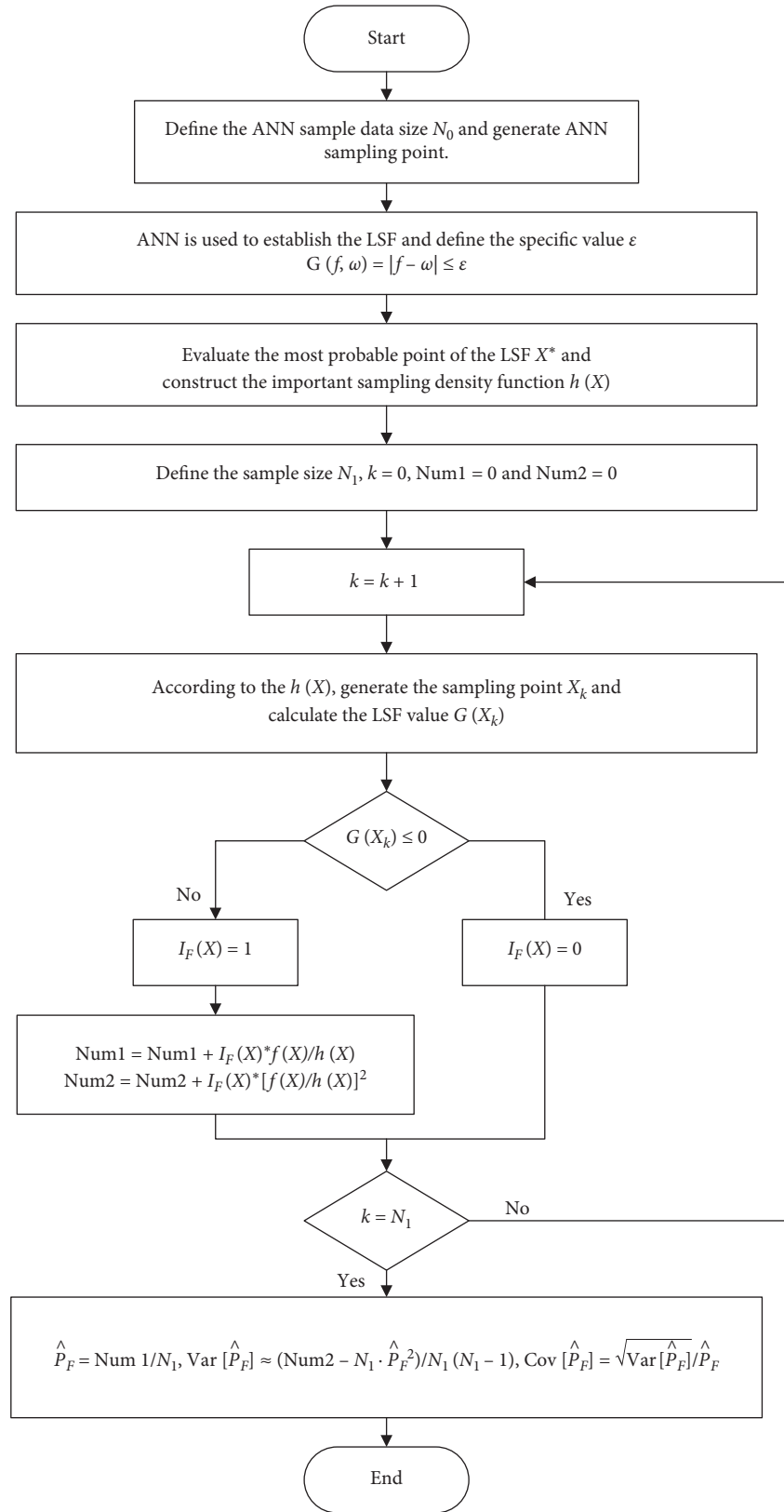


FIGURE 3: The flowchart of reliability analysis of the drum brake based on the ANN-IS method.

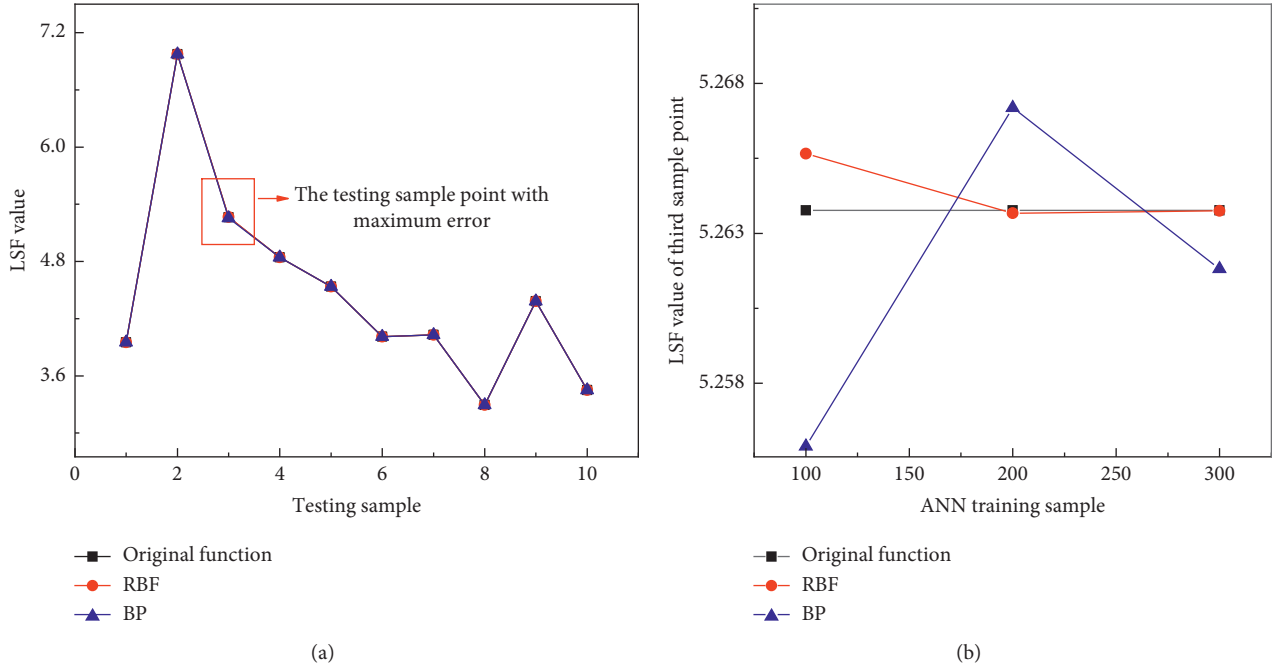


FIGURE 4: Comparison between the original LSF values and the predicted LSF values by ANNs. (a) The LSF value at testing sample for an ANN sample size of 100. (b) The LSF value of the third testing sample point relative to change in the ANN sample data size.

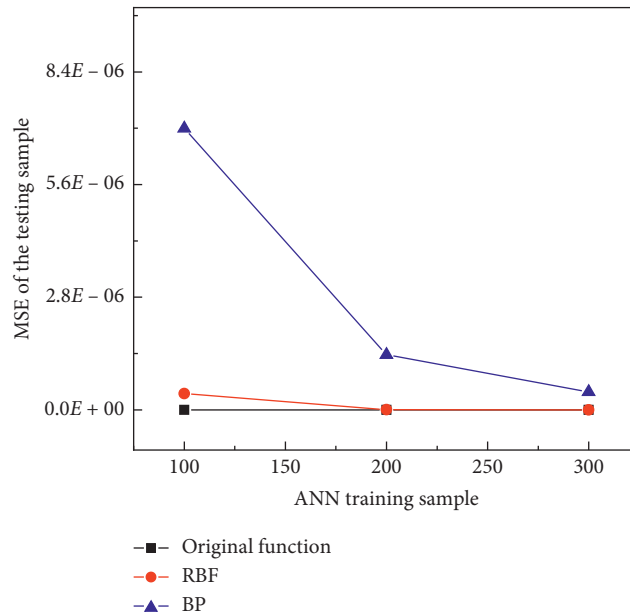


FIGURE 5: Comparison the change of MSE between RBF and BP as the ANN sample data size change.

1.5901%, and 0.4791%, respectively. The results show that the POF of the RBF-IS method is closer to that of MCS than that of the BP-IS method. In addition, the used of number of samples for RBF-IS is smaller than the ANN-based MCS method, so that the RBF-IS method has better calculation efficiency than the ANN-based MCS method.

3.2. *Case 2: Second Verification Problem.* The LSF with random variables is as follows [8]; the mean and standard deviation of the random variables is given as Table 4.

$$G(X) = \left(1 - x_5 \frac{x_4 x_3}{x_6 x_2 x_7}\right) x_4 x_2 x_3 - x_1, \quad (17)$$

TABLE 1: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 100.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3834	$3.58 E-04$	$1 E+7$	—
AFOSM	3.3497	$4.05 E-04$	—	12.989
IS	3.3833	$3.58 E-04$	$1 E+4$	0.0146
BP-MCS	3.3751	$3.69 E-04$	$1 E+7$	3.0726
BP-AFOSM	3.3523	$4.01 E-04$	—	11.914
BP-IS	3.3811	$3.61 E-04$	$1 E+4$	0.8292
RBF-MCS	3.3919	$3.47 E-04$	$1 E+7$	3.0726
RBF-AFOSM	3.3531	$3.99 E-04$	—	11.617
RBF-IS	3.3824	$3.59 E-04$	$1 E+4$	0.3739

TABLE 2: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 200.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3834	$3.58 E-04$	$1 E+7$	—
AFOSM	3.3497	$4.05 E-04$	—	12.989
IS	3.3833	$3.58 E-04$	$1 E+4$	0.0146
BP-MCS	3.3803	$3.62 E-04$	$1 E+7$	1.1173
BP-AFOSM	3.3482	$4.07 E-04$	—	13.607
BP-IS	3.3878	$3.52 E-04$	$1 E+4$	1.5901
RBF-MCS	3.3872	$3.53 E-04$	$1 E+7$	1.3966
RBF-AFOSM	3.3501	$4.04 E-04$	—	12.804
RBF-IS	3.3829	$3.59 E-04$	$1 E+4$	0.1528

TABLE 3: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 300.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3834	$3.58 E-04$	$1 E+7$	—
AFOSM	3.3497	$4.05 E-04$	—	12.989
IS	3.3833	$3.58 E-04$	$1 E+4$	0.0146
BP-MCS	3.3826	$3.59 E-04$	$1 E+7$	0.2793
BP-AFOSM	3.3491	$4.05 E-04$	—	13.251
BP-IS	3.3847	$3.56 E-04$	$1 E+4$	0.4791
RBF-MCS	3.3841	$3.57 E-04$	$1 E+7$	0.2793
RBF-AFOSM	3.3497	$4.05 E-04$	—	13.001
RBF-IS	3.3834	$3.58 E-04$	$1 E+4$	0.0058

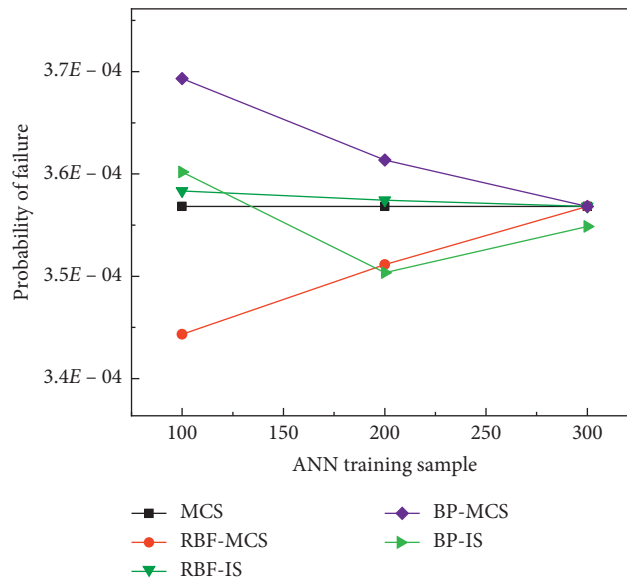


FIGURE 6: Comparison of the POF between ANN-IS (RBF-IS and BP-IS) and various other methods relative to ANN sample data size.

TABLE 4: The mean and standard deviation of the random variables.

Variables	Mean	Standard deviation	Distribution
$x_1$	0.01	0.003	Normal
$x_2$	0.3	0.015	Normal
$x_3$	360	36	Normal
$x_4$	$2.26 E - 4$	$1.13 E - 5$	Normal
$x_5$	0.5	0.05	Normal
$x_6$	0.12	0.006	Normal
$x_7$	40	6	Normal

where  $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$  is the random variable of the normal distribution.

For Case 2, we use the same ANN sample data sizes used in Case 1. The trained RBF and BP are used to establish the LSF on the training samples of three difference sizes, respectively. Figure 7 shows the comparison between the original LSF values and the predicted LSF values by ANNs.

The MSEs of RBF are  $1.72 E - 10$ ,  $2.55 E - 11$ , and  $1.49 E - 11$ ; the MSEs of BP are  $5.88 E - 10$ ,  $1.59 E - 10$ , and  $2.52 E - 11$ , respectively. Figure 8 shows the comparison of the MSE between RBF and BP.

The MSEs of RBF are the smaller than that of BP. Two ANN-IS methods and several reliability analysis methods including AFOSM, IS, BP-MCS, BP-AFOSM, RBF-MCS, and RBF-AFOSM are used to calculate the reliability indices on training samples of three difference sizes. The calculated results are compared with the MCS method. The result is given in Tables 5–7 and Figure 9.

The POF of the MCS, AFOSM, and IS is  $3.383 E - 04$ ,  $3.211 E - 04$ , and  $3.368 E - 04$ , respectively. The number of samples for MCS method is  $1.0 E + 7$  and that of ANN-based MCS is  $1.0 E + 6$ . The POF of the RBF-IS method is  $3.378 E - 04$ ,  $3.382 E - 04$ , and  $3.383 E - 04$ , and relative errors are 0.1546%, 0.0279%, and 0.0136%, respectively. The POF of the BP-IS method is  $3.356 E - 04$ ,  $3.391 E - 04$ , and  $3.381 E - 04$ , and relative errors are 0.8020%, 0.3194%, and 0.0645%, respectively. The results of two verification problems show that the POF of the RBF-IS method is closer to that of the MCS method than that of the BP-IS method. In addition, the number of samples used for the RBF-IS method is smaller than that used for the ANN-based MCS method, demonstrating that the RBF-IS method offers better calculation efficiency than does the ANN-based MCS method. Through two verification problems, the calculation accuracy and efficiency of the proposed method are verified. Therefore, the proposed method is applied to the reliability analysis of the drum brake.

**3.3. Application Problem: Vibration Reliability Analysis of the Drum Brake.** Resonance will occur when the driving frequency is close to the natural frequency of the brake, and the resonance will cause fatigue damage to the brakes and other components and that may lead to failure of the braking system. In this study, the vibration reliability problem of the drum brake including random parameters is researched. The material properties and structural dimensions of drum brakes are random due to the influence of technical conditions, manufacturing and installation errors,

manufacturing environment, and material characteristics. Assuming that the design parameters all obey the normal distribution, the coefficients of variation of the material properties and structural dimensions are determined to be 0.05 and 0.005. In this study, the density ( $D_d$ ), the elastic modulus ( $E_d$ ), and Poisson's ratio ( $P_d$ ) of the brake drum; the density ( $D_s$ ), the elastic modulus ( $E_s$ ), and Poisson's ratio ( $P_s$ ) of the friction plate; the density ( $D_p$ ), the elastic modulus ( $E_p$ ), and Poisson's ratio ( $P_p$ ) of the brake shoe; width of the brake drum ( $W_d$ ), the brake shoe ( $W_s$ ), and the friction plate ( $W_p$ ); and the initial angle ( $\theta_1$ ) and envelop angle ( $\theta_2$ ) of the friction plate are taken as design parameters.

In order to reduce the calculation workload, sensitivity analysis of the design parameters is conducted [15, 35]. Then the "Six sigma analysis" module in the Workbench software is used to conduct the sensitivity analysis on the design parameters of the drum brake. Design of the experiment method is used to generate 50 sample points, and then, modal analysis is conducted to obtain the influence degree of the design parameters on the natural frequency of drum brakes. Figure 10 shows the result of the sensitivity analysis.

Based on the sensitivity analysis results, in order to reduce the calculation workload, the parameters that have little influence on the natural frequency of the brake were screened. Finally, nine parameters were selected as design parameters. The mean and standard deviation of the design parameters is given in Table 8.

According to the theory of Section 2.1, the LSF of the vibration reliability on the drum brake is as

$$G(X) = |f(X) - \omega| < \varepsilon, \quad (18)$$

where  $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$  is the design parameters of the drum brake, and the design parameters are random;  $f(X)$  is the natural frequency of the drum brake;  $\omega$  is the driving frequency of the drum brake and is the 51 Hz in this study; and the specified range  $\varepsilon$  is 0.1 times of the natural frequency.

Figure 11 shows the FEA result at the mean value of the random parameters.

The ANN sample data sizes are defined such as  $N_0 = 100, 200, 300$ , and the design of the experiment method is used to generate training sample points, respectively. FEA is adopted to obtain the natural frequency of drum brake at the training sample points. The ANN sample data can be viewed from the Supplemental Data 2. Trained RBF and BP are used to establish the LSF, respectively. Figure 12 shows the comparison between the FEA values and the predicted LSF values by ANNs.

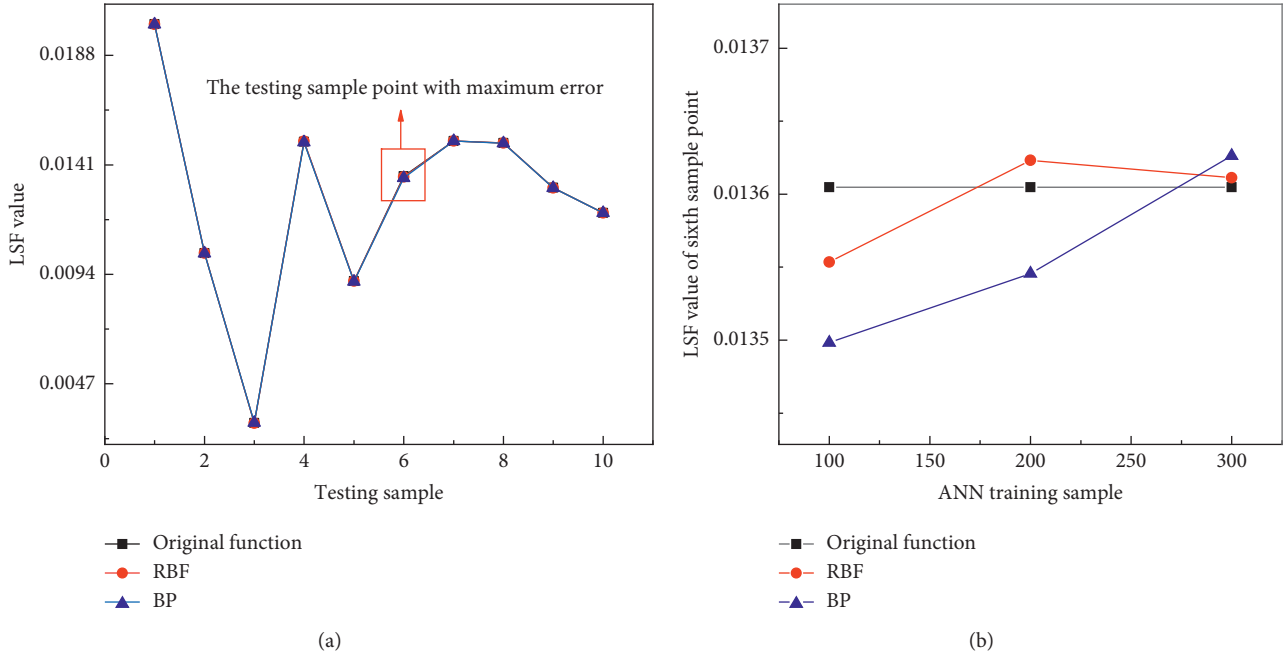


FIGURE 7: Comparison between the original LSF values and the predicted LSF values by ANNs. (a) The LSF value at testing sample for an ANN sample size of 100. (b) The LSF value of the sixth testing sample point relative to change in the ANN sample data size.

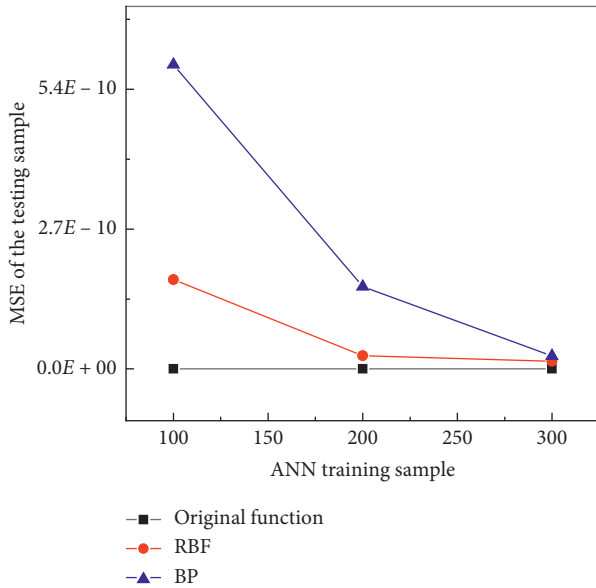


FIGURE 8: Comparison the change of MSE between RBF and BP as the ANN sample data size change.

The MSEs of RBF are  $2.76 E-5$ ,  $3.95 E-6$ , and  $1.79 E-6$ ; the MSEs of BP are  $1.05 E-4$ ,  $2.28 E-5$ , and  $4.19 E-6$ , respectively. Figure 13 shows the comparison the MSE between RBF and BP.

Two ANN-IS methods are used to calculate the reliability indices on training samples of three difference sizes and are compared with ANN-based several reliability analysis methods such as BP-MCS, BP-AFOSM, RBF-MCS, and RBF-AFOSM. The calculated results are given in Tables 9–11 and Figure 14.

TABLE 5: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 100.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3989	$3.383 E-04$	$1 E+7$	—
AFOSM	3.4131	$3.211 E-04$	—	5.0750
IS	3.4001	$3.368 E-04$	$1 E+4$	0.4570
BP-MCS	3.3970	$3.406 E-04$	$1 E+7$	0.6799
BP-AFOSM	3.4117	$3.228 E-04$	—	4.5786
BP-IS	3.4011	$3.356 E-04$	$1 E+4$	0.8020
RBF-MCS	3.3978	$3.390 E-04$	$1 E+7$	0.2069
RBF-AFOSM	3.4163	$3.174 E-04$	—	6.1779
RBF-IS	3.3993	$3.378 E-04$	$1 E+4$	0.1546

TABLE 6: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 200.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3989	$3.383 E-04$	$1 E+7$	—
AFOSM	3.4131	$3.211 E-04$	—	5.0750
IS	3.4001	$3.368 E-04$	$1 E+4$	0.4570
BP-MCS	3.3987	$3.385 E-04$	$1 E+7$	0.0591
BP-AFOSM	3.4125	$3.218 E-04$	—	4.8719
BP-IS	3.3983	$3.391 E-04$	$1 E+4$	0.2310
RBF-MCS	3.3991	$3.380 E-04$	$1 E+7$	0.0887
RBF-AFOSM	3.4127	$3.217 E-04$	—	4.9172
RBF-IS	3.3989	$3.382 E-04$	$1 E+4$	0.0279



TABLE 7: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 300.

Method	Reliability index	POF	Number of samples ( $N_1$ )	Relative error (%)
MCS	3.3989	$3.383 E - 04$	$1 E + 7$	—
AFOSM	3.4131	$3.211 E - 04$	—	5.0750
IS	3.4001	$3.368 E - 04$	$1 E + 4$	0.4570
BP-MCS	3.3991	$3.381 E - 04$	$1 E + 7$	0.0591
BP-AFOSM	3.4129	$3.213 E - 04$	—	5.0216
BP-IS	3.3991	$3.381 E - 04$	$1 E + 4$	0.0645
RBF-MCS	3.3989	$3.382 E - 04$	$1 E + 7$	0.0296
RBF-AFOSM	3.4129	$3.214 E - 04$	—	4.9997
RBF-IS	3.3989	$3.383 E - 04$	$1 E + 4$	0.0136

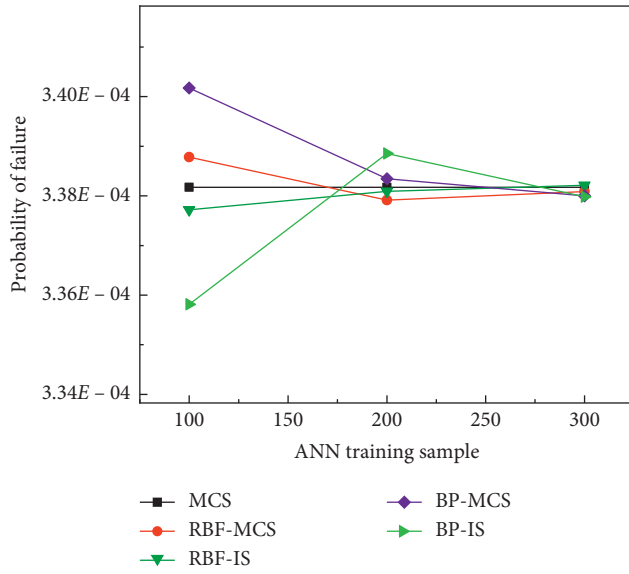


FIGURE 9: Comparison of the POF between ANN-IS (RBF-IS and BP-IS) and various other methods relative to ANN sample data size.

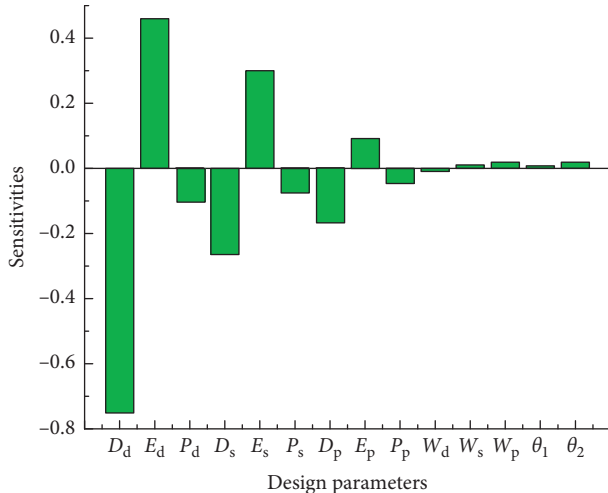


FIGURE 10: The result of sensitivity analysis.

TABLE 8: The mean and standard deviation of the design parameters.

Design parameters	Mean	Standard deviation	Distribution	Units
$x_1 (D_d)$	7330	366.5	Normal	$\text{kg/m}^3$
$x_2 (E_d)$	$1.48 E + 5$	7400	Normal	MPa
$x_3 (P_d)$	0.31	0.0155	Normal	—
$x_4 (D_s)$	2540	127	Normal	$\text{kg/m}^3$
$x_5 (E_s)$	$1.0 E + 5$	5000	Normal	MPa
$x_6 (P_s)$	0.25	0.0125	Normal	—
$x_7 (D_p)$	7100	355	Normal	$\text{kg/m}^3$
$x_8 (E_p)$	$7.252 E + 4$	3626	Normal	MPa
$x_9 (P_p)$	0.32	0.016	Normal	—

The POF of the RBF-IS method is  $1.184 E - 04$ ,  $1.185 E - 04$ , and  $1.184 E - 04$  and that of the BP-IS method is  $1.187 E - 04$ ,  $1.185 E - 04$ , and  $1.184 E - 04$ , respectively. The results show that the COV (POF) values of different methods are less than  $3 E - 2$ , and the calculated results from different reliability analysis methods are basically consistent. The POF of the RBF-IS method relative to change ANN sample data size is smaller than those of other methods. In addition, the ANN-IS method has better calculation efficiency than the ANN-based MCS method from number of samples. The number of samples of the ANN-based MCS method is  $1 E + 7$  and that of the ANN-IS method is  $1 E + 4$ .

#### 4. Discussion and Conclusions

This research evaluated the calculation accuracy and efficiency of the ANN-IS method on reliability of structures such as drum brakes. By analyzing the results of two numeric verification problems and the drum brake vibration reliability problem, we arrive at the following conclusions.

First, the result of the RBF-IS method is closer to that of the MCS method than is the BP-IS method result. After reviewing the results of two numeric verification problems, the results calculated from the MCS method are found to be consistent with reference MCS results. The relative errors of the RBF-IS method for the two numeric verification problems are 0.0058% and 0.0136%, respectively, which are smaller than the results from other methods for the same ANN sample size.

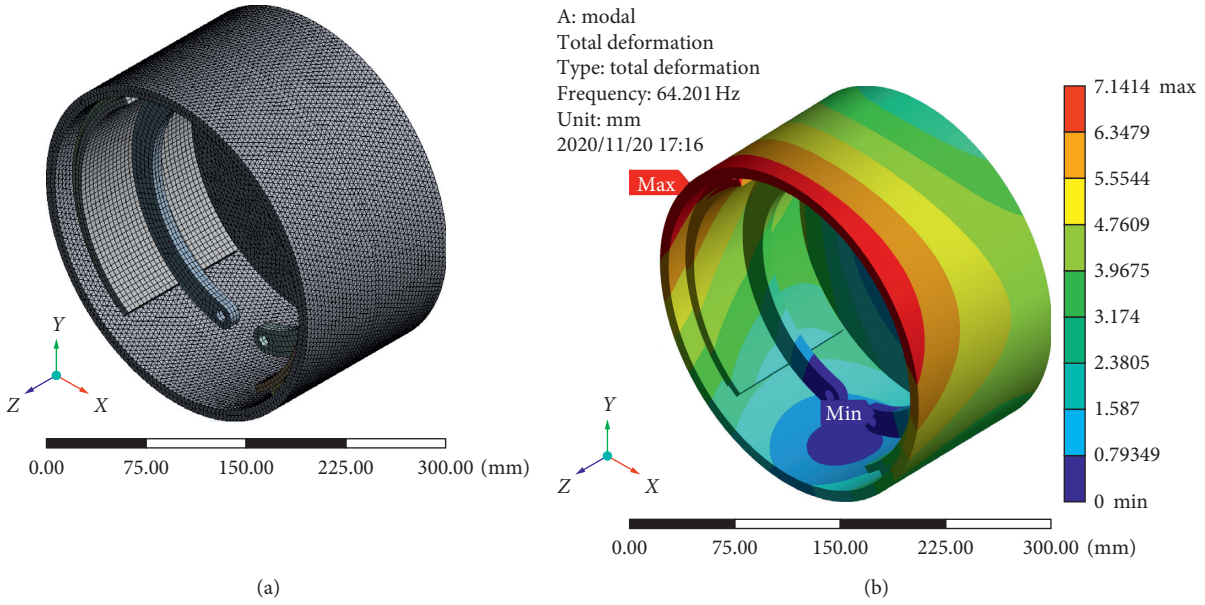


FIGURE 11: The FEA result of the drum brake model. (a) The mesh of the drum brake model. (b) The natural frequency of the drum brake.

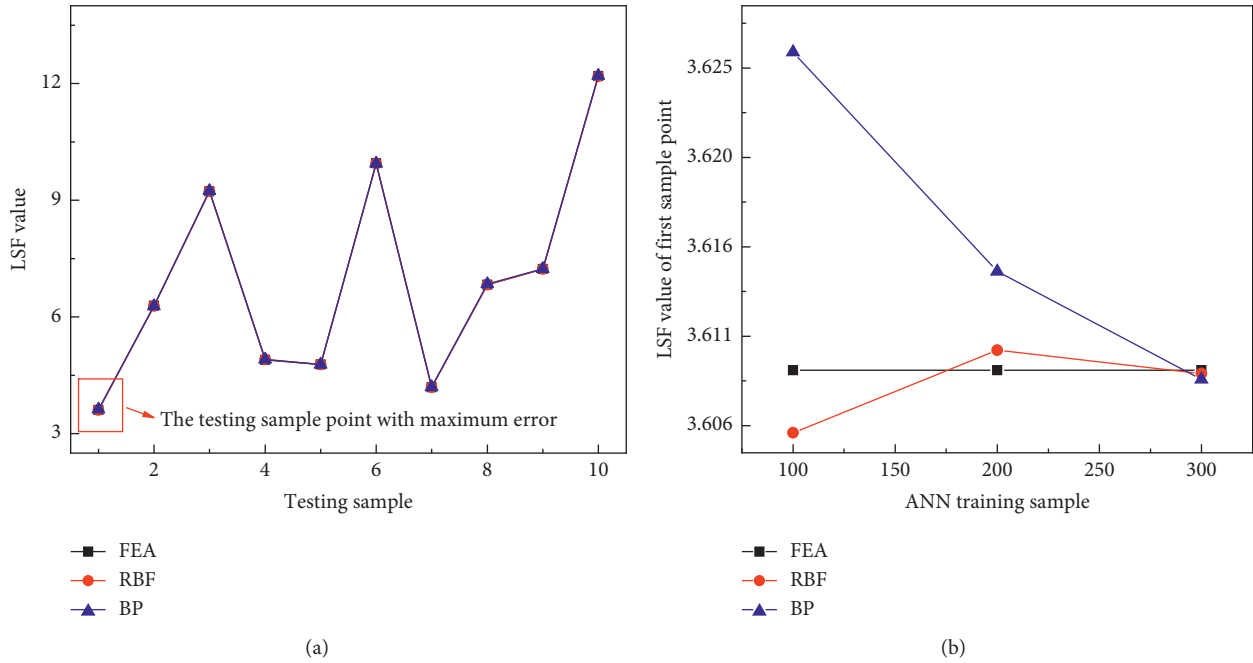


FIGURE 12: Comparison between the FEA values and the predicted LSF values by ANNs. (a) The LSF value at testing sample for an ANN sample size of 100. (b) The LSF value of the first testing sample point relative to change in the ANN sample data size.

Second, the RBF-IS method has a better calculation efficiency than the other methods used in this study. Through numeric analysis, find that use of the RBF-IS method results in smaller changes to the POF as the ANN sample size changes than when other methods are used. Additionally, the POF of the RBF-IS method at an ANN sample size of 200 is found to be closer to that of the MCS method than those of other methods (including BP-IS) at an ANN sample data size of 300. For reliability of structures

such as drum brake, the time required for the FEA increases with both the ANN sample size and the complexity of the structural model. Since the RBF-IS method can use a smaller ANN sample size than other methods while still main training good calculation accuracy, the required calculation time of the RBF-IS method is smaller than those of other methods (including BP-IS). Finally, the number of samples used by the RBF-IS method is  $1 E + 4$  and those used by the ANN-based MCS method is  $1 E + 7$ . This indicates that the

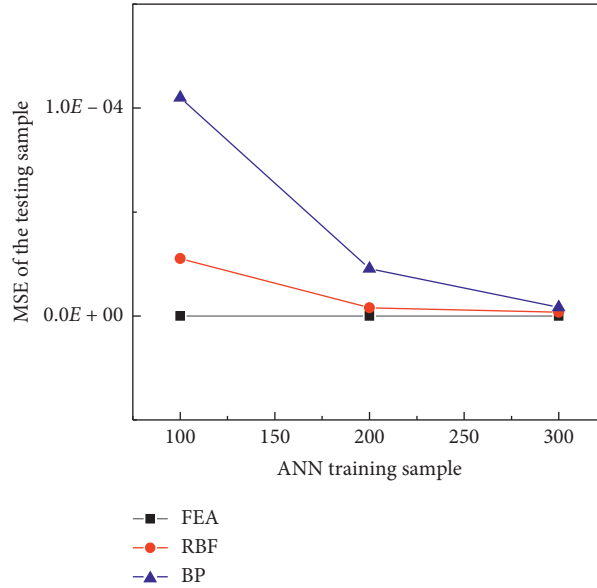


FIGURE 13: Comparison of the change of MSE between RBF and BP as the ANN sample data size change.

TABLE 9: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 100.

Method	Reliability index	POF	Number of samples ( $N_1$ )	COV (POF)
BP-MCS	3.6768	$1.181 E - 04$	$1 E + 7$	$2.910 E - 02$
BP-AFOSM	3.7002	$1.077 E - 04$	—	—
BP-IS	3.6756	$1.187 E - 04$	$1 E + 4$	$2.164 E - 02$
RBF-MCS	3.6768	$1.181 E - 04$	$1 E + 7$	$2.910 E - 02$
RBF-AFOSM	3.6779	$1.176 E - 04$	—	—
RBF-IS	3.6762	$1.184 E - 04$	$1 E + 4$	$2.176 E - 02$

TABLE 10: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 200.

Method	Reliability index	POF	Number of samples ( $N_1$ )	COV (POF)
BP-MCS	3.6766	$1.182 E - 04$	$1 E + 7$	$2.908 E - 02$
BP-AFOSM	3.7011	$1.073 E - 04$	—	—
BP-IS	3.6759	$1.185 E - 04$	$1 E + 4$	$2.230 E - 02$
RBF-MCS	3.6763	$1.183 E - 04$	$1 E + 7$	$2.901 E - 02$
RBF-AFOSM	3.6983	$1.085 E - 04$	—	—
RBF-IS	3.6760	$1.185 E - 04$	$1 E + 4$	$2.194 E - 02$

TABLE 11: Comparison of the reliability index and POF from different reliability analysis methods with an ANN sample size of 300.

Method	Reliability index	POF	Number of samples ( $N_1$ )	COV (POF)
BP-MCS	3.6761	$1.184 E - 04$	$1 E + 7$	$2.906 E - 02$
BP-AFOSM	3.7020	$1.069 E - 04$	—	—
BP-IS	3.6762	$1.184 E - 04$	$1 E + 4$	$2.218 E - 02$
RBF-MCS	3.6761	$1.184 E - 04$	$1 E + 7$	$2.906 E - 02$
RBF-AFOSM	3.7040	$1.061 E - 04$	—	—
RBF-IS	3.6761	$1.184 E - 04$	$1 E + 4$	$2.226 E - 02$

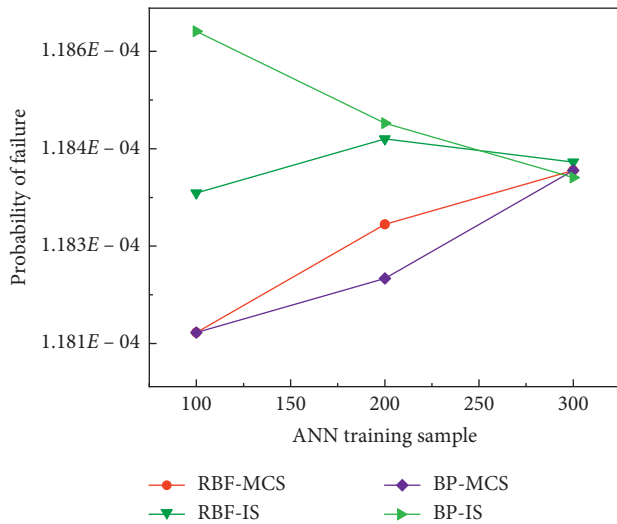


FIGURE 14: Comparison of the POF between ANN-IS (RBF-IS and BP-IS) and various other methods relative to ANN sample data size.

RBF-IS method offers a better calculation efficiency than the ANN-based MCS method.

The disadvantage of the RBF-IS method is that an optimization algorithm such as AFOSM is required to calculate the most probable point of the LSF, which requires the explicit form of the LSF to be known in advance.

Based on this research, find that the RBF-IS method is well suited to reliability problems such as the drum brake vibration reliability problem.

## Data Availability

The datasets used to support the findings of this research are included within the Supplementary Materials.

## Additional Points

In this study, according to the theory of the vibration reliability analysis, the ANN-IS method is used to conduct the vibration reliability analysis on the drum brake. In addition, the calculation results of two ANN-IS methods (RBF-IS and BP-IS) are compared for structure reliability problems. This research demonstrates that the RBF-IS method is well suited to structure reliability problems such as the drum brake vibration reliability problem.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

Z. Yang and U. S. Pak conceptualized the study; U. S. Pak and C. U. Kwon performed finite element analysis; U. S. Pak performed reliability analysis; Z. Yang, U. S. Pak, and C. U. Kwon reviewed and edited the article. All authors have read and agreed to the published version of the manuscript.

## Acknowledgments

The authors are grateful for the support from Chinese National Natural Science Foundation (U1710119 and U1708254) and Fundamental Research Funds for the Central Universities (N2003022).

## Supplementary Materials

Supplementary Data 1. Pseudocode of the proposed method. Supplementary Data 2. ANN sample data and testing data of the drum brake. Table 1. FEA results of drum brake when ANN sample data size is 100. Table 2. FEA results of drum brake when ANN sample data size is 200. Table 3. FEA results of drum brake when ANN sample data size is 300. Table 4. FEA results of ANN testing sample data. (*Supplementary Materials*)

## References

- [1] Y. Zhang, H. Zhang, and C. Lu, "Study on parameter optimization design of drum brake based on hybrid cellular multiobjective genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, Article ID 734193, 23 pages, 2012.
- [2] M. Zhou, Y. Wang, and Q. Huang, "Study on the stability of drum brake non-linear low frequency vibration model," *Archive of Applied Mechanics*, vol. 77, no. 7, pp. 473–483, 2007.
- [3] I. Ahmed, "Modeling of vehicle drum brake for contact analysis using ansys," *SAE Technical Paper Series*, vol. 7, 2012.
- [4] A. Haldar and S. Mahadevan, "First-order and second-order reliability methods," *Probabilistic Structural Mechanics Handbook*, pp. 27–52, 1995.
- [5] D. M. Pedroso, "Form reliability analysis using a parallel evolutionary algorithm," *Structural Safety*, vol. 65, pp. 84–99, 2017.
- [6] F. Homaei and M. Najafzadeh, "A reliability-based probabilistic evaluation of the wave-induced scour depth around marine structure piles," *Ocean Engineering*, vol. 196, Article ID 106818, 2020.
- [7] J. Zhang and X. Du, "A second-order reliability method with first-order efficiency," *Journal of Mechanical Design*, vol. 132, no. 10, 2010.
- [8] X. Huang, Y. Li, and Y. Zhang, "A new direct second-order reliability analysis method," *Applied Mathematical Modelling*, vol. 55, pp. 68–80, 2018.
- [9] A. Hosni Elhewy, E. Mesbahi, and Y. Pu, "Reliability analysis of structures using neural network method," *Probabilistic Engineering Mechanics*, vol. 21, no. 1, pp. 44–53, 2006.
- [10] E. Gobet and P. Turkedjiev, "Adaptive importance sampling in least-squares Monte Carlo algorithms for backward stochastic differential equations," *Stochastic Processes and their Applications*, vol. 127, no. 4, pp. 1171–1203, 2017.
- [11] I. Papaioannou, C. Papadimitriou, and D. Straub, "Sequential importance sampling for structural reliability analysis," *Structural Safety*, vol. 62, pp. 66–75, 2016.
- [12] P. Bjerager, "Probability integration by directional simulation," *Journal of Engineering Mechanics*, vol. 114, pp. 1285–1302, 1988.
- [13] D. A. Alvarez, F. Uribe, and J. E. Hurtado, "Estimation of the lower and upper bounds on the probability of failure using subset simulation and random set theory," *Mechanical Systems and Signal Processing*, vol. 100, pp. 782–801, 2018.

- [14] A. P. van den Eijnden and M. A. Hicks, "Efficient subset simulation for evaluating the modes of improbable slope failure," *Computers and Geotechnics*, vol. 88, pp. 267–280, 2017.
- [15] M. Najafzadeh and F. Saberi-Movahed, "GMDH-GEP to predict free span expansion rates below pipelines under waves," *Marine Georesources & Geotechnology*, vol. 37, no. 3, pp. 375–392, 2018.
- [16] A. Hadidi, B. F. Azar, and A. Rafiee, "Efficient response surface method for high-dimensional structural reliability analysis," *Structural Safety*, vol. 68, pp. 15–27, 2017.
- [17] X.-J. Meng, S.-K. Jing, L.-X. Zhang et al., "A new sampling approach for response surface method based reliability analysis and its application," *Advances in Mechanical Engineering*, vol. 7, no. 1, Article ID 305473, 2014.
- [18] T. Zhang, X. P. Zhou, and X. F. Liu, "Reliability analysis of slopes using the improved stochastic response surface methods with multicollinearity," *Engineering Geology*, vol. 271, 2020.
- [19] H. Zhao, Z. Ru, X. Chang et al., "Reliability analysis of tunnel using least square support vector machine," *Tunnelling and Underground Space Technology*, vol. 41, pp. 14–23, 2014.
- [20] L.-F. You, J.-G. Zhang, S. Zhou et al., "A novel mixed uncertainty support vector machine method for structural reliability analysis," *Acta Mechanica*, vol. 232, no. 4, pp. 1497–1513, 2021.
- [21] K. Cheng and Z. Lu, "Structural reliability analysis based on ensemble learning of surrogate models," *Structural Safety*, vol. 83, 2020.
- [22] K. Yuan, N.-C. Xiao, Z. Wang et al., "System reliability analysis by combining structure function and active learning kriging model," *Reliability Engineering & System Safety*, vol. 195, 2020.
- [23] S. Xiao, S. Oladyshkin, and W. Nowak, "Reliability analysis with stratified importance sampling based on adaptive Kriging," *Reliability Engineering & System Safety*, vol. 197, Article ID 106852, 2020.
- [24] X. Zhang, L. Wang, and J. D. Sørensen, "AKOIS: an adaptive Kriging oriented importance sampling method for structural system reliability analysis," *Structural Safety*, vol. 82, 2020.
- [25] H. M. Gomes and A. M. Awruch, "Comparison of response surface and neural network with other methods for structural reliability analysis," *Structural Safety*, vol. 26, no. 1, pp. 49–67, 2004.
- [26] J. Deng, D. Gu, X. Li et al., "Structural reliability analysis for implicit performance functions using artificial neural network," *Structural Safety*, vol. 27, no. 1, pp. 25–48, 2005.
- [27] Y. Hu, C.-d. Xiao, and Y.-y. Shi, "Reliability analysis for highly non-linear and complex model using ANN-MCM simulation," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, no. 5, 2018.
- [28] Q. Wang and H. Fang, "Reliability analysis of tunnels using an adaptive RBF and a first-order reliability method," *Computers and Geotechnics*, vol. 98, pp. 144–152, 2018.
- [29] J. Cheng, Q. S. Li, and R.-c. Xiao, "A new artificial neural network-based response surface method for structural reliability analysis," *Probabilistic Engineering Mechanics*, vol. 23, no. 1, pp. 51–63, 2008.
- [30] A. A. Chojaczyk, A. P. Teixeira, L. C. Neves et al., "Review and application of Artificial Neural Networks models in reliability analysis of steel structures," *Structural Safety*, vol. 52, pp. 78–89, 2015.
- [31] P. A. M. Lopes, H. M. Gomes, and A. M. Awruch, "Reliability analysis of laminated composite structures using finite elements and neural networks," *Composite Structures*, vol. 92, no. 7, pp. 1603–1613, 2010.
- [32] Y. Zhou, P. Unsong, and Z. Haoyuan, "Reliability sensitivity analysis of stochastic resonance failure of vehicle drum brake," in *Proceedings of the 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, Nanchang, China, 2020.
- [33] C. Su, Y. Zhang, and Q. Zhao, "Vibration reliability sensitivity analysis of general system with correlation failure modes," *Journal of Mechanical Science and Technology*, vol. 25, no. 12, pp. 3123–3133, 2012.
- [34] Y. Mo, S. Guo, and C. Tang, "A vibration reliability analysis method for the uncertain space beam structure," *Shock and Vibration*, vol. 2016, Article ID 9218590, 14 pages, 2016.
- [35] Z. Yang, L. Panxue, W. Hao et al., "Frequency reliability-based sensitivity analysis of motorized spindle by BP neural networks," *Journal of Harbin Institute of Technology*, vol. 49, no. 1, pp. 30–36, 2017.
- [36] C. Kwon, U. Song, U. Pak et al., "Effect of structural parameters on the flow field and power consumption of in-line high shear mixer," *Journal of the Indian Chemical Society*, vol. 98, no. 3, 2021.
- [37] W. Li, F. Xia, S. Zhao et al., "Mixing performance of an inline high-shear mixer with a novel pore-array liquid distributor," *Industrial & Engineering Chemistry Research*, vol. 58, no. 44, pp. 20213–20225, 2019.

## Research Article

# An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification

Aijaz Ahmad Reshi <sup>1</sup>, Furqan Rustam <sup>2</sup>, Arif Mehmood <sup>3</sup>, Abdulaziz Alhossan,<sup>4,5</sup>  
Ziyad Alrabiah,<sup>4</sup> Ajaz Ahmad <sup>4</sup>, Hessa Alsuwailam,<sup>4</sup> and Gyu Sang Choi <sup>6</sup>

<sup>1</sup>Department of Computer Science, College of Computer Science and Engineering, Taibah University, Al Madinah Al Munawarah, Saudi Arabia

<sup>2</sup>Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan

<sup>3</sup>Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Punjab 63100, Pakistan

<sup>4</sup>Department of Clinical Pharmacy, College of Pharmacy, King Saud University, Riyadh 11451, Saudi Arabia

<sup>5</sup>Corporate of Pharmacy Services, King Saud University Medical City, Riyadh, Saudi Arabia

<sup>6</sup>Department of Information & Communication Engineering, Yeungnam University, Gyeongbuk 38541, Republic of Korea

Correspondence should be addressed to Aijaz Ahmad Reshi; [aijazonnet@gmail.com](mailto:aijazonnet@gmail.com)

Received 2 December 2020; Revised 8 February 2021; Accepted 25 April 2021; Published 17 May 2021

Academic Editor: Átila Madureira Bueno

Copyright © 2021 Aijaz Ahmad Reshi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial intelligence (AI) techniques in general and convolutional neural networks (CNNs) in particular have attained successful results in medical image analysis and classification. A deep CNN architecture has been proposed in this paper for the diagnosis of COVID-19 based on the chest X-ray image classification. Due to the nonavailability of sufficient-size and good-quality chest X-ray image dataset, an effective and accurate CNN classification was a challenge. To deal with these complexities such as the availability of a very-small-sized and imbalanced dataset with image-quality issues, the dataset has been preprocessed in different phases using different techniques to achieve an effective training dataset for the proposed CNN model to attain its best performance. The preprocessing stages of the datasets performed in this study include dataset balancing, medical experts' image analysis, and data augmentation. The experimental results have shown the overall accuracy as high as 99.5% which demonstrates the good capability of the proposed CNN model in the current application domain. The CNN model has been tested in two scenarios. In the first scenario, the model has been tested using the 100 X-ray images of the original processed dataset which achieved an accuracy of 100%. In the second scenario, the model has been tested using an independent dataset of COVID-19 X-ray images. The performance in this test scenario was as high as 99.5%. To further prove that the proposed model outperforms other models, a comparative analysis has been done with some of the machine learning algorithms. The proposed model has outperformed all the models generally and specifically when the model testing was done using an independent testing set.

## 1. Introduction

The virus called the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) had been discovered in late 2019. The virus which originated in China became a cause of a disease known as Corona Virus Disease 2019 or COVID-19. The World Health Organization (WHO) declared the disease as a pandemic in March 2020 [1, 2]. According to the reports issued and updated by global healthcare authorities

and state governments, the pandemic affected millions of people globally. The most serious illness caused by COVID-19 is related to the lungs such as pneumonia. The symptoms of the disease can vary and include dyspnea, high fever, runny nose, and cough. These cases can most commonly be diagnosed using chest X-ray imaging analysis for the abnormalities [3].

X-radiation or X-ray is an electromagnetic form of penetrating radiation. These radiations are passed through

the desired human body parts to create images of internal details of the body part. The X-ray image is a representation of the internal body parts in black and white shades. X-ray is one of the oldest and commonly used medical diagnosis tests. Chest X-ray is used to diagnose the chest-related diseases like pneumonia and other lung diseases [4], as it provides the image of the thoracic cavity, consisting of the chest and spine bones along with the soft organs including the lungs, blood vessels, and airways. The X-ray imaging technique provides numerous advantages as an alternative diagnosis procedure for COVID-19 over other testing procedures. These benefits include its low cost, the vast availability of X-ray facilities, noninvasiveness, less time consumption, and device affordability. Thus, X-ray imaging may be considered a better candidate for the mass, easy, and quick diagnosis procedure for a pandemic like COVID-19 considering the current global healthcare crisis.

Deep learning and ANNs have endorsed an exponential research focus over the last decade. The deep ANNs have outperformed other conventional models on many essential benchmarks. Thus, ANNs have generally proved to be the state-of-the-art technology across a wide range of application areas, including NLP, speech recognition, image processing, biological sciences, and other commercial as well as academic areas. The advancement of ANNs has massive potential in healthcare applications, specifically in medical data analysis, diagnosis through medical image processing, and analysis. As seen in recent times, various parts of the world face the healthcare crisis both in terms of the needed number of healthcare professionals and testing equipment. Considering the present pandemic situation, there is an appurtenant relationship between the detection of COVID-19 cases and chest X-ray image analysis and classification. In this work, an automatic diagnostic system has been developed using CNN which uses chest X-ray analysis results to diagnose whether a person is COVID-19-affected or normal. Preliminary analysis of this study has shown promising results in terms of its accuracy and other performance parameters to diagnose the disease in a cost-effective and time-efficient manner. This study used CNN with extra layers to improve the COVID-19 X-ray image classification accuracy. In neural networks, the CNN structure is specially designed to process the two-dimensional image tasks although it can also be used in one- and three-dimensional data. CNN is a type of DNN, inspired by the visual system of the human brain, and is most commonly used in the analysis of visual imagery. To train the CNN model, first, the dataset has been obtained from GitHub [5]. Since the dataset obtained for training the model was very small in size and imbalanced, to solve the problem of having very-limited-sized X-ray image dataset, it has been extended using data augmentation techniques to increase its size and also to make the model training feature rich. Image flipping and rotation at different angles have been used to generate more data. For dataset balancing in terms of proportion of images with different class labels, the dataset has been further extended with some more image instances of the minority class. After data augmentation and dataset balancing, the CNN model has been trained using a total of 800 images (400

COVID-19 and 400 normal) and then the model has been tested by using a test set. The CNN model performance evaluation has then been done using different performance metrics. These metrics include accuracy, precision, sensitivity, specificity, ROC AUC, and  $F_1$  score. Later, the proposed CNN model has also been tested using an independent dataset obtained from the IEEE data port [6] for independent validation of the proposed CNN model. Various machine learning models have also been used for the comparative performance analysis in comparison with the proposed CNN model to show its significance over these models.

The following are some of the key findings of this study:

- (i) CNN with extra convolutional layers (e.g., six layers have been used in the CNN proposed in this study) performs best in COVID-19 diagnosis
- (ii) CNN models require a sufficient amount of images for efficient and more accurate image classification
- (iii) Data augmentation techniques are very effective to improve the CNN model performance remarkably by generating more data from an existing limited-size dataset
- (iv) Data augmentation is also effective in image classification as it gives the ability of invariance to CNNs
- (v) The proposed CNN model performance has been proved statistically significant in the performance of other ML models
- (vi) CNN-based diagnosis using X-ray imaging can be very effective for medical sector to handle the mass testing situations in pandemics like COVID-19

The rest of the paper is divided into various sections. Section 2 constitutes the related work. Section 3 presents the workflow. Section 4 contains the materials and methods used. Section 5 describes the results of the study and discussion, and in the end, Section 6 presents the conclusion.

## 2. Related Work

Deep learning has shown a dramatic increase in the medical applications in general and specifically in medical image-based diagnosis. Deep learning models performed prominently in computer vision problems related to medical image analysis. The ANNs outperformed other conventional models and methods of image analysis [7, 8]. Due to the very promising results provided by CNNs in medical image analysis and classification, they are considered as de facto standard in this domain [9, 10]. CNN has been used for a variety of classification tasks related to medical diagnosis such as lung disease [10], detection of malarial parasite in images of thin blood smear [11], breast cancer detection [12], wireless endoscopy images [13], interstitial lung disease [14], CAD-based diagnosis in chest radiography [15], diagnosis of skin cancer by classification [16], and automatic diagnosis of various chest diseases using chest X-ray image classification [17]. Since the emergence of COVID-19 in December 2019,

numerous researchers are engaged with the experimentation and research activities related to diagnosis, treatment, and management of COVID-19.

Researchers in [18] have reported the significance of the applicability of AI methods in image analysis for the detection and management of COVID-19 cases. COVID-19 detection can be done accurately using deep learning models' analysis of pulmonary CT [18]. Researchers in [19] have designed an open-source COVID-19 diagnosis system based on a deep CNN. In this study, tailored deep CNN design has been reported for the detection of COVID-19 patients using X-ray images. Another significant study has reported on the X-ray dataset comprising X-ray images belonging to common pneumonia patients, COVID-19 patients, and people with no disease [20]. The study uses the state-of-the-art CNN architectures for the automatic detection of patients with COVID-19. Transfer learning has achieved a promising accuracy of 97.82% in COVID-19 detection in this study. Another recent and relevant study has been conducted on validation and adaptability of Decompose-, Transfer-, and Compose-type deep CNN for COVID-19 detection using chest X-ray image classification [21]. The authors have reported the results of the study with an accuracy of 95.12%, sensitivity of 97.91%, and specificity of 91.87%.

Having reviewed the relevant and recent research work on the design, development, and possible applicability of CNNs in COVID-19 detection using medical images, particularly X-ray images, due to the availability of a very less amount of X-ray images of COVID-19 patients and the poor quality of some images in the dataset, the accuracy of the models was affected. This study is particularly focused on dataset preprocessing to fine-tune it, data augmentation, and design of a CNN with extra layers to increase further the performance of the COVID-19 diagnosis using CNNs as described in subsequent sections.

### 3. Workflow

As illustrated in Figure 1, the workflow of this study begins with collection of primary dataset containing two image classes: one class belonged to chest X-rays of COVID-19-confirmed cases and the other class of images belonged to the normal people without the disease. In the next phase of the study, the concerned medical professionals analysed the dataset and removed some of the X-ray images which were not clear in terms of quality and diagnostic parameters. Hence, the resulted dataset was very clean, as each X-ray image was of good quality as well as clear in terms of significant diagnostic parameters according to their expertise. In the third phase, the dataset was augmented using standard augmentation techniques to increase its size. The resulted dataset was used to train the model in the next phase. After training, the model was tested for its performance in the disease detection. The testing of proposed CNN model has been done using test dataset held from the primary dataset as well as using the independent validation dataset. Table 1 contains the details of datasets including the total number of

X-ray images in training set, testing set, validation set, and the proportion of X-ray images in the two prediction classes.

## 4. Materials and Methods

*4.1. Dataset.* In the experiments of this study, a primary dataset containing 178 X-ray images has been used as a base dataset. Of 178 images, 136 X-ray images belonged to confirmed COVID-19 patients and other 42 images belonged to normal or people with other diseases like pneumonia. The dataset used is available on GitHub [5]. The basic dataset consists of two classes of COVID-19 with 136 samples and others with 42 samples. Thus, the dataset was imbalanced and needed preprocessing to achieve promising results. As a first attempt, CNN was trained on the given original dataset and around 54% accuracy was achieved, which was not worthy of the current application domain. The main dataset sources used in this study are enlisted as follows:

- (i) Primary chest X-ray image dataset of COVID-19 patients collected from GitHub. The dataset has been collected by the University of Montreal's Ethics Committee no. CERSES-20-058-D from different hospitals and clinics [5].
- (ii) For dataset balancing, a collection of chest X-ray images were collected from Kaggle [22].
- (iii) Independent validation dataset containing a collection of 100 COVID-19 X-ray images for the real-world testing of the proposed CNN was collected from IEEE DataPort [6].

The experiments have been conducted using Core i7 7th-generation machine with 8 GB RAM, Microsoft Windows 10 platform using Python language with Anaconda 3 software and Jupyter Notebook.

### 4.2. Dataset Preprocessing

*4.2.1. Balancing Dataset Classes.* To balance the given dataset, in order to improve the performance of the proposed CNN models in the detection of COVID-19 cases, 136 normal chest X-ray images have been used. These concatenated extra X-ray images were downloaded from Kaggle [22]. After balancing the dataset when the models have been trained again on the resulted dataset, the accuracy of the given CNN models was improved to 69%. Still, the performance given by the models in terms of accuracy and other measures was not justified as an effective system for COVID-19 detection.

*4.2.2. Analysis of X-Ray Images by Medical Experts.* A deep analysis was done on the X-ray images by medical specialists. Out of 135 X-ray images of confirmed COVID-19 patients, only a set of 90 X-ray images was selected as a perfect candidate to train the models. The resulted dataset now was reduced to 90 COVID-19-confirmed cases and 90 normal X-ray images. The resulted dataset was again used in training the proposed CNN model; there was again an improvement



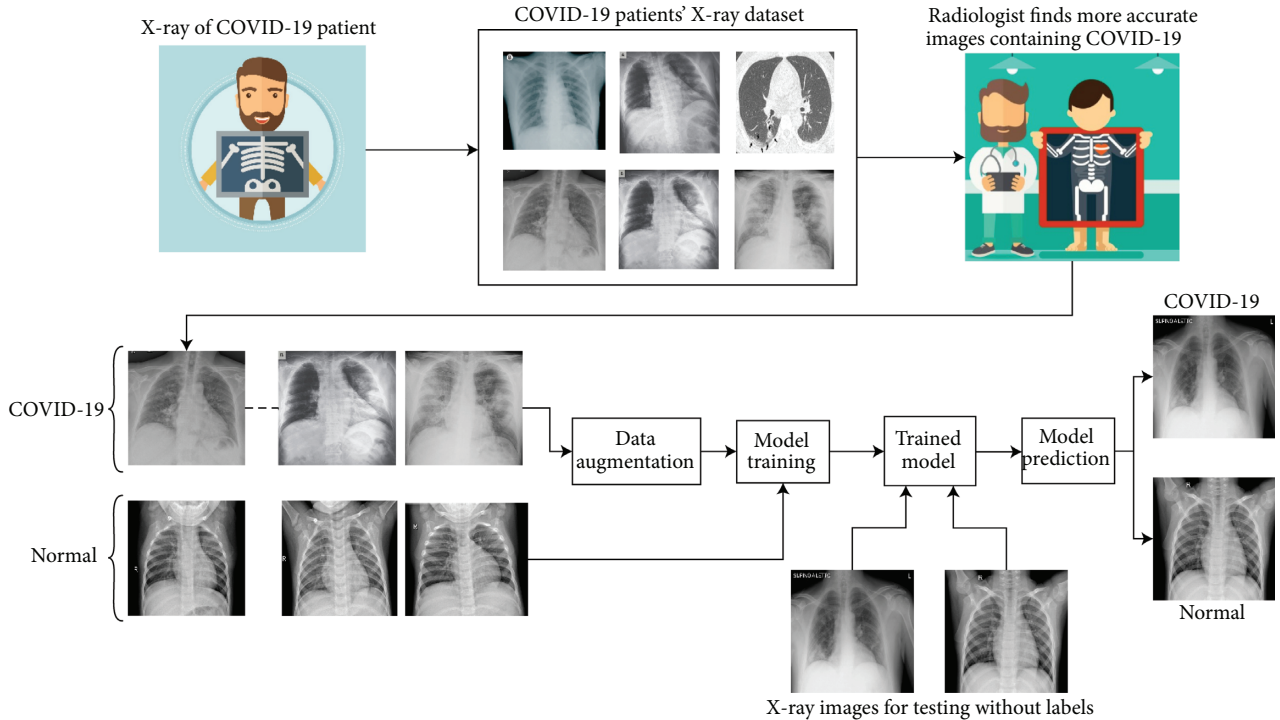


FIGURE 1: System development workflow.

TABLE 1: Dataset image count for training and testing.

Dataset	COVID-19 images	Normal images	Total images
Total data	450	450	900
Training data	400	400	800
Testing data	50	50	100
Independent validation data	100	100	200

in the performance of the model. Specifically, the accuracy was increased to 72% in the given scenario. Still, because the dataset was not containing a sufficient number of images for an effective training, there was not a significant increase in the accuracy and other performance metrics.

**4.2.3. Data Augmentation.** Data augmentation is a technique that can significantly increase the data instances of a dataset to train a model [23]. In the case of image datasets, the technique uses the basic image processing operations, such as flipping, rotating, cropping, or padding for augmentation. The dataset is then extended by these transformed images resulted from the existing image set, which increases the size of dataset to train the neural networks [24]. To solve the problem of the availability of a small size dataset that was affecting the performance of the proposed CNN, the data augmentation method has been used in this study. This technique increased the size of the dataset; in addition, it provides more learning features to the learning model. Two image processing operations, flipping and rotation, have been used in this study for data augmentation. In the first phase of data augmentation, the 90 X-ray images have been flipped to get extra 90 images. The resulted dataset was increased to contain 180 images after applying this

operation. In the second phase, the original 90 images have further been rotated by  $90^\circ$  angle to get 90 more images and then rotated by  $180^\circ$  angle to get 90 more images, and finally, the original 90 images were further rotated by  $270^\circ$  angle to get more 90 images. These operations resulted in a dataset containing 450 COVID-19 X-ray images. Table 2 shows the image processing operations performed on the image types and corresponding count of images resulted from the operation. Figure 2 shows the effect of augmentation techniques applied to the original sample image of the dataset used in this study.

**4.3. Convolutional Neural Networks (CNNs).** The CNNs are inspired by visual system of human brain. The idea behind the CNNs thus is to make the computers capable of viewing the world as humans view it. This way CNNs can be used in the fields of image recognition and analysis, image classification, and natural language processing [25]. CNN is a type of deep neural networks which contain the convolutional, max pooling, and nonlinear activation layers. The convolutional layer, considered as a main layer of a CNN, performs the operation called "convolution" that gives CNN its name. Kernels in the convolutional layer are applied to the layer inputs. All the outputs of the convolutional layers

TABLE 2: COVID-19 image count after data augmentation.

Image type	Count
Original	90
Original flipped	90
Original with a 90-degree rotation	90
Original with 180-degree rotation	90
Original with 270-degree rotation	90
Total	450

are convolved as a feature map. In this study, the Rectified Linear Unit (ReLU) has been used in the activation function with a convolutional layer which is helpful to increase the nonlinearity in input image, as the images are fundamentally nonlinear in nature. Thus, CNN with ReLU in the current scenario is easier and faster. Since the ReLU is zero for all negative inputs, it can be defined as

$$z = \max(0, i). \quad (1)$$

Here, the function implies that the output  $z$  is zero for all negative value and positive value remains the constant as shown in Figure 3.

The pooling layer or subsampling layer is also an important building block of CNN. On each feature map extracted through the convolution layer, the pooling layer operates independently. To minimize overfitting and the number of extracted features, it decreases the spatial size of the feature map and returns the important features. Pooling can be the max, average, and sum in the CNN model. In this study, max pooling has been used because others may not identify the sharp features easily as compared to max pooling. In addition, the batch normalization layer has been used in this study as it involved the training of a very deep neural network. So the technique adjusts the scaling and activation to normalize the input layer and speed up the learning procedure between hidden units. The dropout layer with a 20% dropout rate has also been used, which drops the neurons during the training chosen at random to reduce the overfitting problem. Towards the last stage of the CNN used in the study, there is a flattening layer to convert the output of convolutional layers into a single-dimensional feature vector. In other words, the flattening layer arranges all the pixel data output produced by convolutional layers in one vector. After flattening, the vector data is given as an input to the next layers of the CNN called fully connected layers or dense layers. In a fully connected layer, each neuron of the previous layer is directly connected to each of the neurons in its next layer. The main functionality of dense layers is to take flattened output results from the convolution and pooling layers and as input and classify the image to a specific class label. Each value of the flattened feature set represents the probability of a feature belonging to a specific class. Thus, on the basis of these probabilities, the fully connected network with dense layers finally drives the classification decision.

**4.3.1. The Proposed CNN Architecture.** The proposed CNN model consists of 38 layers in which 6 are convolutional (Conv2D), 6 max pooling layers, 6 dropout layers, 8

activation function layers, 8 batch normalization layers, 1 flatten layer, and 3 fully connected layers; CNN model input image shape is (150, 150, 3), i.e., 150-by-150 RGB image. In all Con2D layers, a  $3 \times 3$  size kernel has been used but the filter size after every two Con2D layers increases. At the 1st and 2nd layers of Con2D, 64 filters have been used to learn from input and the 3rd and 4th layers of Con2D use 128 filters, and at the 5th and 6th layers, 256 filters have been used. After each Con2D layer, the max pooling layer with  $2 \times 2$  pooling size has been used, the batch normalization layer has been used with the axis = -1 argument, the activation layer has been used with the ReLU function, and the dropout layer has been used with 20% dropout rate. The output of 256 output neurons of the final Con2D layer is followed by max pooling, batch normalization, activation, and dropout layer. Since the final pooling and convolutional layer gives a three-dimensional matrix as output, to flatten the matrix, a flattening layer has been used which converts them into a vector that will be input for 3 dense layers.

This study uses CNN for binary classification; that is the reason for using the binary crossentropy (BCE) loss function. In binary classification since only one output node is needed to classify the data to one of the two given classes, so in the case of BCE loss function, the output value is being given to a sigmoid activation function. The output given by the sigmoid activation function lies between 0 and 1. It finds the error between the predicted class and the actual class. The ‘‘Adam’’ optimizer has been used which changes the attribute weight and learning rate to reduce the loss of the learning model. The model parameter values are given in Table 3, and the model architecture is given in Figure 4. During the initial experiments, the CNN has been used with different configurations in terms of the usage of number of convolution layers in the model. The decision of how many convolution layers used in the model was made by using an incremental approach. First, the CNN was tested using only one convolutional layer and the results were analysed. Then, the CNN was built with two layers and results were analysed and so on. The approach had been continued till the results provided by the model were accurate and effective. The final model which was very feasible according to its results consisted of six convolution layers. The results of each increment of the model have been reported in the Results section.

## 5. Results and Discussion

After preprocessing of the dataset, the final dataset consisted of a total of 900 X-ray images. For training and testing the proposed CNN, the dataset was partitioned into two subsets. The training dataset contained 400 COVID-19 X-ray images and 400 normal X-ray images, making a total of 800 X-ray images. The testing dataset similarly contained 100 X-ray images, in which 50 X-ray images were from each class COVID-19 positive and normal. Then, the training subset containing 800 X-ray images has been passed to the model with 25% validation size. So, out of 800 X-ray images, with each epoch, 600 X-ray images train the model, and 200 X-ray Images validate the model. As mentioned in the proposed

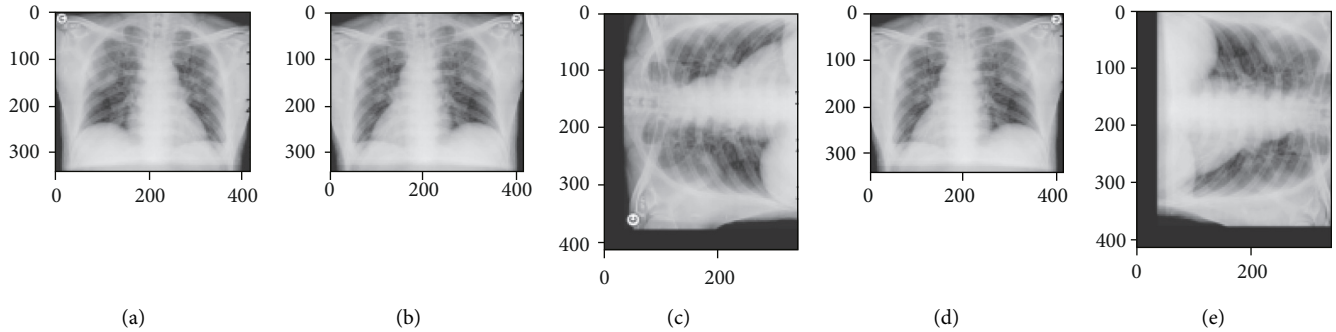


FIGURE 2: Effect of augmentation techniques on an X-ray image. (a) Original. (b) Flipped. (c) Rotated 90°. (d) Rotated 180°. (e) Rotated 270°.

architecture of the CNN model, it consisted of 38 layers in which 6 are convolutional, 6 max pooling layers, 6 dropout layers, 8 activation function layers, 8 batch normalization layers, 1 flattening layer, and 3 fully connected layers. The CNN model thus achieved an extraordinary performance with an accuracy of 100% with the test data subset used from the processed dataset of this study with a precision of 1.0, with the model parameter values given in Table 3. To evaluate the overall performance, in addition to accuracy, other important metrics have been adopted in this study including  $F_1$  score, precision, sensitivity, specificity, and ROC AUC. The scores of these parameters are reported in Table 4.

The confusion matrix of the model is shown in Figure 5. Figures 6 and 7 show the curve of accuracy and loss between training and testing, respectively. According to the confusion matrix, the CNN model test uses the 100 X-ray images from the GitHub dataset, where 50 images belong to the COVID-19 class and 50 to the normal images. The CNN model shows significant performance on testing and predicts all 100 images correctly with 0% error rate as reported in the confusion matrix of Figure 5. Figure 6 shows the model accuracy during the training and validation as a graph where the curve drawn in blue color shows the training accuracy of CNN, while the curve with orange color shows the validation accuracy. Training accuracy of the CNN according to Figure 6 remains consistent after the 5 epochs and the CNN also shows a consistent validation accuracy after the 25 epochs. The plot in Figure 7 shows the loss during the training and validation of CNN. The training loss of CNN is minimum and consistent from the 1st epoch while validation loss becomes minimum after 5 epochs and remains consistent till the last epoch. The above results show the efficiency of the CNN model proposed in this study.

Figure 6 shows the plots drawn from the training and testing accuracy achieved by the proposed CNN model. Figure 7 shows the training and testing loss for the proposed CNN model. As can be observed in Figure 7, the proposed CNN is not taking a lot of time to converge, as in the first epoch the training loss is 31 and right after 5 epochs, it drops to 0.9; then after 23 epochs, it drops again to 0.0011, and at the last epoch, the total loss is 0.000058.

In addition to the above performance measurements,  $K$  fold cross-validation has been applied to further test the

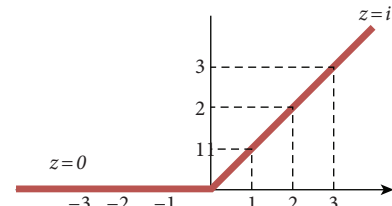


FIGURE 3: ReLU function demonstration.

TABLE 3: Model parameter values.

Parameter	Value
Input dimension	(150, 150, 3)
Filter to learn	64, 128, 256
Max pooling	$2 \times 2$
Batch normalization	Axis = -1
Activation functions	ReLU, sigmoid
Dropout rate	20%
Kernel size	$3 \times 3$
Epochs	50
Optimizer	Adam
Loss function	binary_crossentropy

proposed model for its skill. In this study, a 10-fold cross-validation has been used. The results provided are very effective as the average score of 10 iterations is 99.67% ( $\pm 0.15\%$ ).

**5.1. Testing of the CNN Using COVID-19 Independent Validation Data.** As proof of the significance of the proposed CNN model in the classification for detection of COVID-19 from X-ray images, the trained model has been tested using an independent dataset obtained from the IEEE DataPort [6]. The independent test dataset contained 100 COVID-19 X-rays. This dataset was then extended by adding 100 normal images for testing. The model using the same settings performs very well with an accuracy of 0.995 and precision 1.000 along with other performance parameters reported in Table 5.

Figure 8 shows the confusion matrix of CNN model when tested on the independent test dataset which have been obtained from IEEE DataPort. The CNN model also performed very efficiently on independent test data giving 198

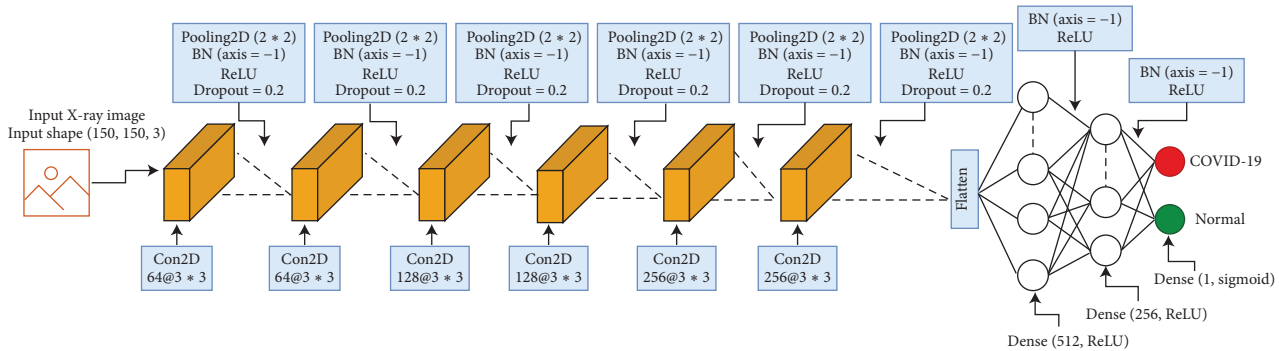


FIGURE 4: CNN model architecture.

correct classification results out of 200 input X-ray images. As can be seen in the results given by the confusion matrix, an equal number of images is obtained from both of the target classes (100 COVID-19 and 100 normal). The model classifies only one image falsely as normal from COVID-19 class.

Falsely classified image was examined by the experts. It was noted that the X-ray image belonged to a person on an early stage of COVID-19. As a result, the image does not contain the prominent patterns with which the image could have been differentiated from normal X-ray image class. Figure 9 shows the actual X-ray image of a COVID-19 positive case which was falsely classified by the model. The comparison of CNN model on test data and independent validation data in terms of different performance metrics is shown in Figure 10.

As mentioned in the proposed CNN model architecture section, the proposed model was constructed in an incremental approach. Starting with single convolutional layer model, in each following increment, a convolutional layer had been added and results were analysed. Table 6 illustrates the results of the model in terms of accuracy after every increment from single convolutional layer to a stable model consisting of six convolutional layers. The incremental approach comparison is shown in Figure 11.

**5.2. Performance Comparison of Machine Learning Models with the CNN Model.** In this study, experiments have also been conducted on some of the relevant machine learning models such as Random Forest (RF) [26, 27], gradient boosting machine (GBM) [28, 29], support vector classifier (SVC) [30], logistic regression (LR) [31], and  $k$ -nearest neighbor (KNN) [32] for comparative analysis of CNN with these models. These models have been used with their best parameter settings as shown in Table 7.

RF has been used with two hyperparameters as shown in the table. The  $n\_estimators$  define how many decision trees are generated under RF to make a prediction. The  $max\_depth$  defines what should be the maximum depth of each decision tree in RF, so in this setting, the  $max\_depth$  parameter restricts the decision tree to a maximum 300 level depth.

GBM has been used with three parameters, two are the same as in RF and one is  $learning\_rate$  which is a tuning

TABLE 4: Model performance on test data.

Evaluation parameters	Score
Accuracy	1.000
Precision	1.000
Sensitivity	1.000
Specificity	1.000
$F_1$ score	1.000
ROC AUC	1.000

Confusion matrix for the GitHub dataset

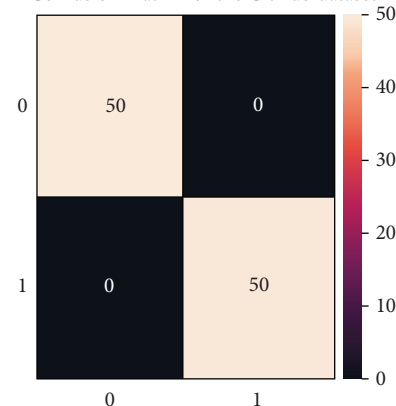


FIGURE 5: Confusion matrix of CNN model for primary/GitHub dataset. Here 0 represents the normal class and 1 represents the COVID-19.

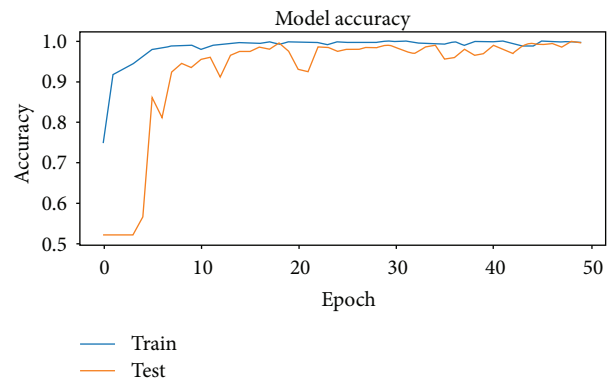


FIGURE 6: Training and testing accuracy plot achieved by the CNN model.

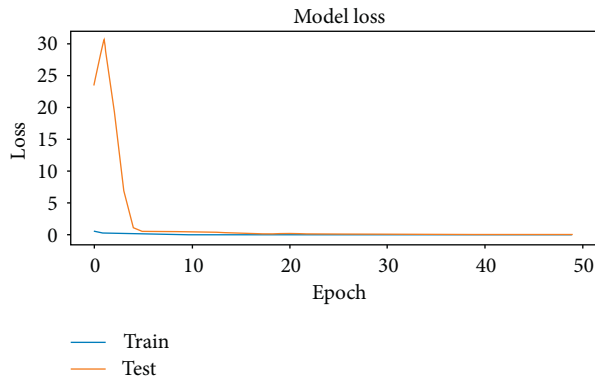


FIGURE 7: Training and testing loss plot by the CNN model.

TABLE 5: Model performance on independent validation data.

Evaluation parameters	Score
Accuracy	0.995
Precision	1.000
Sensitivity	0.990
Specificity	1.000
$F_1$ score	0.994
ROC AUC	0.990

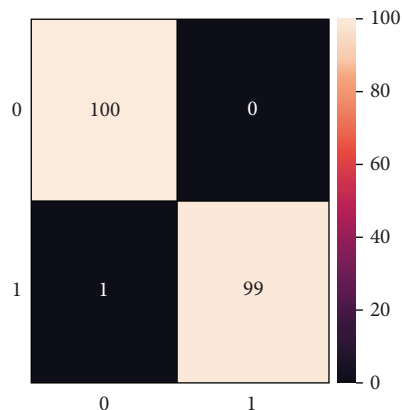


FIGURE 8: Confusion matrix of CNN model for independent validation data (0: normal class, 1: COVID-19).

parameter for optimization of algorithms to reduce the error to a minimum level [28]. SVC has been used with the linear kernel which is best for binary class classification and the second parameter  $C$  conveys SVC optimization the limit to prevent misclassification of the individual training examples. In contrary, even if the hyperplane misclassifies certain points, a very small value of  $C$  would cause the optimizer to search for a wider-margin separating hyperplane. LR has been used with the “liblinear” solver because it is preferred when there is a small dataset, and the second parameter is  $C$  as used in SVC. KNN has been used with all the default parameters setting.

Thus, the CNN and machine learning models have been trained using the original dataset of this study. Then, both CNN and each of the machine learning models have been tested using the training subset of the original dataset. In this

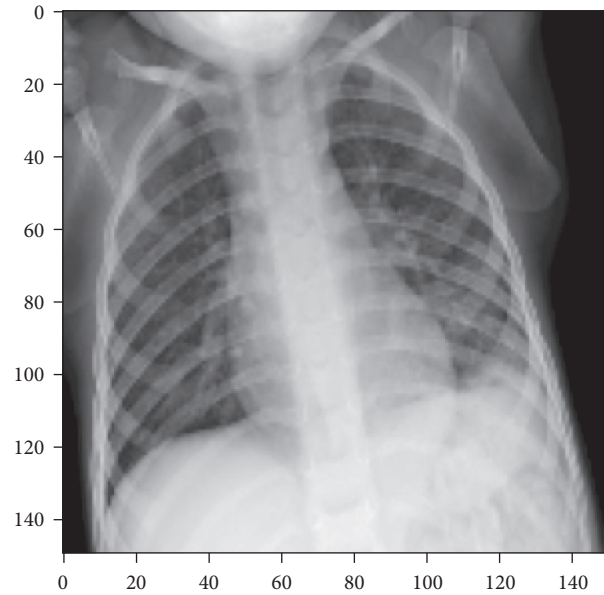


FIGURE 9: The falsely classified X-ray image.

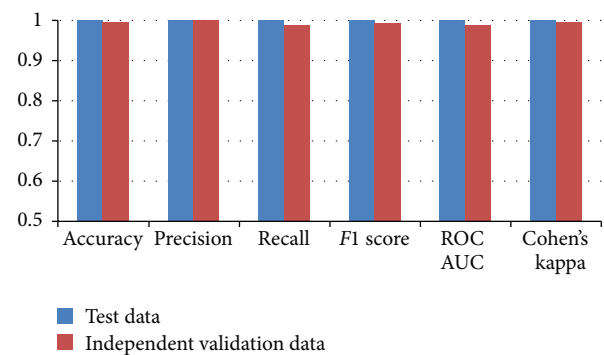


FIGURE 10: CNN model performance comparison on both test data and independent validation data.

TABLE 6: Accuracy score with different number of CNN layers.

Convolutional layer	Test data	Independent validation data
One Conv2D	0.715	0.455
Two Conv2D	0.940	0.895
Three Conv2D	0.995	0.957
Four Conv2D	0.995	0.980
Five Conv2D	0.995	0.995
Six Conv2D	1.000	0.995

scenario, three machine learning models SVC, LR, and KNN performed well almost the same as CNN as reported in Table 8. On the other hand, when the same machine learning models with the same settings were tested on the independent test set as mentioned earlier, their performance was degraded while the proposed CNN maintains the performance metrics in this scenario also. The results of the second scenario in which models were tested using the independent test set are reported in Table 9.

This comparative analysis and also comparing the overall performance results achieved by the proposed CNN model

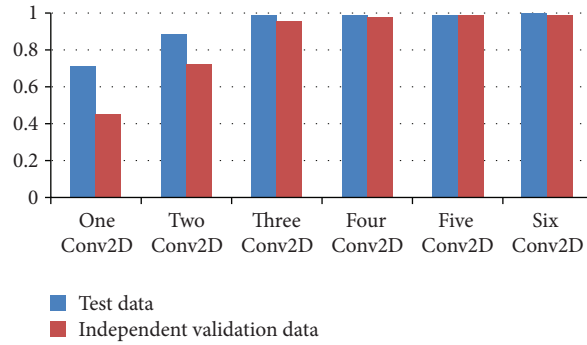


FIGURE 11: Accuracy score comparison with different number of CNN layers.

TABLE 7: Machine learning models hyperparameter setting.

Model	Hyperparameter settings
RF	$N\_estimators = 300, max\_depth = 300$
GBM	$N\_estimators = 300, max\_depth = 300, learning\_rate = 0.2$
SVC	Kernel = linear, $C = 3.0$
LR	Solver = liblinear, $C = 3.0$
KNN	$n\_neighbors = 3$

TABLE 8: Models' performance on test data.

Model	Accuracy	Precision	Sensitivity	Specificity	$F_1$ score	ROC AUC
RF	0.980	0.961	1.000	0.961	0.980	1.000
GBM	0.940	0.960	0.923	0.958	0.941	0.940
SVC	1.000	1.000	1.000	1.000	1.000	1.000
LR	1.000	1.000	1.000	1.000	1.000	1.000
KNN	1.000	1.000	1.000	1.000	1.000	1.000
CNN	1.000	1.000	1.000	1.000	1.000	1.000

TABLE 9: Models' performance on independent validation data.

Model	Accuracy	Precision	Sensitivity	Specificity	$F_1$ score	ROC AUC
RF	0.950	0.960	0.941	0.960	0.950	0.950
GBM	0.920	0.930	0.911	0.928	0.920	0.922
SVC	0.940	0.950	0.931	0.948	0.940	0.943
LR	0.940	0.960	0.923	0.958	0.941	0.944
KNN	0.930	0.990	0.880	0.990	0.930	0.930
CNN	0.995	1.000	0.990	1.000	0.995	0.995

provide strong evidence of possible applicability of the model in the COVID-19 diagnosis using X-ray image classification.

The two bar graphs in Figure 12 shows the comparisons of the performances given by different machine learning models and CNN. The bar graph labeled as test data is the result of the first test scenario, where each of the models was tested on the test subset extracted from the original dataset. The second bar graph labeled as independent validation data is the result of the second test scenario, where each of the models was tested on the independent test set.

To compare the performance results of the proposed CNN-based methodology for the current application domain, the results of other recent studies done by different

researchers have been collected and compared. The comparison of these study results along with the mythology used has been shown in Table 10.

**5.3. Statistical Significance of the Proposed CNN Model.** In order to test whether the proposed CNN model has a statistical significance over the other models,  $t$ -test [36] has been performed. To determine the significance, the alternate hypothesis  $H_a$  and null hypothesis  $H_o$  have been established as follows:

- (i) *Alternate Hypothesis ( $H_a$ ).* There is a statistical significance in the performance given by the

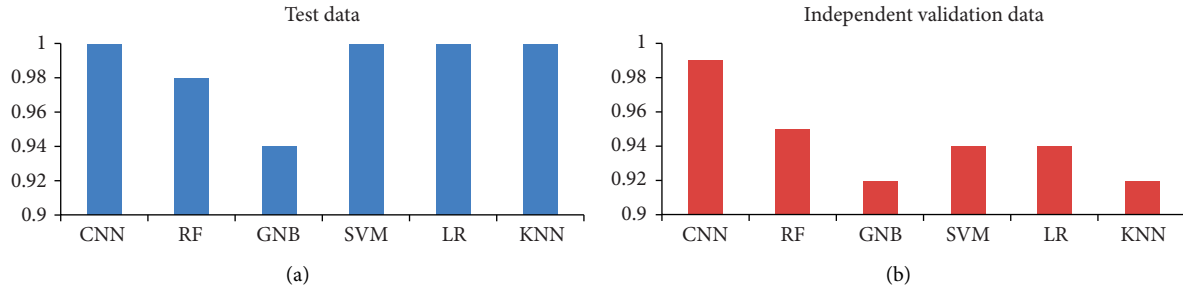


FIGURE 12: Accuracy of different machine learning models along with proposed CNN model. (a) Test data. (b) Independent validation data.

TABLE 10: Comparison with other studies performed on the same dataset.

Study	Model	Accuracy (%)
Sethy et al. [33]	SVM	98.66
Minaee et al. [34]	SqueezeNet	92.2
Das et al. [35]	CNN	97.4
Our study (test data)	CNN	100
Our study (validation data)	CNN	99.2

proposed CNN model over the performance of other models

- (ii) *Null Hypothesis ( $H_0$ )*. There is no statistical significance in the performance given by the proposed CNN model over the performance of other models

First, the significance of the proposed CNN model has been shown on two models, RF and GBM, when these models have been tested on the test data set derived from the primary dataset. The  $H_0$  has been rejected in the case of RF and GBM. That means the proposed CNN has a statistical significance over RF and GBM only in this testing scenario. While  $H_0$  was accepted in the case of SVC, LR, and KNN, that means the proposed CNN has no statistical significance over these models.

Second, the  $t$ -test was performed on the performance results of all the models when these models were tested on the independent validation data set. In this scenario,  $H_0$  was rejected in comparison to all the other models. This implies that the CNN is statistically significant on all the given models in this model testing scenario.

## 6. Conclusion

This study has been conducted to demonstrate the effective and accurate diagnosis of COVID-19 using CNN which was trained on chest X-ray image datasets. The model training was performed incrementally with different datasets to attain the maximum accuracy and performance. The primary dataset was very limited in size and also imbalanced in terms of class distribution. These two issues with the primary dataset affected the performance of the models very badly. To overcome these issues, the dataset was preprocessed using different techniques, including dataset balancing technique, manual analysis of X-ray images by concerned medical

experts, and data augmentation techniques. To balance the dataset for model training and also to test its performance parameters, an ample number of chest X-rays were collected from different available sources. After training and testing the CNN model on the fully processed dataset, the performance results have been reported. In addition, to test further the model performance, particularly the accuracy, the proposed CNN model has been tested using an independent dataset as an independent validation and real-world test obtained from IEEE DataPort [6]. As reported in the results in both the testing scenarios, the proposed CNN model has shown highly promising results. Since this study uses an incremental approach in training the model using different sizes and types of datasets, the approach confirmed the fact that CNN models require an ample amount of image data for the efficient and more-accurate classification. The data augmentation techniques are very effective to significantly improve the CNN model performance by generating more data from an existing limited-size dataset and also by giving the ability of invariance to the CNN. The proposed CNN model's number of convolutional layers was also decided in an incremental approach; that is, in the first increment, only one convolutional layer was used and, then, on the basis of model performance metrics, one layer in each increment was increased till it reaches a stable and efficient stage in terms of its performance. The final version of the CNN consisted of six convolutional layers. A comparative analysis has also been done to further test the scope of the proposed CNN model by performance comparisons with some of the prominent machine learning models such as RF, GBM, SVC, LR, and KNN. The results prove that the proposed CNN has outperformed all the models particularly when each model was tested on the independent validation dataset. Considering the significant effect of data augmentation techniques on model performances, the authors are currently working on the application of other state-of-the-art data augmentation algorithms and techniques. In the future, the results obtained from the study concerned with the applicability of these modern data augmentation techniques in different application domains will be published.

## Data Availability

The data supporting this study are from previously reported studies and datasets, which have been cited.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Authors' Contributions

Aijaz Ahmad Reshi and Furqan Rustam contributed equally to this work.

## Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for supporting this research work through research group no. RG-1441-455.

## References


- [1] D. Cucinotta and M. Vanelli, "WHO declares COVID-19 a pandemic," *Acta Biomedica: Atenei Parmensis*, vol. 91, pp. 157–160, 2020.
- [2] F. Rustam, A. A. Reshi, A. Mehmood et al., "COVID-19 future forecasting using supervised machine learning models," *IEEE Access*, 2020.
- [3] D. J. Cennimo, "Coronavirus disease 2019 (COVID-19) clinical presentation," vol. 8, pp. 101489–101499, 2020, <https://emedicine.medscape.com/article/2500114-clinical#b2>, 2020. Online.
- [4] X-ray (Radiography)-Chest, 2020, <https://www.radiologyinfo.org/en/info.cfm?pg=chestradi#overview>.
- [5] J. P. Cohen, "Github Covid19 X-ray dataset," 2020, <https://github.com/ieee8023/covid-chestxray-dataset>, 2020. Online.
- [6] Z. H. Chen, "Mask-RCNN detection of COVID-19 pneumonia symptoms by employing stacked autoencoders in deep unsupervised learning on low-dose high resolution CT," *IEEE Dataport*, 2020.
- [7] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.
- [8] M. Ahmad, "Ground truth labeling and samples selection for hyperspectral image classification," *Optik*, vol. 230, Article ID 166267, 2021.
- [9] B. Kayalibay, G. Jensen, and P. van der Smagt, "CNN-based segmentation of medical imaging data," 2017, <http://arxiv.org/abs/1701.03056>.
- [10] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 844–848, Singapore, December 2014.
- [11] M. Umer, S. Sadiq, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, "A novel stacked CNN for malarial parasite detection in Thin blood smear images," *IEEE Access*, vol. 8, pp. 93782–93792, 2020.
- [12] R. Rouhi, M. Jafari, S. Kasaei, and P. Keshavarzian, "Benign and malignant breast tumors classification based on region growing and CNN segmentation," *Expert Systems with Applications*, vol. 42, no. 3, pp. 990–1002, 2015.
- [13] M. Sharif, M. Attique Khan, M. Rashid, M. Yasmin, F. Afza, and U. J. Tanik, "Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–23, 2019.
- [14] N. Asada, K. Doi, H. MacMahon et al., "Potential usefulness of an artificial neural network for differential diagnosis of interstitial lung diseases: pilot study," *Radiology*, vol. 177, no. 3, pp. 857–860, 1990.
- [15] S. Katsuragawa and K. Doi, "Computer-aided diagnosis in chest radiography," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 212–223, 2007.
- [16] A. Esteva, B. Kuprel, R. A. Novoa et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [17] Y. Dong, Y. Pan, J. Zhang, and W. Xu, "Learning to read chest X-ray images from 16000+ examples using CNN," in *Proceedings of the 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 51–57, Philadelphia, PA, USA, July 2017.
- [18] D. Dong, Z. Tang, S. Wang et al., "The role of imaging in the detection and management of COVID-19: a review," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 16–19, 2020.
- [19] L. Wang and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," 2020, <http://arxiv.org/abs/2003.09871>.
- [20] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, pp. 635–640, 2020.
- [21] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," 2020, <http://arxiv.org/abs/2003.13815>.
- [22] P. Mooney, "Kaggle X rays dataset," 2020, <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia> Online.
- [23] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, p. 60, 2019.
- [24] D. Ho, E. Liang, and R. Liaw, "1000x Faster Data Augmentation, Berkeley Artificial Intelligence Research," Berkeley, CA, USA, 2019, [https://bair.berkeley.edu/blog/2019/06/07/data\\_aug/](https://bair.berkeley.edu/blog/2019/06/07/data_aug/).
- [25] D. T. Mane and U. V. Kulkarni, "A survey on supervised convolutional neural network and its major applications," *International Journal of Rough Sets and Data Analysis*, vol. 4, no. 3, pp. 71–82, 2017.
- [26] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, pp. 18–22, 2002.
- [27] F. Rustam, A. A. Reshi, I. Ashraf et al., "Sensor-based human activity recognition using deep stacked multilayered perceptron model," *IEEE Access*, vol. 8, pp. 218898–218910, 2020.
- [28] V. K. Ayyadevara, "Gradient boosting machine," in *Pro Machine Learning Algorithms* Springer, Berlin, Germany, 2018.
- [29] F. Rustam, A. Mehmood, S. Ullah et al., "Predicting pulsar stars using a random tree boosting voting classifier (RTB-VC)," *Astronomy and Computing*, vol. 32, Article ID 100404, 2020.
- [30] K. W. Lau and Q. H. Wu, "Online training of support vector classifier," *Pattern Recognition*, vol. 36, no. 8, pp. 1913–1920, 2003.
- [31] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of Biomedical Informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.



- [32] R. Ramteke and K. Y. Monali, "Automatic medical image classification and abnormality detection using k-nearest neighbour," *International Journal of Advanced Computer Research*, vol. 2, p. 190, 2012.
- [33] P. K. Sethy, S. K. Behera, P. K. Ratha, and P. Biswas, "Detection of coronavirus disease (COVID-19) based on deep features and support vector machine," 2020.
- [34] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour Soufi, "Deep-covid: predicting covid-19 from chest x-ray images using deep transfer learning," *Medical Image Analysis*, vol. 65, Article ID 101794, 2020.
- [35] N. N. Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh, "Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays," *IRBM*, 2020.
- [36] E. B. Fatima, B. Omar, E. M. Abdelmajid, F. Rustam, A. Mehmood, and G. S. Choi, "Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: application to fraud detection," *IEEE Access*, vol. 9, pp. 28101–28110, 2021.

## Research Article

# Optimal Economic Modelling of Hybrid Combined Cooling, Heating, and Energy Storage System Based on Gravitational Search Algorithm-Random Forest Regression

Muhammad Shahzad Nazir <sup>1</sup>, Sami ud Din <sup>2</sup>, Wahab Ali Shah,<sup>2</sup> Majid Ali,<sup>2</sup> Ali Yousaf Kharal,<sup>3</sup> Ahmad N. Abdalla,<sup>4</sup> and Padmanaban Sanjeevikumar<sup>5</sup>

<sup>1</sup>Faculty of Automation, Huaiyin Institute of Technology, Huai'an 223003, China

<sup>2</sup>Department of Electrical Engineering, NAMAL Institute Mianwali, Mianwali 42250, Pakistan

<sup>3</sup>College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, China

<sup>4</sup>Faculty of Electronics Information Engineering, Huaiyin Institute of Technology, Huai'an 223003, China

<sup>5</sup>CTiF Global Capsule (CGC), Department of Business Development and Technology, Aarhus University, Herning Campus 7400, Denmark

Correspondence should be addressed to Muhammad Shahzad Nazir; [msn\\_bhutta88@yahoo.com](mailto:msn_bhutta88@yahoo.com)

Received 10 February 2021; Revised 12 April 2021; Accepted 29 April 2021; Published 15 May 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Muhammad Shahzad Nazir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The hybridization of two or more energy sources into a single power station is one of the widely discussed solutions to address the demand and supply havoc generated by renewable production (wind-solar/photovoltaic (PV), heating power, and cooling power) and its energy storage issues. Hybrid energy sources work based on the complementary existence of renewable sources. The combined cooling, heating, and power (CCHP) is one of the significant systems and shows a profit from its low environmental impact, high energy efficiency, low economic investment, and sustainability in the industry. This paper presents an economic model of a microgrid (MG) system containing the CCHP system and energy storage considering the energy coupling and conversion characteristics, the effective characteristics of each microsource, and energy storage unit is proposed. The random forest regression (RFR) model was optimized by the gravitational search algorithm (GSA). The test results show that the GSA-RFR model improves prediction accuracy and reduces the generalization error. The detail of the MG network and the energy storage architecture connected to the other renewable energy sources is discussed. The mathematical formulation of energy coupling and energy flow of the MG network including wind turbines, photovoltaic (PV), CCHP system, fuel cell, and energy storage devices (batteries, cold storage, hot water tanks, and so on) are presented. The testing system has been analysed under load peak cutting and valley filling of energy utilization index, energy utilization rate, the heat pump, the natural gas consumption of the microgas turbine, and the energy storage unit. The energy efficiency costs were observed as 88.2% and 86.9% with heat pump and energy storage operation comparing with GSA-RFR-based operation costs as 93.2% and 93% in summer and winter season, respectively. The simulation results extended the rationality and economy of the proposed model.

## 1. Introduction

The government and legislative authorities incentives to use new energies, concerns about the high and rising price of fossil fuels including its scarcity, and environmental issues are the most important motivations for the integration of renewable energy resources into conventional power

systems [1, 2]. In these circumstances, new technologies such as the combined cooling, heating, and power (CCHP) technology can organically combine heating, cooling, and power supply to realize the cascade utilization of energy, help to improve the utilization rate of energy, and reduce the emission of pollutant gases [3, 4]. In the context of promoting the rapid development of clean energy and the

energy coupling with the microgrid (MG) the CCHP system and MG technology are combined [5, 6]. Many researchers dealt with this concern but the considerable research gap still needs to fill with the impact of the energy coupling of the CCHP and the optimization problem of the MG [7]. The traditional heat-fixed electricity can no longer meet the scheduling requirements as the peak-valley load difference continues to increase. How to realize the coordinated optimization of the three kinds of energy of the CCHP system and improve the flexible adjustment capability of the system is considered as the key issue for the proliferation of these useful systems [8, 9]. The CCHP system can make better advantages of the MG platform. The network mainly includes renewable energy units, energy conversion, and storage units [10]. Its object-oriented areas are smart buildings, isolated islands, and other areas. These systems integrate multiple energy inputs and outputs, and the comprehensive energy utilization rate of the system can reach 80%, which is important for promoting the coordination and complementarity of advantages among multiple energy sources, improving the efficiency of energy use and reducing the emission of undesirable gases [11]. Authors in [12] took the energy supply of commercial buildings as an example, taking the three indicators of the operating cost, energy utilization rate, and CO<sub>2</sub> emission reduction of the CCHP system as the objective function modelling and detailed analysis of the system's sensitivity to changes in various indicators. Authors in [13], based on the characteristics of the CCHP system and mathematical model, accurately characterized the conversion relationship between the three types of energy in the network, such as cold, heat, and electricity, and optimized the operation of the CCHP system under different scheduling strategies. Authors in [14] established the optimal flow model of the energy flow of the MG parameter under the two typical operating modes of the CCHP system and reflected the coupling of the power and natural gas parameter through the energy supply rate index.

A component-based analysis for the CCHP system outlined by [15] includes energy generation units, heat recovery units, and thermal storage systems. Authors in [15] presented comprehensive comparisons of technology and proposed a general method of selecting the correct CCHP program for different applications. Authors in [16, 17] included a detailed overview of the energy system and the requirements for selecting an appropriate system structure. In this study, authors in [18] have discussed the CCHP method with two or more renewable sources of electricity. Authors in [19] reviewed the progress of small-scale energy system production within trigeneration systems. The authors stressed the energy and environmental advantages of small-scale systems and pointed out that the main obstacle to market penetration is the high initial cost [20]. It is found that optimizing micro- and small-scale system operations is harder than large-scale systems due to technical, law, and policies restrictions. Authors [21] had identified the profits of polygeneration systems concerning zero energy constructions. In this study, authors [22] presented the CCHP system equipment evaluation, planning, operation, and

component modelling. However, the CCHP system can cool and heat during power supply and these energy sources cannot be absorbed in the system at the same time, resulting in low energy utilization or waste [23]. Therefore, to meet the constraints of the distributed microsource and energy storage unit operating characteristics, energy supply, and demand balance, the advantage of the proposed GSA-RFR model is its generalization ability. However, due to the different energy source characteristics of different systems, the input characteristics of the prediction model should be selected according to the actual condition of energy source so as to improve the accuracy of the model in specific period. So, this paper proposes the GSA-RFR approach in the CCHP operation scheduling model including heat pumps and energy storage units into the MG system. The detailed scheduling model for MGs with heat pumps and energy storage including wind turbines, photovoltaic cells, microgas turbines, ground source heat pumps, and energy storage units is presented. Considering the energy coupling characteristics in the grid and the operating characteristics of each distributed microsource and energy storage unit, the optimum output of each microsource and energy storage unit and the total cost of the system compared with optimized approaches, proposed GSA-RFR based approach, shows comparatively better performance.

## 2. Microgrid Energy System Description

In grid-connected insulated modes, the characteristics of MG are flexible operating and can improve grid efficiency and safety [24]. If the MG unit can acquire refrigeration power from the grid, the unit can maintain a stable system frequency if it monitors the voltage of the grid. However, the MG antennas are designed to be used on remote islands, where a primary concern is the control of this type of frequency [25]. The MG concept is proposed by the consortium for solutions in electric reliability systems (CERTS) [26]. The CERTS can be characterized as a decentralized entity consisting of multienergy resources and controllable electric and thermal charges. These storage devices are connected to the upstream power generation grid using photovoltaic panels, wind farms, fuel cells, CCHP system, and microturbines (e.g., batteries or super-capacitor) [27]. The electrical grid can be seen as a regulated cell of the power system from an electrical utility perspective. As a part of the point of view of the customers, the MG can be designed carefully to meet the requirements for reliability as well as energy savings, improving efficiency, minimizing voltage sag, and powering a continuous current [28].

The MG with the energy storage system has become a promising component of future implementation of the smart grid [29]. Unlike grid power, however, the system's renewables do not supply a steady stream, matching supply needs as they change, and as a result, the MG system oscillators often do not lower loads or alter their frequency. Therefore, to avoid sudden fluctuations in energy from renewable sources, storage systems are needed to help balance out low-power and high-power systems [30]. The MG structure and energy conversion process with heat

pump and energy storage in this paper are depicted in Figure 1. The wind turbines and photovoltaic cells belong to renewable energy units, which use natural energy to generate electricity, and natural gas consumption units including gas boilers, fuel cells, and microgas turbines are depicted in Figure 1. They use natural gas combustion power to provide energy, and microgas turbines and bromine-cooled units have covert the CCHP system and can recycle high-temperature flue gas during power generation [31, 32].

When the supply of cold (heat) power is insufficient, it can be used to make up the shortage while the gas boiler is used as auxiliary supply equipment for cold (heat) load [31, 32]. The energy storage unit consists of three parts, battery, cold storage, and hot water storage tank. The voltage level of the MG system is 380 V, which is connected to the 10 kV medium-voltage distribution parameter through a common node.

### 3. Proposed Model

**3.1. The System Model Description.** The optimal scheduling model of the MG with heat pump and energy storage established is a mixed-integer nonlinear programming problem. The general solution expression of the model is as follows:

$$\begin{cases} \min & f(x, y) \\ \text{s.t.} & g_i(x, y) = 0 \\ & h_i(x, y) = 0 \\ & x_{\min} \leq x \leq x_{\max} \\ & y \in \{0, 1\} \\ & i = 1, 2, \dots, n \\ & j = 1, 2, \dots, m, \end{cases} \quad (1)$$

where the optimized scheduling variable  $x$  represents the output of each microsource and energy storage unit, the power consumed by the ground source heat pump, the transmission power with the external power grid, and so on. The inequality constraints include the maximum power constraint of the power grid and the operating characteristics constraints of each microsource and energy storage unit. To maintain the safety of the operation process of the MG system, it is necessary to take into account the constraints of the system's energy balance, unit start-up and shut down, and so on in actual operation.

$$\min F_M = \sum_{t=1}^T [C_E(t) - C_C(t) + C_M(t) + C_F(t) + C_S(t)], \quad (2)$$

where  $F_M$  is the total operating cost of the system,  $T$  is the optimization period,  $C_F(t)$  is the unit fuel cost,  $C_M(t)$  is the operation and maintenance cost, and  $C_S(t)$  is the unit start-up cost during the  $t$  period, respectively. The interactive cost of electric energy  $C_E(t)$  and cooling (heating)  $C_C(t)$  is presented.

The fuel cost is calculated as follows:

$$C_F(t) = \frac{C_{\text{NG}}}{L_{\text{NG}}} \left[ \frac{P_{\text{MT}}(t)}{\eta_{\text{MT}}} + \frac{P_{\text{FC}}(t)}{\eta_{\text{FC}}} + \frac{Q_{\text{GB}}(t)}{\eta_{\text{GB}}} \right], \quad (3)$$

where  $C_{\text{NG}}$  and  $L_{\text{NG}}$  represent the unit cost of natural gas and low calorific natural gas,  $Q_{\text{GB}}(t)$  is the output thermal power efficiency of the gas-fired boiler,  $P_{\text{FC}}$  is the fuel cell power,  $\eta_{\text{FC}}$  is the fuel cell efficiency, respectively, while  $t$  is the output of the controllable unit at the time  $t$ .

The operation and maintenance cost is calculated as follows:

$$C_M(t) = \sum_{t=1}^N P_t^{\text{CG}}(t)K_{M,j} + \sum_{j=1}^M P_j^{\text{RG}}(t)K_{M,j} + [P_{\text{ES,ch}}(t) + P_{\text{ES,dis}}(t)]K_{\text{ES}}, \quad (4)$$

where  $K_{M,t}$  is the unit maintenance cost for generating unit;  $K_{M,j}$  is the number of controllable units;  $P_{\text{CG},i}(t)$  is the output of renewable energy unit  $t$  period;  $K_{\text{ES}}$  is the unit maintenance cost of the energy storage device; and  $M$  indicates the controllable unit  $j$  in the start-stop state during time  $t$ .

The unit start-up cost is calculated as follows:

$$C_S(t) = \sum_{i=1}^N \max\{0, V_i(t) - V_i(t-1)\}C_{S,i}, \quad (5)$$

where  $V_i(t) = 1$  is the start-up state; otherwise, it is the shutdown state;  $C_{S,i}$  is the controllable unit start-up cost.

The electricity cost is calculated as follows:

$$C_E(t) = K_{\text{sup}}(t)P_{\text{grid,sup}}(t) - K_{\text{dem}}(t)P_{\text{grid,dem}}(t), \quad (6)$$

where  $P_{\text{grid,sup}}(t)$  and  $P_{\text{grid,dem}}(t)$  are the power supply and demand power of the power grid during the period  $t$  and  $K_{\text{dem}}(t)$  is the electricity demand price.

The refrigeration (heat) profit is calculated as follows:

$$C_C(t) = K_c Q_{\text{co}}(t) + K_h Q_{\text{bc}}(t), \quad (7)$$

where  $K_c$  and  $K_h$  are the predicted costs of cooling and heat load during the period, respectively;  $Q_{\text{co}}(t)$  is the unit cooling and  $Q_{\text{he}}(t)$  is the heating source.

The energy balance constraints are calculated as follows:

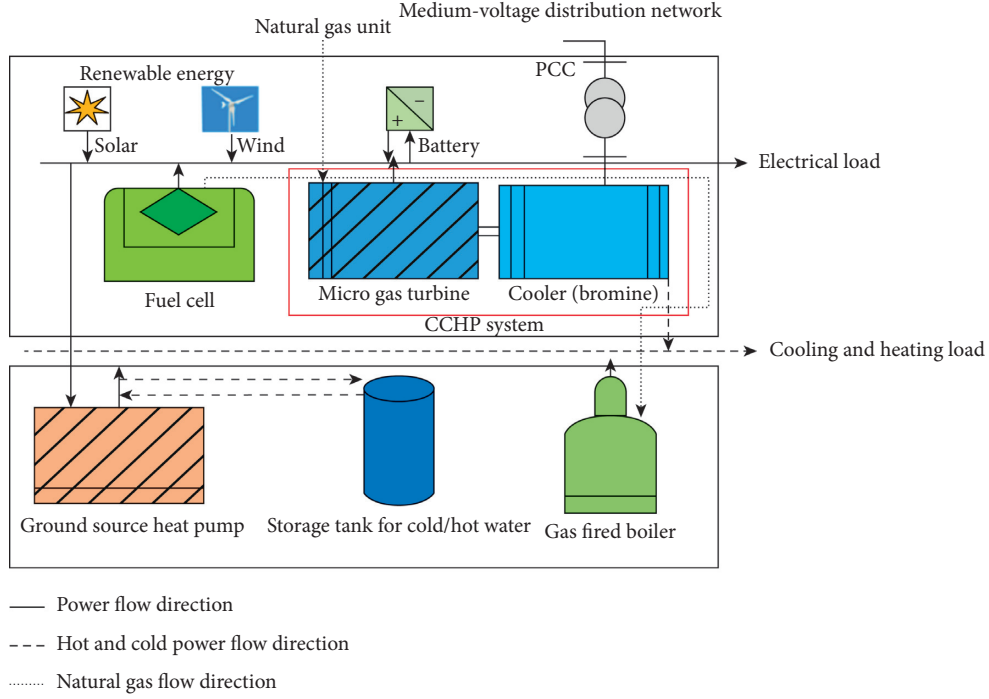


FIGURE 1: Microgrid energy design and power flow.

$$\sum_{i=1}^N P_i^{CG}(t) + P_{\text{grid}}(t) + \sum_{j=1}^M P_j^{RG} + P_{\text{BS},\text{dis}}(t) - P_{\text{BS},\text{ch}}(t) = P_{\text{load}}(t) + P_{\text{HP}}(t), \quad (8)$$

$$Q_{\text{MTC}}(t) + Q_{\text{HPC}}(t) + Q_{\text{CS},\text{dis}}(t) - Q_{\text{CS},\text{ch}}(t) = Q_{\text{co}}(t),$$

$$Q_{\text{MTH}}(t) + Q_{\text{HPH}}(t) + Q_{\text{HS},\text{dis}}(t) - Q_{\text{HS},\text{ch}}(t) = Q_{\text{he}}(t),$$

where  $P_{\text{BS}}(t)$  is the power of the storage battery,  $Q_{\text{CS}}(t)$  is the cold storage, and  $Q_{\text{HS}}(t)$  is hot water storage tank in the  $t$  period, while  $P_{\text{load}}(t)$  is the predicted cost of the electric load in the  $t$  period.

The grid power constraints are calculated as follows:

$$0 \leq P_{\text{grid},\text{sup}}(t) \leq V_{\text{sup}}(t)P_{\text{sup},\text{max}},$$

$$0 \leq P_{\text{grid},\text{dem}}(t) \leq V_{\text{dem}}(t)P_{\text{dem},\text{max}}, \quad (9)$$

$$V_{\text{sup}}(t) + V_{\text{dem}}(t) \leq 1,$$

where  $P_{\text{sup},\text{max}}$  is the maximum power supply and  $P_{\text{dem},\text{min}}$  is the demand power for the grid. The multiple energy sources coordinate the coupling and conversion to minimize the total operating cost of the system.

The controllable unit constraints are calculated as follows:

$$P_{i,\text{min}}^{CG} \leq P_i^{CG}(t) \leq P_{i,\text{max}}^{CG},$$

$$(T_i^{\text{CG}}(t-1) - T_{V,f})(V_i(t-1) - V_f(t)) \geq 0,$$

$$(T_i^{\text{off}}(t-1) - T_{D,i})(V_i(t) - V_i(t-1)) \geq 0, \quad (10)$$

where  $T_i^{\text{on}}$  is the minimum start-up time and  $T_i^{\text{off}}$  is the minimum shutdown time of unit  $i$  and  $j$ ,  $C_{\text{ES}}$  is the capacity

of the energy storage unit,  $P_{\text{ch},\text{max}}$  is the maximum input power, and  $P_{\text{dis},\text{max}}$  is the maximum output power of the energy storage.

The energy storage unit constraints are calculated as follows:

$$\lambda_{\text{min}}C_{\text{ES}} \leq E_{\text{ES}}(t) \leq \lambda_{\text{max}}C_{\text{ES}},$$

$$0 \leq P_{\text{ES},\text{dis}}(t) \leq V_{\text{dis}}(t)P_{\text{dis},\text{max}}, \quad (11)$$

$$0 \leq P_{\text{ES},\text{ch}}(t) \leq V_{\text{ch}}(t)P_{\text{ch},\text{max}},$$

$$V_{\text{dis}}(t) + V_{\text{ch}}(t) \leq 1,$$

where  $\lambda_{\text{max}}$  and  $\lambda_{\text{min}}$  are the maximum and minimum state of the energy storage, respectively.

**3.2. The GSA-RFR Algorithm.** The GSA algorithm needs to determine the fitness function to evaluate the advantages and disadvantages of the RFR model corresponding to each node. The crossover will produce a new particle as follows:

$$X_{\text{inew}}^k = rX_i^k + (1-r)X_j^k,$$

$$V_{\text{inew}}^k = \frac{V_i^k + V_j^k}{|V_i^k| + |V_j^k|} |V_j^k|, \quad (12)$$

where  $r$  is a random number between 0 and 1;  $V_i$  and  $V_j$  are the velocities of particles  $X_i$  and  $X_j$ ; and  $X_i$  new and  $V_i$  new are the positions and velocities of the new particle, which will replace  $X_i$ . For excellent particles, the strategy of dynamic updating inertia factor is adopted.

In the early stage,  $w$  is selected to enhance the global searchability. The smaller  $w$  was selected in the later stage to achieve a more sophisticated search [33]. The update formula of the inertia factor is shown in the following equation:

$$w(t) = (w_1 - w_2) \times \frac{(T - t)}{T + w_2} \quad (13)$$

The mean square of residual was selected as follows:

$$R_{RF}^2 = 1 - \frac{MSE_{\text{ooB}}}{\sigma_y^2}, \quad (14)$$

where  $\sigma_y^2$  is the variance of the predicted cost and  $R_{RF}^2$  is the mean square of residual error. The random forest can calculate the importance of each input feature [33], as shown in the following equation:

$$f_i = \frac{\sum_{j \in \text{feature}_i} n_j}{\sum_k n_k}, \quad (15)$$

$$n_k = w_k M_k - w_{k1} M_{k1} - w_{k2} M_{k2}, \quad (16)$$

where  $n_k$  is the importance of node  $k$ ;  $n_j$  is the node with the feature  $I$  as the feature division;  $W_k$ ,  $W_1$ , and  $W_2$  are the proportion of the number of samples in node  $k$  and its subnodes to all the samples, respectively; and  $M_k$ ,  $M_1$ , and  $M_2$  are the mean square errors of node  $k$  and its subnodes.

The procedure of the proposed algorithm is shown in Figure 2.

## 4. Result and Discussion

**4.1. System Parameter.** In this paper, the MG system under grid connection is selected as a case study and is optimized. The network mainly includes wind turbine, photovoltaic cells, fuel cells, CCHP, and energy storage units. The specific model shows the advantages of clean energy, wind turbines, and photovoltaic cells which are given priority in their output, and the MPPT operation mode is adopted. Figure 3 shows the predictive curves of wind, photovoltaic output, cooling, heating, and electrical loads assuming that both the heating and cooling power in summer in the MG parameter are zero where the optimal scheduling period is  $T = 24$  h and the unit time  $\Delta t = 1$  h.

The power generation cost of the unit is not included, only the operation and maintenance cost of the new energy unit is considered. For the different energy supply requirements in summer and winter in the MG parameter, two typical seasons were selected for analysis, and corresponding optimization scheduling strategies were formulated. The operating parameters of the microenergy parameter are shown in Table 1, the energy storage unit parameters are shown in Table 2, and the other parameters are shown in Table 3. Assuming that the microturbine, fuel cell, and

ground source heat pump are in the shutdown state at the initial operation, the start-up costs of the unit are 1.94, 2.21, and 1.32 \$, respectively.

### 4.2. Analysis of Typical Season

**4.2.1. Summer Season.** To validate the economic and energy-saving effects of the proposed model, the operation optimization model without heat pump and cold storage unit was selected for comparison. On a certain summer day, the optimization results without ground source heat pump and energy storage unit are shown in Figure 4. In Figure 4, the net load cost is the predicted cost of the electrical load minus the wind turbine and photovoltaic output without ground source heat pump and cold storage unit.

The CCHP system must first supply the demand for cooling load, and the microgas turbine operates in the mode of "fixing electricity with cold." If the cooling load is insufficient, the gas boiler will make up the shortfall. In this mode of operation, the CCHP system needs to follow the changes in cooling power at all times. Due to the restriction of the energy coupling relationship, the adjustment capability of its electric power is greatly restricted, and it cannot be added to the optimized operation of the system autonomously. Therefore, the cost of the system in this operating mode is relatively high. The optimization results of electricity and cooling power of the MG with heat pump and cold storage in summer are shown in Figure 5.

The results show that the cooling load in the MG is jointly met by the CCHP system, energy storage unit, and heat pump device. The power required by the heat pump and the electrical load in the grid is composed of wind turbines, photovoltaic cells, grid power, CCHP systems, fuel cells, and battery equipment connect during the period of 0–7 and 23–24, and the demand for electricity and cooling load of the MG is relatively low. At this time, the electricity supply price of the external power grid is the lowest, and the power generation costs of distributed microsources (microturbines and fuel cells) in the grid are higher than the electricity supply price of the power grid. Therefore, the electrical load is first to supply from the grid to supply the user's electrical energy needs, and the shortage is supplemented by the fuel cell, the cold load is supplied by the ground source heat pump, and the CCHP system is in a shutdown state during this period. Since the heat pump cooling consumes less electric power, the supply of electrical load can fully meet the requirements during the periods of 7–10, 15–17, and 20–22, and the supply and sale price of external power grids were moderate. The generation cost of distributed microsources in the grid is higher than the power supply price and lower than the price of power demand. The insufficient part chooses to supply power from the grid microsources of the system which is preferentially called to meet the user's electrical energy demand. If the demand for the cooling load is small, the heat pump is driven by electric energy for cooling; if the demand for the cooling load is large, the CCHP system will start for cooling energy during the period of 10–15 and 18–21, and the price of supply electricity and

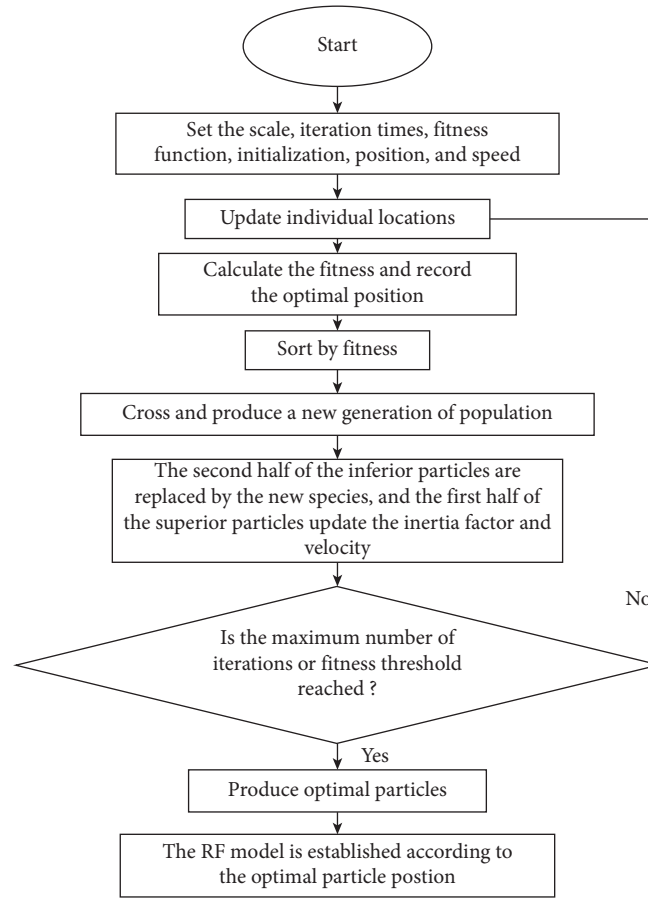


FIGURE 2: GSA-RFR algorithm procedure.

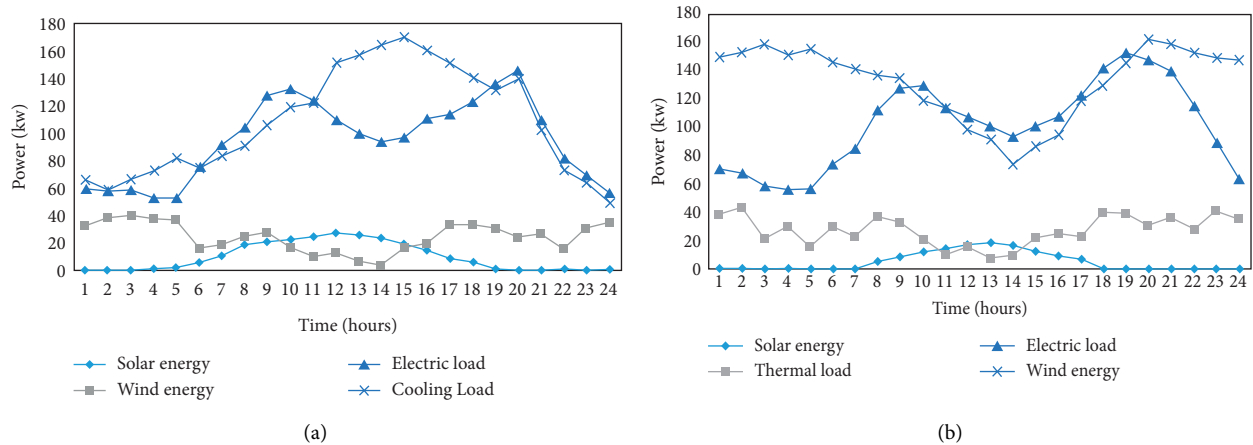


FIGURE 3: Predictive curves of the wind, photovoltaic output, cooling, heating, and electrical loads in (a) summer season and (b) winter season.

TABLE 1: Operating parameters of the MG network.

Source type	Minimum power (kW)	Maximum power (kW)	Minimum power (h)	Minimum shutdown (h)	Maintenance cost (\$/kW.h)
Wind	0	50	—	—	0.029
Photovoltaic	0	30	—	—	0.025
Power grid	-60	60	—	—	—
Gas turbine	15	65	3	2	0.025
Fuel cell	5	40	3	2	0.028
GSHP	0	30	2	2	0.027

TABLE 2: Energy storage unit parameters.

Self-consumption of energy storage	Charging and discharging	Self-consumption	Maximum input power (kW)	Maximum output power (kW)	Minimum state	Maximum state	Initial cost (kW.h)	Capacity (kW.h)	Maintenance cost (\$/kW.h)
Electric energy storage	0.99	0.001	37.5	37.5	0.2	0.8	30	150	0.0018
Thermal energy storage	0.89	0.01	25	25	0	0.9	10	100	0.0016
Cold energy storage	0.90	0.01	25	25	0	0.9	10	100	0.0015

TABLE 3: Parameter of MG network.

Parameter	Cost	Parameter	Cost	Parameter	Cost
$C_C$	1.40	$\lambda_L$	0.16	$\lambda_{GB}$	0.9
$C_H$	1.31	$\lambda_{rec}$	0.87	$\lambda_{co}/\$/kWh$	0.1
$C_{HPC}$	3.0	$\lambda_{HPH}$	2.98	$\lambda_{he}/\$/kWh$	0.1

sold by the external power grid is the highest. The electricity demand price of the power grid is higher than the generation cost of distributed microsources, and each microsource is at rated power as much as possible and demands electricity to obtain economic profits. The CCHP system is selected to supply the cooling load demand, and the heat pump equipment is shut down during the entire optimized dispatch period. When the external electricity price is low, releases the electrical energy is released; when the external electricity price is high, the battery stores electrical energy. The charging and discharging state is determined by the grid electricity price. The cold storage unit can absorb the cold energy output by the ground source heat pump at night and release it when the cold load demand is high during the day. The energy storage unit can realize the peak-valley difference transfer of the load within the grid, thereby reducing the total operating cost of the MG.

*4.2.2. Winter Season.* According to the heating needs of users in winter, the operating state of the heat recovery device in the MG parameter CCHP system is switched to operate in the combined heat and power mode, and the cycle state of the heat pump compressor is changed for heating. On typical winter days, the optimization results of MG power and thermal power without heat pump and heat storage are shown in Figure 6, and the optimization results with the heat pump and heat storage are shown in Figure 7. A brief analysis of the operation optimization results of the winter MG system is presented in Table 4 and Figure 7.

*4.2.3. Optimum of Economic.* To test the advantages of introducing ground source heat pumps and energy storage units in the MG, the peak-cutting, valley-filling, and energy utilization indexes in the grid were analysed. The peak-cutting and valley-filling index  $F_n$  is evaluated by the minimum square sum of various load change rates during the dispatch period, and the energy utilization index  $F_m$  is evaluated by the total energy input and output ratio

presented in (10). The heat pumps and energy storage units into the MG system have significant advantages in terms of economics, peak and valley filling, and energy utilization. The main energy source of the MG is the external power grid and natural gas power generation. The time-phase electricity cost curve of electric energy is shown in Figure 8, the unit cost of natural gas is 2.52  $\$/m^3$ , and the low heating cost is 9.68  $kWh/m^3$ . The electricity cost curve of the microgrid indicated and authenticated the proposed GSA-RFR optimization, while without optimization and GA, PSO approaches show comparatively higher values, which are not in favour of system operation (Figure 8). The cost of the MG parameter under the two scheduling methods in summer is shown in Table 5. It can be seen from the comparison of the operating costs of the MG in Table 5 that the introduction of heat pumps and cold storage units has good economic profits for the MG. The gas cost of the microsource during the optimization period is reduced by 24.9%, and the total operating cost of the system is reduced by 15.8%.

The demand for cold load in summer is low, and the demand for heat load in winter is high. There is a certain difference between the two. If the heat pump is used to supply electricity from the power grid for heating at night and the CCHP system is not started, the energy supply needs of users cannot be met. If the unit capacity of the ground source heat pump is further increased, there will be a situation where the equipment utilization rate is low and the initial investment cost is higher. The costs of the MG parameter under the two scheduling methods in winter are shown in Table 5. From the optimization results in Table 5, it can be seen that the gas cost of the microsource during the winter optimization period is reduced by 18.9%, and the total operating cost of the system is reduced by 17.6%.

The peak-shaving, valley-filling, and energy utilization indicators of the MG are shown in Table 4. On the analysis of the data, the peak-shaving and valley-filling index is higher than that of GSA-RFR presented in Table 4. The energy storage unit can shift a load of electricity, cooling, and heat from the peak period to the valley period, thus effectively reducing the load on the network peak-valley difference. The heat pump CCHP cost is higher, which can output heat energy that consumes three times the electrical energy. Its efficiency is much higher than that of common energy equipment, so the energy utilization rate is higher. Due to the high energy efficiency coefficient of CCHP cooling in summer, the cost of obtaining the same energy is lower, so



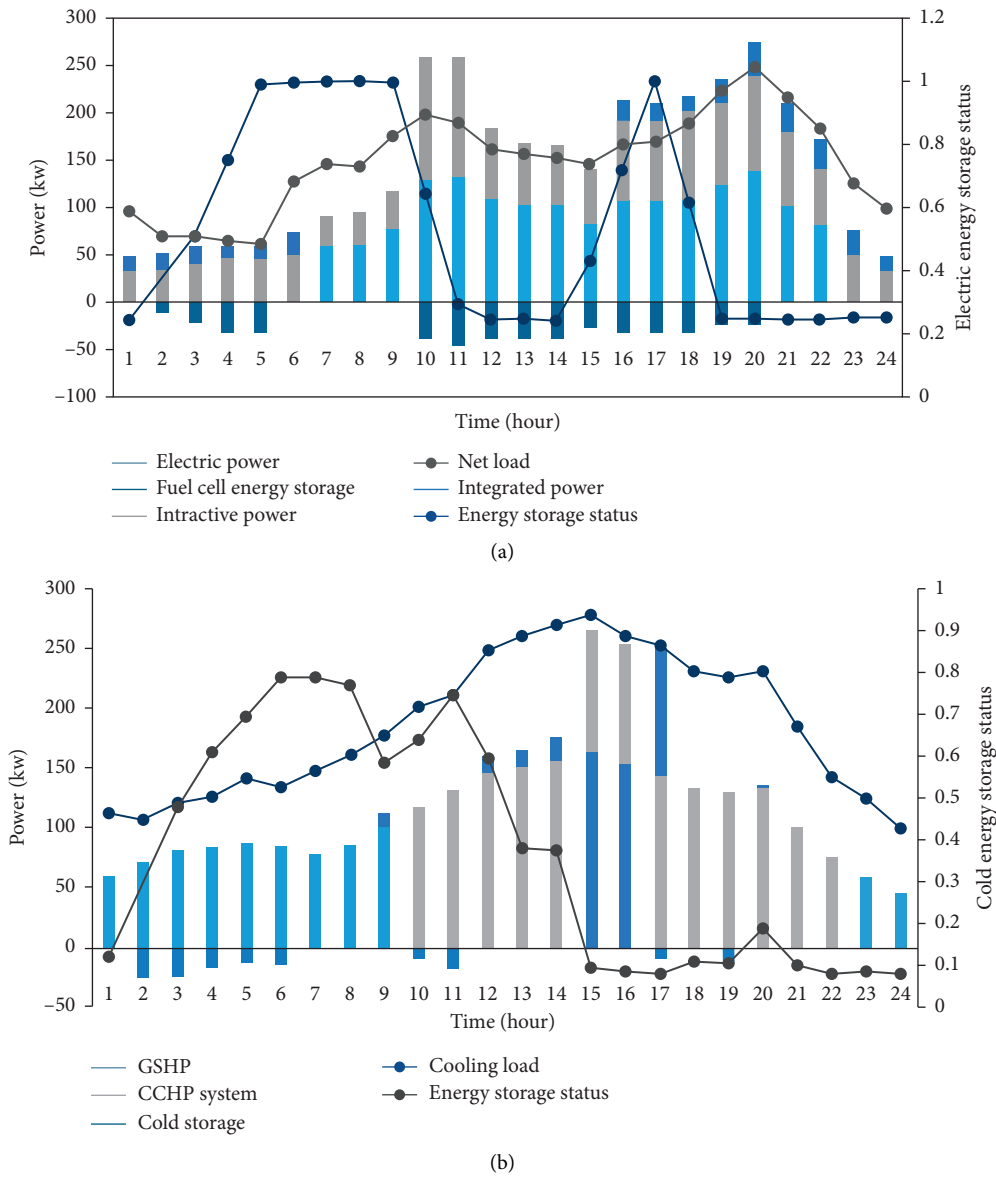


FIGURE 4: Summer heat pump and cold storage optimization.

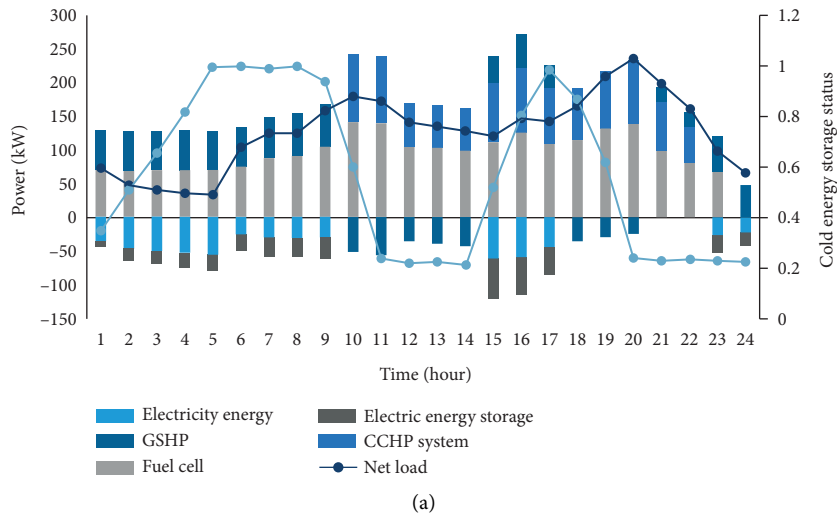


FIGURE 5: Continued.

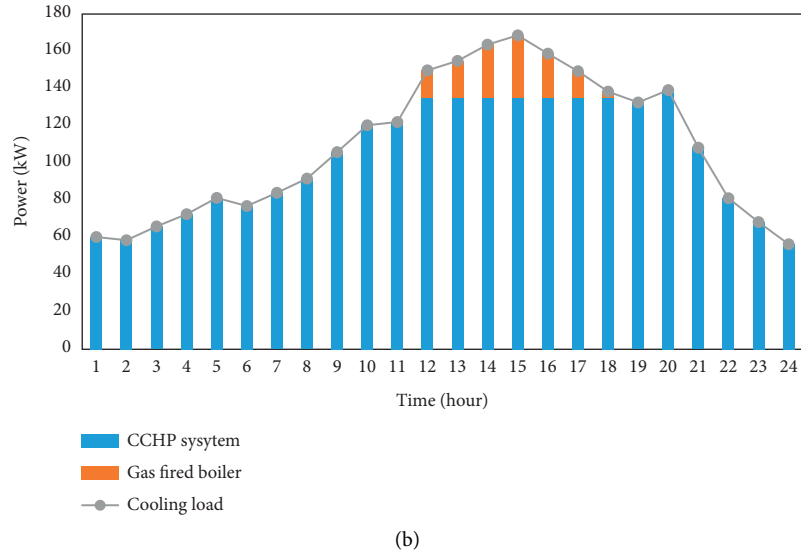


FIGURE 5: Optimization results with heat pump and cold storage in summer: (a) electric power and (b) cooling power.

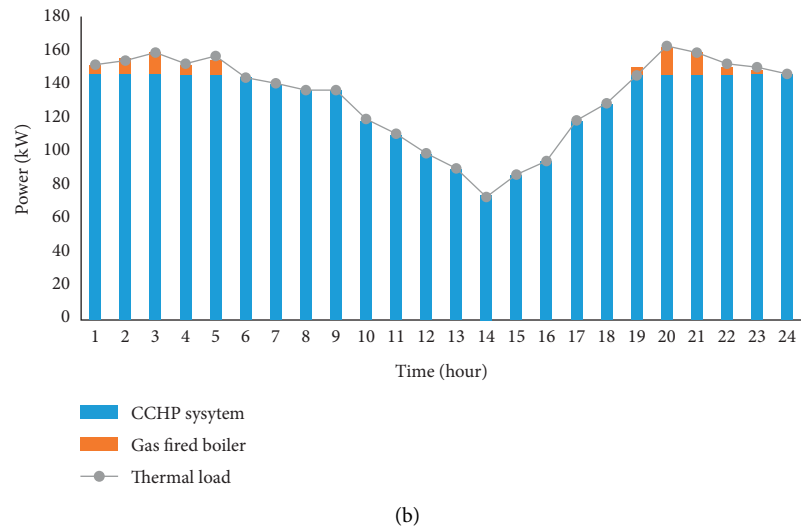
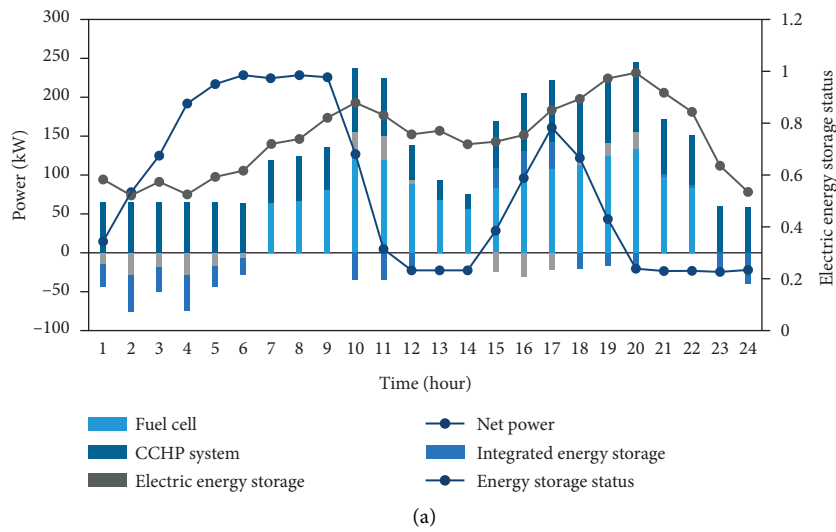


FIGURE 6: Optimization results in winter without heat pump and heat storage: (a) electric power; (b) thermal power.

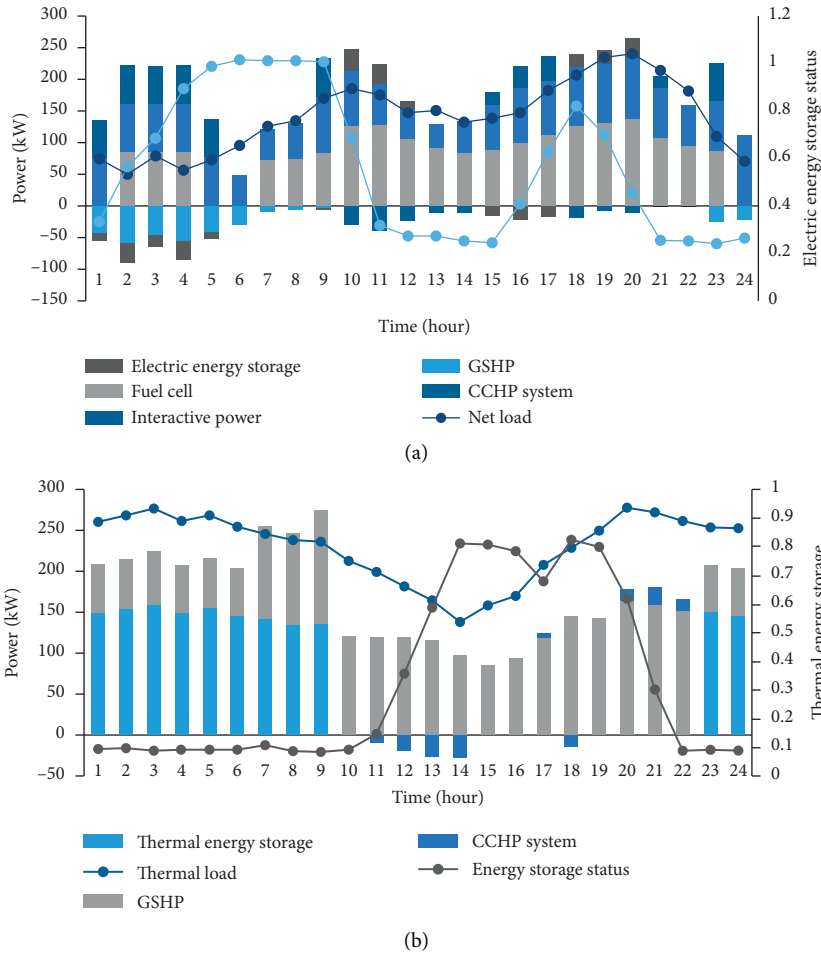


FIGURE 7: Optimization results in winter with heat pump and heat storage: (a) electric power; (b) thermal power.

TABLE 4: Peak-shaving, valley-filling, and energy efficiency indicators under the two modes.

Operation mode	Season	Peak shaving and valley filling/kW.h	Energy efficiency (%)
With heat pumped	Summer	460	88.2
	Winter	369.5	86.9
GSA-RFR	Summer	428.9	93.2
	Winter	343	93

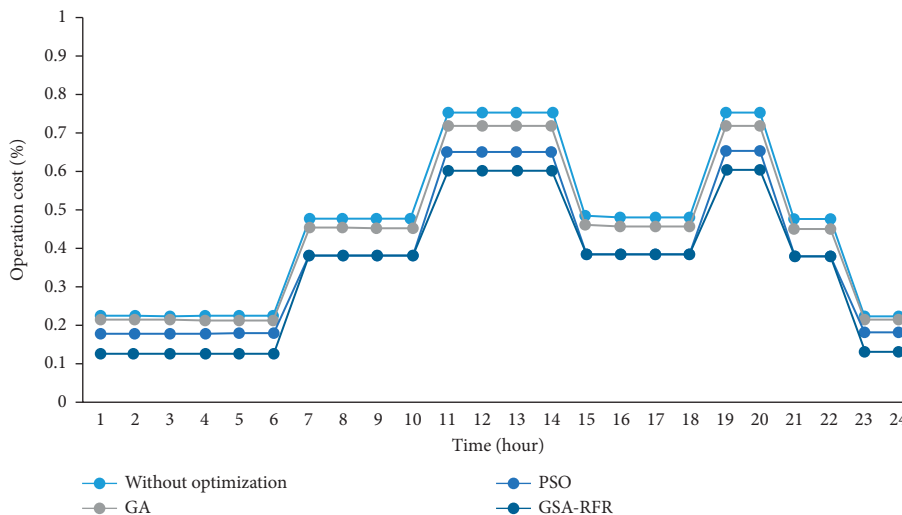


FIGURE 8: Electricity cost of microgrid.

TABLE 5: Costs of the MG parameter under the two scheduling methods.

Parameters	Cost (\$/day)			
	Summer		Winter	
Unit fuel cost	With heat pump and cold storage	GSA-RFR optimization	With heat pump and cold storage	GSA-RFR optimization
Gas turbine	220.9	275.2	1241	947
Fuel cell	39.4	0	209	245
Gas-fired boiler	89.6	263.9	24.5	0
Energy interaction cost	181.8	202.7	68	141.2
Electricity supply cost	63.8	68.8	122	127.2
Electricity demand				
Operation and maintenance cost	3.98	9	73	71.9
The unit starts and stop cost	258.7	260.3	4.19	9.3
Heating profit	1103.8	842.7	316.9	317
Total operation cost	1030.5	689.7	1182	970

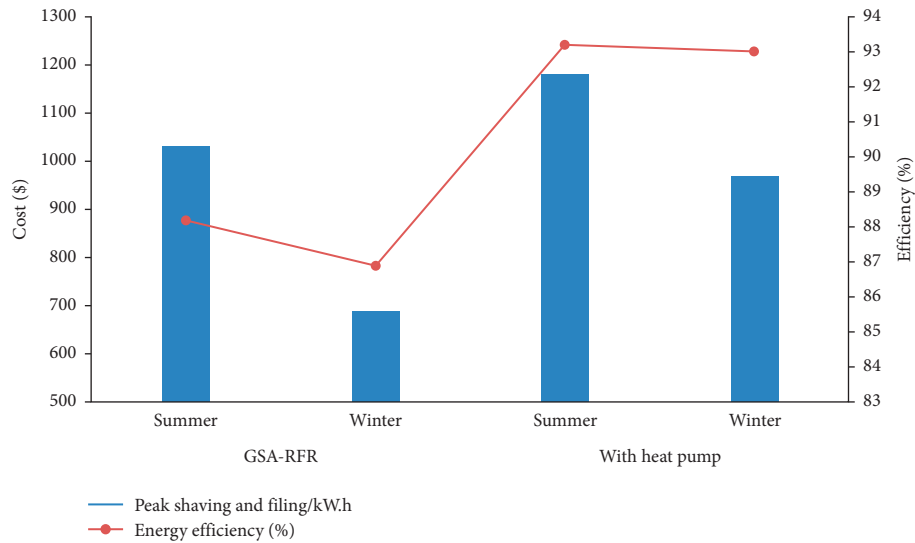


FIGURE 9: A final comparison of GSA-RFR and with heat pump.

the energy utilization rate in summer is slightly higher than that in winter.

To analyse the advantages of the different modes, a comparative analysis of the comprehensive cost of the MG under the two scheduling modes is carried out, as shown in Figure 9. It can be seen from depicted results that GSA-RFR brings significant economic profit to the MG parameters compared with other modes and reduces the operation cost by 3.6%.

## 5. Conclusion

The perfection in the modelling of the energy storage system with economic optimization characteristics is the key features of next-generation energy technologies. Nonetheless, there are still issues to developing a physically attractive/efficient and energy storage system that is cost-effective for electronic as well as hybrid vehicles. The model we are going to use to test this is a mixed-integer program. By given data,

the integrated parameters, the output cost, and the total cost of the grid are obtained. The simulation verification shows that the integration of the heat pumps and energy storage units into the MG parameter can improve the coupling relationship of the three energy sources of the CCHP system. The MG sources including wind, photovoltaic, CCHP systems, fuel cells, and energy storage unit complementary and coordinated operation are also realized. According to the analysis of the peak-cutting, valley-filling, and energy utilization indexes, the heat pump can improve the energy utilization rate and reduce the natural gas consumption of the microgas turbine, and the energy storage unit has the function of realizing load peak cutting and valley filling. Energy efficiency's numerical costs validated the proposed GSA-RFR optimization, which is calculated as 93.2% and 93%, comparing 88.2% and 86.9% of heat pump and energy storage costs in the summer and winter season, respectively. Owing to the limitations of the RFR model, if the actual cost exceeds the range, the prediction result may produce a

greater deviation. This problem can be improved by expanding the range nodes. The proposed model provided a certain reference for the modelling planning and optimal scheduling of the MG parameter. This study is expected to be a significant contribution concerning the maturity of energy storage technologies for microgrid application, which is likely to dominate the electricity market need.

## Abbreviations

$C_{CH}$ :	Cooling (heating) profit of the system
CCHP:	Combined cooling, heating, and power
$C_E$ :	Interactive cost of electric energy
CERTS:	Consortium for solutions in electric reliability systems
$C_{ES}$ :	The capacity of the energy storage
$C_F$ :	Unit fuel cost
$C_M$ :	Operation and maintenance cost
$C_{NG}, L_{NG}$ :	Unit cost of natural gas and low calorific natural gas
$C_S$ :	Unit start-up cost
$C_{S,i}$ :	Controllable unit start-up cost
$F_M$ :	Total operating cost of the system
$K_c, K_h$ :	Predicted costs of cooling and heat load
$K_{dem}$ :	Electricity demand price
$K_{ES}$ :	Unit maintenance cost of the energy storage device
$K_{M,j}$ :	Number of controllable units
$M$ :	Controllable unit
MG:	Microgrid
$P_{BS}$ :	Power of the storage battery
$P_{CGi}$ :	Output thermal power efficiency of the gas-fired boiler
$P_{ch, max}, P_{dis, max}$ :	Charging/discharging output power
$P_{dem, min}$ :	Demand power
$P_{grid}$ :	Active power
$P_{grid, pur}$ :	Supply and demand power
$P_{grid, dem}$ :	
$P_{load}$ :	Predicted cost of the electric load
$P_{pur, max}$ :	Maximum supply and demand of power
PV:	Photovoltaic
$Q_{co}$ :	Unit cooling
$Q_{CS}$ :	Cold storage
$Q_{GB}$ :	Low calorific cost of natural gas
$Q_{grid}$ :	Reactive power
$Q_{he}$ :	Heating source
$Q_{HS}$ :	Hot water storage tank
$t$ :	Time
$T$ :	Optimization period
$T_i^{off}$ :	Shutdown time
$T_i^{on}$ :	Start-up time
$\lambda_{max}$ and $\lambda_{min}$ :	Maximum and minimum state of the energy storage.

## Data Availability

The data used to support this study are cited in the manuscript.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

M. S. N. was responsible for conceptualization, formal analysis, and methodology of the study and wrote the original draft. S. D. reviewed and edited the manuscript and developed software. W. A. S. validated the study and reviewed and edited the manuscript. M. A. performed formal analysis. A. Y. K. was responsible for original draft preparation and investigation. A. N. A. was involved in conceptualization, investigation, formal analysis, and reviewing and editing of the manuscript. P. S. was involved in investigation and reviewing and editing of the manuscript.

## Acknowledgments

All the authors are highly grateful to their affiliated institutes and universities for providing all the necessary services.

## References

- [1] J. A. Aguilar-Jiménez, L. Hernández-Callejo, V. Alonso-Gómez et al., "Techno-economic analysis of hybrid PV/T systems under different climate scenarios and energy tariffs," *Solar Energy*, vol. 212, pp. 191–202, 2020.
- [2] M. S. Nazir, F. Alturise, S. Alshmrany et al., "Wind generation forecasting methods and proliferation of artificial neural network: a review of five years research trend," *Sustainability*, vol. 12, no. 9, p. 3778, 2020.
- [3] M. Ameri and Z. Besharati, "Optimal design and operation of district heating and cooling networks with CCHP systems in a residential complex," *Energy and Buildings*, vol. 110, pp. 135–148, 2016.
- [4] M. S. Nazir, Y. Wang, A. J. Mahdi, X. Sun, C. Zhang, and A. N. Abdalla, "Improving the performance of doubly fed induction generator using fault tolerant control-A hierarchical approach," *Applied Sciences*, vol. 10, no. 3, p. 924, 2020.
- [5] P. R. Díaz, Y. Benito, and J. Parise, "Thermoeconomic assessment of a multi-engine, multi-heat-pump CCHP (combined cooling, heating and power generation) system—a case study," *Energy*, vol. 35, no. 9, pp. 3540–3550, 2010.
- [6] A. M. Bonanos and E. V. Votyakov, "Analysis of thermocline thermal energy storage systems with generic initial condition algebraic model," *Solar Energy*, vol. 213, pp. 154–162, 2021.
- [7] S. Javan, V. Mohamadi, P. Ahmadi, and P. Hanafizadeh, "Fluid selection optimization of a combined cooling, heating and power (CCHP) system for residential applications," *Applied Thermal Engineering*, vol. 96, pp. 26–38, 2016.
- [8] M. Liu, Y. Shi, and F. Fang, "Combined cooling, heating and power systems: a survey," *Renewable and Sustainable Energy Reviews*, vol. 35, pp. 1–22, 2014.
- [9] W. Stanek, W. Gazda, and W. Kostowski, "Thermo-ecological assessment of CCHP (combined cold-heat-and-power) plant supported with renewable energy," *Energy*, vol. 92, pp. 279–289, 2015.
- [10] M. Soltani, M. Chahartaghi, S. Majid Hashemian, and A. Faghieh Shojaei, "Technical and economic evaluations of combined cooling, heating and power (CCHP) system with

- gas engine in commercial cold storages,” *Energy Conversion and Management*, vol. 214, p. 112877, 2020.
- [11] X. Zuo, M. Dong, F. Gao, and S. Tian, “The modeling of the electric heating and cooling system of the integrated energy system in the coastal area,” *Journal of Coastal Research*, vol. 103, no. 1, pp. 1022–1029, 2020.
- [12] B. Cao, W. Dong, Z. Lv, Y. Gu, S. Singh, and P. Kumar, “Hybrid microgrid many-objective sizing optimization with fuzzy decision,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 11, pp. 2702–2710, 2020.
- [13] J. Król and P. Ochoń, “Economic analysis of heat and electricity production in combined heat and power plant equipped with steam and water boilers and natural gas engines,” *Energy Conversion and Management*, vol. 176, pp. 11–29, 2018.
- [14] M. Sheykhi, M. Chahartaghi, M. M. Balakheli, S. M. Hashemian, S. M. Miri, and N. Rafiee, “Performance investigation of a combined heat and power system with internal and external combustion engines,” *Energy Conversion and Management*, vol. 185, pp. 291–303, 2019.
- [15] J. Ji, Z. Ding, X. Xia, et al., “System Design and Optimisation Study on a Novel CCHP System Integrated with a Hybrid Energy Storage System and an ORC,” *Complexity*, vol. 2020, Article ID 1278751, 14 pages, 2020.
- [16] V. Evely, W. Karunkeyoon, P. Rodgers, and A. Al Alili, “Energy, exergy and economic analysis of an integrated solid oxide fuel cell - gas turbine - organic Rankine power generation system,” *International Journal of Hydrogen Energy*, vol. 41, no. 31, pp. 13843–13858, 2016.
- [17] M. Jradi and S. Riffat, “Tri-generation systems: energy policies, prime movers, cooling technologies, configurations and operation strategies,” *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 396–415, 2014.
- [18] F. A. Al-Sulaiman, F. Hamdullahpur, and I. Dincer, “Tri-generation: a comprehensive review based on prime movers,” *International Journal of Energy Research*, vol. 35, no. 3, pp. 233–258, 2011.
- [19] G. Angrisani, C. Roselli, and M. Sasso, “Distributed micro-trigeneration systems,” *Progress in Energy and Combustion Science*, vol. 38, no. 4, pp. 502–521, 2012.
- [20] P. Mancarella, “MES (multi-energy systems): an overview of concepts and evaluation models,” *Energy*, vol. 65, pp. 1–17, 2014.
- [21] A. Rong and Y. Su, “Polygeneration systems in buildings: a survey on optimization approaches,” *Energy and Buildings*, vol. 151, pp. 439–454, 2017.
- [22] D. Maraver, A. Sin, F. Sebastián, and J. Royo, “Environmental assessment of CCHP (combined cooling heating and power) systems based on biomass combustion in comparison to conventional generation,” *Energy*, vol. 57, pp. 17–23, 2013.
- [23] M. J. Dehghani and C. Kyoo Yoo, “Modeling and extensive analysis of the energy and economics of cooling, heat, and power trigeneration (CCHP) from textile wastewater for industrial low-grade heat recovery,” *Energy Conversion and Management*, vol. 205, p. 112451, 2020.
- [24] M. S. Nazir, A. N. Abdalla, Y. Wang et al., “Optimization configuration of energy storage capacity based on the microgrid reliable output power,” *Journal of Energy Storage*, vol. 32, p. 101866, 2020.
- [25] T. Peng, C. Zhang, J. Zhou, and M. S. Nazir, “An integrated framework of Bi-directional Long-Short Term Memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting,” *Energy*, vol. 221, Article ID 119887, 2021.
- [26] X. Tan, Q. Li, and H. Wang, “Advances and trends of energy storage technology in microgrid,” *International Journal of Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 179–191, 2013.
- [27] P. Puranen, A. Kosonen, and J. Ahola, “Technical feasibility evaluation of a solar PV based off-grid domestic energy system with battery and hydrogen energy storage in northern climates,” *Solar Energy*, vol. 213, pp. 246–259, 2020.
- [28] W. Du, R. H. Lasseter, and A. S. Khalsa, “Survivability of autonomous microgrid during overload events,” *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3515–3524, 2018.
- [29] B. M. Eid et al., “Control methods and objectives for electronically coupled distributed energy resources in microgrids: a review,” *IEEE Systems Journal*, vol. 10, no. 2, pp. 446–458, 2014.
- [30] M. V. Jiwanrao, “An efficient fault detection algorithm for micro-grid,” *Power Research*, vol. 10, no. 2, pp. 253–264, 2014.
- [31] X. Chu, D. Yang, and J. Li, “Sustainability assessment of combined cooling, heating, and power systems under carbon emission regulations,” *Sustainability*, vol. 11, no. 21, p. 5917, 2019.
- [32] Y. Wang, Y. Ma, F. Song et al., “Economic and efficient multi-objective operation optimization of integrated energy system considering electro-thermal demand response,” *Energy*, vol. 205, p. 118022, 2020.

## Research Article

# Improving Transformer-Based Neural Machine Translation with Prior Alignments

Thien Nguyen <sup>1</sup>, Lam Nguyen <sup>1</sup>, Phuoc Tran <sup>1</sup> and Huu Nguyen <sup>2</sup>

<sup>1</sup>Natural Language Processing and Knowledge Discovery Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

<sup>2</sup>Faculty of Information Technology, Ho Chi Minh City University of Food Industry, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Thien Nguyen; [nguyenchithien@tdtu.edu.vn](mailto:nguyenchithien@tdtu.edu.vn)

Received 30 January 2021; Accepted 30 April 2021; Published 8 May 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Thien Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Transformer is a neural machine translation model which revolutionizes machine translation. Compared with traditional statistical machine translation models and other neural machine translation models, the recently proposed transformer model radically and fundamentally changes machine translation with its self-attention and cross-attention mechanisms. These mechanisms effectively model token alignments between source and target sentences. It has been reported that the transformer model provides accurate posterior alignments. In this work, we empirically prove the reverse effect, showing that prior alignments help transformer models produce better translations. Experiment results on Vietnamese-English news translation task show not only the positive effect of manually annotated alignments on transformer models but also the surprising outperformance of statistically constructed alignments reinforced with the flexibility of token-type selection over manual alignments in improving transformer models. Statistically constructed word-to-lemma alignments are used to train a word-to-word transformer model. The novel hybrid transformer model improves the baseline transformer model and transformer model trained with manual alignments by 2.53 and 0.79 BLEU, respectively. In addition to BLEU score, we make limited human judgment on translation results. Strong correlation between human and machine judgment confirms our findings.

## 1. Introduction

There was a long period of time when statistical machine translation (SMT) was a dominant translation paradigm. The most effective SMT model is phrase-based. Phrase-based SMT is interpretable, intuitive, and reminiscent of the human translation process. It consists of several separate steps of processing concatenating together in a sequence. For example, a famous phrase-based SMT system with the name Moses created by Koehn [1] contains 9 separate steps including token alignment, lexical translation table creation, and phrase-table creation. The explicitly modular architecture of phrase-based SMT has both advantages and disadvantages. It allows us to easily modify any module to improve the overall system, but it requires us to study multiple modules to create an effective phrase-based SMT system. State-of-the-art neural machine

translation (NMT) based on deep learning, on the other hand, adopts an end-to-end approach different from traditional SMT. The whole NMT model is represented as a large neural network consisting of millions of trained parameters, taking as input a sequence of source tokens and returning a sequence of target tokens. NMT does not require us to study each stage of translation separately since it can function as a black box, i.e., if we enter a source sentence, then it will perform some complex numerical operations and return a predicted target sentence for us. Nevertheless, it has been reported that different parts of SMT actually improve NMT models. Han et al. [2] concatenated source token embeddings with their corresponding lexical translation embeddings as an additional input feature. Their experiments show the improvement in translation accuracy for the Chinese-English language pair. Song et al. [3] replaced source phrases with their

corresponding one-to-one target phrases in a phrase table. Their experiments on Chinese-English and English-Russian language pairs demonstrate that hybrid source sentences consistently lead to better translations. Chen et al. [4] proposed the use of prior alignments to guide NMT models. Their experiments with recurrent NMT models in translating from German to English and from English to French reveal large gains in translation quality of recurrent NMT models trained with prior alignments. Garg et al. [5] proposed an adjustment to the state-of-the-art transformer NMT model [6, 7], making the model capable of learning statistical prior alignments. Their experiments for the three language pairs German-English, Romanian-English, and English-French exhibit that the adjusted transformer model consistently produces better posterior alignments, compared with the baseline transformer model. However, an improvement in translation quality does not materialize. There are two possible reasons that the improvement does not occur. First, their statistical prior alignments are perhaps not good enough. Second, the studied language pairs are rich resources; consequently, the state-of-the-art transformer NMT model successfully captures their properties without the help of prior alignments. Nonetheless, there are many machine translation tasks without the luxury of available rich resources. The problem of translating news articles from Vietnamese into English that we are interested in is one of those tasks. Vietnamese-English is a low-resource language pair, and fortunately, a Vietnamese-English bilingual dataset with manually annotated prior alignments is publicly available by Ngo and Winiwarer [8, 9]. Based on these conditions, in this work, we first verify whether manual prior alignments (MA) improve the translation quality for the Vietnamese-English transformer-based NMT model. Second, we experiment different Vietnamese-English transformer-based NMT models trained with statistical prior alignments (SAs), with the objective of approaching the quality of the model trained with manual prior alignments.

The rest of the paper is divided into six sections. The first section reviews related works. The second section introduces the proposed transformer-based neural machine translation models guided by prior alignments. The third section presents the raw material and the preprocessing steps applied on it to get datasets for our study. The fourth section describes the experiments and discussion on their results. The fifth section unveils a limitation of the proposed models and a future work on improvement. The final section gives conclusions from this work.

## 2. Related Works

In this section, we briefly review works related to our study on improving transformer-based neural machine translation with prior alignments.

**2.1. Token Alignments.** Token alignments for a pair of sentences are a relation from the set of token positions in the source sentence to a set of token positions in the target

sentence. An alignment can be intuitively represented in Pharaoh format [10] as a tuple  $(j - i)$ , where the first element indicates  $j$ -th source token and the second element indicates  $i$ -th target token. Preparing token alignments is a crucial part of the traditional SMT models. The most popular token alignment tool is Giza++ [11], which is used by default in the famous SMT system Moses [1]. Giza++ implements the IBM Model 4 [12]. In addition to Giza++, there is another efficient token alignment tool `fast_align` by Dyer et al. [13], which effectively implements the IBM Model 2 [12]. Dyer et al. reported that the `fast_align` tool provides alignment as well as Giza++ does, while running significantly faster. Based on the efficiency and alignment quality, in this study, we prefer `fast_align` to Giza++ for statistically aligning source and target tokens.

### 2.2. Recurrent NMT Models Trained with Prior Alignments.

While modern NMT models outperform SMT models in terms of translation quality, the task of token alignment is still dominant by traditional statistical tools [5]. Chen et al. [4] combined the advantages of two approaches by using statistical prior alignments to train recurrent NMT models. For German-English and English-French tasks, they experiment two recurrent NMT models trained with prior alignments which have been generated with Giza++ [11]. Their experiment results show that the proposed models significantly improve over baseline recurrent NMT models. Chen et al. also introduced alignment cost for the mismatch between prior alignments and computed single-head attention mechanism of the recurrent models. Further developments on using prior alignments to improve recurrent NMT models can be found in [14–17]. Moreover, a recurrent neural network model trained with prior alignments has also been proved effective in speech synthesis task [18], which has sequence-to-sequence pattern similar to machine translation task.

**2.3. Baseline Transformer Model.** Recently, a novel deep neural network model, transformer [6], with an innovative multihead attention mechanism has been introduced. It has become the state-of-the-art model for many artificial intelligence tasks, including machine translation [19–22]. In comparison with other NMT models, including recurrent ones, transformer not only provides better translation results but also can be trained in a shorter period of time [6]. In this work, we use the transformer model as the baseline translation system. The transformer model is composed of encoder and decoder modules. The output probability distribution  $p_t = (p_{t1}, p_{t2}, \dots, p_{t\Psi})$  of the decoder is then used to predict the next target token.

Given a reference target sentence containing  $T$  tokens, the mathematical formulation of the optimization criterion for training the transformer model is presented in equation (1), revised from the one provided by Muller et al. [23]:

$$\mathcal{L}_1 = -\frac{1}{T} \sum_{i=1}^T \sum_{j=1}^{\Psi} (r_{ij} \times \log(p_{ij})). \quad (1)$$



In equation (1), the symbol  $r_{ij}$  indicates whether  $j$ -th token in the dictionary is the true value at the  $i$ -th position in the target sentence.

**2.4. Transformer Model Guided by Prior Alignments.** Garg et al. [5] altered the state-of-the-art transformer NMT model [6, 7] for joint alignment and translation tasks, making use of prior alignments in training the model. The revised transformer model has the same architecture as the baseline transformer model with a slightly different training procedure. They replace the optimization criterion with a modified one including prior alignments. Specifically, for a pair of source and target sentences of length  $K$  and  $T$ , respectively, and a prior alignment set  $\mathcal{A} \subseteq \{(j-i): j = 1, \dots, K; i = 1, \dots, T\}$ , they randomly take the output of just a head ( $n$  can be any number from 1 to 8) of the fifth decoder layer and then project it into a sequence of  $T$  probability distributions  $(q_{ij})_{i=1, j=1}^{i=T, j=K}$  over  $K$  tokens of the corresponding source sentence for every target token. They compare the probability distributions  $q_{ij}$  with the reference probability generated from prior alignments via cross-entropy:

$$\mathcal{L}_2 = -\frac{1}{T} \sum_{i=1}^T \sum_{j=1}^K (a_{ij} \times \log(q_{ij})). \quad (2)$$

In equation (2), the symbol  $a_{ij}$  indicates the probability of whether the  $i$ -th target token is correctly aligned with the  $j$ -th source token.

Taken together, the optimization criterion for the transformer-M model is the sum of cross-entropy for tokens and a weighted cross-entropy for alignments between source and target sentences in the training dataset:

$$\mathcal{L} = \mathcal{L}_1 + 0.05\mathcal{L}_2. \quad (3)$$

**2.5. Proposed Transformer-Based Models Trained with Prior Alignments.** In experiments for German-English, Romanian-English, and English-French translation tasks, Garg et al. used prior alignments created with Giza++ to train the revised transformer models. The models generate better posterior alignments but do not provide better translations. Motivated by the improvement in translation quality of recurrent NMT models trained with prior alignment [4], we experiment training transformer models with manually constructed alignments (transformer-M) for our Vietnamese-English translation task. The availability of manual token alignments  $\mathcal{A}_M$  allows us to assess the statement on whether prior alignments help us to build a better transformer model. Unfortunately, the approach is labor-consuming and does not provide us the freedom to make a choice of token type other than the one used in manual token alignments. Consequently, aside from the transformer-M model, we build other transformer models trained on statistically constructed prior alignments (transformer-S). Transformer-S models employ different token types and are trained on statistically constructed prior alignments instead of

manually annotated prior alignments, while keeping the same architecture and training procedure as for the transformer-M model.

**2.6. Syllable-to-Word Transformer Model.** The first transformer-S model (transformer-S1) is guided by alignments  $\mathcal{A}_{S1}$  constructed with the fast\_align token aligner in the place of Giza++ as in the study by Garg et al. [5]. In addition to the change of aligner, we adapt their procedure for constructing statistical alignments to suit the Vietnamese-English translation task. The adapted procedure is presented as Algorithm 1.

**2.7. Word-to-Subword Transformer Model.** Influenced by the work of Nguyen et al. [25] for Russian-Vietnamese NMT, we create the second transformer-S model (transformer-S2). While utilizing the same architecture, training procedure, and procedure to construct statistical alignments (Algorithm 1) of Transformer-S1 model, we tokenize the sentences differently in the transformer-S2 model. On the Vietnamese source side, we segment sentences into words, and on the English source side, we divide the sentences into subwords. We decide to adopt this mixed model due to the difference in linguistic morphology between Vietnamese and English. While Vietnamese is a noninflectional language, English is an inflectional language although not as morphologically rich as Russian. We use the VnCoreNLP tool developed by Vu et al. [26] and further improved by Nguyen et al. [27] to segment Vietnamese sentences into words. There is a popular phenomenon that, in Vietnamese, a syllable appears in many different words; therefore, these syllables are ambiguous to recognize by classifiers. We deploy segmentation of Vietnamese sentences into words to reduce ambiguity and, consequently, to enhance the quality of the transformer-S2 model. An example of a Vietnamese sentence and the result of its segmentation into words are presented in Table 1.

The VnCoreNLP tool employs character “\_” to inform that neighboring syllables are concatenated into a word. In Table 1, two syllables “lãnh” and “thổ” are concatenated into a word “lãnh\_thổ.”

On the English target side, we divide sentences into subwords with BPE tool proposed by Sennrich et al. [28]. An example of an English sentence and the result of its segmentation into subwords are presented in Table 2.

The BPE tool uses a pair of characters “@@” to inform that a containing token is a subword and should be concatenated with the next token to form a word in the inference phase of the transformer-S2 model. For some words, segmentation into subwords is interpretable, such as the word “personally” is divided into 2 subwords “person” and “ally” (Table 2). Subword “person” is the root part of many other words, such as “personal,” “personalize,” and “personality.” The segmentation actually has some grammatical meaning. A similar meaningful segmentation can be found for the word “ignorant” divided into “ignor” and “ant.” Meanwhile, there are other words where their segmentation is not

- (1) We tokenize both Vietnamese source sentences and English target sentences. We apply the types of tokens in the Transformer-S1 model as in the case of the Transformer-M model. A token in both source and target sentences is a sequence of characters delimited by spaces. Linguistically, Vietnamese-English Transformer-M and Transformer-S1 models are syllable-to-word models since spaces in Vietnamese delimit syllables and spaces in English delimit words.
- (2) We construct many-to-one alignments from Vietnamese to English, using the fast\_align token aligner.
- (3) We repeat step 2 in the reverse direction from English to Vietnamese.
- (4) We merge the bidirectional alignments generated in steps 2 and 3, following grow-diagonal heuristics proposed by Koehn et al. [24].

ALGORITHM 1: Procedure to construct statistical alignments  $\mathcal{A}_{S1}$ .

TABLE 1: A Vietnamese sentence and the result of its segmentation into words.

Feature	Example
Vietnamese sentence	“lãnh thổ Trung Quốc rộng bao nhiêu và diện tích đất của nó đứng hàng thứ mấy ?”
Segmentation into words	“lãnh_thổ Trung_Quốc rộng bao_nhiều và diện_tích đất của nó đứng hàng thứ mấy ?”

TABLE 2: An English sentence and the result of its segmentation into subwords.

Feature	Example
English sentence	“I personally like to call them mob youth or ignorant angry youth.”
Segmentation into subwords	“I person@@ ally like to call them mo@@ b youth or ignor@@ ant angry youth.”

understandable. In Table 2, the word “mob” is divided into two meaningless subwords “mo” and “b.”

Overall, the transformer-S2 model is a variant of the transformer-S1 model with different token representations on the source and target side.

Moreover, the imperfect segmentation of English sentences into subwords stimulates us to propose a novel transformer-S3 model without the use of English subwords, which puts more focus on the linguistic aspects of machine translation, such as the use of lemmas.

*2.8. Hybrid Word-to-Word Transformer Model Trained with Statistical Word-to-Lemma Alignments.* The transformer-S3 model can be seen as a hybrid of transformer-S1 and transformer-S2 models. Specifically, the transformer-S3 model is a word-to-word model. On the Vietnamese source side, we segment sentences into words, such as in the transformer-S2 model, while on the English target side, we choose to divide sentences into words, such as in the transformer-S1 model. Nevertheless, in preparing prior alignments  $\mathcal{A}_{S3}$ , we revise the procedure to construct statistical alignments (Algorithm 1), replacing English words with their lemmas. Step-by-step procedure to construct  $\mathcal{A}_{S3}$  alignments is presented as Algorithm 2.

In Algorithm 2, we replace English words with their lemmas, using Stanza tool created by Qi et al. [29]. A word is a surface form of a lemma according to its grammatical role in sentences. For example, words “life” and “lives” are inferred from the same lemma “life,” depending on the grammatical number. An example of an English sentence and the result of its lemmatization are shown in Table 3.

We adopt lemmatization of English words to reduce the size of vocabulary of the training dataset. The English side of

the training dataset contains 36672 distinct tokens inflected from a smaller number of 28583 lemmas. We hope that a reduced vocabulary and an unchanged number of tokens will allow the fast\_align aligner to produce better alignments and, consequently, lead to a better translation model trained on them. The relation between English words and their lemmas is one-to-one (see index sequences in Table 4); therefore, Vietnamese-word-to-English-lemma alignments can be employed in training the word-to-word transformer-S3 model.

We want to restate an important characteristic of the transformer-S3 model. The lemmatization of English target words is only applied in the construction of statistical alignments. We still use English words in the translation model.

### 3. Materials

In this work, we use English-Vietnamese Word Alignment Corpus (EVWACorpus) provided by Ngo et al. [9]. The dataset consists of 1000 news articles with 45,531 sentence pairs. These sentence pairs are already tokenized and manually aligned at the token level. A token is a sequence of characters delimited by spaces.

We apply the following processing procedures to the original EVWACorpus so that it fits our study.

*3.1. True-Cased Corpus.* First, we use true-case sentences in the dataset with Moses tool of Koehn et al. [1]. The term “true-case” means to convert a token to its most possible case. For example, the true-cased form of the token “The” is “the.” An example of a sentence in its natural form and its converted true-cased form is presented in Table 5.

- (1) We tokenize both Vietnamese source sentences and English target sentences into words
- (2) We replace English words with their lemmas
- (3) We construct many-to-one alignments from Vietnamese words to English lemmas, using the fast\_align token aligner
- (4) We repeat step 2 in the reverse direction from English lemmas to Vietnamese words
- (5) We merge the bidirectional alignments generated in steps 2 and 3, following grow-diagonal heuristics proposed by Koehn et al. [24]

ALGORITHM 2: Procedure to construct statistical alignments  $\mathcal{A}_{S3}$ .

TABLE 3: Overview of the datasets.

Vietnamese/English	Training	Validation	Testing
Number of news articles	930	40	30
Number of sentences	42,026	1,482	1,527
Average sentence length	26.2/19.2	24.5/17.8	28.3/20.6
Alignments per sentence	22.4	20.8	23.1
Number of unique tokens	16441/36672	2720/4981	3462/6211
Number of alignments	942001	30821	35291
Number of tokens	1099205/806456	36276/26315	43286/31513

TABLE 4: An English sentence and the result of its lemmatization

Feature	Example
English sentence	“teaching English to primary students is very different from secondary or high school students, thus training teachers at primary schools needs careful attention, said John A. Scacco, at the US embassy in Bangkok.”
Result of its lemmatization	“teach English to primary student be very different from secondary or high school student, thus training teacher at primary school need careful attention, say John A. scacco, at the US embassy in bangkok.”
Word index sequence	“1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37”
Lemma token index sequence	“1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37”

True-casing procedure focuses on capitalized tokens (in Table 1, they are “The,” “Fenqing,” and “China.” Based on the frequency calculated from the corpus, these tokens will be converted to lower-cased form or stay unchanged.

**3.2. Filtered Corpus.** Secondly, we leave some sentence pairs out of our work. We filter out wrongly aligned sentence pairs. Sentence pairs are considered wrongly aligned if the indices of tokens are greater than the length of sentences. Due to the computational reasons, we also remove sentence pairs containing any sentence of length greater than 80 tokens. Moreover, we transform the alignment representation in EVWACorpus into Pharaoh format for later use. Finally, we get 45,035 sentence pairs with manually annotated alignments. An example of a sentence pair in the filtered corpus is presented in Table 6.

**3.3. Datasets Extracted from Filtered Corpus.** We divide the filtered corpus into three datasets: training, validation, and testing dataset for training and evaluating different translation models. We apply a dividing procedure similar to the one used by Nguyen et al. [30]. Specifically, we randomly take 1,527 sentence pairs from 30 news articles and use them as the testing dataset. Then, we randomly take the other 1,482 sentence pairs from the other 40 news articles and use them as the validation dataset. The remaining 42,026

sentence pairs from 930 news articles form the training dataset. Overview of the datasets is shown in Table 3.

## 4. Experiments and Discussion

Google Brain team releases an implementation of the transformer model in the Tensor2tensor library [7]. The library is now replaced by its successor Trax (download at <https://github.com/google/trax>). The transformer model is implemented in other popular NMT libraries, such as opennmt [31, 32] and Fairseq [33] of Facebook AI Research team. To carry out our experiments, we choose to use Fairseq library because it allows us to build both transformer models trained with/without prior alignments.

Following the architecture and training procedure for transformer models presented in previous sections, we apply Adam optimizer with learning rate 0.0002 to train them in 10,000 steps of 3200 tokens. After completing each epoch of the training dataset, we save the model. Among all saved models, we choose the one with the best performance in the validation dataset.

We use the testing dataset to evaluate the translation models. Each model translates all Vietnamese sentences from the testing dataset, deploying a beam search of size 5. The predicted English sentences are then compared with the corresponding reference English sentences from the testing dataset via BLEU score [34]. We apply the script multi-

TABLE 5: An example of a sentence in its natural form and its converted true-cased form.

Sentence	Example
In natural form	“The fact that most Fenqing are ignorant of many things determines their opinions and views of the world and China.”
In true-cased form	“the fact that most Fenqing are ignorant of many things determines their opinions and views of the world and China.”

TABLE 6: A Vietnamese-English sentence pair with manually annotated alignments.

Feature	Example
Vietnamese sentence	“cô con gái đưa sự việc ra tòa để cố gắng lấy lại số tiền.”
English sentence	“the daughter took the case to court in an effort to recover the funds.”
Alignments	“0-1 1-1 2-1 3-2 4-4 5-4 6-5 8-7 9-9 10-9 11-11 12-11 13-13 14-13”

bleu.perl (download at <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>) in Moses program [1] to calculate the score. Since BLEU score is a statistical metric designed to be applied on the dataset level, we also make complementary human judgments on the sentence level. Specifically, we randomly take 5 Vietnamese-English sentence pairs from the testing dataset, where the source sentence is composed of 10, 15, 20, 25, and 30 tokens, sequentially. We then make human judgment on the selected sentence pairs to complement the automatic machine judgment in the form of BLEU scores.

Figure 1 shows BLEU scores of the translation results of the testing dataset by the transformer models. We can find that the transformer-M model trained with manual prior alignments significantly outperforms the baseline transformer model by  $16.26 - 14.52 = 1.74$  BLEU ( $\approx 12\%$ ) on the overall dataset level. The first question of our work already has an answer. Prior alignments actually help improve the translation quality of the transformer model.

Figure 1 also reveals a surprising result. Performance of the statistical transformer-S3 model is even better than expected. It not only outperforms other statistical models but also exceeds our expectation of approaching the result of the manual transformer-M model by giving the highest BLEU score. The statistical transformer-S3 model improves the manual transformer-M model by  $17.05 - 16.26 = 0.79$  BLEU. This can be explained by the fact that the quality of manual alignments relies on human, and human does not always provide correct alignments. It is worth to notice that it is difficult to manually align tokens between the source and target sentences. This language-related task is generally ambiguous, which is stated by Lambert et al. [35]. Moreover, the highest BLEU score of the translation result by the transformer-S3 model demonstrates the power of the statistical approach and its flexibility.

We now examine whether human judgment on translation results is correlated with automatic machine judgment on the sentence level. Here are five testcases which we randomly take and study.

Table 7 shows the translation results of a Vietnamese sentence comprising 10 tokens by transformer models. Clearly, the three presented translation models fail to translate the Vietnamese source sentence. However, from the semantic standpoint, the transformer-S3 model is better than others, successfully translating the subject “siêu nhân”

of the Vietnamese source sentence into the reference target word “Superman.” Nevertheless, from the technical standpoint, the baseline transformer model performs better by providing the most number of reference target tokens “only,” “can,” “do,” while the transformer-S3 model misunderstands the source phrase “làm được” and translates them into a passive verb phrase “are done.” This incorrect translation is very interesting because Vietnamese token “được” is mostly used in passive voice. Thus, the transformer-S3 model does make the same mistake as foreign learners of Vietnamese usually do.

Table 8 presents the translation results of a Vietnamese sentence consisting of 15 tokens by transformer models. This test case actually proves the superiority of the transformer-S3 model in comparison with other models. Translation by the transformer-S3 model bears the most resemblance in meaning to the full English reference target sentence. Nevertheless, the transformer-S3 model chooses a wrong tense of the verb “stop.” Instead of the reference verb phrase of the past perfect tense “had stopped,” the transformer-S3 model uses the verb of simple present tense “stop.” It is understandable, considering the fact that Vietnamese verbs, such as “ngừng” in the source sentence, usually do not appear in tense; hence, translation models or even human translators find it difficult to translate Vietnamese verbs.

Table 9 shows the translation results of a Vietnamese sentence consisting of 20 tokens by transformer models. All three presented translation models perform pretty well in this case. Their translations generally reflect the meaning of the source sentence. Still, the translation by the transformer-S3 model is semantically closest to the reference target sentence. The transformer-S3 model translates the key phrase “nhàm chán” into the correct target word “boring.” However, it repeats the error of translating Vietnamese verbs as in testcase 2. It mistranslates the source verb phrase “không biết” into the target verb phrase of the present simple tense “don’t know,” while the reference target phrase “didn’t know” is of past simple tense. At the same time, the baseline transformer model correctly identifies the tense, producing the target phrase “didn’t know.”

Translations of a Vietnamese sentence comprising 25 tokens are presented in Table 10. This test case unveils the positive effect of the flexibility of the statistical alignment approach. We can apply a statistical aligner to different kinds of tokens without limiting ourselves to a preselected kind of

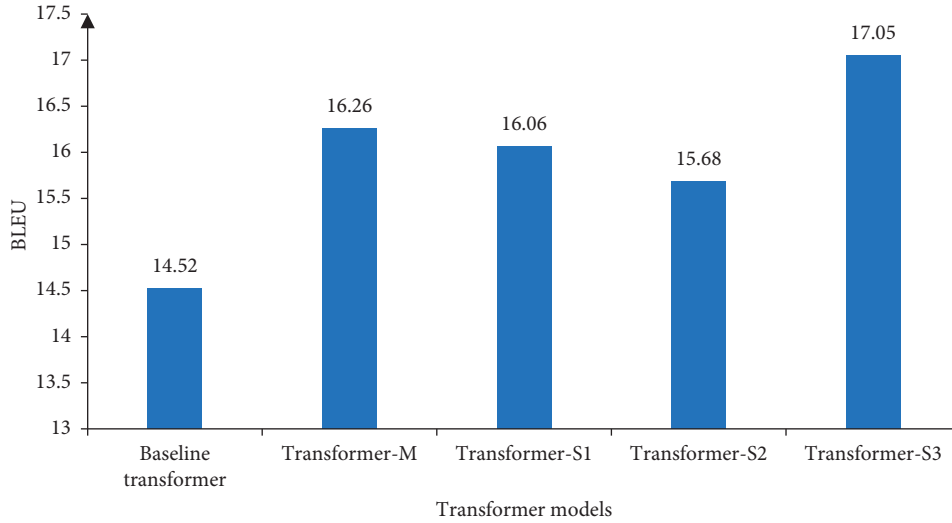


FIGURE 1: Translation results by transformer models.

TABLE 7: Translations of a Vietnamese sentence comprising 10 tokens.

Testcase 1	
Vietnamese source	“chỉ có siêu nhân mới làm được như vậy.”
English reference	“only Superman can do that.”
Translation by transformer	“only a new scan can do so.”
Translation by transformer-M	“only an ultrasound is done as well.”
Translation by transformer-S3	“only Superman are done.”

TABLE 8: Translations of a Vietnamese sentence comprising 15 tokens.

Testcase 2	
Vietnamese source	“do tác dụng phụ nên 10% bệnh nhân ngừng uống thuốc trong hai năm.”
English reference	“as a result of the side effects, 10% of the patients had stopped taking the drug within two years.”
Translation by transformer	“due to the effects of 10% of the patient who took 10% of drinking pills for two years.”
Translation by transformer-M	“due to side side effects should stop 10% of the patient who stopped the medicine for two years.”
Translation by transformer-S3	“due to side effects, 10% of patients stop taking drugs in two years.”

TABLE 9: Translations of a Vietnamese sentence comprising 20 tokens.

Testcase 3	
Vietnamese source	“buổi đêm ở đây khá nhàm chán vì chúng tôi không biết phải làm gì trước khi đi ngủ.”
English reference	“night can be quite boring because we didn’t know what to do before sleeping.”
Translation by transformer	“it was pretty tired because we didn’t know what to sleep before bed.”
Translation by transformer-M	“tonight is quite tired because we don’t know what to sleep before going to sleep.”
Translation by transformer-S3	“this night is quite boring because we don’t know what to do before going to bed.”

tokens as in the case of manual alignments. Specifically, the transformer-S3 model successfully produces the target word “appearance,” having concatenated two neighboring syllables “ngoại” and “hình” into one word “ngoại\_hình” (see Table 10). This happens due to the fact that we choose to build the transformer-S3 model as a linguistics-informed word-to-word model, while the baseline transformer model and transformer-M model are syllable-to-word models. These models require tokenization of Vietnamese sentences into syllables and English sentences into words.

Table 11 displays the translations of a Vietnamese sentence comprising 30 tokens. All three presented translation models fail to translate the key phrases of the source sentence. The subject “bang Gujarat” (meaning: the state of Gujarat) of the source sentence is mistranslated into different things: “federal federal government,” “the federal states,” and “the state of states.” Nevertheless, the translation by the baseline transformer model is smoother, consisting of many reference tokens. Unfortunately, it misses two key words “illegal” and “toxic”; therefore, its meaning is totally

TABLE 10: Translations of a Vietnamese sentence comprising 25 tokens.

Testcase 4	
Vietnamese source	“ngoại hình của chúng ta là yếu tố đầu tiên mà mọi người để ý đến và giúp họ hình dung về chúng ta.”
English reference	“our appearance is the first thing people notice, and it gives them an idea of who we are.”
Translation by transformer	“our foreign image is the first factor that people will take attention to their ideas and help them figure out.”
Translation by transformer-M	“our foreign form is the first factor that people come to attention and help them figure out us.”
Translation by transformer-S3	“our appearance is the first factor that people notice and help them figure out of us.”

TABLE 11: Translations of a Vietnamese sentence comprising 30 tokens.

Testcase 5	
Vietnamese source	“tuần trước, bang Gujarat đưa ra một điều luật mới quy định rằng việc sản xuất và bán rượu độc bất hợp pháp sẽ bị phạt tử hình.”
English reference	“last week, the state of Gujarat brought in a new law making the illegal manufacture and sale of toxic alcohol there punishable by death.”
Translation by transformer	“last week, the federal federal government issued a new law that the production of manufacturing and selling alcohol would be charged with death.”
Translation by transformer-M	“last week, the federal states issued a new law that production and illegal consumption would be charged with death.”
Translation by transformer-S3	“last week, the state of states introduced a new law that the production of production and illegal alcohol will be charged with death.”

different from the reference. While the transformer-S3 model delivers stutters (“state of states” and “production of production”), it yields a correct key word “illegal,” making the translation result better resemble the reference in meaning.

On the whole, human judgment is in line with automatic machine judgment on the quality of the translation models. From the semantic point of view, the transformer-S3 model is the best model. Moreover, we discover that the transformer-S3 model does not succeed at handling the verb tenses.

*4.1. Limitation and Future Work.* Despite many advantages of training transformer-based NMT models with prior alignments, especially statistical ones, it still has a noticeable disadvantage. The models trained with them poorly handle verb tenses in translation. Translations of the best transformer-S3 model may reflect the meaning of the source sentences; however, they do not guarantee a high BLEU score since they generate verbs in an incorrect tense.

This work is the first step towards enhancing translation quality of transformer-based NMT models trained with prior alignments. Future work will address the pitfall of the word-to-word transformer-S3 model trained with statistical word-to-lemma alignments. Research into solving this problem is in progress. We will explore the selection of a head in the multihead attention mechanism, whose output is compared with prior alignments.

## 5. Conclusions

In this study, we have proved that prior alignments help better train the Vietnamese-English transformer-based neural machine translation model. Experiment results

show the improvement of translation quality in terms of BLEU score. Moreover, to free ourselves from dependence on costly manual alignments, we have proposed a novel hybrid word-to-word transformer model trained on statistical word-to-lemma alignments. Unlike strict manual alignments, the flexible statistical aligner allows us to construct word-to-lemma alignments, representing a Vietnamese source sentence as a sequence of words and the corresponding English target sentence as a sequence of lemmas. Statistically constructed word-to-lemma alignments are then used to train a word-to-word transformer-S3 model instead of word-to-word alignment. Experiments have demonstrated that the novel word-to-word transformer-S3 model trained with statistical word-to-lemma alignments outperforms the transformer-M model trained with manual alignments in terms of BLEU score. In addition to machine judgment, we have made limited human judgments on translation results. Strong correlation between human and machine judgment has validated our findings.

Based on the experiment results, we recommend the use of statistical prior alignments in training the transformer-based neural machine translation models at least in the context of low-resource translation tasks.

## Data Availability

Readers can obtain the datasets used in this work by contacting the corresponding author Thien Nguyen via e-mail: [nguyenchithien@tdtu.edu.vn](mailto:nguyenchithien@tdtu.edu.vn).

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors truly appreciate Ms. Trang Nguyen, a translator and scientist. She provided the authors invaluable recommendations and encouragement when they prepared the manuscript and chose a journal to submit our work to.

## References

- [1] P. Koehn, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on interactive Poster and Demonstration Sessions*, pp. 177–180, Stroudsburg, PA, USA, June 2007.
- [2] D. Han, J. Li, Y. Li, M. Zhang, and G. Zhou, "Explicitly modeling word translations in neural machine translation," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 1, pp. 1–17, 2020.
- [3] K. Song, Y. Zhang, H. Yu, W. Luo, K. Wang, and M. Zhang: Code-switching for enhancing NMT with pre-specified translation, <https://www.aclweb.org/anthology/N19-1044.pdf>.
- [4] W. Chen, E. Matusov, S. Khadivi, and J. T. Peter, "Guided alignment training for topic-aware neural machine translation," in *Proceedings of the AMTA 2016: 12th Conference of the Association for Machine Translation in the Americas*, pp. 121–134, Austin, Texas, USA, October 2016.
- [5] S. Garg, S. Peitz, U. Nallasamy, and M. Paulik, "Jointly learning to align and translate with transformer models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4453–4462, Hong Kong, China, November 2019.
- [6] A. Vaswani, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, MIT Press, Cambridge, MA, USA, 2017.
- [7] A. Vaswani, "Tensor2tensor for neural machine translation," in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pp. 193–199, Boston, MA, USA, March 2018.
- [8] Q. H. Ngo and W. Winiwarter, "Building an English-Vietnamese bilingual corpus for machine translation," in *Proceedings of the 2012 International Conference on Asian Language Processing*, pp. 157–160, Hanoi, Vietnam, November 2012.
- [9] Q. H. Ngo, W. Winiwarter, and B. Wloka, "EVBCorpus—a multi-layer English-Vietnamese bilingual corpus for studying tasks in comparative linguistics," in *Proceedings of the 11th Workshop on Asian Language Resources*, pp. 1–9, Nagoya, Japan, October 2013.
- [10] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Machine Translation: From Real Users to Research*, pp. 115–124, Springer, Berlin, Germany, 2004.
- [11] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.
- [12] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.
- [13] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 644–648, Atlanta, GA, USA, June 2013.
- [14] J.-T. Peter, A. Nix, and H. Ney, "Generating alignments using target foresight in attention-based neural machine translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 27–36, 2017.
- [15] T. Alkhouli and H. Ney, "Biasing attention-based recurrent neural networks using external alignment information," in *Proceedings of the Second Conference on Machine Translation*, pp. 108–117, Florence, Italy, August 2017.
- [16] J. Zeng, "Multi-domain neural machine translation with word-level domain context discrimination," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 447–457, Brussels, Belgium, October 2018.
- [17] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Neural machine translation with supervised attention," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3093–3102, Osaka, Japan, December 2016.
- [18] X. Zhu, Y. Zhang, S. Yang, L. Xue, and L. Xie, "Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis," *IEEE Access*, vol. 7, pp. 65955–65964, 2019.
- [19] J. Zhang, "Improving the transformer translation model with document-level context," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 533–542, Brussels, Belgium, November 2018.
- [20] I. V. Tetko, P. Karpov, R. Van Deursen, and G. Godin, "State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis," *Nature Communications*, vol. 11, no. 1, pp. 1–11, 2020.
- [21] N. Parmar, "Image transformer," in *Proceedings of the International Conference on Machine Learning*, pp. 4055–4064, Jinan, China, May 2018.
- [22] T. Wolf, "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Suzhou, China, November 2020.
- [23] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pp. 4696–4705, Vancouver, Canada, December 2019.
- [24] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proceedings of the IWSLT-2005*, Pittsburgh, PA, USA, 2005.
- [25] T. Nguyen, H. Nguyen, and P. Tran, "Mixed-level neural machine translation," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8859452, 7 pages, 2020.
- [26] T. Vu, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese natural language processing toolkit," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 56–60, New Orleans, LA, USA, June 2018.
- [27] D. Q. Nguyen, T. Vu, M. Dras, and M. Johnson, "A Fast and Accurate Vietnamese Word Segmenter," in *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*, pp. 2582–2587, Miyazaki, Japan, May 2019.
- [28] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, Berlin, Germany, August 2016.

- [29] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning: Stanza: A {Python} Natural Language Processing Toolkit for Many Human Languages, 2020, <https://nlp.stanford.edu/pubs/qi2020stanza.pdf>.
- [30] T. Nguyen, H. Le, and V.-H. Pham, “Source-word decomposition for neural machine translation,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 4795187, 10 pages, 2020.
- [31] G. Klein, Y. Kim, Y. Deng et al., “Neural machine translation toolkit,” in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pp. 177–184, Boston, MA, USA, March 2018.
- [32] G. Klein, F. Hernandez, V. Nguyen, and J. Senellart, “The OpenNMT neural machine translation toolkit: 2020 edition,” in *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pp. 102–109, October 2020.
- [33] M. Ott, “Fairseq: a fast, extensible toolkit for sequence modeling,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, MN, USA, June 2019.
- [34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, PA, USA, July 2002.
- [35] P. Lambert, A. Gispert, R. Banchs, and J. B. Mariño, “Guidelines for word alignment evaluation and manual alignment,” *Language Resources and Evaluation*, vol. 39, no. 4, pp. 267–285, 2005.



## Research Article

# Intelligent Computation Offloading for IoT Applications in Scalable Edge Computing Using Artificial Bee Colony Optimization

Mohammad Babar <sup>1</sup>, Muhammad Sohail Khan,<sup>1</sup> Ahmad Din <sup>2</sup>, Farman Ali <sup>3</sup>,  
Usman Habib <sup>4</sup>, and Kyung Sup Kwak <sup>5</sup>

<sup>1</sup>Department of Computer Software Engineering, University of Engineering and Technology, Mardan 23200, Pakistan

<sup>2</sup>Department of Computer Science, COMSATS University Islamabad (CUI), Abbottabad Campus, Islamabad 22010, Pakistan

<sup>3</sup>Department of Software, Sejong University, Seoul 05006, Republic of Korea

<sup>4</sup>National University of Computer & Emerging Sciences, Chiniot-Faisalabad Campus, Islamabad, Pakistan

<sup>5</sup>Department of Information and Communication Engineering, Inha University, Incheon 22212, Republic of Korea

Correspondence should be addressed to Mohammad Babar; mbabarcs@gmail.com and Kyung Sup Kwak; kskwak@inha.ac.kr

Received 23 January 2021; Accepted 24 April 2021; Published 4 May 2021

Academic Editor: Ning Cai

Copyright © 2021 Mohammad Babar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most of the IoT-based smart systems require low latency and crisp response time for their applications. Achieving the demand of this high Quality of Service (QoS) becomes quite challenging when computationally intensive tasks are offloaded to the cloud for execution. Edge computing therein plays an important role by introducing low network latency, quick response, and high bandwidth. However, offloading computations at a large scale overwhelms the edge server with many requests and the scalability issue originates. To address the above issues, an efficient resource management technique is required to maintain the workload over the edge and ensure the reduction of response time for IoT applications. Therefore, in this paper, we introduce a meta-heuristic and nature-inspired Artificial Bee Colony (ABC) optimization technique that effectively manages the workload over the edge server under the strict constraints of low network latency and quick response time. The numerical results show that the proposed ABC algorithm has outperformed Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Round-Robin (RR) Scheduling algorithms by producing low response time and effectively managing the workload over the edge server. Furthermore, the proposed technique scales the edge server to meet the demand of high QoS for IoT applications.

## 1. Introduction

Internet of Things (IoT) is reshaping the technological landscape of traditional systems. The concept of IoT-based smart system is making its way from dreams to reality [1]. Smart city, smart healthcare, and smart industry have grasped the researchers' attention at a monumental scale [2–5]. However, smart systems generate high volume of data in a short period of time and create several challenges such as data management, security, storage, and energy consumption [6]. In addition, the applications pertaining to these IoT-based systems are resource-constrained and require a

crisp response, low latency, and high bandwidth, which are beyond their capabilities [7].

Cloud computing is considered as a resource-rich solution to the above problems. However, the inherent longer latencies of cloud computing make it nonviable. These longer latencies hinder the performance of IoT-based smart systems [8]. Edge computing offers computation, storage, and communication services at the edge of a network, resulting in low latency, high bandwidth, and energy-efficiency [9]. Both the edge and fog computing architectures have been used to handle resource-scarcity of IoT devices [10]. In this work, the objective behind the utilization of

edge-based architecture is to deploy edge as a micro data center that has the potential to provide cloud like services, even in the absence of the cloud. However, fog computing provides computing, storage, and other services through intermediate nodes such as routers and gateways, which are resource-limited. Edge computing utilizes computation offloading concept, where the resource-constrained IoT devices handle compute-intensive tasks to the edge server, execute the task, and send back the result to IoT devices. Computation offloading not only saves energy in IoT devices but also extends the lifetime of these devices [11]. Figure 1 shows edge computing architecture for IoT.

One way to achieve the required high QoS is via computation offloading application of edge computing [12]. However, computation offloading is rather a complex job, which enfoldes the complexity of task scheduling, partitioning, migration, and latency. In addition, the offloading has a vital role in edge discovery, as well as selecting an appropriate edge node for computation.

The current IoT-based smart systems use large-scale sensors that generate a huge amount of data at the IoT deployment layer. The rapid processing of the generated data is very substantial. Therefore, computation offloading for the resource-constrained devices is quite significant. The objective of this research is to scale the edge server for delay sensitive tasks that demand stringent QoS requirements. A computation offloading technique selects a task from the IoT layer generated by IoT devices and offloads it to the edge server for execution. However, computation offloading at a large scale creates congestion on the edge server, which provides low QoS. Therefore, a resource scheduling mechanism for load balancing over edge servers is required to ensure the effective utilization of the edge resources, while considering communication cost and response time of the tasks. On the other hand, computation offloading is a nontrivial, challenging, and NP-hard problem, where its complexity is directly proportional to the increasing number of offloading tasks. Thereby, several studies proposed greedy algorithms to tackle computation offloading problem [13, 14], but the computation offloading still grows exponentially and became very challenging for traditional greedy algorithms. Therefore, in this paper, we proposed an Artificial Bee Colony- (ABC-) based computation offloading algorithm that effectively and seamlessly performs the process of computation offloading. The major contributions of this study are as follows:

- (i) We devised a classical three-tier framework by integrating edge and cloud to simulate computation offloading process following strict energy and latency constraint for delay-sensitive tasks. An edge server is used in conjunction with cloud due to its higher computing power than edge servers. In the proposed framework, the inclusion of cloud further scales the edge server efficiently.
- (ii) To effectively balance the workload over edge servers, we propose ABC optimization technique based on swarm intelligence, where the objective function is set to achieve the minimum computation cost and low latency for the offloaded tasks.
- (iii) A computation offloading algorithm based on ABC is implemented for seamless computation offloading. The results exhibit that the proposed technique shows notable improvement in reducing the response time and efficient load balancing over edge nodes, compared to Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Round-Robin (RR) Scheduling.

The rest of the paper is structured as follows. Section 2 presents a detailed overview of the related work. Section 3 shows the edge-cloud integration framework that spans over the system model. In addition, it also shows the computation offloading algorithm based on Artificial Bee Colony. Section 4 presents the results and discussion of the proposed technique. Finally, Section 5 comprehends the conclusion of this research study.

## 2. Related Work

Smart systems are built using a large number of IoT devices, which generate huge amount of data in a short period of time. The generated data is sent to the cloud for aggregation, analytics, and computation [15–17]. Computation offloading to the cloud decreases the computation load from IoT devices. However, it produces excessive delay that violates the QoS requirements for real-time systems and applications [18]. Furthermore, it leads to inefficient use of resources and unnecessarily overloads back-haul links [19]. Edge computing brings down computation, storage, and other services in the closed proximity of the users, thereby meeting the strict QoS requirements for delay-sensitive applications.

Computation offloading has largely attracted the researcher's attention, and a number of studies are conducted on edge computing domain. In [20], a three-tier edge-cloud integration framework has been deployed to reduce energy and latency of IoT devices. To identify the best edge server for computation offloading, the expected offloading and the propagation delay of different edge servers are considered to determine a threshold. If the expected delay is below the threshold, the request is accepted for offloading; otherwise, the task is given to the next feasible edge server. The edge server is further connected to the cloud for scalability, where the task is offloaded to the cloud if the edge server reaches its maximum. This work produces desirable results, but they did not consider the communication cost.

In [21], the authors focused on scheduling the problem of computation offloading over the edge server placed in the proximity of users. The offloading and computation decisions are made by the IoT device, while considering the battery life and response time for better quality of experience (QoE). They formulated the offloading problem as mixed-integer nonlinear programming (MINLP) and resolved it using branch and bound reinforcement learning technique. The proposed solution did not consider the load balancing over edge and it leads to scalability issue.

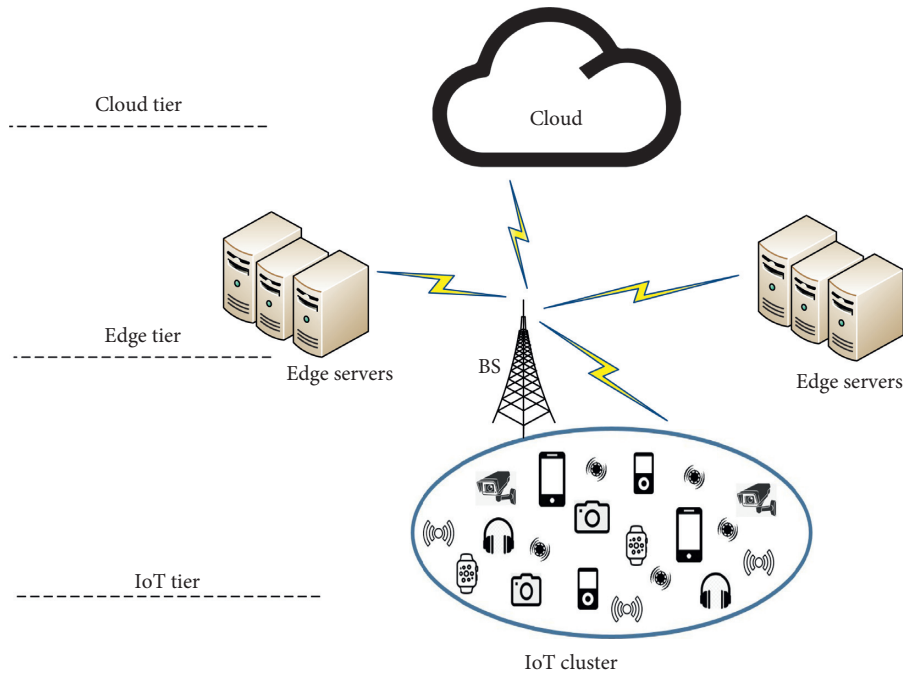


FIGURE 1: Edge computing architecture for IOT.

A distributed and scalable framework for wearable IoT devices is proposed in [22]. This existing system considered metrics such as response time, bandwidth, storage, and a number of tasks successfully completed for effective resource provisioning using recurrent learning. The response time is reduced by introducing a control layer between cloud and IoT layer, which generalizes the edge computing model but lacks practical computation offload environment. A dynamic computation offloading algorithm based on stochastic optimization technique is presented to deal with computation offloading problem [23]. The computation offloading problem is further divided into subproblems to achieve a minimum cost of the offloading process. They reduced the cost but did not consider the load balancing over multiple edge servers while migrating tasks.

The existing computation offloading frameworks comprised over cloud, edge, and IoT have been devised for seamless computation offloading [24–26]. These studies focused on reducing the communication overhead, response time, and bandwidth using machine learning algorithms such as Lyapunov optimization, Deep Supervised Learning (DSP), and Discrete Particle Swarm Optimization (DPSO). These frameworks have a well-defined objective for low latency and minimize energy consumption of IoT devices. However, this existing system focuses on edge-cloud framework instead of computation offloading process. A load balancing framework based on directed graph partitioning algorithms is proposed to balance load over edge node for in-network flexible resource provisioning and allocation [27]. The devised algorithm is inappropriate in the dynamic workload conditions. A cloud-edge integration architecture is introduced to deal with the scheduling problem of bag-of-tasks applications using Modified Particle Swarm Optimization (MPSO) [28]. The similar problem is

handled using Genetic Algorithm (GA) [29]. The main goals of these studies were to reduce the operating cost and remote processing time of the task. However, these studies have not focused on communication cost and latency incurred by the computation offloading process.

Several studies use different clustering techniques to protect the edge server from bottleneck and cope with the scalability issue. For instance, a CNN-based fused tile partitioning (FTP) is presented to distribute the workload over edge servers [30]. A PSO-based multiclustering technique in a semiautonomous edge-IoT environment is proposed in the account of reducing processing and communication delays to distribute the load over edge servers [31]. A graph-based edge clustering technique and software defined network- (SDN-) based multicluster overlay (MCO) are utilized to optimize task size, number of servers, required channels for communications, best channel allocation for effective load distribution, and scaling the edge server [32, 33]. However, these studies produce additional communication overhead and add more latency while making computation offloading decision.

The existing studies reveal that the computation offloading process is very complex and challenging. It consumes extra energy and incurs latency, while intercommunicating between devices and servers [5, 6]. A single device is responsible for making computation offloading decisions, which consumes more energy and causes fast battery drain. In addition, the existing edge server makes computation offloading decision independently in the IoT environment [34–37], where the edge server is overwhelmed with many requests, creating congestion over the edge server, and originates scalability issue. The existing approaches do not attain the high QoS requirements of IoT applications. Nonetheless, they reflect a trade-off between QoS and the scale of offloading requests.

In our work, we design a dynamic and decentralized task execution through computation offloading. To accomplish the above-mentioned objective, a three-tier edge-cloud integration framework is designed for a successful computation offloading process. One of the major advantages of the proposed layered-based architecture is a robust service discovery. The IoT system is designed using a large number of devices and servers, where searching for the right resource for the IoT device is quite challenging. A social Internet of things (SIoT) clustering approach [38] is deployed at IoT layer that performs the task of aggregation and resource management. The SIoT not only controls the number of offloading tasks sent to the edge server and protects the edge server from bottleneck but also creates an association between offloading task and the resource allocation. This association finds the right resource for executing a particular task, hence reducing the latency of the offloading task.

We propose an ABC optimization technique that balances the workload over the edge server, provides the right resource for offloading device, and exploits low latency interconnections between IoT device and server. The proposed framework can be effectively utilized for IoT devices, where the task is executed under strict energy with the required latency to meet the high QoS requirements, which is very unlikely to get using the traditional cloud. Finally, a computation offloading algorithm based on ABC is proposed to provide an efficient computation offloading facility that searches to find new resources for task execution. The objective function measures the network latency and execution time of the task to achieve minimum service time. The detailed discussion on ABC Algorithm and computation offloading technique is provided in Section 3.

### 3. Edge-Cloud Integration Framework for Computation Offloading

In this section, we briefly describe the system model. In addition, we also provide the detailed overview of the Artificial Bee Colony optimization technique and a novel ABC-based computation offloading algorithm.

**3.1. System Model.** There are three layers in the IoT-based edge infrastructure as shown in Figure 2. The first layer is IoT layer, where a large number of sensors are connected to LAN in clusters. These clusters are connected to the second layer called edge layer, which contains multiple edge nodes. These nodes perform basic analytics on the data received from the sensors. However, the computational and storage capabilities of these nodes are limited. The third layer is the cloud layer. The master node of second layer is responsible for deciding whether to offload the task to the cloud layer or not. It has powerful computing and storage resources to perform heavy analytics and large/long-term data storage. Table 1 expresses the list of symbols and notations used in the system model.

The IoT layer has  $N$  nodes  $\{S_1, S_2, \dots, S_n\}$ , where each sensor/mote  $S_i$  is working at frequency  $\lambda_i$ . Edge layer has  $M$  nodes  $\{Eg_1, Eg_2, \dots, Eg_m\}$ . If the computation job is

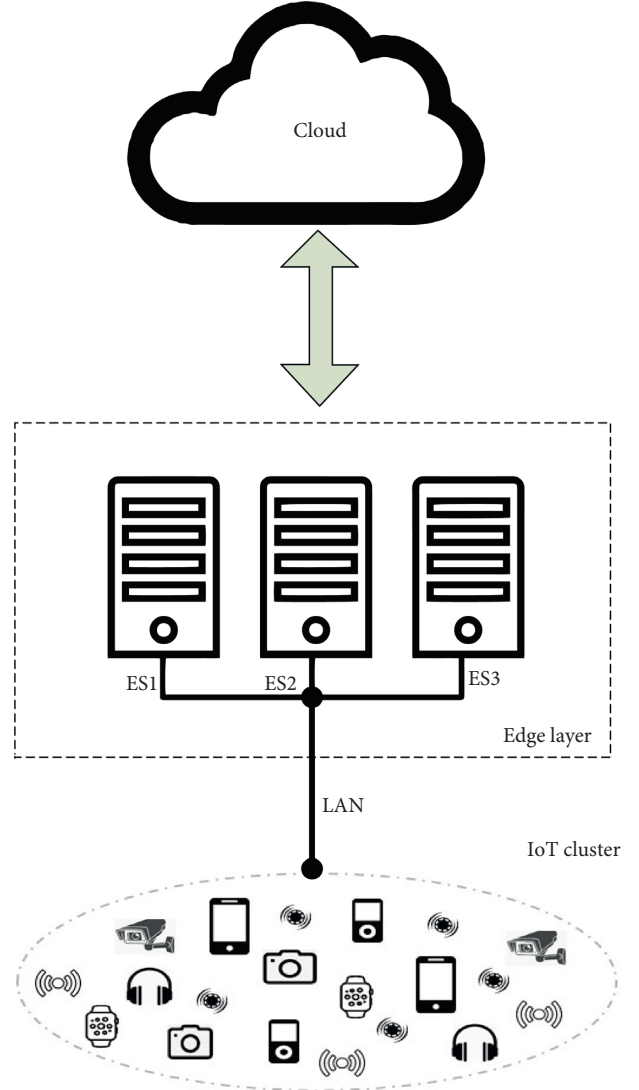


FIGURE 2: Edge-computing-based system model.

TABLE 1: Symbols and notations used in the system model.

Symbols	Descriptions
$S_n$	IoT nodes $n$
$Eg_m$	Edge servers $m$
$ST_i$	Service time of IoT node $i$
$T_{com_i}$	Computation requirement of task
$\lambda_i$	Frequency of sensor
$C_{sen}$	Computation capacity of sensor
$NL_{me}$	Network latency (IoT to edge)
$B_{me}$	Bandwidth (IoT node to edge)
$D_m$	Data size
$C_{mote}$	Energy required for task execution
$EC_{BW}$	Bandwidth (edge to cloud)

performed in the IoT node without offloading it to the fog or cloud, then the service time for a job is computed using the following equation:

$$ST_n = \frac{T_{\text{com}_i}}{C_{\text{sen}}}, \quad (1)$$

where  $ST_n$  is a service time of the task,  $T_{\text{com}_i}$  is computation required to complete the task  $i$ , and  $C_{\text{sen}}$  is computational capability of the mote (IoT sensor node). The communication cost between mote and edge node is computed using equation (2), which depends on transfer capacity and network latency.

$$\text{MoteEdgeComm} = NL_{\text{me}} + \frac{D_m}{B_{\text{me}}}, \quad (2)$$

where  $NL_{\text{me}}$  is network latency between mote and edge,  $D_m$  is data generated by the mote/sensor, and  $B_{\text{me}}$  is the bandwidth between mote and the edge. If the computation is performed in the IoT node, the energy ( $E_m$ ) required for the computation of the job is given by the following equation:

$$E_m = ST_n * C_{\text{mote}}, \quad (3)$$

where  $C_{\text{mote}}$  is an energy consumed by processing units in unit time. The service time of the job offloaded to the edge node is calculated using the following equation:

$$ST_{EN} = \text{MoteEdgeComm} \times \frac{T_{\text{com}_i}}{C_{\text{EdgeNode}}}. \quad (4)$$

In the above equation,  $C_{\text{EdgeNode}}$  is clock frequency of the edge node. The task offloaded to the edge node required energy for its completion, which is calculated using the following equation:

$$E_{\text{edge}} = 2 \times \frac{D_m}{B_{\text{me}}} + \frac{T_{\text{com}_i}}{C_{\text{EdgeNode}}}. \quad (5)$$

The service time for job offloading to the cloud is computed using the following equation:

$$ST_{\text{cloud}} = \text{EdgeCloudComm} + \frac{T_{\text{com}_i}}{C_{\text{cloud}}}, \quad (6)$$

where  $\text{EdgeCloudComm}$  is composed of the cloud latency and time required to send data to cloud from the edge node. It is calculated using the following equation:

$$\text{EdgeCloudComm} = NL_{EC} + \frac{T_{\text{com}_i}}{C_{\text{EdgeNode}}}, \quad (7)$$

where  $NL_{EC}$  is network latency between the edge node and cloud, while  $EC_{BW}$  is bandwidth between edge and cloud. Therefore, the energy consumption for the offloaded task to the cloud is calculated using the following equation:

$$E_{\text{cloud}} = 2 \times \frac{D_m}{EC_{BW}} + \frac{T_{\text{com}_i}}{C_{\text{cloud}}}. \quad (8)$$

The total communication cost for offloading the job to cloud using IoT node is computed by the following equation:

$$\text{TotalCommCost} = \text{EdgeCloudComm} + \text{MoteEdgeComm}. \quad (9)$$

**3.1.1. Objective Function.** It is important to handle two main decisions of whether to offload the task to edge or cloud. Therefore, the objective function is used to minimize the energy consumption ( $E$ ) and service time delay ( $ST$ ) of each offloading scheme. The first decision is about offloading the job to the edge:

$$s_i = \begin{cases} 0, & \text{if job } i \text{ is offloaded to the edge,} \\ 1, & \text{if job } i \text{ is executed using mote computational resource.} \end{cases} \quad (10)$$

Similarly, the edge might have limited resources to fulfil the computational requirements of the job. Therefore, it may decide to further offload the task to the cloud. The following variables in decision allow offloading the job to the cloud:

$$t_i = \begin{cases} 0, & \text{if job } i \text{ is offloaded to the cloud,} \\ 1, & \text{if job } i \text{ is offloaded to the edge.} \end{cases} \quad (11)$$

For ABC optimization algorithm,  $ST_i$  and  $E_i$  are required to be normalized using the two following equations:

$$ST_{\text{current}} = \frac{ST_{\text{current}} - ST_{\text{min}}}{ST_{\text{max}} - ST_{\text{min}}}, \quad (12)$$

$$E_{\text{current}} = \frac{E_{\text{current}} - E_{\text{min}}}{E_{\text{max}} - E_{\text{min}}}. \quad (13)$$

For all  $K$  jobs, total energy consumption  $E_{\text{total}}$  is calculated using equation (15), and the total delay for offloading the task to the cloud is computed using the following equation:

$$ST_{\text{total}} = \sum_{n=1}^K (ST_n(s_i) + ST_{\text{edge}}(s_i) + ST_{\text{cloud}}(1 - s_i)(1 - t_i)), \quad (14)$$

$$E_{\text{total}} = \sum_{n=1}^K (E_n(s_i) + E_{\text{edge}}(s_i) + E_{\text{cloud}}(1 - s_i)(1 - t_i)). \quad (15)$$

Therefore, the objective function is

$$F_{\text{obj}} = \delta(ST) + (1 - \delta)(E), \quad (16)$$

where  $ST$  is the service time and  $E$  is energy consumption for the offloading scheme.

$$ST = \min(ST(l_j, \text{job}_r)), \quad \text{for } 1 \leq j \leq M, 1 \leq r \leq k, \quad (17)$$

$$E = \min(E(l_j, \text{job}_r)), \quad \text{for } 1 \leq j \leq M, 1 \leq r \leq k, \quad (18)$$

where  $\delta \in [0, 1]$  is weight to prioritize the elements of the objective function and  $(E(l_j, \text{job}_r))$  is energy consumption of the task  $\text{job}_r$  by the node at level  $l_j$ . The objective function has following constraint on the edge layer:

$$\frac{\sum_{j=1}^k H_k}{H_k - H_{\text{avg}}} \leq Eg_{\text{cap}}, \quad (19)$$

where  $Eg_{cap}$  is a load capacity of the edge node  $j$  and  $H_k$  is an arithmetic mean of response time if it is the only edge that serves all the offloaded jobs.

**3.2. Artificial Bee Colony (ABC) Algorithm.** The ABC was proposed by Dervis Karaboga [39]. It is a swarm-intelligence-based optimization algorithm, which contains three types of bees. The first type is scout bees that search for new sources of food randomly, thus ensuring exploration. The second type is onlooker bees that choose a food source by observing the dance of employed bee. The third type is employed bees that are linked to the food source, thus ensuring exploitation. Scout bees and onlooker bees are not linked to any specific food source. Therefore, they are usually called unemployed bees. A general outline of the ABC algorithm is shown in Algorithm 1.

In this section, we presented the ABC algorithm for computation offloading at IoT edge. The objective function measures the service time (network latency and time required for job completion) and energy consumption for the solution provided by the optimization algorithm. The main purpose is to minimize the objective function, which searches for minimum computational cost and job latency. There are three decision variables:  $s_j$ ,  $t_j$ , and  $\xi$  (contains a list of jobs). The input for the algorithm is a set of jobs and nodes.

**3.2.1. Initialization Phase.** The population (nodes) is represented by vectors  $x_n$ , which is initialized by bees using the following equation:

$$x_{nj} = lb_j + \text{rand}(0, 1) \times (ub_j - lb_j), \quad (20)$$

where  $ub_j$  and  $lb_j$  are the upper and the lower bounds of the parameter, respectively.

### 3.2.2. Employed Bee Phase

**(1) New Solution.** The employed bee searches for new nodes ( $y_{mi}$ ) with more resources in a neighbourhood. The new neighbour node  $y_{mi}$  can be found by the the following equation:

$$y_{nj} = x_{nj} + \tau_{nj} \times (x_{nj} - x_{pj}), \quad (21)$$

where  $\tau_{nj}$  is a function that generates a random number between  $-1$  and  $1$  and  $x_{pj}$  is a randomly chosen node in a neighbourhood.

**(2) Greedy Selection.** Fitness of new node  $y_{mi}$  is calculated. If its fitness is high, then  $x_{nj}$   $y_{mi}$  is memorized.

### 3.2.3. Onlooker Bee Phase

**(1) Probability Calculation Based on Fitness.** Onlooker bees choose the node probabilistically based on information provided by the employed bee nodes. The onlooker bees choose node  $x_n$  using the probability  $p_n$  as shown in the following equation:

$$p_{nj} = \frac{F_{nj}(x_n)}{\sum_{n=1}^k (F_k)(x_n)}, \quad (22)$$

where  $F_n$  is fitness function, which is computed by using the following equation:

$$F_n = \begin{cases} \frac{1}{1 + F_{obj}}, & \text{if } F_{obj} \geq 0, \\ 1 + |F_{obj}|, & \text{if } F_{obj} < 0. \end{cases} \quad (23)$$

**(2) New Solution for Onlooker Bee.** Once a node is chosen for the onlooker bee probabilistically, a new neighborhood node  $y_{mi}$  is determined using equation (20).

**(3) Greedy Selection.** At this stage,  $y_{mi}$  and  $x_{nj}$  are compared to each other. If a new node in neighborhood  $y_{mi}$  has high fitness value, then current node  $y_{mi}$  is memorized.

**3.2.4. Scout Bee Phase.** Scout bees ensure exploration and choose a node randomly. An employed bee becomes a scout bee if it fails to improve its solution in a limit (number of trails). Figure 3 and Algorithm 2 present the flowchart and algorithm of ABC-based computation offloading technique, respectively.

## 4. Results and Discussion

In this section, we discuss the results achieved using our proposed framework. A three-tier edge-cloud integration framework is proposed, where IoT devices are placed at Tier-1, where data are generated from multiple devices in multitasking manner. Tier-2 comprises edge servers, and Tier-3 includes a resource-rich cloud. The hierarchical representation of the proposed framework helps efficiently utilize the resources and distinguish the responsibility of each tier. A simulation-based edge-cloud integration test-bed is designed using MATLAB. The simulation setup not only provides the opportunity to conduct the experiment in the control environment using a preferred set of parameters but also allows us to repeat the experiment under different scenarios and constraints. Thereby, we have evaluated the performance of the proposed Artificial Bee Colony computation offloading algorithm against the metaheuristic algorithms such as Particle Swarm Optimization, Ant Colony Optimization, and Round-Robin Scheduling [10]. A list of parameters with their corresponding values acquired by conducting several preliminary experiments is provided in Table 2.

The metrics for the evaluation of the proposed offloading algorithm are degree of imbalance and standard deviation in response time, while observing the load of edge node. The degree of imbalance among edge nodes is calculated using the following equation:

- (1) Step 1: Initialization Phase
- (2) **repeat**
- (3) Step 2: Employed bees' phase
- (4) Step 3: Onlooker bees' phase
- (5) Step 4: Scout bees' phase
- (6) Step 5: Memorize the best solution achieved so far
- (7) **until** maximum cycle number reached
- (8) Output the best solution identified

ALGORITHM 1: ABC algorithm.

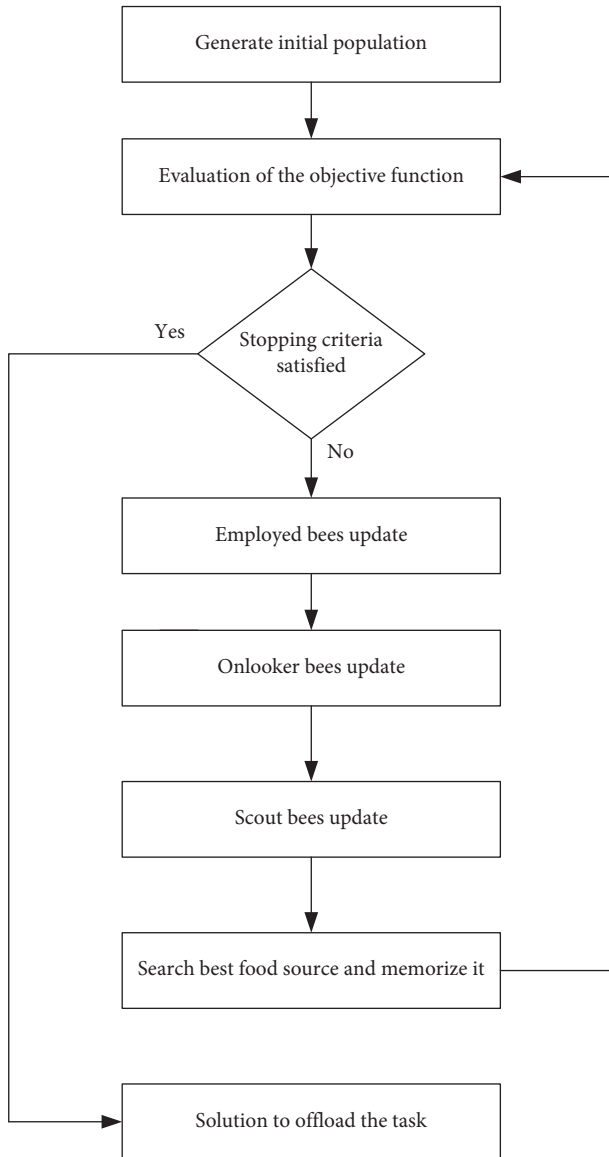


FIGURE 3: Flow chart of computation offloading framework.

$$LI = \frac{\max(H_k) - \min(H_k)}{H_{avg}} \quad (24)$$

The IoT devices at Tier-1 generate the tasks, selected for computation offloading to the edge server. This approach

leverages the IoT devices to save energy and make them capable of handling compute-intensive tasks. The edge tier is placed between IoT tier and cloud, which takes the data load generated by IoT devices and executes the offloaded tasks using a number of edge servers. However, each server has several virtual machine (VM) instances that ensure the successful execution of the offloaded tasks, where each VM handles a different class of IoT applications. The cloud is a resource-rich solution to IoT and is connected to the edge server through Internet for achieving scalability in the edge server. The performance analysis of computation offloading based on ABC algorithm is evaluated using three-tier edge-cloud framework for seamless and successful execution of task offloading between IoT, edge, and cloud. In this experiment, we have considered the response time, standard deviation, and degree of imbalance as a set of parameters. Using these parameters, we tested the performance of ABC-based computation offloading algorithm under different scenarios. However, the degree of imbalance reflects the inequality among multiple edge servers, and the standard deviation of the response time exhibits the load balance between edge servers. The degree of imbalance is expressed using equation (24).

Figure 4 shows the performance of the ABC task offloading algorithm in scenario 1, where the number of edge server  $Eg_n$  is 3, total number of IoT nodes  $M_n$  is 250, and primary server rate is 100. The two delay-sensitive and delay-tolerant applications are generated from IoT nodes. It has been observed that the proposed ABC task offloading algorithm outperforms the counterparts ACO, PSO, and RR Algorithms by keeping the response time well below the defined latency threshold. The ABC algorithm satisfies the QoS requirement of both delay-sensitive and delay-tolerant application. However, the RR scheduler degrades its performance by violating the latency requirements of 100 ms. In scenario 2, experimental parameters are changed by increasing the number of edge servers  $Eg_n$  to 5 and server rate to 300 with a gradual increase in number of IoT devices. In Figure 5, the proposed ABC task offloading algorithm is compared with ACO, PSO, and RR scheduler. The achieved results show that the proposed algorithm maintains the low response time even with the increasing number of computation offloading requests of IoT devices.

In Figure 6, we present the performance of ABC algorithm. It is witnessed that the proposed algorithm maintains lower response time of the offloaded tasks in the single run of

```

(1) Step1: Initialization
(2)  $q \leftarrow \#$  of employed bees,  $r \leftarrow \#$  of onlooker bees
(3)  $Dp \leftarrow \#$  dimension of problem
(4) StoppingCriteria  $\leftarrow$  Max. # of iterations allowed
(5) Create an initial population using Equation (20)
(6) Evaluate the fitness of the population
(7) repeat
(8)   Step 2: Employed bees' phase
(9)    $k = 1$ 
(10)  while  $k < q$  do
(11)    Compute new solution using Equation (21)
(12)    Compute the fitness value of new solution using Equation (23)
(13)    if  $\text{fit}(y_{nk}) > \text{fit}(x_{nk})$  in a neighborhood then
(14)       $x_{nk} = y_{nk}$ , and  $\text{trail}_n = 0$ 
(15)    else
(16)      Increase  $\text{trail}_n$  by 1
(17)    end if
(18)     $k = k + 1$ 
(19)  end while
(20)  Step 3: Onlooker bees' phase
(21)   $Dp = r$ ,  $s = 1$ ,  $k = 0$ 
(22)  while  $s < Dp$  do
(23)    Generate a random number  $pr$  such that  $r \in [0, 1]$ 
(24)    Calculate the probability  $p_{nj}$  using Equation (22)
(25)    if  $pr < p_{nj}$  then
(26)      Compute new solution using Equation (21)
(27)      Compute the fitness value of new solution using Equation (23)
(28)      if  $\text{fit}(y_{nk}) > \text{fit}(x_{nk})$  in a neighborhood then
(29)         $x_{nk} = y_{nk}$ , and  $\text{trail}_n = 0$ 
(30)      else
(31)        Increase  $\text{trail}_n$  by 1
(32)      end if
(33)    end if
(34)     $s = s + 1$ 
(35)     $k = k + 1$ 
(36)    if  $k > Dp$  then
(37)       $k = 1$ 
(38)    end if
(39)  end while
(40)  Step 4: Scout bees' phase
(41)  if  $\text{trail} > \text{limit}$  then
(42)    Initialize randomly chosen solution using Equation (20)
(43)  end if
(44)  Step 5: Memorize the best solution achieved so far
(45) until maximum cycle number reached
(46) Output the best solution identified

```

ALGORITHM 2: ABC to offload computation to the edge/cloud.

TABLE 2: Simulation parameters.

S. no.	Parameters	Value
1	No. of IoT devices	250–2000
2	Edge server	Core i7 (2.6 GHz, 8 GB RAM)
3	IoT cluster radius	100–300 meters
4	Task size	250 Kb–1 MB
5	Latency	100 ms
6	No. of servers	3–8
7	Communication parameters	3GPP



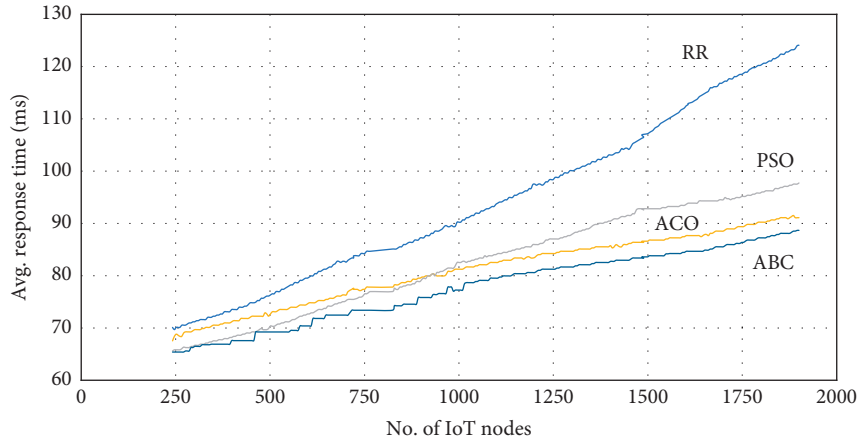


FIGURE 4: Average response time of the offloading tasks.

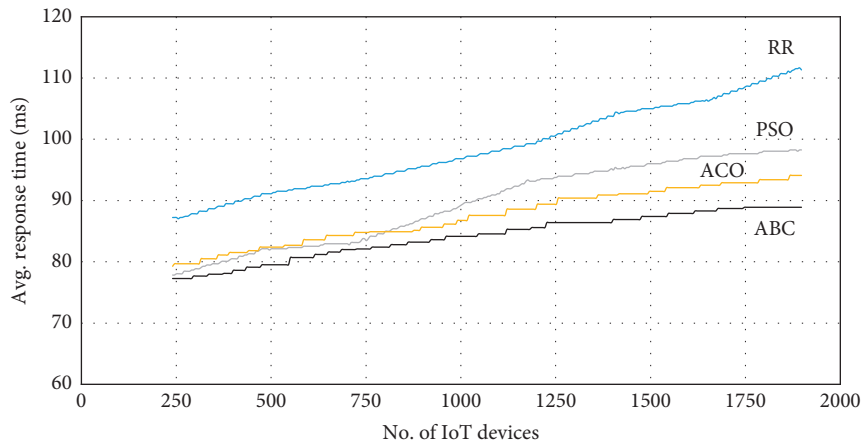


FIGURE 5: Average response time of the offloaded tasks.

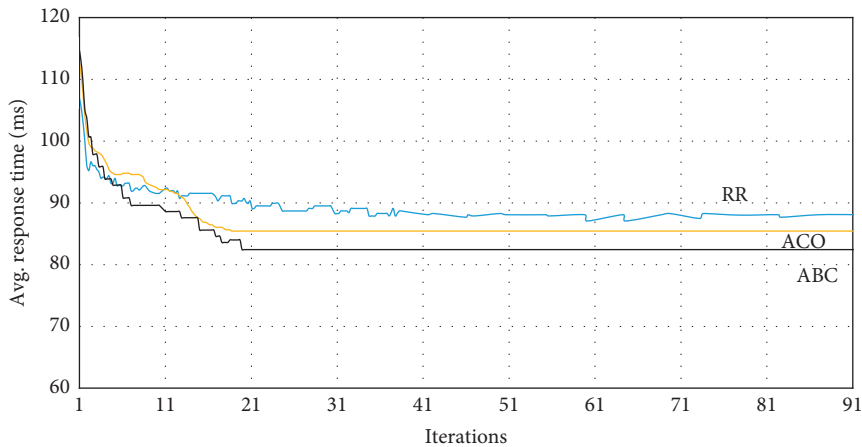


FIGURE 6: Average response time of the offloaded tasks.

the ABC algorithm. In scenario 3, the experimental setup is deployed with the following parameter settings:  $Eg_n = 8$ , server rate = 500, and  $M_n = 2000$ . However, the number of applications is increased to 3. This setup is objectively designed to simulate real IoT-based smart system environment, where heavy traffic is generated by IoT nodes.

We recorded the average response time of the bees. The ABC algorithm minimizes the objective function by discovering minimum computational cost and low latency of each offloaded task. The probabilistic calculation and memorization of the fitness value reach the minimum fitness value immediately in the 11<sup>th</sup> iteration. The proposed ABC

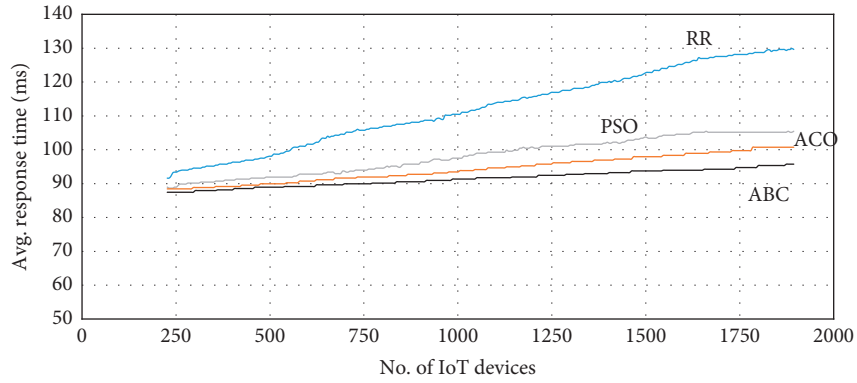


FIGURE 7: Average response time of the offloaded tasks.

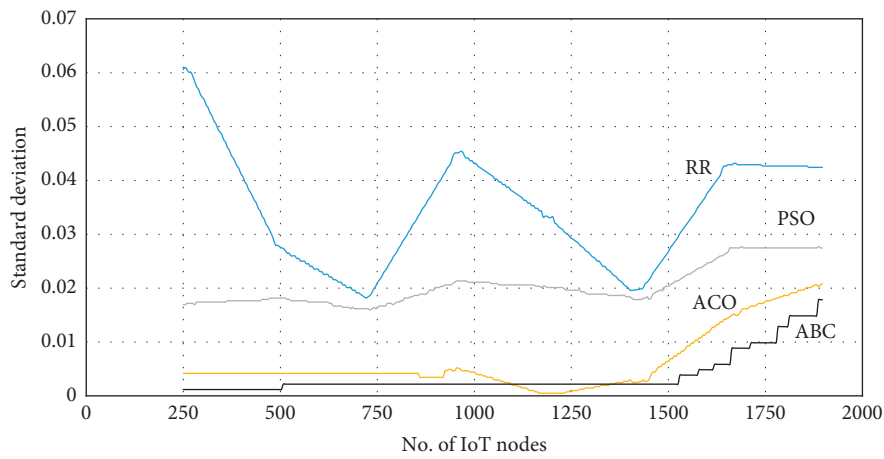


FIGURE 8: Standard deviation of the response time on the edge nodes.

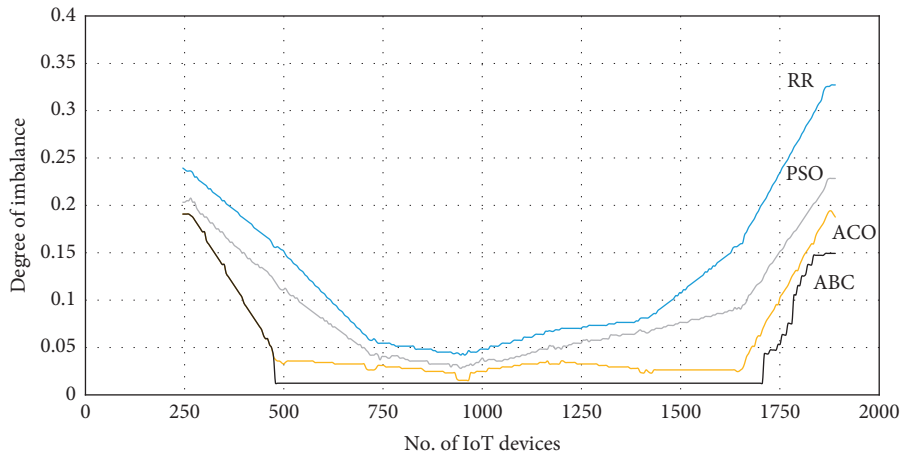


FIGURE 9: Degree on imbalance of the offloaded tasks.

algorithm explores the search space quickly, resulting in faster convergence compared to the ACO and PSO for the best possible solution. In addition, it achieves the best value in a short period of time.

In scenario 4, we changed a set of parameters by introducing three different classes of applications using three

different types of sensors. Each sensor generates different data rate according to the task with having number of edge servers  $Eg_n=8$ , the server rate = 500, and number of IoT nodes  $M_n=2000$ .

In Figure 7, the results exhibit that the ABC task offloading algorithm maintains the low response time because

the onlooker bee probabilistically selects and memorizes the successful node while looking for other probabilistic solutions and outperforms the RR, PSO, and ACO algorithms. Figure 8 reflects the behaviour of the proposed ABC algorithm by considering the standard deviation. The standard deviation is the variation between the average response times of all the tasks that are offloaded for remote execution. As the number of IoT devices increases, the proposed algorithm shows an improved performance in the standard deviation. However, the standard deviation grows exponentially while exceeding 1750 IoT devices, which is due to the inherent issue of scalability in edge computing. These results are achieved through the same parameters mentioned in scenario 4. Figure 9 exhibits the degree of imbalance of the offloaded tasks over edge servers while increasing the number of offloading tasks from IoT layer using scenario 4. It is observed that the proposed ABC algorithm minimizes the objective function, produces low values, and reflects that the workload is effectively distributed among the edge servers.

## 5. Conclusion and Future Work

Smart city is designed using a large number of IoT devices. These devices are resource-limited and their applications are resource-intensive, which require high QoS. Nonetheless, they produce a large amount of data in a short period of time. Cloud can be a feasible solution for it, but the inherent longer latency makes it nonviable. Edge computing is a potential solution that resides in the close proximity of the users. It offers low latency, high bandwidth, crisp response, and reliability for the resource-limited IoT devices. Therefore, in this paper, we proposed a three-tier edge-cloud integration architecture and utilized a classical computation offloading technique for seamless task offloading process. A metaheuristic and nature-inspired ABC optimization technique is used to balance the workload over edge servers while considering network latency and service rate of the edge servers. The numerical results exhibit that the proposed ABC algorithm produced notable improvement in response time of IoT applications. In addition, the results also ensure that the workload over edge servers is managed effectively, which scales the edge server for entertaining maximum number of offloaded tasks.

In the future, we aim to extend this work to LTE and 5G communication architecture using multiobjective optimization, including communication cost and energy consumption cost.

## Data Availability

The data used to support the findings of this study are included within the article. The used simulations software and its details are mentioned in the Results section that help to reach conclusions.

## Disclosure

Mohammad Babar and Farman Ali are the co-first authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Mohammad Babar and Farman Ali contributed equally to this work.

## Acknowledgments

This work was supported by the National Research Foundation of Korea-Grant funded by the Korean Government (Ministry of Science and ICT, NRF-2020R1A2B5B02002478).

## References

- [1] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for iot security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, 2020.
- [2] L. U. Khan, I. Yaqoob, N. H. Tran, S. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge computing enabled smart cities: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10200–10232, 2020.
- [3] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 25, pp. 1–24, 2020.
- [4] C. K. M. Lee, Y. Z. Huo, S. Z. Zhang, and K. K. H. Ng, "Design of a smart manufacturing system with the application of multi-access edge computing and blockchain technology," *IEEE Access*, vol. 8, pp. 28659–28667, 2020.
- [5] W. Yan, Z. Wang, H. Wang, W. Wang, J. Li, and X. Gui, "Survey on recent smart gateways for smart home: systems, technologies, and challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 31, Article ID e4067, 2020.
- [6] I. Lee and K. Lee, "The internet of things (IoT): applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.
- [7] M. Babar, M. S. Khan, F. Ali, M. Imran, and M. Shoaib, "Cloudlet computing: recent advances, taxonomy, and challenges," *IEEE Access*, vol. 9, pp. 29609–29622, 2021.
- [8] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. ur Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1761–1804, 2020.
- [9] K. Jain and S. Mohapatra, "Taxonomy of edge computing: challenges, opportunities, and data reduction methods," *Edge Computing*, Springer, New York, NY, USA, pp. 51–69, 2019.
- [10] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020.
- [11] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *Journal of Network and Computer Applications*, vol. 169, Article ID 102781, 2020.
- [12] Q.-H. Nguyen and F. Dressler, "A smartphone perspective on computation offloading—a survey," *Computer Communications*, vol. 159, pp. 133–154, 2020.
- [13] F. Wei, S. Chen, and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," *China Communications*, vol. 15, no. 11, pp. 149–157, 2018.

- [14] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2450–2463, 2018.
- [15] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, "A survey of computation offloading in edge computing," in *Proceedings of the 2020 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–6, Hangzhou, China, October 2020.
- [16] F. Ali, S. El-Sappagh, S. M. R. Islam et al., "An intelligent healthcare monitoring framework using wearable sensors and social networking data," *Future Generation Computer Systems*, vol. 114, pp. 23–43, 2021.
- [17] F. Ali, S. El-Sappagh, S. M. R. Islam et al., "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion," *Information Fusion*, vol. 63, pp. 208–222, 2020.
- [18] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: a survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.
- [19] G. Orsini, D. Bade, and W. Lamersdorf, "Context-aware computation offloading for mobile cloud computing: requirements analysis, survey and design guideline," *Procedia Computer Science*, vol. 56, pp. 10–17, 2015.
- [20] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: towards minimizing delay in the internet of things," in *Proceedings of the 2017 IEEE International Conference on Edge Computing (EDGE)*, pp. 17–24, Honolulu, HI, USA, June 2017.
- [21] J. Luo, X. Deng, H. Zhang, and H. Qi, "QoE-driven computation offloading for edge computing," *Journal of Systems Architecture*, vol. 97, pp. 34–39, 2019.
- [22] H. Fouad, N. M. Mahmoud, M. S. E. Issawi, and H. Al-Feel, "Distributed and scalable computing framework for improving request processing of wearable IoT assisted medical sensors on pervasive computing system," *Computer Communications*, vol. 151, pp. 257–265, 2020.
- [23] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242–4251, 2019.
- [24] Z. Zhao, R. Zhao, J. Xia et al., "A novel framework of three-hierarchical offloading optimization for MEC in industrial IoT networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5424–5434, 2020.
- [25] H. Zhao, S. Deng, C. Zhang, W. Du, Q. He, and J. Yin, "A mobility-aware cross-edge computation offloading framework for partitionable applications," in *Proceedings of the 2019 IEEE International Conference on Web Services (ICWS)*, pp. 193–200, Milan, Italy, July 2019.
- [26] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: a deep learning approach," in *Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, Montreal, Canada, October 2017.
- [27] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Communications*, vol. 13, no. 3, pp. 156–164, 2016.
- [28] B. M. Nguyen, H. Thi Thanh Binh, T. The Anh, and D. Bao Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment," *Applied Sciences*, vol. 9, no. 9, p. 1730, 2019.
- [29] H. T. T. Binh, T. T. Anh, D. B. Son, P. A. Duc, and B. M. Nguyen, "An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment," in *Proceedings of the 2018 Ninth International Symposium on Information and Communication Technology*, pp. 397–404, Da Nang, Vietnam, December 2018.
- [30] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: distributed adaptive deep learning inference on resource-constrained IoT edge clusters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2348–2359, 2018.
- [31] S. Azimi, C. Pahl, and M. H. Shirvani, "Particle swarm optimization for performance management in multi-cluster IoT edge architectures," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science, CLOSER 2020*, pp. 328–337, Prague, Czech Republic, February 2020.
- [32] M. Bouet and V. Conan, "Mobile edge computing resources optimization: a geo-clustering approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 787–796, 2018.
- [33] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "A multi-clustering approach to scale distributed tenant networks for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 499–514, 2019.
- [34] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. H. Tsang, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 72–84, 2015.
- [35] R. Yu, X. Huang, J. Kang et al., "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, 2015.
- [36] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.
- [37] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 794–801, Abu Dhabi, UAE, October 2015.
- [38] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between internet of things and social networks: review and research challenges," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 206–215, 2014.
- [39] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

## Review Article

# Deep CNN and Deep GAN in Computational Visual Perception-Driven Image Analysis

**R. Nandhini Abirami** <sup>1</sup>, **P. M. Durai Raj Vincent** <sup>1</sup>, **Kathiravan Srinivasan** <sup>1</sup>,  
**Usman Tariq** <sup>2</sup>, and **Chuan-Yu Chang** <sup>3</sup>

<sup>1</sup>*School of Information Technology and Engineering, Vellore Institute of Technology (VIT), Vellore 632014, India*

<sup>2</sup>*College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*

<sup>3</sup>*Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan*

Correspondence should be addressed to P. M. Durai Raj Vincent; [pmvincent@vit.ac.in](mailto:pmvincent@vit.ac.in) and Chuan-Yu Chang; [chuanyu@yuntech.edu.tw](mailto:chuanyu@yuntech.edu.tw)

Received 7 January 2021; Revised 11 February 2021; Accepted 18 March 2021; Published 20 April 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 R. Nandhini Abirami et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Computational visual perception, also known as computer vision, is a field of artificial intelligence that enables computers to process digital images and videos in a similar way as biological vision does. It involves methods to be developed to replicate the capabilities of biological vision. The computer vision's goal is to surpass the capabilities of biological vision in extracting useful information from visual data. The massive data generated today is one of the driving factors for the tremendous growth of computer vision. This survey incorporates an overview of existing applications of deep learning in computational visual perception. The survey explores various deep learning techniques adapted to solve computer vision problems using deep convolutional neural networks and deep generative adversarial networks. The pitfalls of deep learning and their solutions are briefly discussed. The solutions discussed were dropout and augmentation. The results show that there is a significant improvement in the accuracy using dropout and data augmentation. Deep convolutional neural networks' applications, namely, image classification, localization and detection, document analysis, and speech recognition, are discussed in detail. In-depth analysis of deep generative adversarial network applications, namely, image-to-image translation, image denoising, face aging, and facial attribute editing, is done. The deep generative adversarial network is unsupervised learning, but adding a certain number of labels in practical applications can improve its generating ability. However, it is challenging to acquire many data labels, but a small number of data labels can be acquired. Therefore, combining semisupervised learning and generative adversarial networks is one of the future directions. This article surveys the recent developments in this direction and provides a critical review of the related significant aspects, investigates the current opportunities and future challenges in all the emerging domains, and discusses the current opportunities in many emerging fields such as handwriting recognition, semantic mapping, webcam-based eye trackers, lumen center detection, query-by-string word, intermittently closed and open lakes and lagoons, and landslides.

## 1. Introduction

Computer vision (CV), the core component of machine intelligence, is an interdisciplinary field enabling computers to achieve a visual understanding of digital images. For a machine to view as animals or people do, it relies on computer vision. Table 1 contains the list of abbreviations and their expansion used in the manuscript. CV is a

booming field and is applied to many of our everyday activities; some of them are face detection, object detection, biometrics, a medical diagnosis from faces, self-checkout kiosk, autonomous vehicles, image recognition, image enhancement, image deblurring, motion tracking, video surveillance, control of robots, analysis of mammography, and X-rays [22–26]. The fundamental goal of all these applications is to create a human observer replica in interpreting the

TABLE 1: List of abbreviations used in this manuscript along with their expansion.

Abbreviation	Full form	Authors
CV	Computer vision	Roberts, 1963 [1]
D-CNN	Deep convolutional neural network	Lecun et al., 1998 [2]
RNN	Recurrent neural network	Graves, 2006 [3]
DNN	Deep neural network	Ivakhnenko, 1971 [4]
AI	Artificial intelligence	John McCarthy, 1956
MNIST	Mixed National Institute of Standards and Technology	Lecun et al. ( <a href="http://yann.lecun.com/exdb/mnist/">http://yann.lecun.com/exdb/mnist/</a> )
ReLU	Rectified linear unit	Hahnloser et al., 2000 [5]
COCO	Common objects in context	Lin et al., 2014 [6]
D-GAN	Deep generative adversarial network	Goodfellow et al., 2014 [7]
DCGAN	Deep convolutional GAN	Radford et al., 2015 [8]
SRGAN	Super-resolution generative adversarial networks	Ledig et al., 2017 [9]
APGAN	Laplacian pyramid GAN	Denton et al., 2015 [10]
SAPGAN	Self-attention generative adversarial networks	Zhang, et al., 2019 [11]
GRAN	Generating images with recurrent adversarial networks	Im, et al., 2016 [12]
GPF-CNN	Gated peripheral-foveal convolutional neural network	Hahnloser et al., 2000 [5]
PSGAN	Pose and expression robust spatial-aware GAN	Jiang et al., 2019 [13]
ResNet	Residual neural network	He Zhang et al., 2016 [28]
CRGAN	Conditional recycle GAN	Li et al., 2018 [14]
ACGAN	Auxiliary classifier GAN	Odena et al, 2017 [15]
CGAN	Conditional GAN	Gauthier et al., 2014 [16]
InfoGAN	Information maximizing GAN	Chen et al., 2016 [17]
LAPGAN	Laplacian pyramid of adversarial networks	Denton et al., 2015 [10]
SAGAN	Self-attention GAN	Zhang et al., 2018 [11]
VAEGAN	Variational autoencoder GAN	Larsen et al., 2016 [12]
BIGAN	Bidirectional GAN	Rui et al., 2020 [18]
AAE	Adversarial autoencoders	Makhzani et al., 2016 [19]
MCGAN	Mean and covariance feature matching GAN	Mroueh et al., 2017 [20]
GRAN	Generative recurrent adversarial networks	Daniel et al., 2016 [12]
LSGAN	Least squares generative adversarial networks	Mao et al., 2016 [29]
WGAN	Wasserstein GAN	Martin et al., 2017 [21]

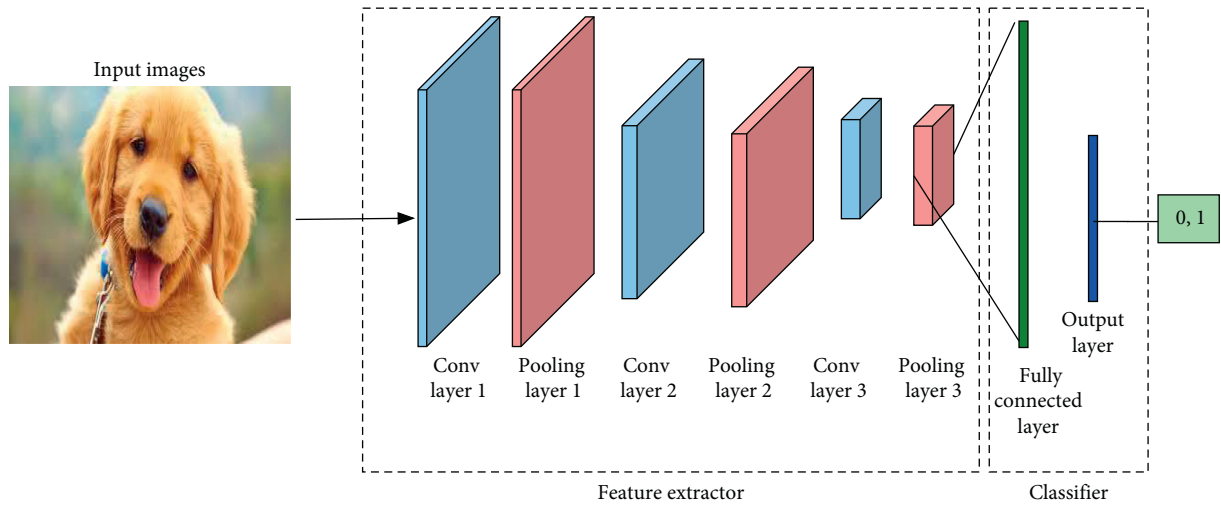
scene in a broad sense and perform decision-making for the task at hand [27]. CV and image processing are confusing terms and are often used interchangeably. Image processing receives images as the input, processes them, and outputs images, while CV receives images as the input, processes them, and interprets the images. The output generated by CV is an abstract representation of the image's constituent or the entire image. D-GAN is proposed as unsupervised learning, but adding a certain number of labels in practical applications can improve its generating ability. However, it is challenging to acquire many data labels, but a small number of data labels can be obtained. Therefore, combining semisupervised learning and GAN is one of the future directions.

Figure 1(a) represents the general architecture of the deep convolutional neural network (D-CNN). D-CNN is similar to a neural network where D-CNN is built with neurons having learning weights and biases [30]. However, in recent times, D-CNN is widely used over a standard neural network as it is faster and computationally inexpensive compared to neural networks. An image, which is a matrix of pixels, is flattened and fed into the neural network. Furthermore, to flatten an image of size  $40 \times 30 \times 1$ , 1200 neurons are required at the input layer. Here, the complexity is manageable using a neural network. Colored images have layers corresponding to RGB. A total of 3 layers for each color make the number of neurons required at the input

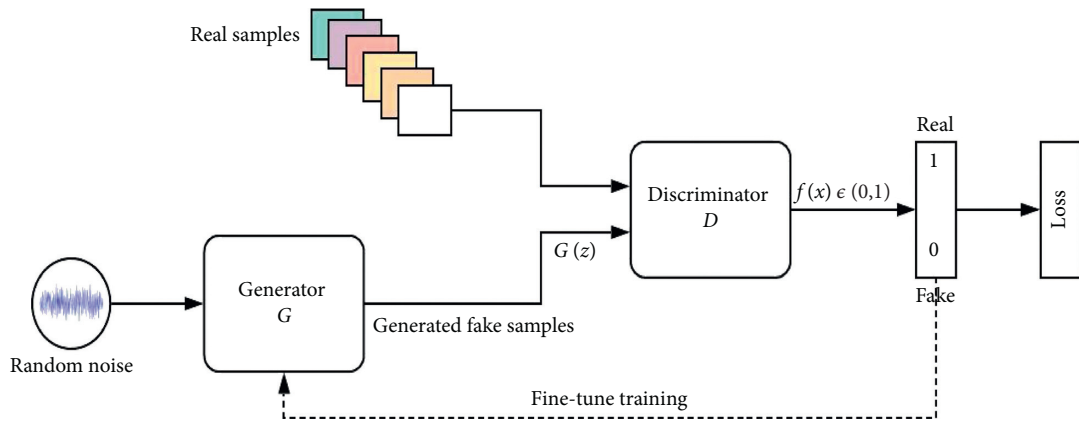
layer very high. When an image of size  $1024 \times 1024 \times 3$  has to be fed into the neural network, 3,145,728 neurons are required at the input layer, which is computationally expensive. The number of neurons needed at the input layer increases exponentially as the size of the image increases.

Figure 1(b) represents the architecture of deep generative adversarial networks (D-GAN), where  $G$  captures the data distribution and generates fake data  $G(z)$  whose distribution is  $p_z(z)$ . The generative model improvises to generate distributions similar to  $p$  data( $x$ ), the real data distribution. The discriminator  $D$  is fed either with an actual data sample or generated data sample  $G(z)$ . The discriminator outputs a probability  $f(x)$  belonging to  $(0, 1)$ , indicating the data source. The generative model is trained to capture the data distribution from the original data. The discriminative model predicts the probability that data have originated from the generator  $G$  rather than the original training data. The generator's goal is to generate data close to the distribution of real data and deceive the discriminator. The purpose of the discriminator is to identify the fake data generated by the generator. Model distributions are generated by feeding random noise as the input through a multilayer perceptron. The discriminator has a multilayer perceptron with a classifier at the end [31].

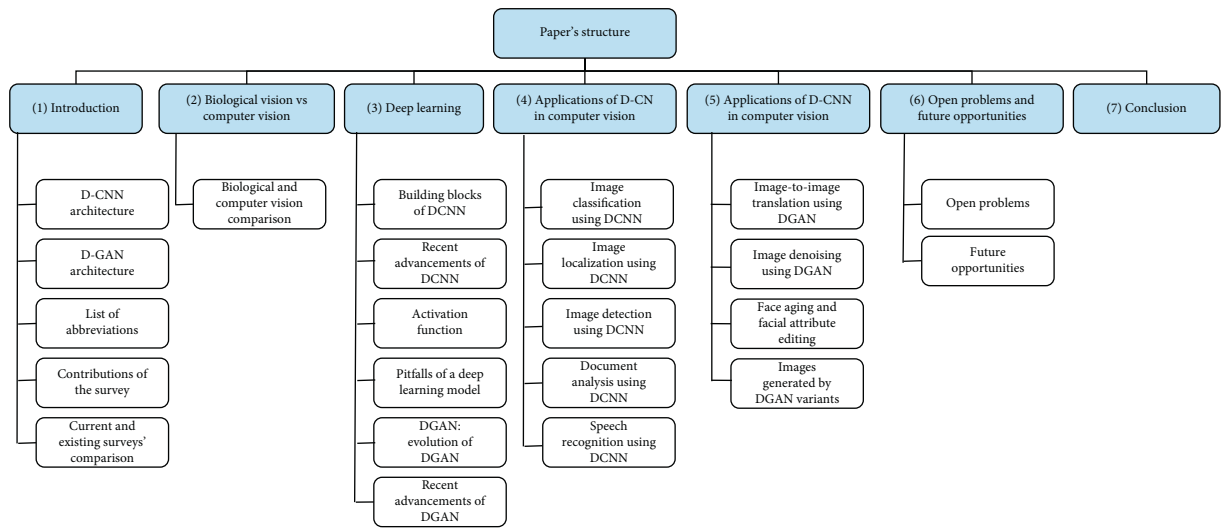
The generator and discriminator compete until the counterfeits generated by the generator are indistinguishable from the data distribution. Through adversarial training, the



(a)



(b)



(c)

FIGURE 1: (a) The general architecture of deep CNN. (b) The architecture of deep generative adversarial networks. (c) Structure of this survey.

quality of data generated by the generator gradually improves [32]. The quality of data samples generated by the generator and the discriminator's identifying capability improves each iteration interactively. The generator can be any neural network such as artificial neural network, convolutional neural network, recurrent neural network, or long short-term memory, whose task is to learn the data distribution. Simultaneously, a discriminator is essentially a binary classifier capable of classifying the input to be real or fake. The entire network is trained using backpropagation to fine-tune the training. The error is estimated using the sample label, and the discriminator output and the parameters of  $G$  are updated using an error backpropagation algorithm. D-GAN is inspired by two-player minimax game theory, which has two players, one benefitting at the loss of the other, and is represented by the following equation [7]:

$$\min G \max D V(D, G) = E_{x \sim p \text{ data}(x)} [\log D(x)] + E_{z \sim p z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where  $p \text{ data}(x)$  is the model data distribution and is the  $G(z)$  generated data distribution.

The output of a discriminator is a probability indicating the origin of the data sample. A probability of 1 or a number very close to 1 represents that the data sample is real data. A probability of 0 or a number close to 0 represents the fake data. When the probability is close to 0.5, it indicates that the discriminator finds it hard to identify counterfeit samples.  $G$  is trained repeatedly to make  $D$ 's output approach 1 for the data samples generated by  $G$ . The model is trained until Nash equilibrium is achieved where a change in strategy does not change the game anymore. The Nash equilibrium is achieved when the generator has gained the capability to generate data close to the real data. The discriminator does not distinguish the real data and generator data. The generator is now considered to have to learn the real-data distribution.

*1.1. Contributions of This Survey.* This paper presents the general architecture of D-CNN, its application, various methodologies adopted, and its application-based performance. An overview of D-GANs is also discussed with their existing variants and their application in different domains. Furthermore, this paper identifies GANs' advantages, disadvantages, and recent advancements in the field of computer vision. Also, it aims to investigate and present a comprehensive survey of the essential applications of GANs, covering crucial areas of research with their architectures. Figure 1(c) shows the structure of the survey. This survey presents a detailed description of D-CNN and D-GAN with their architectures. Recent advancements of D-CNN and D-GAN are discussed with their applications. Activation functions used for the CNN are discussed, and various pitfalls of deep learning with their possible solutions are discussed in detail. Applications of D-CNN and D-GAN are analyzed in Sections 4 and 5. Table 2 shows a comparison between the current survey and existing surveys on D-CNN and D-GAN.

## 2. Biological Vision vs. Computer Vision

Biological vision has tremendous capabilities in retrieving vital information from visual data and analyzing them for functional needs. The perceptual mechanism used by people and animals to interpret the visual world is diverse. Research on biological vision is an excellent source of inspiration for CV and focuses on computationally understanding brain functions' mechanism for visual interpretation. Understanding the perceptual mechanism of biological vision is the initial step towards interpreting the visual data. Computational understanding of biological vision in the current research studies is based on the framework defined by David Marr [40]. Biological vision can perform tasks with high reliability, even if the visual data are noisy, cluttered, and ambiguous. It can efficiently solve computationally complex problems and that are still challenging for CV. The fundamental goal of CV, the science of image analysis, is to automate computational methods to extract visual information and understand the image's content for decision-making [41, 42]. From CV's perspective, an image is a sequence of square pixels that may be aligned as an array or matrix. At a higher level, the structure of both biological and computer vision is the same. Nevertheless, both systems' objective is the same: to extract and represent the visual data into useful information for making actions.

## 3. Deep Learning

Deep learning or hierarchical learning has emerged as a subfield of machine learning, which, in turn, is a subfield of artificial intelligence [43]. Artificial intelligence is an effort to make machines think and automatically perform intellectual tasks otherwise performed by humans. AI is a classical programming paradigm where humans craft rules for the data, and the machine outputs the answers. Questions arose as if a machine could automatically learn data processing rules by looking at the data. Machine learning, a new programming paradigm, came into existence as an answer to this question. With machine learning, data and the solutions were fed for the machines to craft the rules. A machine learning model is trained rather than being explicitly programmed. Machine learning and deep learning came into existence when a need arose to solve fuzzy and more complex problems such as language translation, speech recognition, and image classification [44, 45]. At its core, deep learning and machine learning are about learning the representation of the data at hand to get the expected output. Deep learning models are capable of learning complex relationships existing between the inputs and the outputs. Deep learning uses multiple processing layers to discover the data's intricate structure with multiple abstraction levels [46]. The deep in deep learning is a reference to successive layers of representation. Weights parameterize the multiple nonlinear hidden layers in a deep learning model. For a network to correctly map the inputs to its targets in a deep learning model, proper values are to be set for the weights of all layers present in a network.



TABLE 2: Comparison between the current and the existing surveys in the literature.

S. no	Paper title	Survey objective (existing)	Survey objective (current)
1	A survey of generative adversarial networks [33]	State-of-the-art GAN architectures are surveyed, and their application domains on natural language processing and computer vision are discussed. The loss functions of the GAN variants are discussed.	State-of-the-art GAN is discussed along with its performance on the MNIST dataset. Generator and discriminator losses are visually represented for the GAN variants.
2	Recent progress on generative adversarial networks (GANs): A survey [34]	Basic theory and different GAN models are summarized. The models derived from the GAN are classified, and evaluation metrics are discussed.	Variants of the GAN, their application, architecture, methodology, advantage, and disadvantages are analyzed and summarized. Evolution of the GAN with conditions, encoders, loss functions, and process discrete data are separately discussed.
3	A survey of the recent architectures of deep convolutional neural networks [35]	An overview of different layers of D-CNN, namely, the convolutional layer and pooling layer, is discussed. An outline of the pitfalls of deep learning is briefed.	Different layers of D-CNN, namely, the convolution layer, pooling layer, and the operations performed in the convolution and pooling layers, are discussed in detail. A detailed review of deep learning pitfalls, namely, overfitting, underfitting, and data insufficiency, is discussed along with their possible solutions.
4	Deep learning for generic object detection: A survey [36]	Recent achievements in the field of object detection have been discussed.	Recent advancements of the D-CNN in computer vision have been tabulated and discussed with their methodology and performance. Activation functions that are used for computer vision problems are tabulated.
5	A survey on image data augmentation for deep learning [37]	This survey presents the existing methods for data augmentation.	Advantages of data augmentation and comparing results showing the model's performance with and without data augmentation are accomplished.
6	Adversarial-learning-based image-to-image transformation: A survey [38]	This survey presents an overview of adversarial learning-based methods by focusing on the image-to-image transformation scenario.	The existing survey mainly focused on image-to-image translation. This survey discusses several applications based on adversarial learning.
7	Survey of convolutional neural networks for image captioning [39]	This survey presents a shallow overview of image captioning performed using D-CNN.	This survey elaborately discusses various applications using the D-CNN.

Any deep learning problem has an actual target value and the predicted value. The difference between the actual and the predicted value is called the loss function. A distance score, which represents the network performance, is computed based on real and predicted values. Initially, the input's weights are randomly assigned, and the expected output is far from the actual output, and accordingly, the distance score is very high. The weights are then adjusted, the training loop is repeated, and the distance score decreases. Tens and hundreds of iterations are performed over thousands of examples, and a minimal loss value represents that the outputs are close to the target. Deep learning has exponentially improved state-of-the-art object detection, speech recognition, and many other domains [47]. Figure 2 shows the deep learning models. D-CNN has excelled in processing images, speech, audio, and video, while RNN has brought a breakthrough in processing sequential data.

*3.1. Deep Convolutional Neural Network.* The deep convolutional neural networks, popularly known as D-CNN or

D-ConvNets, are a robust ANN class and are the most established deep learning algorithm that have become dominant in computer vision and tons of other applications [48]. Convolutional layers, pooling layers, and fully connected layers are the D-CNN [49] building blocks [50]. D-CNN is designed to process the data that arrive in multiple arrays or grids through its numerous building blocks. The convolutional and pooling layers' role is to extract the features, while fully connected layers map the extracted features to the output. Deep CNN has multiple convolutions and pooling layers, followed by single or multiple fully connected layers. The input passes through these layers and gets transformed into output through forwarding propagation. Convolution operation and activation function are the backbones of the D-CNN [51].

Tensor, kernel, and feature map are the three essential terminologies to perform convolution operation. Tensor is the input, which is a multidimensional array, the kernel is a small array of numbers, and the feature map is the output tensor, shown in Figure 3. The convolution operation is a linear process where a dot product is performed between the

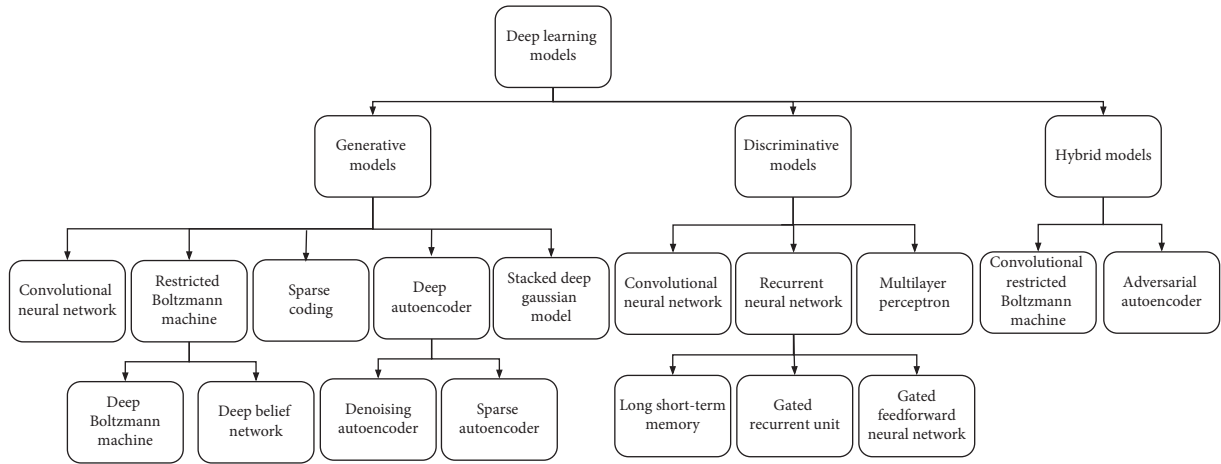


FIGURE 2: Deep learning models—taxonomy.

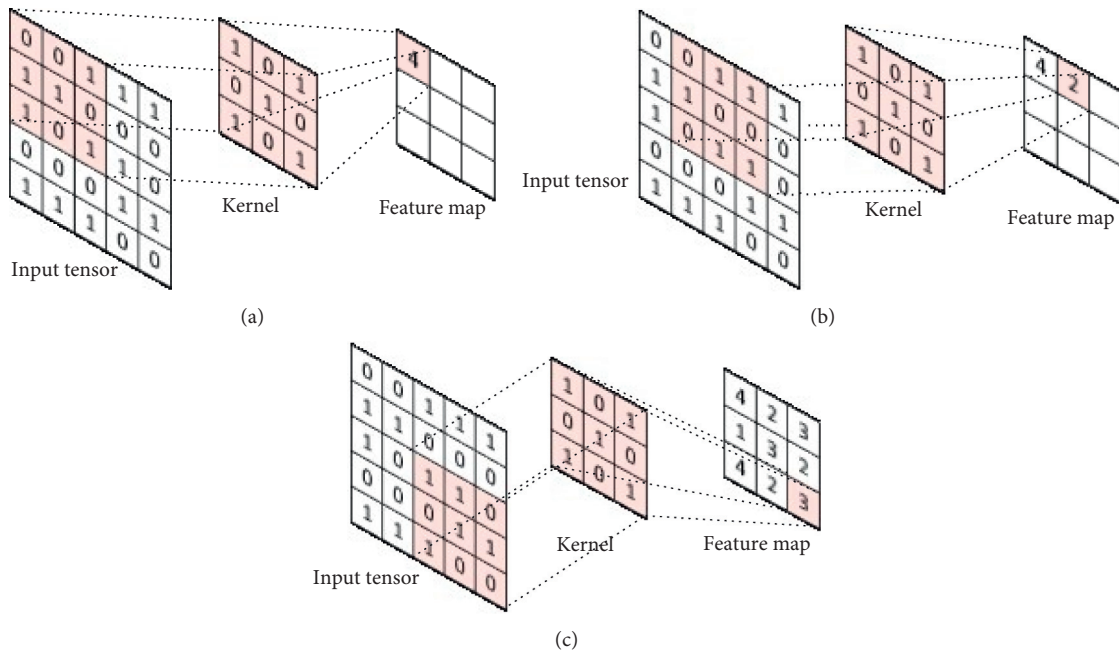


FIGURE 3: Convolution operation.

kernel and the input tensor. Each element of the kernel is multiplied with the corresponding tensor element and summed up to arrive at the output value placed in the feature map's corresponding position [52]. The convolution operation is defined by kernel size and the number of kernels. The kernel size may be  $3 \times 3$ ,  $5 \times 5$ , or  $7 \times 7$  based on the size of the input tensor. The number of kernels is arbitrary, each kernel representing various characteristics of the input. The convolution operation is repeated for each kernel. Figures 3(a)–3(c) show an example of a convolution operation. Here, the kernel size is  $3 \times 3$ , which is applied across each input tensor element to perform a dot product and return the corresponding value for the output tensor. A drawback of the convolution operation is that the feature map shrinks in size compared to the input tensor. Moreover,

this is because the kernel center is not applied across the bordering elements at the input tensor's right. With a  $5 \times 5$  input tensor, the feature map size shrunk to  $3 \times 3$ . The size of the feature map  $f_s$  for a  $t \times t$  tensor and a  $k \times k$  kernel is determined using the following formula:

$$f_s = (t - k + 1) * (t - k + 1). \quad (2)$$

Applying this formula, a 49-pixel input will shrink into a 25-pixel feature map, which will further shrink when the process is repeated, resulting in the loss of essential features of the input. Furthermore, to address this issue in deep CNN models with more layers, padding is used, where columns and rows of zeroes are added on all the input tensor sides. Moreover, this is performed to fit the center of the kernel to

the input tensor's rightmost bordering elements for maintaining the size of the feature map the same as that of the input tensor. Figure 4 shows zero padding where rows and columns are added to all the sides of the input tensor. As a result of zero padding, the feature map's size is  $5 \times 5$ , the same as the input tensor.

The number of pixel shifts performed by the kernel over the input tensor is called stride. When the value of stride is 1, the kernel shifts 1 pixel at a time. When the value of stride is 2, the kernel shifts 2 pixels at a time, and so on. Activation functions that are frequently used are logistic sigmoid, hyperbolic tangent, and ReLU. Table 3 presents the recent advancements of the D-CNN in computer vision.

**3.2. Activation Functions.** The deep learning mechanism is the input is fed into the network, and to the product of input and the weights, a bias is added. An activation function is then applied to the result, and the same process is repeated until the last layer is reached. Activation functions play a significant role in a neural network to define a neuron's output for a given set of inputs. The activation function takes up the weighted sum of inputs and performs a transformation operation to compress the output between a lower and upper limit. Activation functions are of two types: linear and nonlinear. Deep learning uses nonlinear activation functions for all its classification problems as the output lies between 0 and 1. Without nonlinearity, each layer of the network would execute linear transformations, in which case an equivalent single layer can replace the hidden layers. For a back-propagation to be executed, it is required that the activation function be differentiable. For deep learning, an activation function has to be both nonlinear and differentiable. Some of the standard activation functions in deep learning are sigmoid, tanh, ReLU, and leaky ReLU.

Table 4 shows the most frequently used activation functions. Sigmoid transforms the output between 0 and 1. In recent times, sigmoid has become one of the least used activation functions because of its drawbacks. First, it causes gradients to vanish when the neuron's activation saturates close to 0 or 1; the gradients in this region are close to zero. The second drawback is that the output is not zero-centered. tanh is another activation function that performs better than the sigmoid activation function. The output lies in the range of  $-1$  and  $1$  [69]. ReLU is the most popular and frequently used activation function in deep learning. The two problems overcome with ReLU are slow training time of the S-type activation function and vanishing gradient [70, 71]. The mathematics behind ReLU is when the output is 0, conversely if, the output is a linear function [72]. The range of the output is between 0 and infinity.

Since ReLU has zero output for the input's negative values, the gradient will be zero at this point because the network will not respond to any variations in the input or the error. This problem can make part of the network passive because of dead neurons. This problem called dying leaky ReLU can overcome ReLU. Leaky ReLU is similar to ReLU, except that leaky ReLU does not make the negative input to

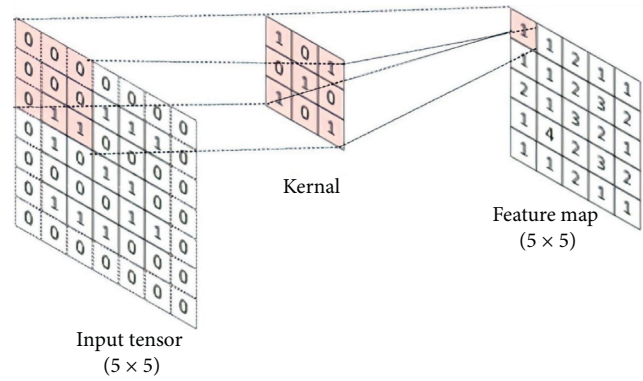


FIGURE 4: Convolution operation with zero padding.

zero. Instead, it gives a small nonzero value of 0.01 in case of a negative regime of the input. The range is between  $-\infty$  and  $\infty$ . The purpose of leaky ReLU is to minimize the dying neuron input problem. In multiclass classification problems, the output layer employs softmax as the classification function [73]. Softmax produces output that sums up to a numerical value of 1. So, the output of softmax specifies the probability distribution for  $n$  different classes of the target. Moreover, this is why softmax is used explicitly in multiclass classification problems. Due to the gradient disappearance problem, tanh and sigmoid activation functions are not employed on the D-CNN.

**3.3. Pitfalls of a Deep Learning Model.** The model's performance indicates how well the model is trained and how well it generalizes new or unseen data. Evaluating the performance of a model is a crucial step in data science. The common barriers in creating high-performance models are overfitting, underfitting, and significantly few training data. Figure 5 shows overfitting, and it is said to occur when a model performs so well on a training set. The performance of the model depreciates on the validation set—loss during the training phase decreases, but the loss during the validation phase increases. Furthermore, this is because the model learns even the unnecessary information from the training set; hence, the model's performance is too good on the training set.

Nevertheless, the model fails to perform on the validation set. Overfitting can be addressed by improving the model and obtaining more training data. The model can be enhanced by randomly omitting feature detectors from the model's architecture. This technique is called dropout, developed by Geoff Hinton [74]. A vast number of different networks can be trained in a reasonable time using random dropouts. Thus, different networks are presented for each training case. In a nutshell, the dropout technique assumes that a randomly selected portion of the network is muted for each training case [75]. Furthermore, this is a useful technique as it prevents any single neuron within the network from becoming excessively influential. Thus, the model does not rely too much on any specific feature of the data. Dropout is used when there is an overfitting. Dropout can improve the validation accuracy in later epochs, even if there is no overfitting. Dropout is added based on experimentation, and the usual dropout range is

TABLE 3: Recent advancements of the D-CNN in computer vision.

S. no	Application	The objective of the study	Methodology/ network architecture	Performance	Dataset	Depth/layer sizes
1	Olive fruit variety classification [53]	To provide computer vision methodology for automatic classification of seven different olive fruit varieties using an image processing technique	Six different D-CNN architectures, namely, AlexNet, residual neural network-50, residual neural network-101, inception-residual neural network V2, Inception V1, and Inception V3, were employed.	AlexNet: 89.90%, residual neural network-50: 94%, residual neural network-101: 95.91%, inception-residual neural network V2: 91.81%, Inception V1: 94.86%, and Inception V3: 95.33%.	The dataset was generated from 400 photographs of olive fruits of each variety.	Five convolutional layers
2	Fabric defect detection [54]	On-loom fabric defect detection combines image preprocessing, candidate defect map generation, fabric motif determination, and D-CNN	Seven-layer D-CNN with pairwise potential activation layer as a third hidden layer.	Accuracy for predicting the presence of a defect in an image: 95%. Accuracy for counting the number of defects in an image: 98%.	Fabric defect dataset created using an on-loom fabric imaging system.	Seven-layer CNN, which includes pairwise potential activation layer
3	Polarimetric synthetic aperture radar image classification [55]	Polarimetric synthetic aperture radar image classifies image pixels of terrain types, namely, forest, water, grass, and sand	Complex-valued D-CNN and real-valued D-CNN.	Complex-valued D-CNN: 93.4%; real-valued D-CNN: 89.9%.	The performance is analyzed with the airborne steered array radar dataset and the electronically steered array radar dataset.	Six-layer CNN
4	Visual aesthetic quality assessment [56]	To present a biological model for three tasks: aesthetic score regression, aesthetic quality classification, and aesthetic score distribution prediction	A double-subnet gated peripheral-foveal convolutional neural network: a foveal and a peripheral subnet. The peripheral subnet mimics peripheral vision, while foveal extracts fine-grained features.	Gated peripheral-foveal convolutional neural networks (VGG16): 80.70%.	Standard aesthetic visual assessment datasets and photo. Net datasets are used for unified aesthetic prediction tasks.	Nine-layer CNN
5	3D object recognition [57]	To take multiview images captured from partial angles as the input and perform 3D object detection using the 3D CNN	3D object information is encoded from the 3D spatial dimension. 3D kernel, the view images are applied to perform 3D convolution.	ModelNet10 dataset: 94.5%. ModelNet40 dataset: 93.9%.	Pearl image dataset with 10,500 images split into seven classes.	Eight layers
6	Medical image classification [58]	To develop a feature extractor using a fine-tuned D-CNN and to classify medical images	Fine-tuned AlexNet and GoogLeNet D-CNN architectures were used in two ways: (i) as an image feature extractor and to train multiclass support vector machines; (ii) as a classifier to generate softmax probabilities.	GoogLeNet: 81.03%. AlexNet: 77.55%.	Image CLEF 2016 medical image public dataset with 6776 training images and 4166 testing images.	Eight layers with five convolutional layers followed by three fully connected layers and max pooling layers

TABLE 3: Continued.

S. no	Application	The objective of the study	Methodology/ network architecture	Performance	Dataset	Depth/layer sizes
7	Detection and recognition of dumpsters [59]	Visual detection of dumpsters using a twofold methodology with minimal labeling of the dataset to have census of their type and numbers	Google Inception v3 is used, and a D-CNN is pretrained with 1,500,000 images corresponding to 1000 different classes. ReLU is used as an activation function.	94%.	Dumpsters dataset with 27,624 labeled images provided by Ecoembes.	27-layer deep CNN
8	Classification of rice grain images [60]	Localization and classification of rice grain images using contrast-limited adaptive histogram equalization technique	Region-based D-CNN is used to localize and classify rice grains. Dropout regularization and transfer learning were used to avoid overfitting.	81% accuracy is achieved as against 50–76% accuracy of human experts.	MIMR1 to MIMR8 datasets to classify rice into sticky and paddy rice.	Residual neural network-50 is used as the prominent architecture, which is 50 layers deep
9	Object recognition [61]	To evaluate the performance of the inception, recurrent residual convolutional neural network model on benchmark datasets, namely, Tiny ImageNet-200, Canadian Institute for Advanced Research-100, CU3D-100, and Canadian Institute for Advanced Research-10	An inception recurrent residual convolutional neural network, a deep CNN model, was introduced. It utilizes the power of the residual network, inception network, and recurrent convolutional neural network.	72.78% accuracy was achieved, which is 4.53% better than a recurrent CNN.	The model's performance is evaluated on different benchmark datasets, namely, Canadian Institute for Advanced Research, Canadian Institute for Advanced Research-100, Tiny ImageNet-200, and CU3D-100.	Five-layer CNN
10	3D object classification [62]	To successfully classify 3D objects for mobile robots irrespective of starting positions of object modeling	A novel 3D object representation using the 3D CNN with a row-wise max pooling layer and cylinder occupancy grid was introduced.	The results showed that the cylindrical occupancy grid performed better than the existing rectangular algorithms with accuracy 91%.	The performance assessment is done on the ModelNet10 dataset and dataset with six classes collected using the mobile robot.	Three-layer CNN
11	Product quality control [63]	To automatically detect defects in products using fast and robust deep CNN	A simplified D-CNN architecture consisting of nested convolutional and pooling layers with the ReLU activation function is used. It has two parts: a classification frame and a detection frame.	An accuracy of 99.8% is achieved on a benchmark dataset provided by the German Association of Pattern Recognition.	Deutsche Arbeitsgemeinschaft fur Mustererkennung e.V German dataset with six classes. Each class consists of 1000 defect-free images and 150 defective images.	11-layer CNN
12	Recognition of Chinese food [64]	To propose an efficient D-CNN architecture for Chinese food recognition	A 5-layer deep CNN architecture performs a pipeline of processing to optimize the entire network through backpropagation.	97.12% accuracy was achieved, which is better than other bag feature methods.	Chinese food image dataset composed of 8734 images under 25 food categories.	Five-layer deep CNN

TABLE 3: Continued.

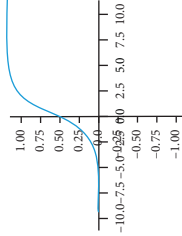
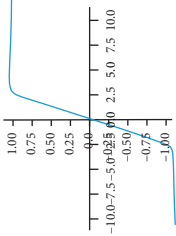
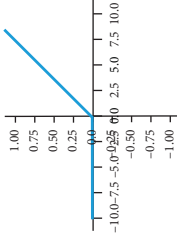
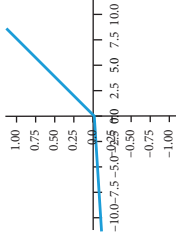
S. no	Application	The objective of the study	Methodology/ network architecture	Performance	Dataset	Depth/layer sizes
13	Age recognition from facial images [65]	Recognition of age from the facial image using pretrained models	MobileNetV2, residual neural network, and pretrained models such as soft stagewise regression networks were used for age recognition. Multiclass classification is performed, which is followed by regression to calculate the age.	Residual neural networks performed better than the other two network models.	The dataset contains 460,723 photographs from the internet movie database cinema website and another dataset with 62,328 pictures from Wikipedia.	Three-layer CNN
14	Person recognition [66]	To develop an effective and efficient multiple-person recognition system for face recognition in random video sequences using the D-CNN	Multiple video faces are detected, and the VGG-19 D-CNN classifier is trained to identify the facial images. The model is tested using standard labeled faces in the wild database.	96.83% accuracy is achieved using the VGG-19 D-CNN classifier.	In the training phase, images from the Chinese Academy of Sciences WebFace database with 9000 classes were utilized. In the validation phase, labeled faces in the wild were used.	Three-layer CNN
15	Multiformat digit recognition [67]	To recognize unconstrained natural images of digits using DIGI-Net	DIGI-Net D-CNN architecture is trained and tested on the MNIST, CVL single digit dataset, and the Chars74K dataset digits.	MNIST: 99.11%, Computer Vision Lab single digit dataset: 93.29%, and digits of the Chars74K dataset: 97.60%.	The performance evaluation is done on MNIST, CVL single digit dataset, and the Chars74K dataset digits.	Seven-layer deep CNN
16	Liver image classification [68]	To develop a perpetual hash-based D-CNN to classify liver images to reduce the time taken to classify liver computed tomography images	A fused perceptual hash-based D-CNN, a hybrid model, was designed to identify malignant and benign masses using computer tomography images.	98.2% accuracy was achieved using the fused perceptual hash-based D-CNN.	The dataset is obtained from Elazig Education and Research Hospital Radiology Laboratory.	Seven-layer deep CNN
17	Object detection [49]	Hardware-oriented ultrahigh-speed object detection using the D-CNN	Two-stage high-speed object detection bounding boxes in the first stage and D-CNN-based classification in the second stage.	MNIST dataset: 98.01%. Self-built dataset: 96.5%.	MNIST and self-built dataset.	Five-layer deep CNN

between 20 and 50 percent of the neurons. The graph is shown in Figures 5(a) and 5(b) which show the accuracy and loss curves after two dropouts with a dropout rate of 0.25 and 0.5 have been added. It can be seen that accuracy has improved, and loss decreased after adding dropout.

In many cases, the data available may not be sufficient to train a model. With significantly few training data, the model may not learn patterns from the training data and inhibit the model's capability to generalize unseen data [76]. It is challenging, expensive, and time-consuming to collect the required new data to train the model [77]. Under such circumstances, data augmentation is warranted, which is a powerful technique for mitigating overfitting. It is a powerful and computationally

inexpensive technique of artificially inflating training data size with the data in hand without collecting new data [71, 78]. One or more deformations are applied to the data, while the labels' semantic meaning is preserved during the transformation [79]. With more data provided to the model, it will generalize well on the validation data. Some popularly used data augmentation techniques are rotation, flip, skew, crop, contrast, brightness adjustment, and zoom in/out [80, 81]. Figure 6 shows different augmentations applied to a single image. Here, flip, rescale, zoom, height, and width shift augmentations are applied to a cat image. Training the model with the additional deformed data makes the model generalize better with unseen data. Figure 7 shows an improvement in accuracy after

TABLE 4: Activation functions used in deep learning.

Name	Plot	Equation	Derivative	Characteristics	Advantages	Disadvantages
Sigmoid		$f(x) = \sigma(x) = (1/(1 + e^{-x}))$	$f'(x) = f(x)(1 - f(x))$	(i) It takes real-value inputs and outputs a value in the range (0, 1)	(i) Smooth gradient (ii) Have activations within a specific range	(i) Vanishing gradient problem (ii) Not zero-centered
tanh		$f(x) = \tanh(x) = (\sinh(x)/\cosh(x)) = ((e^x - e^{-x})/(e^x + e^{-x}))$	$f'(x) = 1 - f(x)^2$	(i) It ranges from -1 to 1 (ii) Resembles a sigmoidal shape	(i) Derivatives are steeper when compared to sigmoid	(i) Vanishing gradient problem
ReLU		$f(x) = \begin{cases} 0, & \text{for } x < 0, \\ x, & \text{for } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0, & \text{for } x < 0, \\ x, & \text{for } x \geq 0. \end{cases}$	(i) ReLU is half rectified, ranging from 0 to infinity	(i) Biological plausibility (ii) Better gradient propagation	(i) Negative values become zero
Leaky ReLU		$f(x) = \begin{cases} 0.01x, & \text{for } x < 0, \\ x, & \text{for } x \geq 0, \end{cases}$	$f'(x) = \begin{cases} 0.01x, & \text{for } x < 0, \\ x, & \text{for } x \geq 0, \end{cases}$	(i) Allows a small nonzero value for negative x	(i) Solves dying ReLU problems	(i) Cannot be used for complex classification problems

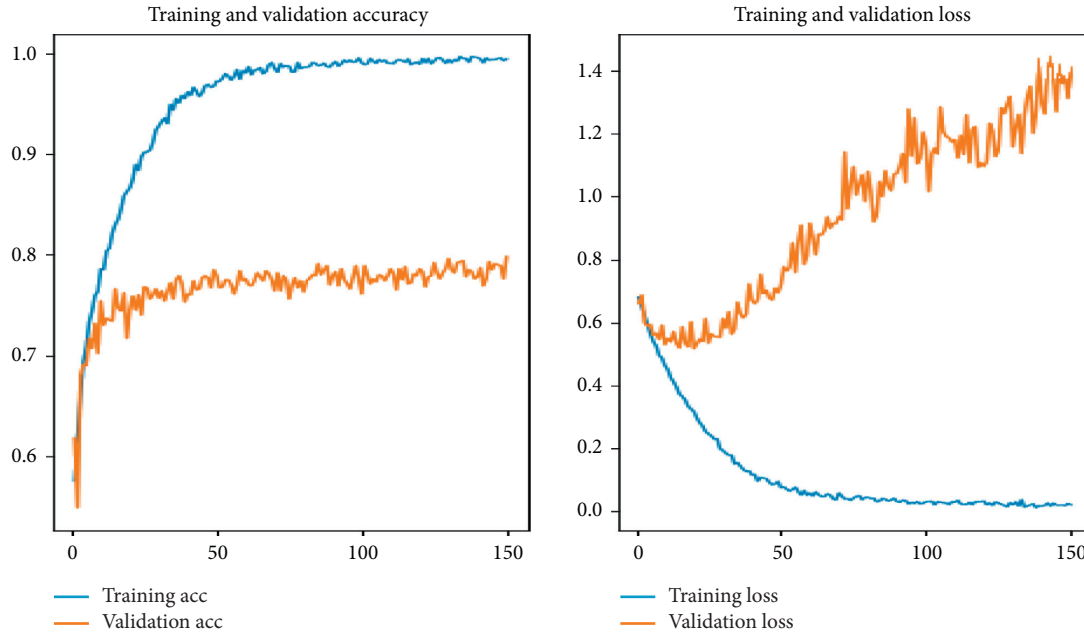


FIGURE 5: Overfitting. (a) Accuracy curve after dropout. (b) Loss curve after dropout.

adding dropout and applying different augmentations to the data.

The next problem usually faced by AI practitioners is underfitting. The model is said to underfit when it cannot learn the patterns even from the training set and exhibits poor performance on the training set. Moreover, this can be addressed by increasing the training data, improving the model complexity, and increasing the training epochs. Deeper models with more neurons per layer can avoid underfitting [82]. Imbalanced datasets are another crucial problem faced as they widely exist in real-world situations and have proven to be the greatest challenge in classification problems. Data access has become comfortable with the advancement of technology; however, data imbalance has become ubiquitous in most of the collected datasets. For example, in medical data, most people are healthy, and unhealthy people are less in proportion than healthy people, significantly affecting classification accuracy. Here, the classes with adequate samples are called the majority class, and classes with inadequate samples are called the minority class. Prediction of minority class becomes problematic as it has a fewer number of samples or insufficient samples.

Several techniques are adopted to handle data imbalance in the dataset. Some of them are weight balancing, over- and undersampling (resampling), and penalizing algorithms [83]. Weight balancing is performed by modifying the weights carried by the training samples when computing the loss. Resampling is one of the frequently adopted techniques where undersampling is done to remove samples from the majority class or oversampling is done to add more samples to the minority class. Oversampling is done by duplicating random records from the minority class. Penalizing algorithm is another technique where the cost of classification mistakes is increased on the minority

class. Label noise is another problem in deep learning, and some of its sources are nonexpert labeling, automatic labeling, and data poisoning, adversaries. Dan Hendrycks et al. [84] recommended a loss correction technique to utilize trusted data with clean labels. The authors effectively used trusted data to overcome the effects of label noise on classification.

**3.4. Deep Generative Adversarial Network.** Generative adversarial networks are a framework proposed by Goodfellow et al. in the year 2014 [7]. Significant improvements were achieved in computer vision applications such as image super-resolution [85], image classification [86], image steganography [87], image transformation [88], video generation, image synthesis [89], video super-resolution [90], and image style transformation. Variants of the D-GAN model were also proposed in recent times.

Figure 8 shows the D-GAN architecture where the generator generates fake images of human faces, and the discriminator's job is to distinguish the real faces from the fake faces. In general, plausible data are generated by the generator. These data generated by the generator become negative examples of training the discriminator. The discriminator, a binary classification neural network, takes in the real samples and the samples from the generator and learns to distinguish the real samples from the generator's fake samples. Two loss functions, generator loss and discriminator loss, are backpropagated to the generator and discriminator. The discriminator ignores the generator loss. The generator and the discriminator update the weights based on the loss, where the noise samples  $i$  ranging from 1 to  $m$  are represented by



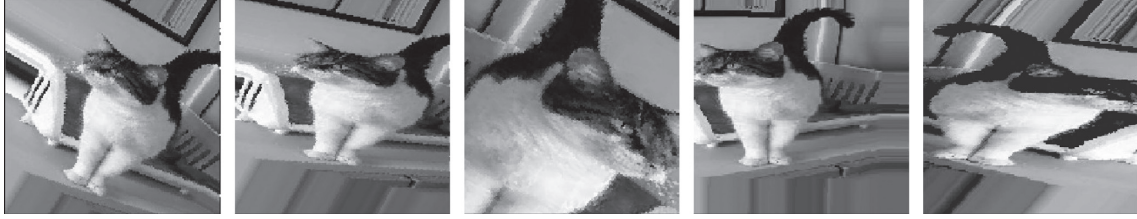


FIGURE 6: Data augmentation of a single image. (a) Original image. (b) Rescale. (c) Height shift augmentation. (e) Width shift augmentation.

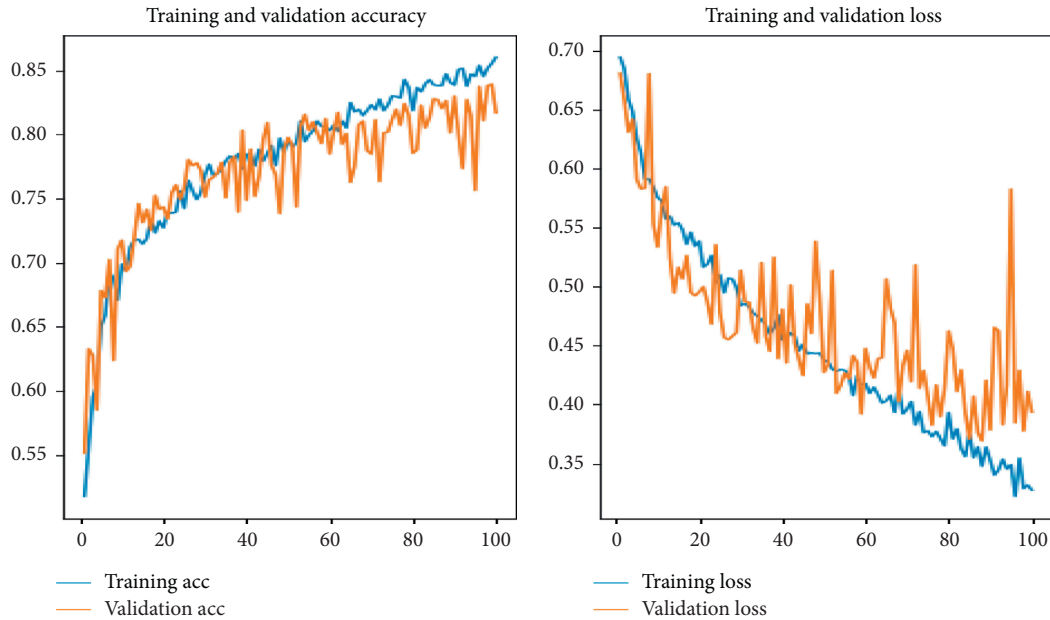


FIGURE 7: Accuracy and loss after adding dropout and applying augmentations.

$$D_{\text{loss}} = \frac{1}{m} \sum_{i=1}^m [\log(D(x^i)) + \log(1 - D(G(z^i)))], \quad (3)$$

$$G_{\text{loss}} = \frac{1}{m} \sum_{i=1}^m -\log(D(G(z^i))).$$

**3.5. Evolution of the Deep GAN.** With deep GAN's advent by Goodfellow, several variants of the deep GAN were proposed for various CV applications. These deep GAN variants have their own architecture, methodology, advantages, and disadvantages but with the same two-player minimax game theory as the base. Figure 9 shows D-GAN's evolution with conditions, encoders, loss functions, and process discrete data. Table 5 shows D-GAN's evolution with its application, architecture, methodology, advantage, and disadvantage.

D-GAN is successfully used in many computer vision applications, and image generation is at the forefront of all these. D-GAN generates images, gradually enhancing the resolution and the quality of images generated. Variants of D-GAN are used for various applications such as image transformation, image deraining [88], increasing image resolution, facial attribute transformation, and fusion of the image. Table 6 shows some of the progressively increasing applications of the D-GAN in computer vision.

## 4. Applications of the D-CNN in Computer Vision

Most of the D-CNN applications are related to images, while applications of the D-GAN are related to data generation. This section will progress through the essential applications of the D-CNN.

**4.1. Image Classification Using the D-CNN.** There are several image classification tasks performed using D-CNN [11, 104–109]. One of the vital image classification tasks is handwritten digit recognition which recognizes numbers between 0 and 9, where the data from the MNIST database are obtained to predict the correct label for the handwritten digits. MNIST is a database of handwritten numbers widely used as a testbed for various deep learning applications. It has 70,000 images, of which 60,000 are training images and 10,000 are testing images [110]. Figure 10 shows sample images from the MNIST dataset. The images are greyscaled with  $28 \times 28$  pixels, as represented in Figure 10. The  $28 \times 28$  pixels are flattened into a 1D vector of size 784 pixels, and each of these pixels has values between 0 and 255. The black pixel takes the value 255, while the white pixel takes the value one, and various other shades of grey take values between 0 and

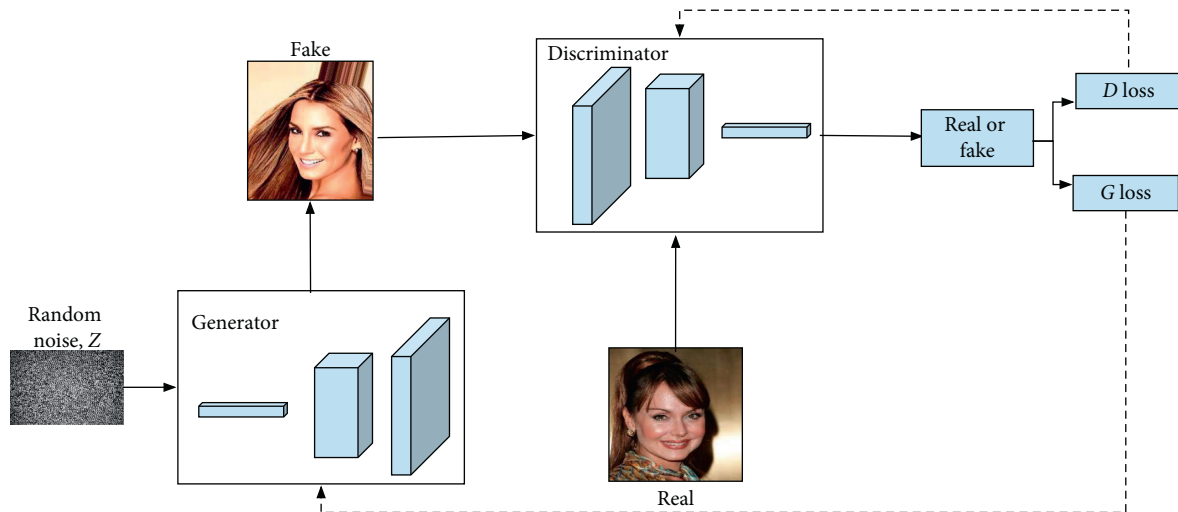


FIGURE 8: Deep generative adversarial network—human face generation.

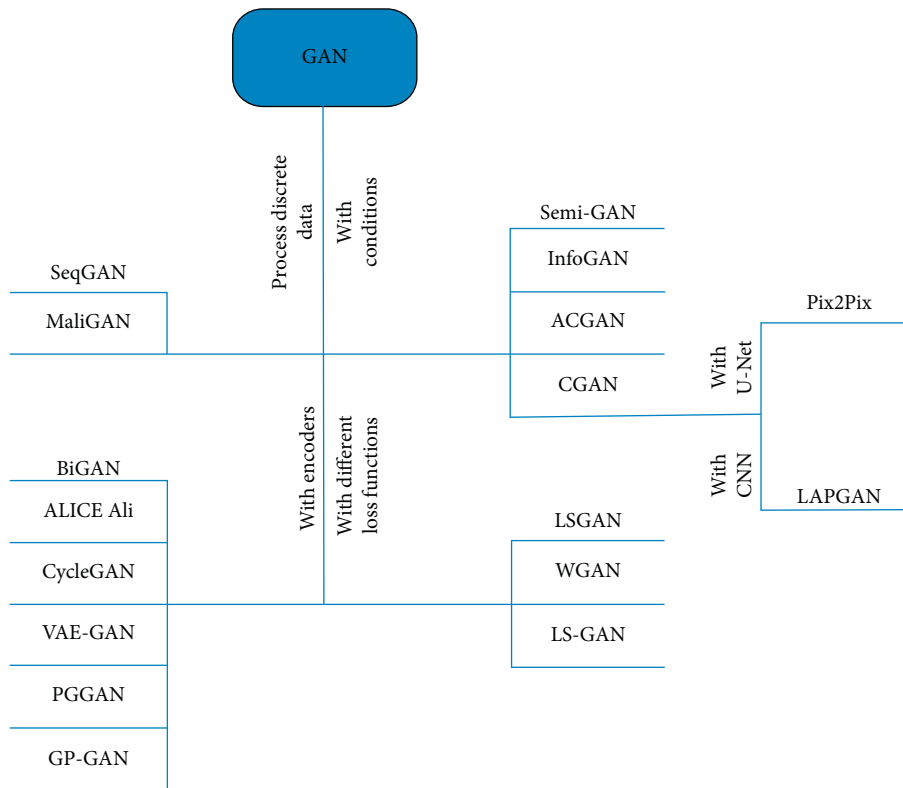


FIGURE 9: Evolution of the deep GAN with conditions, encoders, loss functions, and process discrete data separately.

255. Handwritten digits are recognized using the D-CNN, which is considered the most suitable model for performing this task.

Data are downloaded from the MNIST database, and it takes some time to download the data during the first run, and the subsequent runs fetch the cached data. The data obtained from the MNIST database have features and labels. The features range from 0 to 225, corresponding to pixels of  $28 \times 28$  images representing digits 0 through 9. The labels represent digits 0–9 of the respective image. It is normalized

to scale the data to be between 0 and 1. The actual image from the MNIST dataset and the normalized image are represented in Figures 11(a) and 11(b).

Handwritten digit recognition is implemented using eight hidden layers, where the first layer is the convolutional layer used for feature extraction. ReLU is used as the activation function with 32 filters and a kernel size of  $3 \times 3$  pixels. Another convolutional layer is used with ReLU as the activation function, 64 filters, and kernel size of  $3 \times 3$  pixels. The next hidden layer is a pooling layer where max pooling

TABLE 5: Evolution of the deep GAN.

S. no	Model	Application	Network architecture	Methodology	Key advantage	Major limitation
1	SRGAN [9]	Image super-resolution	Generator: two convolutional layers with small $3 \times 3$ kernels and 64 feature maps, batch normalization layers, and parametric ReLU. Discriminator: leaky ReLU activation function.	A perceptual loss function is proposed, which is composed of content loss and adversarial loss	Low-resolution images are converted into a high-resolution image for 4x upscaling factors	Texture information is not real enough, accompanied by some noise
2	ACGAN [15]	Image synthesis	Generator: has a series of deconvolutional layers, also known as transposed convolutional layers. Discriminator: has a set of 2D convolutional layers with leaky ReLU followed by linear layers and softmax and sigmoid functions for each of its outputs.	Two variants of the model were trained to generate images of resolution $64 \times 64$ and $128 \times 128$ spatial resolutions	Accuracy can be assessed for individual classes	Ignores the loss component arising from class labels when a label is unavailable for a given training image
3	CGAN [16]	Image-image translation, image tagging, and face generation (Gauthier, J. (2014))	Generator and discriminator are conditioned on some arbitrary external information with the ReLU activation function and sigmoid for the output layer.	Minimize the value function for $G$ and maximize the value function for $D$	CGAN could easily accept a multimodal embedding as the conditional input	CGAN is not strictly unsupervised; some kind of labeling is required for them to work
4	InfoGAN [1, 2]	Facial image generation	Generator: upconvolutional architecture and normal ReLU activation function. Discriminator: leaky ReLU with leaky rate 0.1 applied to hidden layers as nonlinearity.	Learn disentangled representations by maximizing mutual information	InfoGAN is capable of learning disentangled and interpretable representation	Sometimes, it requires adding noise to the data to stabilize the network
5	DCGAN [91]	General image representations	Generator: the ReLU activation except for the output layer, which uses the tanh function and batch norm. Discriminator: leaky ReLU for high-resolution modeling and batch norm.	The hierarchy of representations is learned from object parts	Stable, good representations of images, easy convergence	Gradients disappear or explode
6	LAPGAN [10]	Generation of images	Laplacian pyramid of convolutional networks.	Image generation in a coarse-to-fine fashion	Independent training of each pyramid level	Nonconvergence and mode collapse
7	SAGAN [11]	Generation of images	Spectral normalization is applied to the GAN generator.	Realistic images are generated	Inception score is boosted, Fréchet inception distance is reduced, and images are generated sequentially	Attention is not extended

TABLE 5: Continued.

S. no	Model	Application	Network architecture	Methodology	Key advantage	Major limitation
8	GRAN [12]	Generating realistic images	Recurrent CNN with constraints.	Images are generated by incremental updates to the canvas using a recurrent network	Sequential generation of images	Samples collapse on training for a long duration
9	VAE-GAN [92]	Facial image generation	GAN discriminator is used in place of a VAE's decoder to learn the loss function.	Combines GAN and VAE to produce an encoder of data into the latent space	Its advantage, GAN, and VAE are combined in a single model	The major drawback of the VAE is the blurry output it generates
10	BIGAN [18]	Image generation	Generator: in addition to $G$ , it has an encoder to map the data to latent representations. Discriminator: discriminates both data and latent spaces.	Learn features for related semantic tasks and use them in unsupervised settings	Minimization of the reconstruction loss	Generator and discriminator are highly dependent on each other in the training phase
11	AAE [19]	Dimensionality reduction, data visualization, disentangling the style of the image, and unsupervised clustering	Generative autoencoder.	Variational inference is performed by matching the arbitrary prior distribution with the autoencoder's aggregated posterior of the hidden code vector	Balanced approximation; this method can be extended to semisupervised learning and is better than variational autoencoders	Samples generated are blurry and smoothed (Zhang, J. et al., 2018)
12	Pix2Pix [93]	Image-image translation	Generator: UNET architecture. Discriminator: PatchGAN classifier. ReLU activation function. Batch normalization.	The network learns the loss associated with the data and the task, making it applicable for a wide variety of tasks	Parameter reduction, and realistic images are generated	The required images are to be one-one paired

with  $2 \times 2$  pixels as pool size is used. Next to the pooling layer is the dropout, a technique adapted to prevent overfitting in the neural network [111]. The dropout technique's key idea is to randomly drop a few units from the network and its connections during the training to reduce overfitting significantly. A dropout rate of 0.25 randomly drops out 1 in 4 units from the network. Between the convolutional layer and the fully connected output layer is the flatten layer. The flatten layer aims to transform the 2D matrix into a 1D vector fed into the fully connected layer. The flattened 1D vector is then passed on to the fully connected layer with ReLU as the activation function. Another dropout with a dropout rate of 0.50 is used. Finally, the output layer with the softmax activation function is used. The softmax activation function is used as its role is to specify the probability distribution for ten different classes. Since the task in hand is a multiclass classification, the output layer has ten nodes or perceptron corresponding to each of the ten categories to predict each class's probability distribution. The perceptron with the highest probability is picked, and the label associated with it is returned as the output. The model is fit over 12 epochs. The test accuracy achieved is 98.56%, and the test loss is 0.0513, as represented in Figures 12(a) and 12(b). It can be seen that the accuracy increases with the increase in epoch, and loss decreases with the increase in accuracy.

Image classification is a classical problem of computer vision and deep learning. It is challenging because of the image's variations due to light effects and misalignments [112]. Image classification in a computer sense is a course of action for grouping and categorizing images and labeling them based on their features and attributes [80]. It trains computers to use a well-defined dataset to interpret and classify images to narrow the gap between human vision and computer vision [113]. Some of the existing use cases of image classification are gender classification, social media applications such as Facebook and Snapchat which use image classification to enhance the user experience, and self-driving cars where various objects on their path, namely, vehicle, people, and other moving objects, are recognized [114].

Amerini et al. [115] proposed a novel framework called FusionNet by combining two D-CNN architectures to identify the source social network based on the images. 1D-CNN learns discriminative features, while 2D-CNN architecture infers unique attributes from the image. The learned features are fused using FusionNet, and then the classification is performed. Distinctive traces of social networks embedded in the images are exploited to identify the source. The full-frame images are broken into fixed dimension patches, and the patches are then classified independently. Each of the image patches is processed with D-CNN, and the predictions

TABLE 6: Recent advancements of the deep GAN in computer vision.

S. no	Application	The objective of the study	Methodology/network architecture	Performance	Dataset
1	Facial attribute transformation [14]	To develop a novel conditional recycle D-GAN that can transform high-level face attributes retaining the face's identity	The developed conditional recycle D-GAN model has two phases. In the first phase, conditional D-GAN attempts to generate fake facial images with a condition. In the second phase, recycling D-GAN is used to generate facial images to modify the attributes without changing the identity.	The results were compared with existing D-GAN architectures to prove the efficiency of CRGAN. CRGAN performed better than existing D-GAN architectures.	CelebA dataset
2	Fusion of images [94]	To propose a method to fuse images belonging to different spectra using D-GAN	FusionNet architecture was developed using the Pix2Pix architecture to generate fused images from different spectra fragments of images.	The proposed FusionNet model was compared with existing fusion methods such as the cross bilateral filter, the weighted least squares, and the sparse joint representation. The FusionNet technique performed equally well with the existing methods.	Dataset provided by experts
3	Synthesis of high-quality faces [95]	To propose a D-GAN-based method to synthesize high-quality images from polarimetric images	The proposed model has a generator subnetwork built based on an encoder-decoder network and a discriminator subnetwork. The generator is trained by optimizing identity loss, perceptual loss, and identity preserving loss.	The qualitative and quantitative performance of the developed model is compared with state-of-the-art methods. The use of perceptual loss generated visually pleasing results.	A dataset with polarimetric and visible facial signatures from 111 subjects
4	Vehicle detection in aerial images [96]	To develop a lightweight deep CNN model to detect vehicles in aerial images using D-GAN effectively	The architecture has two parts: lightweight deep CNN was developed to accurately detect vehicles and a multicondition-constrained GAN to generate samples to cope with data insufficiency.	The model tested on the Munich dataset achieved a mean average precision of 86.9%.	Performance evaluation is done on Munich public dataset and the collected dataset.
5	Image deraining [97]	To develop a deep learning model to remove rain streak from images	A feature supervised D-GAN was developed to remove rain from a single image. Feature supervised D-GAN has two subnets to generate derained images that are very close to the real image.	The developed model was tested on synthetic and real-world images. It showed better performance than the existing state-of-the-art deraining methods.	Performance evaluated on real-world images and two synthetic datasets.
6	Scene generation [12]	To develop a model to generate scenes based on the conditional D-GAN	A model named PSGAN was developed to generate a multidomain particular scene. The quality of the images is improved by spectral normalization.	The developed model is compared with Pix2Pix and StarGAN. 97% accuracy is achieved using PSGAN as against 95% accuracy achieved using StarGAN.	The performance of the model is evaluated on MNIST, CIFAR-10, and LSUN.

TABLE 6: Continued.

S. no	Application	The objective of the study	Methodology/network architecture	Performance	Dataset
7	Human pose estimation [98]	To develop self-attention D-GAN to perform human pose estimation	The D-GAN model used hourglass networks as its backbone. Hourglass architecture has Conv-Deconv architecture and residual blocks. The generator predicts the pose, while the discriminator enforces structural constraints to refine the postures.	The model outperformed the state-of-the-art methods on benchmark datasets.	The performance of the model is evaluated on Leeds Sports Pose and MPII human pose dataset.
8	Automatic pearl classification [57]	To develop deep learning models to perform automatic classification of pearls	Multiview GAN is used to expand the pearl images dataset. Multistream CNN is trained using the expanded dataset.	The image generated using the multiview GAN is used to reduce the existing multistream CNN significantly.	The dataset includes 10,500 pearls, with seven categories and each category containing 1,500 pearls.
9	Image dehazing [99].	To develop a deep learning model to recover the image's texture information and enhance hazy scenes' visual performance	Attention-to-attention generative network model is developed to map hazy images to haze-free images. All the instance normalization layers are removed to generate high-quality images.	The developed model performed better state-of-the-art methods for both real-world and synthetic images	NYU2 synthetic dataset with 1300 images and SUN3D synthetic dataset with 150 images.
10	Gesture recognition [100]	To propose a new gesture recognition algorithm based on D-CNN and DCGAN	For a particular gesture, the model recognizes the meaning of the gesture. DCGAN is used to solve overfitting in case of data insufficiency. Preprocessing is done to improve illumination conditions.	An accuracy of 90.45% is achieved.	Data collected using a computer containing 1200 images for each gesture.
11	Face depth estimation [101]	To develop a D-GAN-based method to estimate the depth map for a given facial image	D-GAN architecture is used to estimate the depth of a 2D image for 3D reconstruction. Data augmentation is done to improve the robustness of the models. Transformations such as slight rotation clockwise, Gaussian blur, and histogram equalization were applied to the image.	Several variants of the D-GAN were evaluated for depth estimation. Wasserstein GAN was found to be the most robust model for depth estimation.	The Texas 3D face recognition database and Bosphorus database for 3D face analysis.
12	Image enhancement [102]	To propose an image enhancement model using the conditional D-GAN based on the nonsaturating game	The super-resolution method is combined with the D-GAN to generate a clearer image. The architecture has 23 layers composed of convolution layers with the ReLU activation function.	The model is compared with existing methods, which showed an improvement in peak signal-to-noise ratio by 2.38 dB.	Images from Flickr and ImageNet datasets were used without augmentation.
13	Retinal image synthesis [103]	To propose multiple-channels-multiple-landmarks, a preprocessing pipeline to synthesize retinal images from optic cup images	Residual neural network and U-Net were integrated to form residual U-Net architecture. Residual U-Net is capable of capturing finer-scale details. Multiple-landmark maps comprise of batch normal layer, convolution layer, and ReLU activation. The final layer has a sigmoid activation function.	The proposed multiple-channels-multiple-landmarks model outperformed the existing single vessel-based methods. Pix2Pix, using the proposed method, generated realistic images.	Public fundus image datasets DRIVE and DRISHTI-GS were used.

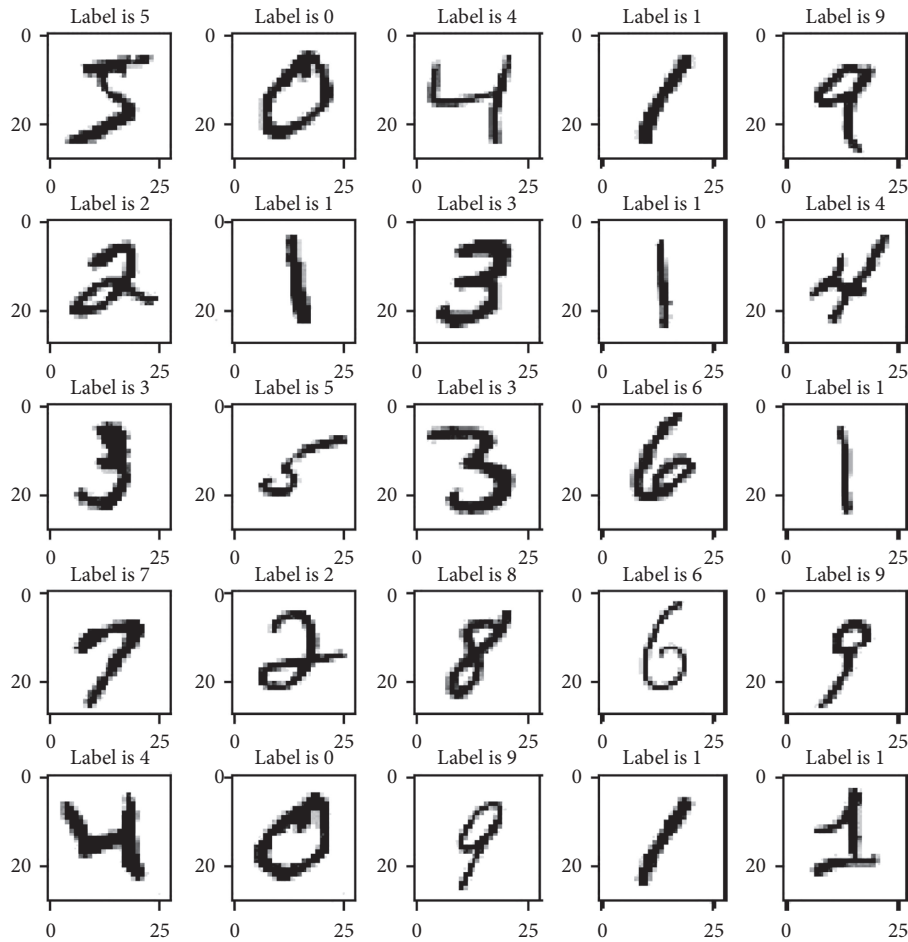


FIGURE 10: Sample images from the MNIST dataset.

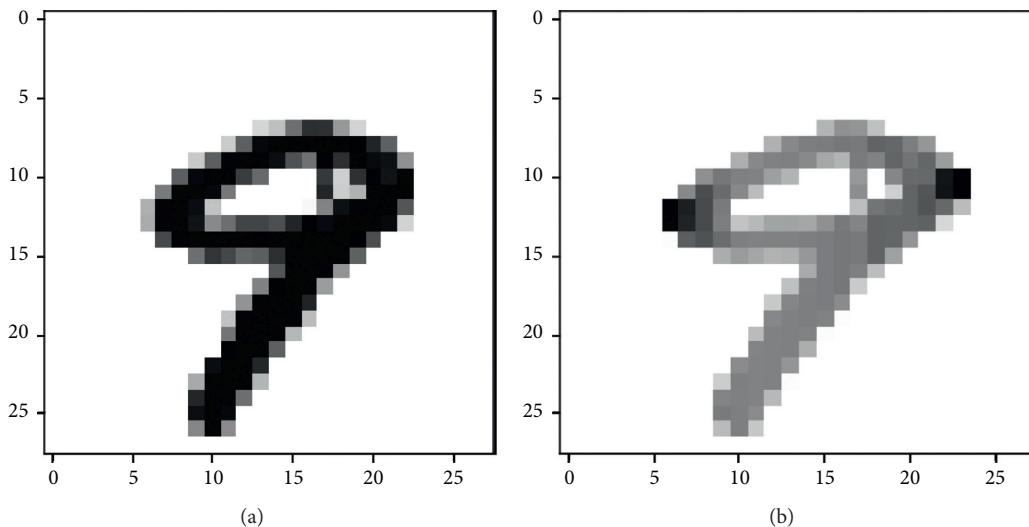


FIGURE 11: (a) Image from the MNIST dataset. (b) Normalized image.

are obtained. Furthermore, to get the prediction at the image level, a voting strategy is applied at each patch. The label with the majority vote is assigned as the final prediction label. The average accuracy of 94.77% is achieved at the patch level.

*4.2. Image Localization and Detection Using the D-CNN.* On a glance over the object, human vision is capable of detecting the object, its size, location, and various other features. Object detection using deep learning allows

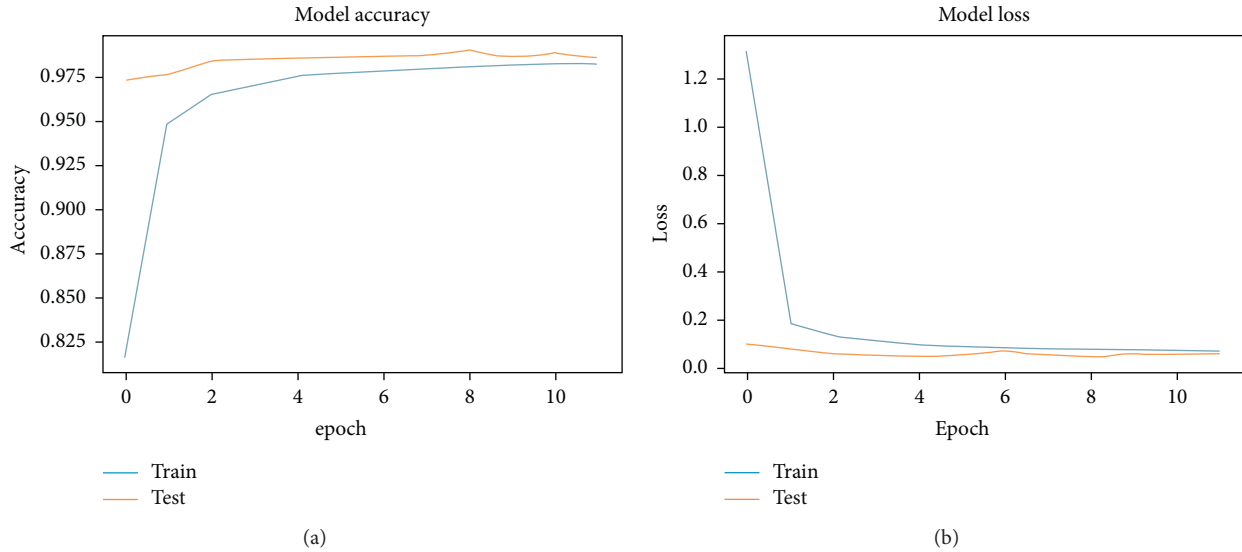


FIGURE 12: Accuracy and loss curves. (a) Accuracy vs. epoch. (b) Loss vs. epoch.

computers to play a crucial role in many real-world applications such as robots, smart vehicles, and self-driving cars [116]. Object detection is one of the most challenging problems and the most important goal of computer vision. Object detection involves identifying different objects in an image using a bounded box. The identified objects can be further analyzed at a granular level to digging deeper into the image. Earlier object identification was performed by splitting the image into multiple pieces and then passing them into a classifier for object detection. Splitting the images into multiple pieces is performed using a sliding window algorithm. In this approach, the detection window is slid through the actual image at multiple positions, and each grid is a smaller piece of the image. Robust visual descriptors needed for object detection are extracted from the image using image processing. The convolutional neural network used visual descriptors to make object or nonobject decisions [117]. Since the process had to be repeated multiple times, it was computationally expensive.

Moreover, to overcome the sliding window algorithm's shortcomings, object detection was performed using image segmentation. Segmentation is categorized into boundary-based, thresholding, region-based, and boundary-based. When a digital image is passed, the neurons are synchronized based on pixels with similar intensities to form a connected region [118]. It can be contextual if spatial relationships in an image are considered or noncontextual if spatial relationships are not considered. The goal of image segmentation is to alter the representation of an image into a form that is meaningful and easier for analysis. The accuracy of object detection is based on the quality of image segmentation. Similar to image classification problems, networks that are deeper exhibited better performance in object detection. Object localization is the next level of object classification where the objects' position was also determined with a bounding box and labeling the objects. The difference between localization and detection is that

classification with localization handles only one object, whereas detection finds multiple objects in an image and labels.

Tu et al. [119] proposed a method to detect passion fruits based on multiple-scale faster region-based CNN. The detection phase involves multiple-scale feature extractors that extract low- and high-level features. Data augmentation is done to enlarge the training data size. Pretrained residual neural network-101 architecture is used for object detection.

*4.3. Document Analysis Using the D-CNN.* Documents are the source of information for several cognitive processes, namely, graphic understanding, document retrieval, and OCR. Document analysis plays a crucial role in cognitive computing to extract information from document images. Document analysis is performed by identifying and categorizing images based on regions of interest. There are several existing methods for document analysis, such as pixel-based classification methods, region-based classification method, and connected component classification method. The region-based classification method segments document images into zones and classifies them into semantic classes. Pixel-based classification methods perform document analysis by taking each pixel and generate labeled images using the classifier. The connected component method creates the object hypothesis using local information, further inspected, refined, and classified [120].

D-CNN is widely adopted for document analysis to reduce computational complexity, cost, and data without compromising accuracy. With D-CNN, it is possible to classify images directly from segmented objects without extracting handcrafted features. Maryem Rhanoui et al. [121] performed document-level sentiment analysis using a combination of D-CNN and bidirectional long short-term memory and achieved an accuracy of 90.66%. The features are extracted by the D-CNN, and the extracted features are



passed as the input to long short-term memory. Vectors' built-in word embedding is passed as the input to the CNN. Four filters are applied, and the layer of max pooling is applied after each filter. The results of max pooling were concatenated and passed as the input to binary long short-term memory. The output of binary long short-term memory is passed as the input to a fully connected layer. The fully connected layer connects each piece of information from the input with output information. Finally, the softmax function is applied as an activation function to produce the desired output by assigning classes to articles.

*4.4. Speech Recognition Using the D-CNN.* Human-machine interaction for intelligent devices, namely, domestic robots, smartphones, and autonomous cars, is becoming increasingly common in daily life. Hence, noise robust automatic speech recognition has become very crucial for the human-machine interface. The basic idea behind speech recognition is to utilize the speaker's lip movement's visual information to complement the corrupted audio speech inputs. Automatic speech recognition models the relationship between phones and acoustic speech signals by extracting features and classifying speech signals. Furthermore, this is usually performed in two steps, where in the first step, the raw speech signal is transformed into features using dimensionality reduction and information selection. The second step estimates phonemes using generative or discriminative models. Phoneme class conditional probabilities can be estimated using the D-CNN through the raw speech signal as the input. The features are learned from the raw speech signal in both continuous speech recognition and phoneme recognition tasks.

Kuniaki Noda et al. [122] proposed a CNN-based approach for audiovisual speech recognition. Here, the authors first used a denoising autoencoder to acquire noise features. Then, the authors used the CNN to extract features from mouth area images. The training data for the CNN were raw images and their corresponding phoneme outputs. Lastly, the authors applied the multistream hidden Markov model to integrate audio and visual hidden Markov models trained with corresponding features. The model achieved a 65% word recognition rate with denoised mel-frequency cepstral coefficients with the signal-to-noise ratio under 10 dB for the audio signal input.

## 5. Applications of the D-GAN in Computer Vision

*5.1. Image-to-Image Translation Using the Deep GAN.* Remarkable progress has been achieved in image-image translation with the advent of the deep GAN. The image-image translation aims to learn the mapping to translate the image within two different domains, from the source to a target domain, without losing the original image's identity and reducing the reconstruction loss. Some of the essential image-image translations are converting the real-world images into cartoon images, coloring the greyscale images, and changing a nighttime picture to a daylight picture.

D-GAN's role is to confuse the discriminator by generating images that are close to the real images. D-GAN is incredibly successful in super-resolution, representation learning [123], image generation [124, 125], and image-image translation.

Kim et al. proposed a novel method for image-image translation by incorporating a learnable normalization function and a new attention module. Existing attention-based models lacked behind in handling the geometric changes. This model is incredibly successful in translating images with massive shape changes. The auxiliary classifier is used to obtain an attention map to distinguish between source and target domains. Furthermore, this is done to focus on the region of interest, ignoring other minor regions. Attention maps are inserted both in the generator and discriminator to focus on the region of interest. The attention map embedded in the generator focuses on the essential areas that distinguish the two domains. In contrast, the attention map embedded in the discriminator focuses on distinguishing the target domain's real and fake image. The choice of the normalization mechanism dramatically improves the quality of the transformed images. Adaptive Layer-Instance Normalization is used to select a ratio between layer normalization and instance normalization adaptively, and the parameters are learned during the training process. The class activation map gives discriminative image regions to determine the class. The model's performance is superior to the existing state-of-the-art methods on both style transfer and object transfiguration.

*5.2. Image Denoising.* Image denoising removes noise from images retaining the detailing of the images. Image denoising is significantly improved with the advancement in the D-CNN [126]. However, D-CNN models focus mainly on reducing the mean squared error resulting in images lacking high-frequency details. Furthermore, to overcome this issue, D-GAN is applied to remove noise from images [127, 128]. Zhong et al. proposed a method to remove noise from images using the D-GAN. The architecture of the generator in the D-GAN has a convolutional block and eight dense blocks. Each block comprises a convolutional layer, batch normalization, and ReLU activation. Each layer, except the last layer in the network, is fed with each of the previous layers using skip connections. This method effectively reduces the vanishing gradient problem. The convolutional layer extracts low-level features, while the dense blocks extract the high-level features. The generator network is capable of learning the residual difference between the ground truth and the noisy image. The final  $3 \times 3$  convolutional layer generates the output images. The discriminator network differentiates the fake and the ground truth image, making the final denoised visually appealing image. The model can handle different types of noise, but it cannot handle unknown real noises.

*5.3. Face Aging and Facial Attribute Editing.* Deep GAN-based methods have been proposed to alter facial attributes to anticipate a person's future look. Conditional GAN has been widely adopted to perform face aging [129]. D-GANs

are also incorporated in facial attribute editing, manipulating facial images' attributes to generate face with the desired attribute, retaining other facial images' details. The latent representation of the facial images is decoded to edit the facial attributes. GAN-based methods have been proposed for facial attribute editing, which changes only the desired attributes and preserves the other identities of the facial images, retaining the facial image's identity [130]. The work uses reconstruction learning to preserve the attribute details and "*only change what you want.*" The authors applied attribute classification constraints to the generated image rather than imposing constraints on the latent representation to warrant the desired attributes' correct change. The facial attributes are manipulated to change the facial image with and without a beard, black hair and brown hair, mouth open and mouth close, brown hair and blond hair, and young and old. Yujun Shen et al. [131] interpreted the latent codes of trained models such as StyleGAN and progressive GAN and encoded various semantics in the interpreted latent space. Given a synthesized face, different face attributes such as pose, age, and expression are edited without having to retrain the D-GAN model. Table 7 shows the comparison of handwritten digits generated by D-GAN variants.

## 6. Open Problems and Future Opportunities for Computational Visual Perception-Driven Image Analysis

This paper discussed the development of computational visual perception with D-GAN and D-CNN. The advantages and disadvantages of various architectures of D-GAN models are discussed. Future research with D-GAN can be performed on model collapse, nonconvergence, and training difficulties. Also, various other shortcomings of the CNN and their solution are reviewed. Table 8 lists the challenges and open problems for computational visual perception-driven image analysis.

- (i) Handwriting recognition mainly relies on the language model that we furnish to the system and the character modeling quality. This research can be performed to obtain a better handwriting recognition system that is faster and more accurate.
- (ii) Semantic mapping is promising in autonomous vehicles, but state-of-the-art methods still need improvement to produce reliable tools. To do this, better 3D geometry must be included in mapping to achieve more accurate results in the semantic segmentation process. Moreover, map updating has to be done to ensure that maps are always coherent with reality.
- (iii) The biggest problem with calibration in webcam-based eye trackers is a variation in the head pose. Furthermore, this has to be handled without modifying the base components of the system. The future works can be directed towards a 3D model-based head tracking, gaze estimation calculated geometrically, and accurate iris segmentation method.
- (iv) Lumen center detection is based on the geometry and appearance of the lumen. Future works can be directed towards lumen segmentation using the center point of the lumen computed previously as the seed and filling tracheal ring discontinuities to improve segmentation accuracy.
- (v) In query-by-string word, spotting the latent semantic model's performance improves when more samples are used to build the model. However, acquiring the transcription of handwritten documents can be tricky so that synthetic information can train the whole framework. Another problem that requires a solution is the vast possible parameter combinations of the bag-of-visual-words model. An adaptive framework can automatically generate spatial representation and codebook based on the model training phase's indexation errors.
- (vi) Coastal lagoons alternate between being open and closed to the ocean, resulting in intermittently closed and open lakes and lagoons. These are features by a berm, originated from sediments and sand deposited by tides, winds, and waves from the ocean. Moreover, this helps prevent ocean water from flow further into the lagoon, but rain can cause the lagoon to overflow. Hence, future research can explore the adoption of existing computer vision technology and techniques to monitor ICOLLs, including obtaining water level measurement and berm height and improving the decisions for when to open/close a lagoon entrance.
- (vii) Presently, scientists are working towards the development of a vaccine to control coronavirus disease. The clinicians and scientists are making all the attempts to prevent, treat, and control the spread of coronavirus. However, there is an urgent need to perform research into the use of SARS-Cov-2 in suitable animal models to examine the replication, transmission, and pathogenesis. The task remains in identifying the source of coronavirus, the immunological basis of the virus, the immune responses, and whether the mutation-prone positive-sense ssRNA virus of coronavirus will become endemic or alters into forms that are more lethal shortly.
- (viii) The medical imaging community utilizes transfer learning where large-scale annotated data are required. Here, the transfer learning model's performance can be improved by carefully selecting the source and target domain. Residual learning can make significant advancements if effectively utilized in medical image analysis tasks.
- (ix) With the potential use of image processing techniques in computer vision, disaster management can be improved. Satellite images of areas that are prone to landslide are taken during a specific

TABLE 7: Comparisons of images generated by D-GAN variants.

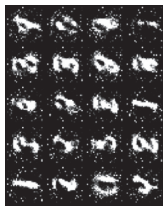
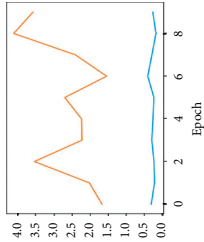

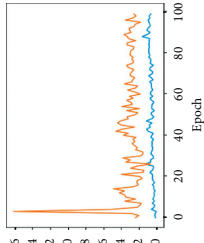

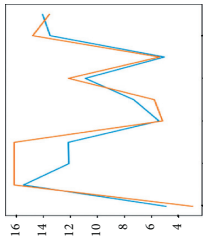
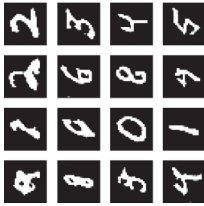
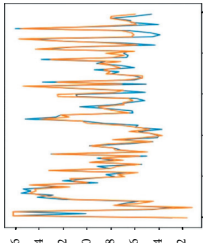
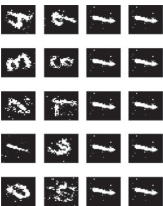
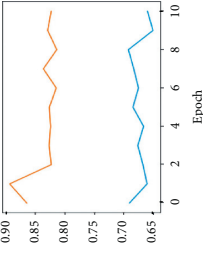
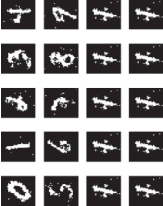
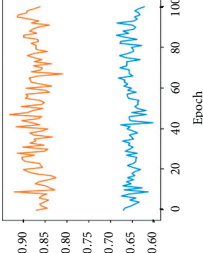
Type of D-GAN	Epoch 10	Training losses	Epoch 100	Training losses	Characteristics	Shortcomings
D-GAN					Generator-discriminator framework via a minimax game where samples are directly generated.	The model parameters oscillate, destabilize, and never converge.
DCGAN					It uses convolutional stride and transposed convolution for the downsampling and the upsampling.	Gradients disappear or explode.
CGAN					Conditional generation of images.	CGAN is not strictly unsupervised. Some labeling is required for it to work strictly.

TABLE 7: Continued.

Type of D-GAN	Epoch 10	Training losses	Epoch 100	Training losses	Characteristics	Shortcomings
LSGAN					It creates high-quality images compared to the GAN. It is more stable during training.	Additional computational cost.
	WGAN					Stability of learning and overcomes the mode collapse problem.

TABLE 8: Challenges and open problems for computational visual perception-driven image analysis.

Research direction	Key references	Dataset	Challenges and open problems	Future research
Water level measurement	[132, 133]	GNSS dataset	Image-based in situ water level measurement faces several challenges: image distortions, low visibility, and ambient noises.	The image can be changed from 2D to 3D, and the water level measurement can be done by utilizing advanced computer vision techniques.
Image processing	[134, 135]	Sequential image captured with the camera	Classical image processing techniques are based on a controlled environment. Environmental changes require image processing techniques such as custom filters, thresholding, and limited site coverage.	The model is generalized using deep learning techniques in a dynamic environment.
Flood detection	[136]	Manually collected dataset	Lack of open-source data to train computer vision algorithms.	The data can be collected and opened to train the proposed model.
UAV image processing	[137, 138]	Datasets of AOGCM simulations	Proposed solutions have limited generalizability.	Advanced convolutional neural networks can be used. Instead of using image processing techniques, the model's generalizability can be assessed using real-world data for the testing phase.

interval. Image registration techniques can be used to register these to each other, and the movement can be recorded, which can help predict future landslides. This technique will be beneficial when satellite images of areas that are prone to landslides are available. Deviations in the positions of hills are estimated to predict the number of future landslides.

- (x) The research related to underwater imaging is evolving to expose undiscovered species underwater. It is not possible to identify all the underwater species by continuously visualizing the recorded videos underwater. Therefore, an automated system to classify or detect the species underwater is required.

## 7. Conclusions

This paper summarized a comprehensive survey of deep CNN and deep GAN, their basic principles, GAN variants, and their computational visual perception applications. A comparison between biological and computer vision is made to better understand the background of computer vision and understand the architecture of neural networks. This survey presents an extensive comparison of current and existing surveys. This survey extensively surveys the applications of D-GAN and D-CNN. The building blocks of the CNN, recent advancements of the CNN, activation functions, D-GAN evolution, and its recent advancement are discussed in detail. The pitfalls of deep learning and its solutions are discussed briefly. Both state-of-the-art and classical applications of deep GAN and deep CNN are evaluated. Developments in computational visual perception or computer vision with D-GAN and D-CNN in recent years are reviewed. D-GAN is proved to solve the problem of insufficient data and improve the quality of image generation. Experimental results are discussed to explore the ability of

variants of D-GAN models. The advantages, disadvantages, and network architectures of different D-GAN models are discussed. Besides, D-CNN and D-GAN applications that have achieved remarkable achievements in various computer vision applications are discussed. Furthermore, D-GAN has a wide variety of applications combined with other deep learning algorithms.

This article extensively surveyed the current opportunities and future challenges in all the emerging domains. This article discussed the current opportunities in many emerging domains such as handwriting recognition, semantic mapping, webcam-based eye trackers, lumen center detection, query-by-string word, intermittently closed and open lakes and lagoons, and landslides. Future research with the D-GAN has to be directed towards model collapse, nonconvergence, and training difficulties. Though there are vast improvements such as weight regularization, weight pruning, and Nash equilibrium, future research in this area is still mandatory. D-GAN in the security domain has more research scope as adversarial attacks on neural networks have become very common. Slight perturbation in samples may lead to the wrong classification by neural networks. Furthermore, to outdo adversarial attacks, it is necessary to make D-GANs more robust to adversarial attacks. Though D-GAN is put forward as unsupervised learning, adding labels to the data will significantly improve the D-GAN's data quality. Modifying the D-GAN in this way is one of the future research directions.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

N. A. R, P. D. R. V, and C-Y. C. performed the conceptualization and supervised the data. C-Y. C. carried out the funding acquisition. N. A. R, P. D. R. V, K. S, and C-Y.

C. investigated the data and performed the methodology. C.-Y. C. and K. S. carried out the project administration and validated the data. N. A. R, P. D. R. V, K. S., and U. T wrote, reviewed, and edited the manuscript. All authors read and agreed to the published version of the manuscript.

## Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan (Grant no. MOST 103-2221-E-224-016-MY3). This research was partially funded by the “Intelligent Recognition Industry Service Research Center” from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, and the APC was funded by the aforementioned project.

## References

- [1] L. Roberts, *Machine Perception of Three-Dimensional Solids*, IEEE, New York, NY, USA, 1963.
- [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [3] A. Graves, “Connectionist temporal classification,” in *Studies in Computational Intelligence*, pp. 61–93, Springer, Berlin, Germany, 2012.
- [4] A. G. Ivakhnenko, “Polynomial theory of complex systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 4, pp. 364–378, 1971.
- [5] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.
- [6] T.-Y. Lin, M. Maire, S. Belongie et al., “Microsoft COCO: common objects in context,” in *Proceedings of the Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 740–755, Springer International Publishing, Cham, Switzerland, 2014.
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680, Curran Associates, Inc., Red Hook, NY, USA, 2014.
- [8] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016.
- [9] C. Ledig, L. Theis, F. Huszar et al., “Photo-realistic single image super-resolution using a generative adversarial network,” 2017, <https://arxiv.org/abs/1609.04802>.
- [10] E. L. Denton, S. Chintala, a. szlam, and R. Fergus, “Deep generative image models using a Laplacian pyramid of adversarial networks,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, pp. 1486–1494, Curran Associates, Inc., Red Hook, NY, USA, 2015.
- [11] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *Proceedings of the International Conference on Machine Learning: PMLR*, pp. 7354–7363, Long Beach, CA, USA, June 2019.
- [12] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, “Generating images with recurrent adversarial networks,” 2016, <https://arxiv.org/abs/1602.05110>.
- [13] M. Xudong, L. Qing, H. Xie, and Y. K. Raymond, “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2250, Venice, Italy, October 2017.
- [14] H.-Y. Li, W.-M. Dong, and B.-G. Hu, “Facial image attributes transformation via conditional recycle generative adversarial networks,” *Journal of Computer Science and Technology*, vol. 33, no. 3, pp. 511–521, 2018.
- [15] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” 2016, <https://arxiv.org/abs/1610.09585>.
- [16] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, <https://arxiv.org/abs/1411.1784>.
- [17] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: interpretable representation learning by information maximizing generative adversarial nets,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2172–2180, Barcelona, Spain, December 2016.
- [18] D. Rui, G. Guo, X. Yan, B. Chen, Z. Liu, and X. He, “BiGAN: collaborative filtering with bidirectional generative adversarial networks,” in *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 82–90, Cincinnati, OH, USA, May 2020.
- [19] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” 2016, <https://arxiv.org/abs/1511.05644>.
- [20] Y. Mroueh, T. Sercu, and V. Goel, “Mcgan: mean and covariance feature matching gan,” in *Proceedings of the International Conference on Machine Learning*, pp. 2527–2535, Sydney, Australia, August 2017.
- [21] A. Martin, C. Soumith, and B. Léon, “Wasserstein generative adversarial networks,” *Proceedings of the 34th International Conference on Machine Learning*, PMLR, vol. 70, pp. 214–223, 2017.
- [22] D. Gerónimo, J. Serrat, A. M. López, and R. Baldrich, “Traffic sign recognition for computer vision project-based learning,” *IEEE Transactions on Education*, vol. 56, no. 3, pp. 364–371, 2013.
- [23] J. Thevenot, M. B. López, and A. Hadid, “A survey on computer vision for assistive medical diagnosis from faces,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1497–1511, 2018.
- [24] H. Hassan, A. K. Bashir, M. Ahmad et al., “Real-time image dehazing by superpixels segmentation and guidance filter,” *Journal of Real-Time Image Processing*, 2020.
- [25] M. Ahmad, A. K. Bashir, and A. M. Khan, “Metric similarity regularizer to enhance pixel similarity performance for hyperspectral unmixing,” *Optik*, vol. 140, pp. 86–95, 2017.
- [26] H. Hassan, A. K. Bashir, R. Abbasi, W. Ahmad, and B. Luo, “Single image autofocus estimation by modified Gaussian function,” *Transactions on Emerging Telecommunications Technologies*, Wiley, Hoboken, NJ, USA, 2019.
- [27] G. Danuser, “Computer vision in cell biology,” *Cell*, vol. 147, no. 5, pp. 973–978, 2011.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [29] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, Venice, Italy, October 2017.

- [30] Z. Li, W. Yang, S. Peng, and F. Liu, "A survey of convolutional neural networks: analysis, applications, and prospects," 2020, <https://arxiv.org/abs/2004.02806>.
- [31] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 660–674, 2017.
- [32] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: a review," *Medical Image Analysis*, vol. 58, p. 101552, 2019.
- [33] K. Zhu, X. Liu, and H. Yang, "A survey of generative adversarial networks," in *Proceedings of the 2018 chinese automation congress (CAC)*, pp. 2768–2773, Xi'an, China, November–December 2018.
- [34] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): a survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.
- [35] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [36] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," 2019, <https://arxiv.org/abs/1809.02165>.
- [37] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, p. 60, 2019.
- [38] Y. Chen, Y. Zhao, W. Jia, L. Cao, and X. Liu, "Adversarial-learning-based image-to-image transformation: a survey," *Neurocomputing*, vol. 411, pp. 468–486, 2020.
- [39] S. Kalra and A. Leekha, "Survey of convolutional neural networks for image captioning," *Journal of Information and Optimization Sciences*, vol. 41, no. 1, pp. 239–260, 2020.
- [40] N. V. K. Medathati, H. Neumann, G. S. Masson, and P. Kornprobst, "Bio-inspired computer vision: towards a synergistic approach of artificial and biological vision," *Computer Vision and Image Understanding*, vol. 150, pp. 1–30, 2016.
- [41] J. Sanchez-Riera, K. Srinivasan, K.-L. Hua, W.-H. Cheng, M. A. Hossain, and M. F. Alhamid, "Robust RGB-D hand tracking using deep learning priors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2289–2301, 2018.
- [42] A. Kumar, K. Srinivasan, W.-H. Cheng, and A. Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Information Processing & Management*, vol. 57, no. 1, Article ID 102141, 2020.
- [43] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.
- [44] W. Zhou, Y. Niu, and G. Zhang, "Sensitivity-oriented layer-wise acceleration and compression for convolutional neural network," *IEEE Access*, vol. 7, pp. 38264–38272, 2019.
- [45] S. Lin and G. C. Runger, "GCRNN: group-constrained convolutional recurrent neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4709–4718, 2018.
- [46] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory DSP]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.
- [47] "Deep Learning | Nature," October 2020, <https://www.nature.com/articles/nature14539>.
- [48] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, Article ID 258619, 12 pages, 2015.
- [49] J. Li, X. Long, S. Hu, Y. Hu, Q. Gu, and D. Xu, "A novel hardware-oriented ultra-high-speed object detection algorithm based on convolutional neural network," *Journal of Real-Time Image Processing*, vol. 17, pp. 1703–1714, 2020.
- [50] S. Hijazi, R. Kumar, and C. Rowen, *Using Convolutional Neural Networks for Image Recognition*, pp. 1–12, Cadence Design Systems, Inc., San Jose, CA, USA, 2015.
- [51] F. H. C. Tivive and A. Bouzerdoum, "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 541–556, 2005.
- [52] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [53] J. M. Ponce, A. Aquino, and J. M. Andújar, "Olive-fruit variety classification by means of image processing and convolutional neural networks," *IEEE Access*, vol. 7, pp. 147629–147641, 2019.
- [54] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.
- [55] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.
- [56] X. Zhang, X. Gao, W. Lu, and L. He, "A gated peripheral-foveal convolutional neural network for unified image aesthetic prediction," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2815–2826, 2019.
- [57] Q. Xuan, Z. Chen, Y. Liu, H. Huang, G. Bao, and D. Zhang, "Multiview generative adversarial network and its application in pearl classification," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 8244–8252, 2018.
- [58] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 31–40, 2016.
- [59] I. Ramírez, A. Cuesta-Infante, J. J. Pantrigo et al., "Convolutional neural networks for computer vision-based detection and recognition of dumpsters," *Neural Computing and Applications*, vol. 32, pp. 13203–13211, 2018.
- [60] K. Aukkapinyo, S. Sawangwong, P. Pooyoi, and W. Kusakunniran, "Localization and classification of rice-grain images using region proposals-based convolutional neural network," *International Journal of Automation and Computing*, vol. 17, pp. 233–246, 2019.
- [61] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Improved inception-residual convolutional neural network for object recognition," *Neural Computing and Applications*, vol. 32, no. 1, pp. 279–293, 2020.
- [62] J. Moon, H. Kim, and B. Lee, "View-point invariant 3d classification for mobile robots using a convolutional neural network," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2888–2895, 2018.
- [63] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal*

- of *Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3465–3471, 2018.
- [64] J. Teng, D. Zhang, D.-J. Lee, and Y. Chou, “Recognition of Chinese food using convolutional neural network,” *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11155–11172, 2019.
- [65] D. V. Pakulich, S. A. Yakimov, and S. A. Alyamkin, “Age recognition from facial images using convolutional neural networks,” *Optoelectronics, Instrumentation and Data Processing*, vol. 55, no. 3, pp. 255–262, 2019.
- [66] N. Puhalanthi and D.-T. Lin, “Effective multiple person recognition in random video sequences using a convolutional neural network,” *Multimedia Tools and Applications*, vol. 79, no. 15–16, pp. 11125–11141, 2020.
- [67] A. Madakannu and A. Selvaraj DIGI-Net, “DIGI-Net: a deep convolutional neural network for multi-format digit recognition,” *Neural Computing and Applications*, vol. 32, no. 15, pp. 11373–11383, 2020.
- [68] F. Özyurt, T. Tuncer, E. Avci, M. Koç, and İ. Serhatlioğlu, “A novel liver image classification method using perceptual hash-based convolutional neural network,” *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3173–3182, 2019.
- [69] K. Chauhan and S. Ram, “Image classification with deep learning and comparison between different convolutional neural network structures using tensorflow and keras,” *International Journal of Advanced Research and Development*, vol. 5, pp. 533–538, 2018.
- [70] Z. Qiumei, T. Dan, and W. Fenghua, “Improved convolutional neural network based on fast exponentially linear unit activation function,” *IEEE Access*, vol. 7, pp. 151359–151367, 2019.
- [71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.
- [72] A. F. Agarap, “Deep learning using rectified linear units (relu),” 2018, <https://arxiv.org/abs/1803.08375>.
- [73] W. Zhao and S. Du, “Spectral-spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016.
- [74] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, <https://arxiv.org/abs/1207.0580>.
- [75] J. Ba and B. Frey, “Adaptive dropout for training deep neural networks,” in *Advances in Neural Information Processing Systems* 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 3084–3092, Curran Associates, Inc., Red Hook, NY, USA, 2013.
- [76] X. Yu, X. Wu, C. Luo, and P. Ren, “Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework,” *GIScience & Remote Sensing*, vol. 54, no. 5, pp. 741–758, 2017.
- [77] L. Taylor and G. Nitschke, “Improving deep learning with generic data augmentation,” in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1542–1547, Bangalore, India, November 2018.
- [78] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, “A bayesian data augmentation approach for learning deep models,” 2017, <https://arxiv.org/abs/1710.10564>.
- [79] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.
- [80] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” 2017, <https://arxiv.org/abs/1712.04621>.
- [81] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: learning augmentation policies from data,” 2019, <https://arxiv.org/abs/1805.09501>.
- [82] M. A. Alsheikh, D. Niyato, S. Lin, H.-p. Tan, and Z. Han, “Mobile big data analytics using deep learning and Apache spark,” *IEEE Network*, vol. 30, no. 3, pp. 22–29, 2016.
- [83] R. A. Bauder, T. M. Khoshgoftaar, and T. Hasanin, “Data sampling approaches with severely imbalanced big data for medicare fraud detection,” in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 137–142, Volos, Greece, November 2018.
- [84] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, “Using trusted data to train deep networks on labels corrupted by severe noise,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 10456–10465, Montréal, Canada, December 2018.
- [85] C. You, W. Cong, M. W. Vannier et al., “CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE),” *IEEE Transactions on Medical Imaging*, vol. 39, no. 1, pp. 188–203, 2020.
- [86] Y. Zhan, D. Hu, Y. Wang, and X. Yu, “Semisupervised hyperspectral image classification based on generative adversarial networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2018.
- [87] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, “A novel image steganography method via deep convolutional generative adversarial networks,” *IEEE Access*, vol. 6, pp. 38303–38314, 2018.
- [88] C. Wang, C. Xu, C. Wang, and D. Tao, “Perceptual adversarial networks for image-to-image transformation,” *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4066–4079, 2018.
- [89] N. Li, Z. Zheng, S. Zhang, Z. Yu, H. Zheng, and B. Zheng, “The synthesis of unpaired underwater images using a multistyle generative adversarial network,” *IEEE Access*, vol. 6, pp. 54241–54257, 2018.
- [90] A. Lucas, A. K. Katsaggelos, S. Lopez-Tapia, and R. Molina, “Generative adversarial networks and perceptual losses for video super-resolution,” in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 51–55, Athens, Greece, October 2018.
- [91] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016, <https://arxiv.org/abs/1511.06434>.
- [92] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in *Proceedings of the International Conference on Machine Learning; PMLR*, pp. 1558–1566, Boston, MA, USA, October 2016.
- [93] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, Honolulu, HI, USA, July 2017.



- [94] Yu.V. Vizil'ter, O. V. Vygolov, D. V. Komarov, and M. A. Lebedev, "Fusion of images of different spectra based on generative adversarial networks," *Journal of Computer and Systems Sciences International*, vol. 58, pp. 441–453, 2019.
- [95] H. Zhang, B. S. Riggan, S. Hu, N. J. Short, and V. M. Patel, "Synthesis of high-quality visible faces from polarimetric thermal faces using generative adversarial networks," *International Journal of Computer Vision*, vol. 127, no. 6-7, pp. 845–862, 2019.
- [96] J. Shen, N. Liu, H. Sun, and H. Zhou, "Vehicle detection in aerial images based on lightweight deep convolutional network and generative adversarial network," *IEEE Access*, vol. 7, pp. 148119–148130, 2019.
- [97] P. Xiang, L. Wang, F. Wu, J. Cheng, and M. Zhou, "Single-image de-raining with feature-supervised generative adversarial network," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 650–654, 2019.
- [98] X. Wang, Z. Cao, R. Wang, Z. Liu, and X. Zhu, "Improving human pose estimation with self-attention generative adversarial networks," *IEEE Access*, vol. 7, pp. 119668–119680, 2019.
- [99] W. Wang, A. Wang, Q. Ai, C. Liu, and J. Liu, "AAGAN: enhanced single image dehazing with attention-to-attention generative adversarial network," *IEEE Access*, vol. 7, pp. 173485–173498, 2019.
- [100] W. Fang, Y. Ding, F. Zhang, and J. Sheng, "Gesture recognition based on CNN and DCGAN for calculation and text output," *IEEE Access*, vol. 7, pp. 28230–28237, 2019.
- [101] A. T. Arslan and E. Seke, "Face depth estimation with conditional generative adversarial networks," *IEEE Access*, vol. 7, pp. 23222–23231, 2019.
- [102] C. Xu, Y. Cui, Y. Zhang, P. Gao, and J. Xu, "Image enhancement algorithm based on generative adversarial network in combination of improved game adversarial loss mechanism," *Multimedia Tools and Applications*, vol. 79, no. 13-14, pp. 9435–9450, 2020.
- [103] Z. Yu, Q. Xiang, J. Meng, C. Kou, Q. Ren, and Y. Lu, "Retinal image synthesis from multiple-landmarks input with generative adversarial networks," *BioMedical Engineering OnLine*, vol. 18, no. 1, p. 62, 2019.
- [104] M. Ahmad, M. Mazzara, R. A. Raza et al., "Multiclass non-randomized spectral-spatial active learning for hyperspectral image classification," *Applied Sciences*, vol. 10, no. 14, p. 4739, 2020.
- [105] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-D CNN for hyperspectral image classification," 2020, <https://arxiv.org/pdf/2004.14152.pdf>.
- [106] M. Ahmad, S. Shabbir, D. Oliva, M. Mazzara, and S. Distefano, "Spatial-prior generalized fuzziness extreme learning machine autoencoder-based active learning for hyperspectral image classification," *Optik*, vol. 206, Article ID 163712, 2020.
- [107] M. Ahmad, "Ground truth labeling and samples selection for hyperspectral image classification," *Optik*, vol. 230, Article ID 166267, 2021.
- [108] M. Ahmad, M. A. Alqarni, A. M. Khan, R. Hussain, M. Mazzara, and S. Distefano, "Segmented and non-segmented stacked denoising autoencoder for hyperspectral band reduction," *Optik*, vol. 180, pp. 370–378, 2019.
- [109] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, Nevada, USA, June-July 2016.
- [110] F. Siddique, S. Sakib, and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in Python with tensorflow and comparison of performance for various hidden layers," in *Proceedings of the 2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 541–546, Dhaka, Bangladesh, September 2019.
- [111] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. D. Salakhutdinov, "A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [112] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: a simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.
- [113] A. Saxena, "Image classification using convolutional neural networks," *Journal of the Gujarat Research Society*, vol. 21, pp. 72–80, 2019.
- [114] D. V. A. Bharadi, A. I. Mukadam, M. N. Panchbhai, and N. N. Rode, "Image classification using deep learning," *International Journal of Engineering Research and Technology*, vol. 6, 2017.
- [115] I. Amerini, C.-T. Li, and R. Caldelli, "Social network identification through image classification with CNN," *IEEE Access*, vol. 7, pp. 35264–35273, 2019.
- [116] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1*, pp. 656–664, Curran Associates Inc., Red Hook, NY, USA, December 2012.
- [117] S. ul Hussain and B. Triggs, "Feature sets and dimensionality reduction for visual object detection," in *Proceedings of the British Machine Vision Conference 2010*, pp. 112.1–112.10, British Machine Vision Association, Aberystwyth, UK, August-September 2010.
- [118] H. Zhuang, K.-S. Low, and W.-Y. Yau, "Multichannel pulse-coupled-neural-network-based color image segmentation for object detection," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3299–3308, 2012.
- [119] S. Tu, J. Pang, H. Liu et al., "Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images," *Precision Agriculture*, vol. 21, no. 5, pp. 1072–1091, 2020.
- [120] D. Augusto Borges Oliveira and M. Palhares Viana, "Fast CNN-based document layout analysis," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1173–1180, Venice, Italy, October 2017.
- [121] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, "A CNN-BiLSTM model for document-level sentiment analysis," *Machine Learning and Knowledge Extraction*, vol. 1, no. 3, pp. 832–847, 2019.
- [122] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.
- [123] F. Ghorban, N. Milani, D. Schugk et al., "Conditional multichannel generative adversarial networks with an application to traffic signs representation learning," *Progress in Artificial Intelligence*, vol. 8, no. 1, pp. 73–82, 2019.
- [124] T. Zhang, P. Jiang, and M. Zhang, "Inter-frame video image generation based on spatial continuity generative adversarial networks," *Signal, Image and Video Processing*, vol. 13, no. 8, pp. 1487–1494, 2019.

- [125] Z. Wang, G. Healy, A. F. Smeaton, and T. E. Ward, "Use of neural signals to evaluate the quality of generative adversarial network performance in facial image generation," *Cognitive Computation*, vol. 12, no. 1, pp. 13–24, 2020.
- [126] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020.
- [127] G. Zhao, J. Liu, J. Jiang, and W. Wang, "A deep cascade of neural networks for image inpainting, deblurring and denoising," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29589–29604, 2018.
- [128] J. Zhang, H. Dang, H. K. Lee, and E.-C. Chang, "Flipped-adversarial auto encoders," 2018, <https://arxiv.org/pdf/1802.04504.pdf>.
- [129] H. T. Shen, J. Song, J. Zhang, L. Gao, and X. Liu, "Dual conditional GANs for face aging and rejuvenation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, pp. 899–905, Stockholm, Sweden, July 2018.
- [130] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: facial attribute editing by only changing what you want," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5464–5478, 2019.
- [131] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," 2020, <https://arxiv.org/abs/1907.10786>.
- [132] Z. Zhang, Y. Zhou, H. Liu, and H. Gao, "In-situ water level measurement using NIR-imaging video camera," *Flow Measurement and Instrumentation*, vol. 67, pp. 95–106, 2019.
- [133] E. Ridolfi and P. Manciola, "Water level measurements from drones: a pilot case study at a dam site," *Water*, vol. 10, p. 297, 2018.
- [134] S. Udomsiri and M. Iwahashi, "Design of FIR filter for water level detection," *World Academy of Science, Engineering and Technology*, vol. 48, pp. 47–52, 2008.
- [135] S. Park, N. Lee, Y. Han, and H. Hahn, "The water level detection algorithm using the accumulated histogram with band pass filter," *World Academy of Science, Engineering and Technology*, vol. 56, pp. 193–197, 2009.
- [136] W. Kang, Y. Xiang, F. Wang, L. Wan, and H. You, "Flood detection in gaofen-3 SAR images via fully convolutional networks," *Sensors*, vol. 18, no. 9, p. 2915, 2018.
- [137] Y. Hirabayashi, R. Mahendran, S. Koirala et al., "Global flood risk under climate change," *Nature Climate Change*, vol. 3, no. 9, pp. 816–821, 2013.
- [138] A. L. Sumalan, D. A. N. Popescu, and L. Ichim, "Flooded areas detection based on LBP from UAV images," in *Proceedings of the 2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 186–191, Sinaia, Romania, October 2015.

## Research Article

# Design and Implementation of Novel LMI-Based Iterative Learning Robust Nonlinear Controller

Saleem Riaz <sup>1</sup>, Hui Lin <sup>1</sup>, Farkhanda Afzal <sup>2</sup>, and Ayesha Maqbool <sup>3</sup>

<sup>1</sup>School of Automation, Northwestern Polytechnical University, Shaanxi, Xi'an 170072, China

<sup>2</sup>MCS, National University of Sciences and Technology, Islamabad 44000, Pakistan

<sup>3</sup>Department of Computer Science, NUST Balochistan Campus (NBC), Quetta, Islamabad, Pakistan

Correspondence should be addressed to Hui Lin; [linhui@nwpu.edu.cn](mailto:linhui@nwpu.edu.cn)

Received 31 January 2021; Revised 14 March 2021; Accepted 22 March 2021; Published 16 April 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Saleem Riaz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An iterative learning robust fault-tolerant control algorithm is proposed for a class of uncertain discrete systems with repeated action with nonlinear and actuator faults. First, by defining an actuator fault coefficient matrix, we convert the iterative learning control system into an equivalent unknown nonlinear repetitive process model. Then, based on the mixed Lyapunov function approach, we describe the stability of the nonlinear repetitive mechanism on time and trial indices and have appropriate conditions for the repeated control system's stability in terms of linear matrix inequality theory. Through LMI techniques, we have obtained satisfactory results and controller stability, and robustness against fault tolerance is also discussed in detail. Finally, the simulation results of the output tracking control of the two exemplary models verify the effectiveness of the proposed algorithm.

## 1. Introduction

Iterative learning control is suitable for controlled objects with repetitive motion (running) properties in a limited time interval. It uses the data generated during the previous iteration of the system to correct undesirable control signals and generate the control signals used in the current iteration to make the system control. The performance is gradually improved, and finally, the complete tracking in a limited time interval is achieved. In a comparison with other control methods, the iterative learning control method has a simple controller method, a small amount of calculation, and only less knowledge of dynamic characteristics and can get the precise control. The characteristics of complete tracking of the controlled object are as follows: this control technology is applied in industrial applications such as assembly line industrial robots and chemical intermittent processes. The process of the iterative learning control system can generally detect the timeline and batch on the tree of a two-dimensional repetitive process. Over the last few years, based on system 2D generation idea, learning control system analysis and design had tremendous attention. The study in [1]

designed a PD-type iterative learning control algorithm for a linear continuous repetition system based on the 2D method and obtained the system with accurate tracking control; the authors of [2, 3] combined with the stability theory discrete process designed an iterative learning controller. The output error of the system converges monotonically, and the performance in the direction of the batch axis gradually improves with the repeated process. To optimize the replicated system's performance in both time and batch order, the study in [4] designed a good control architecture for the uncertain 2D system. It proposed a balanced optimization algorithm of robust iterative learning, which comprehended the repeated system's stable robustness in both directions [5].

Iterative learning control uses the tracking error and original data of the system to continuously modify the current input target, so that the system output can quickly track the expected output within a limited time [6–9]. Based on iterative learning control system that can be regarded as a kind of repeated processes based on batch axis and the timeline, the system control problem of the research in recent years came to the attention of academia at home and

abroad and was successfully used in multiaxis truss type robot, injection molding machine, and electric motorcycle motor system in the practical repeat operations such as industrial object [3, 10–12]. As the process of industrialization continues to accelerate, industrial systems are becoming more and more complex, and people's requirements for system reliability and safety are getting higher and higher, making fault diagnosis and fault-tolerant control in the past few decades [13–16]. Both academic and practical application fields have received more and more attention. Fault-tolerant control is divided into active fault-tolerant control and passive fault-tolerant control. In fact, most researches pay more attention to active fault-tolerant control. Fault estimation [17–20] is different from fault diagnosis; it can accurately estimate the magnitude and shape of the fault and thus is able to reconstruct the fault signal. Therefore, in existing literature, fault estimation is a prerequisite for fault-tolerant control and has achieved affluence of theoretical research results [21–23]. At present, the methods of fault estimation mainly include observer-based methods [24, 25], signal reconstruction-based methods [26, 27], and artificial intelligence-based methods [28, 29]. The observer-based methods can accurately reflect the impact of faults on system performance and accuracy. The fault information has attracted the attention of domestic and foreign scholars in theoretical research. It mainly includes methods based on adaptive state observers [30–32], methods based on robust observers [33–36], and optimal [9, 37] and sliding mode observers approaches [38–41]. These methods have been continuously applied to network control systems [42–44], nonlinear systems [45], switching systems [46], fuzzy systems [47], parameter changing systems [48], and so on.

In control theory, the RC is a neutral-type delay, and the number is like endless poles. In 1988, "Scientific American" (English) mentioned that RCs could only stabilize unless there are zero degree relative plants [49]. Most technicians need to push the low-level filter on the delay control for a strictly proper plant. On this basis, the established system is improved to modify RCs (MRCs), and its low-level filter can reduce the steady state at the cost of high-frequency tracking signals [50]. That is to say, the stability and tracking functions in MRC have an intermediate point. Therefore, on the one hand, it is a stable system, and on the other hand, it is easily introduced into the distorted system when there is great uncertainty in the way [51–53].

However, it cannot be ignored that actuators in actual industrial systems with repetitive operation characteristics are often in high-frequency responding, and actuators are very prone to mechanical wear, failure, and other faults, which may not only reduce the control performance of the system but also affect the stability and safety of the system [7, 54–58]. B. Cichy and his team [59] discuss the design of robust iterative learning fault-tolerant control law for actuator fault linear systems and its convergence. The paper [60] further analyzes the robustness of the guaranteed iterative learning fault-tolerant control system and finally guarantees the reliability of the iterative learning repetition process on batch axis and time axis. However, the above-stated pieces of literature

are all aimed at completely linear systems, while the real control systems inevitably have some nonlinear characteristics. In papers [61, 62], iterative learning control laws are designed for discrete repeated processes with nonlinear links, and stability analysis is carried out for these nonlinear systems by defining hybrid Lyapunov function and adaptive Lyapunov function based on batch axis and time axis, respectively. However, these methods do not fully consider the robustness of uncertainty in the systems, let alone discuss the fault shadow of nonlinear systems. Therefore, the design of robust iterative fault-tolerant control law for repetitive process systems with nonlinear links and uncertainties is an inevitable choice to improve the performance and reliability of system batches.

In this paper, the iterative learning in fault-tolerant control for a class of uncertain discrete nonlinear repetitive processes with unknown actuator faults is studied. By designing iterative learning fault-tolerant control law and defining the mixed Lyapunov function based on batch axis and time axis, the stability of the system under normal and failure conditions is discussed, respectively, and the sufficient conditions for the existence of robust fault-tolerant controller are given in the form of linear matrix inequality (LMI). Finally, this method is applied to the simulation of single-link manipulator and the injection velocity control process of molding process system to verify the effectiveness of the proposed method.

In this paper, for the convenience of description, the following assumptions are made for the matrix:

- $\mathbf{X}, \mathbf{X}^T \in R^n$  represents its transpose.
- $\mathbf{X} < 0$  represents a negative definite matrix
- $\mathbf{X} > 0$  denotes a positive definite matrix
- $*$  denotes the transpose of elements at the symmetric position of the matrix
- $\|\cdot\|$  represents the Euclidean norm
- $\forall$  any

## 2. Problem Description

Consider the following class of uncertain discrete time-invariant nonlinear systems running repeatedly:

$$\begin{cases} x(t+1, k) = (A + \Delta A)x(t, k) + (B + \Delta B)u(t, k) + f(x(t, k)), \\ y(t, k) = (C + \Delta C)x(t, k), \end{cases} \quad (1)$$

where  $k = 0, 1, \dots, N$  stands for batch, and the repeating time cycle within each batch is  $0 \leq t \leq T$ .  $X(t, k) \in R^n$ ,  $u(t, k) \in R^l$ , and  $y(t, k) \in R^m$  separately represent the state vector, input vector, and output vector of the system. Without loss of generality, suppose the initial boundary conditions of the system  $x(0, k) = x_0$  and  $u(t, 0) = u_0(t)$ . Matrices  $A$ ,  $B$ , and  $C$  are, respectively, the corresponding dimension of the system matrix, and  $\Delta A$ ,  $\Delta B$ , and  $\Delta C$  are uncertainty [63] and satisfy the following relations:  $\Delta A = H_1 \Xi F_1$ ,  $\Delta B = H_1 \Xi F_2$ , and  $\Delta C = H_2 \Xi F_2$ , including  $H_1$ ,  $H_2$ ,  $F_1$ , and  $F_2$  to already know the certainty of the

matrix;  $\Xi \Xi T \leq I$  or less bounded constraint condition where  $I$  is the identity matrix. In addition,  $f(x(t, k)) \in R^n$  represents a nonlinear vector function obtained from the modeling of the system's nonlinear link, assuming that the following conditions are met:

$$\|f(x(t, k+1)) - f(x(t, k))\| \leq \zeta \|x(t, k+1) - x(t, k)\|. \quad (2)$$

Here,  $\zeta > 0$  is Lipschitz constant. For the vector element  $UI(t, k)$  that is input to the system,  $u_i^F(t, k)$  represents the output when an actuator fails in the system. Actuator fault model is further defined:

$$u_i^F(t, k) = \alpha_i u_i(t, k), \quad i = 1, 2, \dots, l, \quad (3)$$

where the actuator fault coefficient  $\alpha_i$  is unknown but meets the following conditions:

$$0 \leq \underline{\alpha}_i \leq \alpha_i \leq \bar{\alpha}_i, \quad i = 1, 2, \dots, l, \quad (4)$$

where the properties  $\underline{\alpha}_i$  ( $\underline{\alpha}_i \leq 1$ ) and  $\bar{\alpha}_i$  ( $\bar{\alpha}_i \geq 1$ ) are known as scalars.

If  $\underline{\alpha}_i = \bar{\alpha}_i = 1$ , it means that the control system actuator is working normally; that is,  $u_i^F(t, k) = u_i(t, k)$ . If  $\alpha_i = 0$ , it represents the complete failure of the actuator due to the damage caused by various emergencies;  $\alpha_i > 0$  represents a partial actuator failure due to normal conditions such as

$$\begin{aligned} \mathbf{u}^F &= [u_1^F, u_2^F, \dots, u_l^F]^T, \\ \bar{\boldsymbol{\alpha}} &= \text{diag}\{\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_l\}, \\ \underline{\boldsymbol{\alpha}} &= \text{diag}\{\underline{\alpha}_1, \underline{\alpha}_2, \dots, \underline{\alpha}_l\}, \\ \boldsymbol{\alpha} &= \text{diag}\{\alpha_1, \alpha_2, \dots, \alpha_l\}. \end{aligned} \quad (5)$$

On this arrangement,  $\beta = \text{diag}\{\beta_1, \beta_2, \dots, \beta_l\}$ ,  $\beta_0 = \text{diag}\{\beta_{10}, \beta_{20}, \dots, \beta_{l0}\}$ ,  $\beta_i = ((\bar{\alpha}_i + \alpha_i)/2)$ , and  $\beta_{i0} = ((\bar{\alpha}_i - \alpha_i)/(\bar{\alpha}_i + \alpha_i))$ ,  $i = 1, 2, \dots, l$ . There must be some unknown matrices  $\alpha_0$ :

$$\boldsymbol{\alpha} = (I + \alpha_0)\boldsymbol{\beta}, \quad (6)$$

$$I\boldsymbol{\alpha}_0 \leq \boldsymbol{\beta}_0 \leq I. \quad (7)$$

Other  $\alpha_0 = \text{diag}\{\alpha_{10}, \alpha_{20}, \dots, \alpha_{l0}\}$  and  $|\alpha_0| = \text{diag}\{|\alpha_{10}|, |\alpha_{20}|, \dots, |\alpha_{l0}|\}$ .

Therefore, discrete time-invariant nonlinear system (1) with actuator fault can be expressed as

$$\begin{cases} x(t+1, k) = (A + \Delta A)x(t, k) + (B + \Delta B)\omega t(t, k) + f(x(t, k)), \\ y(t, k) = (C + \Delta C)x(t, k). \end{cases} \quad (8)$$

Uncertain discrete nonlinear steady-state systems contain actuator failures (7), and the control goal of this paper is to meet the condition (6) in the case of unknown actuator failures. Based on the iterative learning, fault-tolerant control input makes the system output so that the control system can gradually track the output after a certain required number of iterations, i.e.,

$$\sup_{0 \leq i \leq T} (y_x(t) - y(t, k)) < \varepsilon. \quad (9)$$

### 3. Iterative Learning Control Law Design

According to the control objective of this paper, the output tracking error  $e(t, k)$  is defined as follows:

$$e(t, k) = y_k(t) - y(t, k). \quad (10)$$

For system (8), the following control law is designed:

$$u(t, k+1) = u(t, k) + \Delta u(t, k+1), \quad (11)$$

where  $u(t, k+1)$  is the system control input of the current batch;  $u(t, k)$  is the control input of the previous batch;  $\Delta u(t, k+1)$  is to modify control system input amount of updates.

For the convenience of analysis, define

$$\begin{cases} \eta(t+1, k+1) = x(t, k+1) - x(t, k), \\ \varphi(t, k+1) = f(x(t-1, k+1)) - f(x(t-1, k)). \end{cases} \quad (12)$$

Therefore, the modified update quantity in iterative learning control law (11) can be further obtained as

$$\Delta u(t, k+1) = K_1 \eta(t+1, k+1) + K_2 e(t+1, k), \quad (13)$$

where  $K_1$  and  $K_2$  are unknown undetermined matrices.

By substituting the iterative learning fault-tolerant control law form (11) into the nonlinear system (8), the state-space model of discrete repeated processes can be obtained in the following form:

$$\begin{cases} \boldsymbol{\eta}(t+1, k+1) = \hat{\mathbf{A}}\boldsymbol{\eta}(t, \mathbf{k}+1) + \hat{\mathbf{B}}\mathbf{e}(t, \mathbf{k}) + \boldsymbol{\varphi}(t, \mathbf{k}+1), \\ \mathbf{e}(t, \mathbf{k}+1) = \hat{\mathbf{C}}\boldsymbol{\eta}(t, \mathbf{k}+1) + \hat{\mathbf{D}}\mathbf{e}(t, \mathbf{k}) + \hat{\mathbf{k}}_\varphi(t, \mathbf{k}+1). \end{cases} \quad (14)$$

Here,

$$\begin{aligned} \hat{\mathbf{A}} &= (A + \Delta A) + (B + \Delta B)\alpha K_1, \\ \hat{\mathbf{B}} &= (B + \Delta B)\alpha K_2, \\ \hat{\mathbf{C}} &= -(C + \Delta C)((A + \Delta A) + (B + \Delta B)\alpha K_1), \\ \hat{\mathbf{D}} &= I - (C + \Delta C)(B + \Delta B)\alpha K_2, \\ \hat{\mathbf{E}} &= -(C + \Delta C). \end{aligned} \quad (15)$$

Obviously, discrete repetitive process model (14) is a nonlinear repeated process, including  $\eta(t+1, k+1)$  and  $\mathbf{e}(t, \mathbf{k})$ , respectively, and represents the time and batch variable on the axis direction,  $\varphi(t, \mathbf{k}+1)$  is to enter the present process of iterative nonlinear input item, and conventional stability analysis of linear systems of KYP lemma methods cannot be directly used to solve the repetitive process system with nonlinear term.

### 4. Stability Analysis

The following lemma is given in advance for subsequent stability analysis.

**Lemma 1** (see [64]). *If  $\Omega_1$  and  $\Omega_2$  are, respectively, about variable  $\omega \in R^n$  of two dimensions of the same symmetric matrix, then it shows the following equation:*

$\bar{\omega}^T \Omega_1 \omega \leq 0, \forall \omega \neq 0, \omega^T \Omega_2, \omega \leq 0$ . Founded, there is always the  $\exists \tau \geq 0$  can make  $\Omega_1 - \tau \Omega_2 < 0$ .

**Lemma 2** (see [65]). *The matrix  $\Phi$  with a given appropriate dimension is equal to  $\Phi = \Phi^T$ . And then for any matrix  $\Delta^T \Delta \leq I$ , or  $\Delta$  less inequality below  $\Phi + \mathbf{X}\Delta\mathbf{Y} + \mathbf{Y}^T\Delta^T\mathbf{X}^T < 0$ . If and only if there is  $\varepsilon > 0$ , such that  $\Phi + \varepsilon\mathbf{X}\mathbf{X}^T + \varepsilon^{-1}\mathbf{Y}^T\mathbf{Y} < 0$ .*

**Lemma 3** (see [66]). *For matrices of given dimensions  $\Gamma, \Lambda$ , and  $\Sigma$ , existing matrix  $W$  makes linear matrix inequality (LMI), there exists a matrix  $W$ . This makes the following linear matrix inequality true:  $\Gamma + \mathbf{sym}\{\Lambda^T\mathbf{W}\Sigma\} < 0$ , if and only if the following two inequalities about the matrix  $W$  are true  $\Lambda^{\perp T}\Gamma\Lambda^{\perp} < 0, \Sigma^{\perp T}\Gamma\Sigma^{\perp} < 0$ .*

**Theorem 1.** *If the existence of the matrix  $P > 0$  makes the following LMI true, then the nonlinear discrete repeat process (14) is stable.*

$$\begin{bmatrix} \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} - \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{C}} & \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{C}}^T \hat{\mathbf{D}} & \hat{\mathbf{A}}^T \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{D}}^T \hat{\mathbf{C}} & \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{D}}^T \hat{\mathbf{D}} - \mathbf{I} & \hat{\mathbf{B}}^T \mathbf{P} + \hat{\mathbf{D}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{E}}^T \hat{\mathbf{C}} & \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{E}}^T \hat{\mathbf{D}} & \mathbf{P} + \hat{\mathbf{E}}^T \hat{\mathbf{E}} \end{bmatrix} < 0. \quad (16)$$

It is proved that the mixed Lyapunov function for the discrete nonlinear repeat process (14) is constructed in the following form with respect to the time axis and batch axis:

$$\begin{aligned} \mathbf{V}(t, k) &= \mathbf{V}_1(t, k) + \mathbf{V}_2(t, k) = \boldsymbol{\eta}^T(t, k+1)\mathbf{P}_\eta(t, k+1) \\ &\quad + \mathbf{e}^T(t, k)\mathbf{R}\mathbf{e}(t, k), \end{aligned} \quad (17)$$

and define

$$\begin{aligned} \Delta \mathbf{V}_1(t, k) &= \boldsymbol{\eta}^T(t+1, k+1)\mathbf{P}_\eta(t+1, k+1) \\ &\quad - \boldsymbol{\eta}^T(t, k+1)\mathbf{P}_\eta(t, k+1), \\ \Delta \mathbf{V}_2(t, k) &= \mathbf{e}^T(t, k+1)\mathbf{R}\mathbf{e}(t, k+1) - \mathbf{e}^T(t, k)\mathbf{R}\mathbf{e}(t, k), \\ \Delta V(t, k) &= \Delta V_1(t, k) + \Delta V_2(t, k), \end{aligned} \quad (18)$$

where  $P > 0, R = I, b > 0$ . According to stability theorem, when  $\Delta V(t, k) < 0$ , discrete nonlinear repeat procedure (17) is stable: namely,

$$\Delta V_1(t, k) + \Delta V_2(t, k) < 0. \quad (19)$$

Plug in  $\eta(t+1, k+1)$  and  $e(t, k+1)$  into equation (19), and get

$$\begin{aligned} &[\hat{\mathbf{A}}_\eta(t, k+1) + \hat{\mathbf{B}}e(t, k) + \varphi(t, k+1)]^T \mathbf{P}, \\ &[\hat{\mathbf{A}}_\eta(t, k+1) + \hat{\mathbf{B}}e(t, k) + \varphi(t, k+1) - \eta(t, k+1)]^T \mathbf{P}, \\ &\boldsymbol{\eta}(t, k+1) + \hat{\mathbf{C}}\boldsymbol{\eta}(t, k+1) + \hat{\mathbf{D}}e(t, k) + \hat{\mathbf{E}}\varphi(t, k+1)^T, \\ &\hat{\mathbf{C}}\boldsymbol{\eta}(t, k+1) + \hat{\mathbf{D}}e(t, k) + \hat{\mathbf{E}}\varphi(t, k+1)^T - e(t, k)^T e(t, k) < 0. \end{aligned} \quad (20)$$

After transformation, the following inequalities can be obtained:

$$\mathbf{X}^T \begin{bmatrix} \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} + \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{C}} & \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{C}}^T \hat{\mathbf{D}} & \hat{\mathbf{A}}^T \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{D}}^T \hat{\mathbf{C}} & \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{D}}^T \hat{\mathbf{D}} - \mathbf{I} & \hat{\mathbf{B}}^T \mathbf{P} + \hat{\mathbf{D}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{E}}^T \hat{\mathbf{C}} & \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{E}}^T \hat{\mathbf{D}} & \mathbf{P} + \hat{\mathbf{E}}^T \hat{\mathbf{E}} \end{bmatrix} \mathbf{X}. \quad (21)$$

Here,  $\mathbf{X} = \begin{bmatrix} \eta(t, k+1) \\ e(t, k) \\ \varphi(t, k+1) \end{bmatrix}$ , thus directly obtaining the

conclusion of Theorem 1.

There is coupling between unknown matrix variables in the conclusion of Theorem 1, so K1 and K2 of iterative learning fault-tolerant control correction and update quantity (13) cannot be calculated by direct application of the LMI tool. Therefore, based on the conclusion of Theorem 1, it is further discussed and Theorem 2 is obtained.

**Theorem 2.** *Regardless of system (1), the uncertainty, i.e.,  $\Delta A, \Delta \Delta C$ , and  $B$  are assumed to 0, if there is  $A$  positive definite matrix  $Q > 0$ , matrix  $N_1$ , and  $N_2$  and scalar  $\tau > 0$ , epsilon  $1 > 0$  make the matrix inequality  $\Omega < 0$ , the nonzero symmetric matrix elements:  $\Omega_{11} = \Omega_{44} = -Q, \Omega_{14} = (AQ + B\beta N_1)T, \Omega_{15} = [-C(AQ + B\beta N_1)T, \Omega_{16} = Q, \Omega_{18} = (\beta N_1)T, \Omega_{22} = -\tau I, \Omega_{24} = (B\beta N_2)T, \Omega_{25} = (\tau I - CB\beta N_2)T, \Omega_{28} = (\beta N_2)T, \Omega_{33} = -I, \Omega_{34} = I, \Omega_{35} = (-C)T, \Omega_{47} = \varepsilon 1\beta 0B, \Omega_{55} = -\tau I, \Omega_{57} = -\varepsilon 1\beta 0CB, \Omega_{66} = -\rho I, \Omega_{77} = \Omega_{88} = -\varepsilon 1I$ , then the nonlinear discrete repetition process (14) is stable under the action of the actuator fault satisfies the admissible condition (7) and the iterative learning fault-tolerant control law (11), where  $\rho = \zeta^{-2}$ , modify the gain matrix of update quantity (13)  $K_1 = N_1 Q^{-1}, K_2 = \tau^{-1} N_2$ . The proof notes that the following equivalent conditions can be obtained according to equation (2):*

$$\begin{aligned} &[f(x(t-1, k+1)) - f(x(t-1, k))]^T \\ &[f(x(t-1, k+1)) - f(x(t-1, k))] \\ &\leq \zeta^2 [x(t-1, k+1) - x(t-1, k) \\ &\quad \cdot [x(t-1, k+1) - x(t-1, k)]]. \end{aligned} \quad (22)$$

Substituting equation (12) into the above equation, we can obtain  $\varphi(t, k+1)T\varphi(t, k+1) \leq \zeta^2 \eta(t, k+1)T\eta(t, k+1)$ .

That is,

$$\mathbf{X}^T \Omega_2 \mathbf{X} \leq 0, \quad (23)$$

and it can be known from Theorem 1 that  $\mathbf{X}^T \Omega_1 \mathbf{X} \leq 0$ , in which

$$\begin{aligned} \Omega_1 &= \begin{bmatrix} \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} - \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{C}} & \hat{\mathbf{A}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{C}}^T \hat{\mathbf{D}} & \hat{\mathbf{A}}^T \mathbf{P} + \hat{\mathbf{C}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{D}}^T \hat{\mathbf{C}} & \hat{\mathbf{B}}^T \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{D}}^T \hat{\mathbf{D}} - \mathbf{I} & \hat{\mathbf{B}}^T \mathbf{P} + \hat{\mathbf{D}}^T \hat{\mathbf{E}} \\ \hat{\mathbf{P}}\hat{\mathbf{A}} + \hat{\mathbf{E}}^T \hat{\mathbf{C}} & \hat{\mathbf{P}}\hat{\mathbf{B}} + \hat{\mathbf{E}}^T \hat{\mathbf{D}} & \mathbf{P} + \hat{\mathbf{E}}^T \hat{\mathbf{E}} \end{bmatrix}, \\ \Omega_2 &= \begin{bmatrix} -\zeta^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}. \end{aligned} \quad (24)$$

Then, the following inequality can be obtained through Lemma 1:

$$\Omega_1 - \tau\Omega_2 < 0, \quad (25)$$

and  $\tau > 0$  type can be rewritten as  $\tau - 1$  on  $\Omega_1 - \Omega_2 < 0$ ; namely,

$$\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ * & \theta_{22} & \theta_{23} \\ * & * & \theta_{33} \end{bmatrix} < 0, \quad (26)$$

where symmetric elements are

$$\begin{aligned} \theta_{11} &= \tau^{-1} \hat{\mathbf{A}}^T \mathbf{P} \hat{\mathbf{A}} - \tau^{-1} \mathbf{P} + \tau^{-1} \hat{\mathbf{C}}^T \hat{\mathbf{C}} + \zeta^2 \mathbf{I}, \\ \theta_{12} &= \tau^{-1} \hat{\mathbf{A}}^T \mathbf{P} \hat{\mathbf{B}} + \tau^{-1} \hat{\mathbf{c}}^* \mathbf{D}, \theta_{13} = \tau^{-1} \hat{\mathbf{A}}^T \mathbf{P} + \tau^{-1} \hat{\mathbf{C}}^T \mathbf{E}, \\ \theta_{22} &= \tau^{-1} \hat{\mathbf{B}}^T \mathbf{P} + \tau^{-1} \mathbf{D}^T \mathbf{E}, \theta_{23} \\ &= \tau^{-1} \mathbf{P} + \tau^{-1} \mathbf{E}^T \mathbf{E} - \mathbf{I}. \end{aligned} \quad (27)$$

By applying the Schur complement lemma to equation (26), the following inequality is obtained:

$$\begin{bmatrix} -\tau^{-1} \mathbf{P} + \zeta^2 \mathbf{I} & 0 & 0 & \hat{\mathbf{A}}^T & \hat{\mathbf{C}}^T \\ * & -\tau^{-1} \mathbf{I} & 0 & \hat{\mathbf{B}}^T & \hat{\mathbf{D}}^T \\ * & * & -\mathbf{I} & \mathbf{I} & \hat{\mathbf{E}}^T \\ * & * & * & -\tau \mathbf{P}^{-1} & 0 \\ * & * & * & * & -\tau \mathbf{I} \end{bmatrix} < 0. \quad (28)$$

Lemma, the following results can be obtained:

$$\begin{bmatrix} -\mathbf{Q}^{-1} & 0 & 0 & \hat{\mathbf{A}}^T & \hat{\mathbf{C}}^T & \mathbf{I} \\ * & -\tau^{-1} \mathbf{I} & 0 & \hat{\mathbf{B}}^T & \hat{\mathbf{D}}^T & 0 \\ * & * & -\mathbf{I} & \mathbf{I} & \hat{\mathbf{E}}^T & 0 \\ * & * & * & -\mathbf{Q} & 0 & 0 \\ * & * & * & * & -\tau \mathbf{I} & 0 \\ * & * & * & * & * & -\rho \mathbf{I} \end{bmatrix} < 0. \quad (29)$$

Further multiply (29) on both sides by  $\text{diag}\{\mathbf{Q}, \tau \mathbf{I}, \mathbf{I}, \mathbf{I}, \mathbf{I}, \mathbf{I}\}$  and its transpose, and let  $N_1 = K_1 \mathbf{Q}, N_2 = \tau K_2$ ; then  $\mathbf{A} + \mathbf{B} \alpha \mathbf{K}_1, \hat{\mathbf{B}} = \mathbf{B} \alpha \mathbf{K}_2, \hat{\mathbf{C}} = -\mathbf{C}(\mathbf{A} + \mathbf{B} \alpha \mathbf{K}_1), \hat{\mathbf{D}} = \mathbf{I} - \mathbf{C} \mathbf{B} \alpha \mathbf{K}_2, \hat{\mathbf{E}} = -\mathbf{C}$ . If we put in this equation, we will get

$$\begin{bmatrix} -\mathbf{Q} & 0 & 0 & (\mathbf{A} \mathbf{Q} + \mathbf{B} \alpha N_1)^T & [-\mathbf{C}(\mathbf{A} \mathbf{Q} + \mathbf{B} \alpha N_1)] & \mathbf{Q} \\ * & -\tau \mathbf{I} & 0 & (\mathbf{B} \alpha N_2)^T & (\tau \mathbf{I} - \mathbf{C} \mathbf{B} \alpha N_2)^T & 0 \\ * & * & -\mathbf{I} & \mathbf{I} & (-\mathbf{C})^T & 0 \\ * & * & * & -\mathbf{Q} & 0 & 0 \\ * & * & * & * & -\tau \mathbf{I} & 0 \\ * & * & * & * & * & -\rho \mathbf{I} \end{bmatrix} < 0, \quad (30)$$

and then we obtain  $\alpha = (\mathbf{I} + \alpha_0) \beta$  into this equation and get

$$\mathbf{Z} + \mathbf{X} \bar{\mathbf{E}} \mathbf{Y} + \mathbf{Y}^T \bar{\mathbf{E}} \mathbf{X}^T < 0. \quad (31)$$

Here,

$$\mathbf{Z} = \begin{bmatrix} -\mathbf{Q} & 0 & 0 & (\mathbf{A} \mathbf{Q} + \mathbf{B} \beta N_1)^T & [-\mathbf{C}(\mathbf{A} \mathbf{Q} + \mathbf{B} \beta N_1)] & \mathbf{Q} \\ * & -\tau \mathbf{I} & 0 & (\mathbf{B} \beta N_2)^T & (\tau \mathbf{I} - \mathbf{C} \mathbf{B} \beta N_2)^T & 0 \\ * & * & -\mathbf{I} & \mathbf{I} & (-\mathbf{C})^T & 0 \\ * & * & * & -\mathbf{Q} & 0 & 0 \\ * & * & * & * & -\tau \mathbf{I} & 0 \\ * & * & * & * & * & -\rho \mathbf{I} \end{bmatrix}, \quad (32)$$

$$\mathbf{X} = [0 \ 0 \ 0 \ \mathbf{B}^T \ (-\mathbf{C} \mathbf{B})^T \ 0]^T,$$

$$\bar{\mathbf{E}} = \alpha_0,$$

$$\mathbf{Y} = [\beta N_1 \ \beta N_2 \ 0 \ 0 \ 0 \ 0].$$

According to Lemma 2, it is further obtained that

$$Z + \varepsilon_1 \beta_0^2 \mathbf{X} \mathbf{X}^T + \varepsilon_1^{-1} \mathbf{Y}^T \mathbf{Y} < 0, \quad (33)$$

$$Z + \left[ \varepsilon_1^{(1/2)} \beta_0 \mathbf{X} \quad \varepsilon_1^{-(1/2)} \mathbf{Y}^T \right] \begin{bmatrix} \varepsilon_1^{(1/2)} \beta_0 \mathbf{X}^T \\ \varepsilon_1^{-(1/2)} \mathbf{Y} \end{bmatrix} < 0. \quad (34)$$

Using Schur complement lemma again, equation (34) can be written directly as

$$\begin{pmatrix} Z & \varepsilon_1^Z \beta_0 \mathbf{X} \varepsilon_1^{-(1/2)} \mathbf{Y}^T \\ * & -I \end{pmatrix} < 0. \quad (35)$$

Multiplying (35) on both sides by  $\text{diag}\{\mathbf{I}, \mathbf{I}, \varepsilon_1^{(1/2)} \mathbf{I}, \varepsilon_1^{(1/2)} \mathbf{I}\}$  and its transpose, we get  $\Omega < 0$ .

On the basis of Theorem 2, we further consider the stability of discrete nonlinear repetitive process (14) with uncertainty and design of iterative learning robust fault-tolerant controller. Since Lemma 2 cannot be directly used in the proof process after considering uncertainty, we need to make the following equivalent transformation for Theorem 2. First, we define it as follows:

$$\begin{aligned} \mathbf{R}_1 &= [(\mathbf{A}\mathbf{Q} + \mathbf{B}\beta\mathbf{N}_1) \quad \mathbf{B}\beta\mathbf{N}_2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T, \\ J_1 &= [0 \quad 0 \quad 0 \quad 0 \quad -\mathbf{C}^T \quad 0 \quad 0 \quad 0]^T, \\ &\cdot [I \quad J_1] \begin{bmatrix} \Phi_1 & \mathbf{R}_1 \\ \mathbf{R}_1^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ J_1^T \end{bmatrix} < 0. \end{aligned} \quad (36)$$

If the matrix  $\Gamma = \begin{bmatrix} \Phi_1 & \mathbf{R}_1 \\ \mathbf{R}_1^T & 0 \end{bmatrix}$ ,  $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ , and  $\Lambda = \mathbf{I}$ ,  $\Sigma = [J_1^T \quad -I]$ , using Lemma 3, the following inequality can be obtained:

$$\begin{bmatrix} \Phi_1 & \mathbf{R}_1 \\ \mathbf{R}_1^T & 0 \end{bmatrix} + \text{sym} \left\{ \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} [J_1^T \quad -I] \right\} < 0. \quad (37)$$

Here,

$$\mathbf{W}_1 = [W_{11}^T \quad W_{12}^T \quad W_{13}^T \quad W_{14}^T \quad W_{15}^T \quad W_{16}^T \quad W_{17}^T \quad W_{18}^T]^T. \quad (38)$$

Similarly, the above formula is sorted out in the same way again as follows:

$$\mathbf{R}_2 = [0 \quad 0 \quad 0 \quad 0 \quad (-\varepsilon_1 \beta_0 \mathbf{C})^T \quad 0 \quad 0 \quad 0 \quad 0]^T, \quad (39)$$

$$J_2 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \mathbf{B} \quad 0 \quad 0]^T,$$

and make  $\Gamma = \begin{bmatrix} \Phi_2 & \mathbf{R}_2 \\ \mathbf{R}_2^T & 0 \end{bmatrix}$ ,  $W = \begin{bmatrix} W_2 \\ W_3 \end{bmatrix}$ , and  $\Sigma = [J_2^T \quad -I]$ .

**Theorem 3.** *Considers the case where system (1) has uncertainty, where it is false*

$$\Psi = \begin{bmatrix} \Phi_2 & \mathbf{R}_2 \\ \mathbf{R}_2^T & 0 \end{bmatrix} + \text{sym} \left\{ \begin{bmatrix} W_2 \\ W_3 \end{bmatrix} [J_2^T \quad -I] \right\} < 0. \quad (40)$$

In which,  $W_3 = [W_{31}^T \quad W_{32}^T \quad \dots \quad \dots \quad W_{39}^T \quad W_{40}^T]^T$  Theorem 3 (1) the uncertainty of the system, including false set is  $\Delta A = H1 \Xi F1$ ,  $\Delta B = H1 \Xi F2$ ,  $\Delta C = H2 \Xi F2$ , and if there is

a positive definite matrix  $Q > 0$ , matrix  $N1, N2$  and scalar  $\tau > 0$ ,  $\Lambda > 0$ ,  $\varepsilon > 0$  make matrix inequality  $Y < 0$  as follows:

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ * & \mathbf{Y}_{22} \end{bmatrix} < 0, \\ \mathbf{Y}_{11} &= \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ * & \Psi_{22} \end{bmatrix}, \\ \mathbf{Y}_{12} &= [\Psi_{31} \quad \Psi_{32}], \\ \mathbf{Y}_{22} &= \text{diag}\{-\lambda \mathbf{I} - \lambda \mathbf{I} - \lambda \mathbf{I} - \lambda \mathbf{I} - \lambda \mathbf{I} - \lambda \mathbf{I}\}. \end{aligned} \quad (41)$$

The discrete repetition process (14) was stable under the condition of failure tolerance (7) and iterative learning robust fault-tolerant control law (11), and the modified update quantity (13) was the gain matrix  $\mathbf{K}_1 = \mathbf{N}_1 \mathbf{Q}^{-1}$ ,  $\mathbf{K}_2 = \tau^{-1} \mathbf{N}_2$ .

The proof of formula (40) can be transformed into, when the system uncertainty is further considered.  $\mathbf{Y}_1 + \mathbf{M} \Xi \mathbf{N} + \mathbf{N}^T \Xi \mathbf{M}^T < 0$ .

Here,

$$\begin{aligned} \mathbf{Y}_1 &= \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ * & \mathbf{Y}_{22} \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{H}_1^T & 0 & 0 & 0 & 0 & \mathbf{H}_1^T & 0 \\ 0 & 0 & 0 & 0 & \mathbf{H}_2^T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{F}_2 & 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{N} &= \Psi_{32}^T. \end{aligned} \quad (42)$$

Using Lemma 2 and conditions  $\Xi \Xi^T \leq I$  or less if and only if there is  $\lambda > 0$  makes

$$\mathbf{Y}_1 + \lambda \mathbf{M} \mathbf{M}^T + \lambda^{-1} \mathbf{N}^T \mathbf{N} < 0. \quad (43)$$

If true, the above equation can be written as

$$\mathbf{Y}_1 + \left[ \lambda^{(1/2)} \mathbf{M} \quad \lambda^{-(1/2)} \mathbf{N}^T \right] \begin{bmatrix} \lambda^{(1/2)} \mathbf{M}^T \\ \lambda^{-(1/2)} \mathbf{N} \end{bmatrix} < 0. \quad (44)$$

Using Schur complement lemma, the above equation can be rewritten as

$$\begin{bmatrix} \mathbf{Y}_1 & \left[ \lambda^{(1/2)} \mathbf{M} \quad \lambda^{-(1/2)} \mathbf{N}^T \right] \\ * & -I \end{bmatrix} < 0. \quad (45)$$

Finally, multiplying equation (45) on both sides by  $\text{diag}\{\mathbf{I}, \mathbf{I}, \lambda^{(1/2)} \mathbf{I}, \lambda^{(1/2)} \mathbf{I}\}$ , we could obtain  $Y < 0$ .

## 5. Control Algorithm

The control algorithm for the proposed method is described below in the form of step-by-step algorithm implementation process as shown in Figure 1 as a flow diagram.

*Step 1.* Obtain the model parameters of the controlled object and transform them into a discrete time-invariant system as shown in equation (1). If there is a solution, then go to the next process and adopt gain and learning laws applied. If



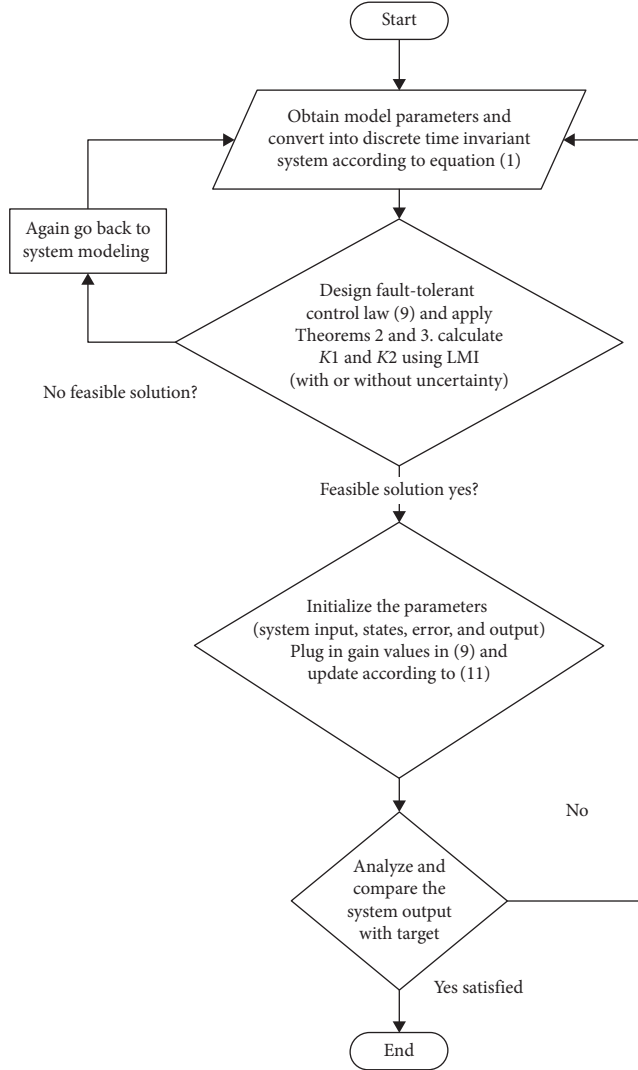


FIGURE 1: Control and design of algorithm step-by-step process.

there is no solution, then obviously, we need to calculate again system parameters accordingly.

*Step 2.* Design iterative learning fault-tolerant control law in the form of formula (11), and apply LMI toolbox in MATLAB to solve Theorems 2 or 3 according to the parameters given in formula (1), and obtain the parameters  $K_1$  and  $K_2$  of the system with and without uncertainty in formula (1), respectively. If there is a feasible solution, proceed to the next step; otherwise, redesign.

*Step 3.* According to the given reference target and system of the initial boundary conditions, the system of Equation (1) is taken after the initial iteration value, the state of the output and error when the initial iteration ended. And then, in Step 2 to solve the parameters  $K_1$  and  $K_2$  plug type (9) in the form of iterative learning control law, according to the type (11) and iterative learning control in the form of reconsideration, iterative learning controller is constantly modified.

*Step 4.* Observe the output response curve of the system, compare it with the expectation curve of a given reference target, and draw the root-mean-square (RMS) error curve. If the system output can track the expectation curve with a small error after several batches, the calculation method will be finished. Otherwise, if the tracking error is too large or the output response curve fails to track the expected curve after multiple batches, the controller shall be redesigned and Step 2 shall be returned.

## 6. Practical Applications of the Proposed Controller

*6.1. Practical Simulation Example 1.* To verify the validity of the method in this paper, the injection velocity control process of the reinjection molding process was taken as the research object [67]:

$$\begin{cases} x(t+1, k) = (A + \Delta A)x(t, k) + (B + \Delta B)u(t, k) + f(x(t, k)), \\ y(t, k) = (C + \Delta C)x(t, k). \end{cases} \quad (46)$$

Among them, system moment matrix is as follows:

$$\begin{aligned} A &= \begin{bmatrix} 1.582 & -0.5916 \\ 1 & 0 \end{bmatrix}, \\ B &= [1 \ 0]^T, \\ C &= [1.69 \ 1.419]. \end{aligned} \quad (47)$$

The expected trajectory is defined as

$$y_d(t) = \begin{cases} \sin(0.01\pi t), & 1 \leq t < 200, \\ 1, & 200 \leq t < 300, \\ 4 - 0.01t, & 300 \leq t \leq 400. \end{cases} \quad (48)$$

At time  $t \in [0, 400]$ , the initial state of each batch is 0. Suppose that the system's operating frequency norm  $[0, \infty]$  is divided into  $[0, \infty] = [0, 6] \cup [6, 25] \cup [25, \infty]$ . After the 20th iteration, the occurrence actuator's fault suddenly appears, and the fault in the system is  $f(t) = 0.7 + 0.3 \times \sin(0.5t)$ . Evaluate performance and introduce performance indicators:

$$\text{RMS}(e(t, k)) = \sqrt{\frac{1}{400} \sum_{t=1}^{400} e^2(t, k)}. \quad (49)$$

Supposing that the coefficients of matrices  $A$  and  $B$  contain uncertainties, the decomposition of the uncertainty is carried out as follows: if the uncertainty matrix of the system  $H_1 = [-0.05, 0.1]$ ,  $H_2 = [-0.12, 0.15]$ ,  $F_1 = I$ ,  $F_2 = [1, 1]^T$ ,  $\Xi = \text{diag}\{\sigma_1, \sigma_2\}$ , and  $s_1$  and  $s_2$  are  $-1 \sim 1$ , then the gain matrix of the fault-tolerant controller is  $K_1 = [-1.5749, 0.7293]$  and  $K_2 [0.1291]$ .

Suppose the injection molding process is assumed to work in the intermediate frequency segment. In that case, the system RMS error's simulation curves and the number of overlaps are shown in Figures 2 and 3. By comparing the iterative fault-tolerant control results, it can be seen that the fault-tolerant control results designed within the specific domain of the system can accelerate the convergence speed

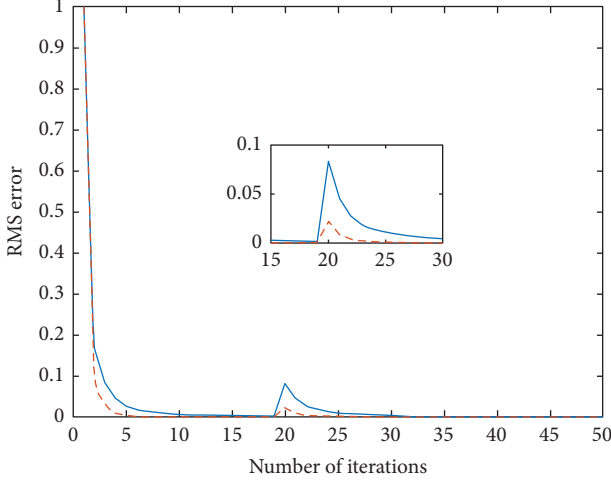


FIGURE 2: RMS error with and without uncertainty.

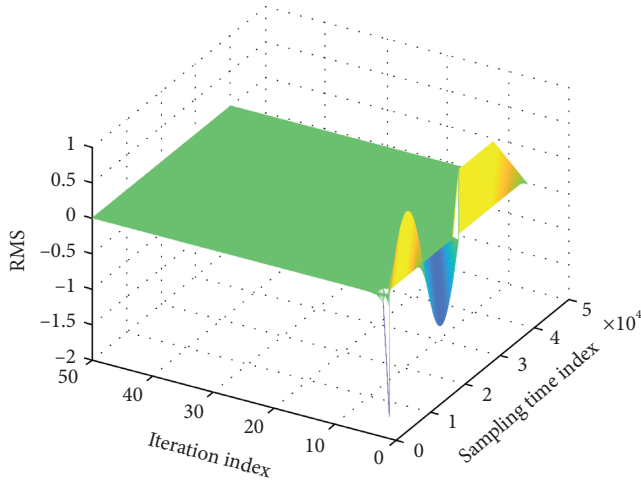


FIGURE 3: RMS error in time and iteration domain.

to some extent and can better restrain the influence of the operator's fault. The controller's uncertainty and fault-tolerant capability are quite robust, limiting the text and not be repeated. As shown in Figure 2, the system tracking error decreases quickly with the increase of iteration time before the actuator fails. After the actuator failed in iteration 20, the system's tracking performance decreased significantly. However, under the control of the fault-tolerant controller, with the increasing number of iterations, the system recovered quickly and steadily, indicating that the system had a good follow-up effect under normal and actuator failure conditions.

The simulation outcomes and effects of the control law are shown in Figures 2 and 3. Even with nonrepetitive disturbances, the tracking error of the first 20 batches of the system before the failure occurred quickly converged to a stable state with respect to time, and the tracking performance continued to improve in the batch direction, as shown in Figure 2, in the 25th batch of process failures. After it happened, the tracking performance of the system was

lower than that of the 25th batch. However, after several batch iterations, the tracking performance reached an ideal level again, as shown in Figure 3. This conclusion is reflected in Figures 2 and 3. Figure 3 shows the daily values of different batches, and Figure 4 shows the tracking error of the system in a three-dimensional graph.

**6.2. Practical Simulation Example 2.** In order to verify the effectiveness of the algorithm proposed in this paper, the single-link manipulator that performs the motion trajectory tracking task repeatedly is considered as the experimental object. The continuous time model of the single-link manipulator system is described as [68]

$$\tau(t) = Mt^2 \frac{d^2\theta(t)}{dt^2} + V \frac{d\theta(t)}{dt} + M_g l \cos(\theta(t)), \quad (50)$$

where  $\theta(t)$  is manipulator joints of angular displacement,  $\tau(t)$  is the driving moment of joints,  $l = 0.6$  m represents the length of the manipulator rod,  $m = 0.6$  kg is the quality of the manipulator,  $V = 1.5$  (kg/m<sup>2</sup>/s) is the mechanical arm joint friction resistance coefficient, and  $g = 9.8$  (m/s<sup>2</sup>) for the acceleration of gravity.

Through Euler's method, discrete sampling is carried out for the continuous system (34) at time interval  $h = 0.2$  s, and  $T$  represents the discrete sampling point. At the same time,  $x_1(t, k) = (ht, k)$  and  $x_2(t, k) = (ht + h, k)$ , respectively, represent the state variable of the  $k$  batch of the repetitive operation of the single-link decentralization system,  $u(t, k) = (hT, k)$  as the control input,  $y(T, k) = (ht + h, k)$  as the system output, and the discrete time-invariant nonlinear model of the single-link system (50) can be represented as

$$\begin{aligned} x_1(t+1, k) &= x_2(t, k) x_2(t+1, k) \\ &= \left(2 - \frac{vh}{Mt^2}\right) x_2(t, k) + \left(\frac{vh}{Mt^2} - 1\right) x_1(t, k) \\ &\quad - \frac{gh^2}{l} \cos(x_1(t, k)) + \frac{h^2}{Mt^2}. \end{aligned} \quad (51)$$

A discrete time-invariant nonlinear model consistent with the form of equation (1) can be further obtained:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 \\ \frac{Vh}{Mt^2} - 1 & 2 - \frac{Vh}{Mt^2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.39 & 0.61 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 \\ \frac{h^2}{Mt^2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.19 \end{bmatrix}, \end{aligned} \quad (52)$$

$$C = [0 \ 1].$$

The nonlinear function is

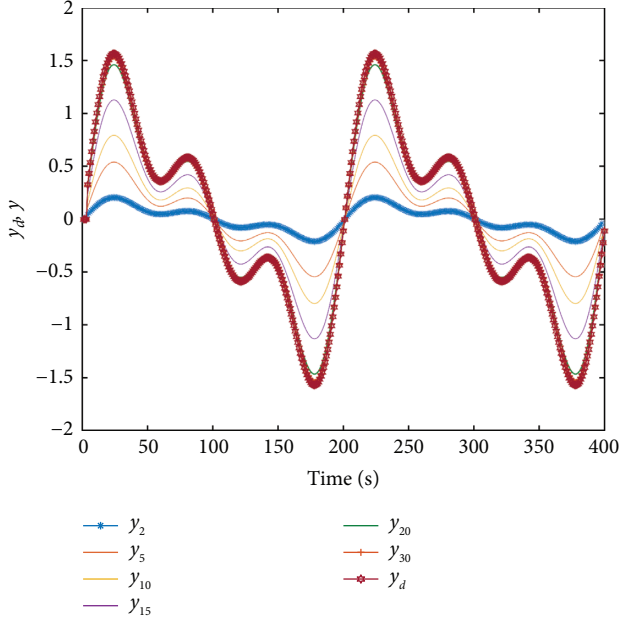


FIGURE 4: Output versus reference trajectory curve without actuator failure.

$$f(x(t, k)) = \begin{bmatrix} 0 \\ \frac{gh^2}{l} \end{bmatrix} \cos(x_1(t, k)) = \begin{bmatrix} 0 \\ 0.653 \end{bmatrix} \cos(x_1(t, k)). \quad (53)$$

Because  $\cos(x_1(t, k))$  has upper and lower bounds, so for nonlinear function,  $f(x(t, k)) = \begin{bmatrix} 0 \\ 0.653 \end{bmatrix} \cos(x_1(t, k))$  satisfies the nonlinear constraint (2). At the same time, it is assumed that the manipulator needs to run 100 batches repeatedly and track the reference trajectory:

$$y_d(t) = \sin\left(\frac{\pi t}{100}\right) + 0.5 \sin\left(\frac{\pi t}{200}\right) + 0.5 \sin\left(\frac{\pi t}{300}\right). \quad (54)$$

Initially, we can assume that our system's initial condition is zero like input  $u(t, k) = (0, 0)$  states  $x_1(t, k) = (0, 0)$  and  $x_2(t, k) = (0, 0)$  and output  $y(T, k) = (0, 0)0$ , respectively. Due to the repeated operation of the industrial application of mechanical arm often, we assume that the 50th connecting rod will be shown after the repeat problem such as mechanical wear. The failure occurs after each batch of the intermittent fault happens in the process, and the failure changes in each iteration with respect to time. And the manipulator joints' input driving moment will change with fault time-varying characteristics. The typical case of this fault is the gear transmission part of the manipulator; after a long period of operation, there will be missing teeth, broken teeth, and other phenomena, resulting in the system in each operating batch of the cycle time time-varying failure. Therefore, it is assumed that the system input is an actuator fault in the form of equation (3), and the fault coefficient =  $(0.15 + 0.85 \sin(0.5t))$ .

Assuming that there is no uncertainty, the gain  $K1 = [-1.873 \ 2 -4.118 \ 6]$  and  $K2 = 0.5673$  of the modified update quantity (13) in the iterative learning fault-tolerant controller (11) is solved according to Theorem 2 and LMI tool.

In addition, it is noted that, in the manipulator system in actual operation, the joint frictional resistance coefficient  $V$ , rod length  $L$ , and mass  $M$  in matrices  $A$  and  $B$  will become uncertain due to wear and other factors, so the variation of these uncertainty factors should be considered in the controller design process. Assuming that the coefficients of matrices  $A$  and  $B$  contain uncertainties, the uncertainty decomposition is performed as follows:

$$\begin{aligned} H_1 &= \begin{bmatrix} 0 & 0 \\ -0.05 & 0.1 \end{bmatrix}, \\ H_2 &= [0 \ 0], \\ F_1 &= I, \\ F_2 &= [0 \ 1]T, \end{aligned} \quad (55)$$

where  $\Xi = \text{diag}\{\sigma_1, \sigma_2\}$ ,  $\sigma_1$  and  $\sigma_2$  are  $-1$  to  $1$  in between numerical values.  $\Delta A = H_1 \Xi F_1 = \begin{bmatrix} 0 & 0 \\ -0.05\sigma_1 & 0.1\sigma_2 \end{bmatrix}$ ,

$$\Delta B = H_1 \Xi F_2 = \begin{bmatrix} 0 \\ 0.1\sigma_2 \end{bmatrix}, \text{ and } \Delta C = H_2 \Xi F_2 = 0.$$

At this point, according to the conclusion of Theorem 3, we know that the system of the gain of modified update quantity (13) in iterative learning robust fault-tolerant controller (11) can be obtained by using LMI tool,  $K_1 = [-2.4149 \ -4.6484]$  and  $K_2 = 1.9840$ .

In order to evaluate the control effect of each batch of manipulator during repeated operation, the RMS value of tracking error  $E(t, k)$  was defined as the tracking energy index

$$\text{RMS}(e(t, k)) = \sqrt{\frac{1}{400} \sum_{i=1}^{400} e^2(t, k)}. \quad (56)$$

In the formula, the smaller RMS ( $E(t, k)$ ) is, the better the tracking effect of lot  $K$  is. The tracking control simulation results of the single-link manipulator system under the nondeterministic conditions are shown in Figures 4–6, respectively.

It can be clearly observed that the output of the system is trying to track the reference tracking trajectory of the system as shown in Figure 4. As can be seen from Figure 5, the tracking error of the system decreases rapidly with the increase of iteration times before the actuator fails. After the failure of the actuator in batch 50, the tracking performance of the system decreased significantly. However, under the action of the fault-tolerant controller, with the increasing number of iterations, the system collected again rapidly and stably, indicating that the system achieved a good tracking effect under both normal and failure conditions of the actuator. Figure 4 shows the system output curve and

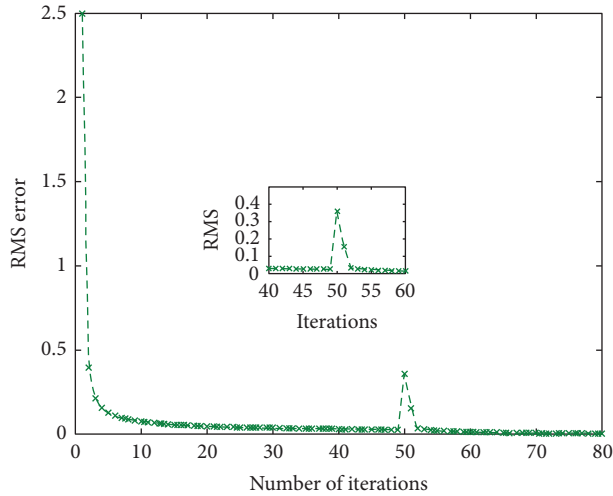


FIGURE 5: RMS error curve without actuator failure.

expectation curve before the actuator failure, and the three-dimensional figure of Figure 6 represents the tracking error change of the system in two dimensions of the batch axis and time axis. It can be seen that, with the increase of the number of iterations, the actual output curve of the manipulator can gradually track the upward periodic curve. Correspondingly, the simulation results of robust fault-tolerant control when the system is uncertain are shown in Figures 7–9, respectively.

The simulation results are shown in Figures 4 and 5. Even if the disturbance is repeated, the tracking error of the first 50 iterations of the system before the fault occurs rapidly converges to a stable state with time, and the tracking performance keeps improving in the direction of the batches, as shown in Figure 7. After the failure of the first batch, the tracking performance of the system decreased compared with that of the 50th batch. However, after several batch iterations, the tracking performance reached an ideal level again, as shown in Figure 8. This conclusion is shown in Figures 8 and 9, which show the regular values of different batches of iterations, and Figure 9, which shows the tracking error of the system in 3D. It can be observed that under the effect of time-varying failure and repetitive disturbance, and actuator failure, the proposed algorithm in the normal situation and therefore failure can guarantee system in both iteration and time direction monotone convergent. According to the asymptotic stability, a designed controller has fault tolerance, and fault tolerance of both performed at the same time and has a robust inhibitory effect on non-recurring interference.

Figures 4–9 shows that, under the use of time-varying faults and nonrepetitive disturbances, the proposed algorithm can guarantee monotonic convergence of the system in both batch and time directions under normal and fault scenarios, and it is asymptotically stable. The designed controller has both fault-tolerant and fault-tolerant performances and has a robust suppression effect on non-repetitive external interference.

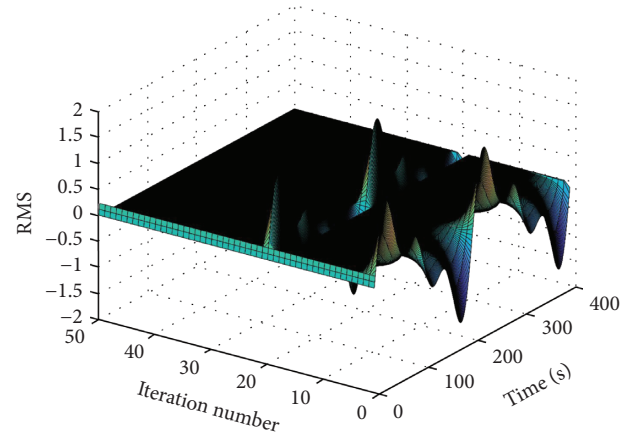


FIGURE 6: Error trajectory in both time and iteration domain (without actuator failure).

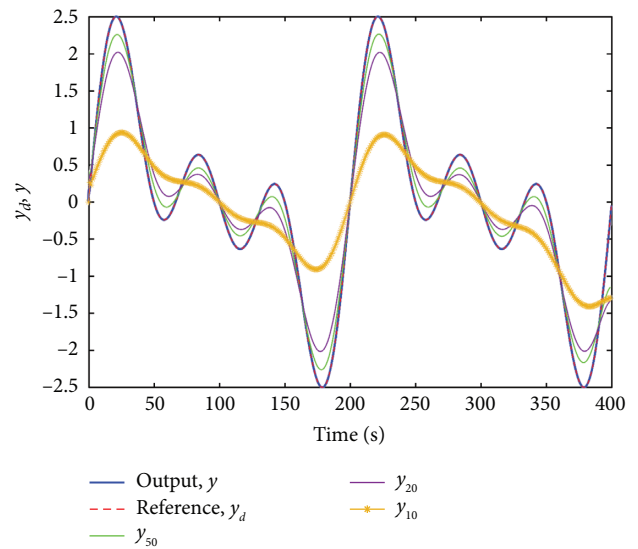


FIGURE 7: Output versus reference trajectory curve with actuator failure.

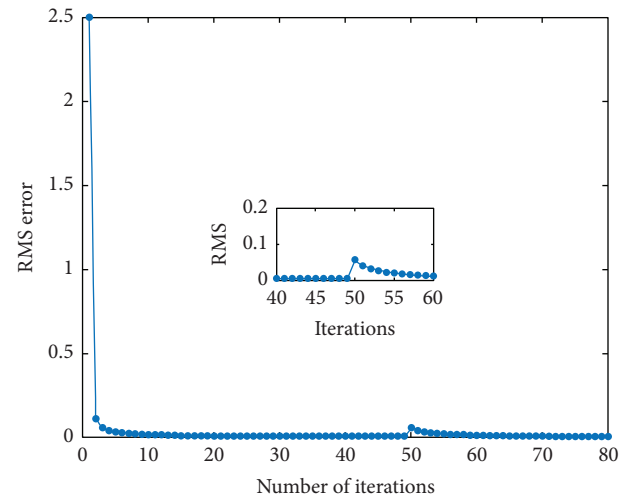


FIGURE 8: RMS error curve with actuator failure.

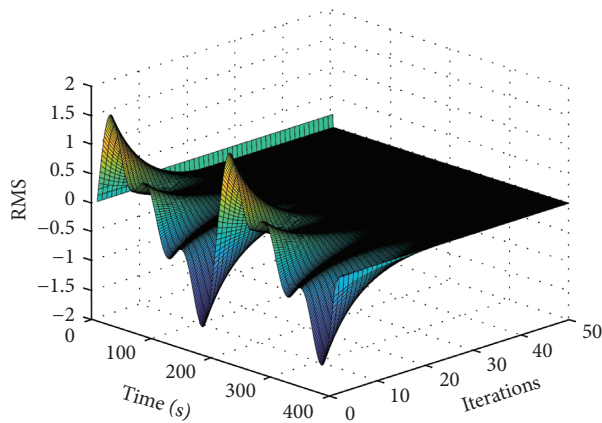


FIGURE 9: Error trajectory in both time and iteration domain (with actuator failure).

## 7. Conclusion

The iterative learning fault-tolerant control algorithm for a class of uncertain discrete time-invariant nonlinear systems with actuator faults is studied in this paper. First, the sufficient conditions for the stability of the nonlinear discrete repetition process are obtained by the Lyapunov method, the gain of the iterative learning fault-tolerant controller is solved by LMI, and the robustness of the system with uncertainty is discussed. Finally, the feasibility of the proposed algorithm is verified by the simulation experiment of the single-link manipulator as well as the electromechanical system for batch process. The method proposed in this paper is a further extension of the iterative learning control method for linear repeating process, which is obviously more adaptable and effective for the repeated operation process with nonlinear links in the industrial process.

The proposed algorithm improves the error identification and estimation of nonlinearly unstable systems, enhances error detection reliability, and reduces false alarms. The faults of nonlinear models based on iterative learning are more up-front and more suitable for technical applications. Based on the method proposed in this paper, we can also use it to identify and estimate uncertain nonlinear systems. The future applications of this proposed algorithm are as follows:

- (i) Aircraft actuator fault-tolerant control system
- (ii) Robotics (soft robotics, elastic actuators, and omnivheel robots) trajectory control
- (iii) Robustification in PMSM motor servo applications
- (iv) Nonlinear servo speed and position control system
- (v) Piezoelectric position and force tracking controls

## Data Availability

The data are available and can be provided upon request. The simulation examples are taken from two sources: (1) J. Shi, F. Gao, and T. J. Wu, "Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system," *Journal of Process Control*, vol. 15, no. 8, pp. 907–924, 2005; (2) D.-H. Hwang, Z. Bien, and S.-R.

Oh, "Iterative learning control method for discrete-time dynamic systems," in *IEE Proceedings D-Control Theory and Applications*, 1991, vol. 138, no. 2, pp. 139–144: IET.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Hamidaoui, C. Shao, and S. Haouassi, "A PD-type iterative learning control algorithm for one-dimension linear wave equation," *International Journal of Control, Automation and Systems*, vol. 18, no. 4, pp. 1045–1052, 2020.
- [2] L. Zhou, L. Jia, Y.-L. Wang, D. Peng, and W. Tan, "An integrated robust iterative learning control strategy for batch processes based on 2D system," *Journal of Process Control*, vol. 85, pp. 136–148, 2020.
- [3] S. Riaz, H. Lin, M. Bilal Anwar, and H. Ali, "Design of PD-type second-order ILC law for PMSM servo position control," *Journal of Physics: Conference Series*, vol. 1707, Article ID 012002, 2020.
- [4] Z. Li, Y. Hu, and D. Li, "Robust design of feedback feed-forward iterative learning control based on 2D system theory for linear uncertain systems," *International Journal of Systems Science*, vol. 47, no. 11, pp. 2620–2631, 2016.
- [5] S. Riaz, H. Lin, and H. Elahi, "A novel fast error convergence approach for an optimal iterative learning controller," *Integrated Ferroelectrics*, vol. 213, no. 1, pp. 103–115, 2020.
- [6] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.
- [7] Y. Wang, J. Shi, D. Zhou, and F. Gao, "Iterative learning fault-tolerant control for batch processes," *Industrial & Engineering Chemistry Research*, vol. 45, no. 26, pp. 9050–9060, 2006.
- [8] B. Pang and C. Shao, "PID-type parameter optimal iterative learning control algorithm based on singular value decomposition [J]," *Information and Control*, vol. 43, no. 4, pp. 483–489, 2014.
- [9] S. Riaz, H. Lin, and M. P. Akhter, "Design and implementation of an accelerated error convergence criterion for norm optimal iterative learning controller," *Electronics*, vol. 9, no. 11, p. 1766, 2020.
- [10] S.-L. Xie, S.-P. Tian, and Z. Xie, *Iterative Learning Control Theory and Application*, Science Press, Beijing, China, 2005.
- [11] C. Freeman and Y. Tan, *Iterative Learning Control and Repetitive Control*, Taylor & Francis, Oxfordshire, UK, 2011.
- [12] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 590–600, 2010.
- [13] Z. Gao, X. Liu, and M. Z. Chen, "Unknown input observer-based robust fault estimation for systems corrupted by partially decoupled disturbances," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2537–2547, 2015.
- [14] D. Rotondo, F. Nejjari, V. Puig, and J. Blesa, "Model reference FTC for LPV systems using virtual actuators and set-membership fault estimation," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 5, pp. 735–760, 2015.
- [15] B. Zhao, D. Liu, and Y. Li, "Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems," *Information Sciences*, vol. 384, pp. 21–33, 2017.

- [16] L. Liu, Z. Wang, and H. Zhang, "Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 299–313, 2016.
- [17] X. Li, D. Lu, G. Zeng, J. Liu, and W. Zhang, "Integrated fault estimation and non-fragile fault-tolerant control design for uncertain Takagi-Sugeno fuzzy systems with actuator fault and sensor fault," *IET Control Theory & Applications*, vol. 11, no. 10, pp. 1542–1553, 2017.
- [18] J. A. Guzmán-Rabasa, F. R. López-Estrada, B. M. González-Contreras, G. Valencia-Palomo, M. Chadli, and M. Perez-Patricio, "Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system," *Measurement and Control*, vol. 52, no. 9-10, pp. 1228–1239, 2019.
- [19] X. Nian, W. Chen, X. Chu, and Z. Xu, "Robust adaptive fault estimation and fault tolerant control for quadrotor attitude systems," *International Journal of Control*, vol. 93, no. 3, pp. 725–737, 2020.
- [20] B. Luo, Y. Yang, and D. Liu, "Adaptive  $Q^*$ -learning for data-based optimal output regulation with experience replay," *IEEE Transactions on Cybernetics*, vol. 48, no. 12, pp. 3337–3348, 2018.
- [21] M. Schmid, E. Gebauer, C. Hanzl, and C. Endisch, "Active model-based fault diagnosis in reconfigurable battery systems," *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 2584–2597, 2020.
- [22] C. Edwards and C. P. Tan, "Sensor fault tolerant control using sliding mode observers," *Control Engineering Practice*, vol. 14, no. 8, pp. 897–908, 2006.
- [23] S. Du and S. Razavi, "Fault-tolerant control of variable speed limits for freeway work zone using likelihood estimation," *Advanced Engineering Informatics*, vol. 45, p. 101133, 2020.
- [24] G. Xiao and F. Liu, "Observer-based cooperative distributed fault-tolerant model predictive control with imperfect network communication and asynchronous measurements," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 12, pp. 4531–4549, 2020.
- [25] X. Feng and Y. Wang, "Fault estimation based on sliding mode observer for Takagi-Sugeno fuzzy systems with digital communication constraints," *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 569–588, 2020.
- [26] G. Li, S. J. Qin, Y. Ji, and D. Zhou, "Reconstruction based fault prognosis for continuous processes," *Control Engineering Practice*, vol. 18, no. 10, pp. 1211–1219, 2010.
- [27] S. Riaz, H. Elahi, K. Javaid, and T. Shahzad, "Vibration feature extraction and analysis for fault diagnosis of rotating machinery—a literature survey," *Asia Pacific Journal of Multi-disciplinary Research*, vol. 5, 2017.
- [28] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique," *Computers & Chemical Engineering*, vol. 134, Article ID 106697, 2020.
- [29] Y. Zhao, T. Li, X. Zhang, and C. Zhang, "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: advantages, challenges and the future," *Renewable and Sustainable Energy Reviews*, vol. 109, pp. 85–101, 2019.
- [30] B. Jiang and M. Staroswiecki, "Adaptive observer design for robust fault estimation," *International Journal of Systems Science*, vol. 33, no. 9, pp. 767–775, 2002.
- [31] S. He and F. Liu, "Adaptive observer-based fault estimation for stochastic Markovian jumping systems," *Abstract and Applied Analysis*, vol. 2012, Article ID 176419, 11 pages, 2012.
- [32] H. Li, F. You, and F. Wang, "A delay dependent approach to robust fast adaptive fault estimation design for uncertain neutral systems with time delay," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 8, pp. 2579–2588, 2018.
- [33] M. Zhou, Z. Wang, and Y. Shen, "Simultaneous fault estimation and fault-tolerant tracking control for uncertain nonlinear discrete-time systems," *International Journal of Systems Science*, vol. 48, no. 7, pp. 1367–1379, 2017.
- [34] J. Sahng, M.-Y. Chen, and D.-H. Zhou, "Incipient fault detection using transformed component statistical analysis," *Journal of Shanghai Jiaotong University*, vol. 6, p. 11, 2015.
- [35] Á. P. Ò, "¿ itüä Ð í äHoo Hoo," *Acta Automatica Sinica*, vol. 36, no. 12, 2010.
- [36] L. Wang, H. Pan, J. Gao, and D. Chen, "Predictive compensation for wireless networked system with time delay and packet dropout based on TS model," in *Abstract and Applied Analysis*, vol. 2014, Article ID 953039, 8 pages, 2014.
- [37] S. Riaz, H. Lin, and H. Elahi, "A novel fast error convergence approach for an optimal iterative learning controller," *Integrated Ferroelectrics*, vol. 213, 2021.
- [38] J. Han, H. Zhang, Y. Wang, and K. Zhang, "Fault estimation and fault-tolerant control for switched fuzzy stochastic systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2993–3003, 2018.
- [39] F. Li, Z. Hu, and G. Zhao, "Fault estimation and adaptive fault tolerant control for dynamic systems based on the second-order sliding mode observer," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 230, no. 3, pp. 222–230, 2016.
- [40] Z. Hu, G. Zhao, L. Zhang, and D. Zhou, "Fault estimation for nonlinear dynamic system based on the second-order sliding mode observer," *Circuits, Systems, and Signal Processing*, vol. 35, no. 1, pp. 101–115, 2016.
- [41] B. Jiang, V. Cocquempot, and M. Staroswiecki, "Fault estimation in nonlinear uncertain systems using robust/sliding-mode observers," *IEE Proceedings—Control Theory and Applications*, vol. 151, no. 1, pp. 29–37, 2004.
- [42] F. Haouari, M. Djemai, and B. Cherki, "Sliding mode observers for TS fuzzy systems with application to sensor fault estimation," in *Proceedings of the 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, pp. 1–5, IEEE, Tlemcen, Algeria, May 2015.
- [43] Z. Ming-Yue, L. He-Ping, L. Zhi-Jun, S. De-Hui, and L. Ke-Ping, "Fault tolerant control for networked control systems with access constraints," *Acta Automatica Sinica*, vol. 38, no. 7, pp. 1119–1126, 2012.
- [44] K. Zhao, P. Li, C. Zhang, X. Li, J. He, and Y. Lin, "Sliding mode observer-based current sensor fault reconstruction and unknown load disturbance estimation for PMSM driven system," *Sensors*, vol. 17, no. 12, p. 2833, 2017.
- [45] J. Mackerle, "Finite element modelling of ceramics and glass, an addendum—a bibliography (1998–2004)," *Engineering Computations*, vol. 22, no. 3, 2005.
- [46] L. Wang, F. Liu, J. Yu, P. Li, R. Zhang, and F. Gao, "Iterative learning fault-tolerant control for injection molding processes against actuator faults," *Journal of Process Control*, vol. 59, pp. 59–72, 2017.
- [47] D. Rotondo, F.-R. López-Estrada, F. Nejjari, J.-C. Ponsart, D. Theilliol, and V. Puig, "Actuator multiplicative fault estimation in discrete-time LPV systems using switched observers," *Journal of the Franklin Institute*, vol. 353, no. 13, pp. 3176–3191, 2016.

- [48] D. Du, S. Xu, and V. Cocquempot, "Actuator fault estimation for discrete-time switched systems with finite-frequency," *Systems & Control Letters*, vol. 108, pp. 64–70, 2017.
- [49] H. Elahi, "The investigation on structural health monitoring of aerospace structures via piezoelectric aeroelastic energy harvesting," *Microsystem Technologies*, pp. 1–9, 2020.
- [50] H. Elahi, M. Eugeni, F. Fune et al., "Performance evaluation of a piezoelectric energy harvester based on flag-flutter," *Micromachines*, vol. 11, no. 10, p. 933, 2020.
- [51] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control system: a new type servo system for periodic exogenous signals," *IEEE Transactions on Automatic Control*, vol. 33, no. 7, pp. 659–668, 1988.
- [52] M. IKEDA and M. TAKANO, "Repetitive control for systems with nonzero relative degree," *Transactions of the Society of Instrument and Control Engineers*, vol. 24, no. 6, pp. 575–582, 1988.
- [53] H. Elahi, K. Munir, M. Eugeni, S. Atek, and P. Gaudenzi, "Energy harvesting towards self-powered IoT devices," *Energies*, vol. 13, no. 21, p. 5528, 2020.
- [54] Y. Zhang and J. Jiang, "Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems," *IFAC Proceedings Volumes*, vol. 39, no. 13, pp. 1437–1448, 2006.
- [55] W. Limin, C. Xi, and G. Furong, "An LMI method to robust iterative learning fault-tolerant guaranteed cost control for batch processes," *Chinese Journal of Chemical Engineering*, vol. 21, no. 4, pp. 401–411, 2013.
- [56] L. Wang, S. Mo, D. Zhou, F. Gao, and X. Chen, "Robust delay dependent iterative learning fault-tolerant control for batch processes with state delay and actuator failures," *Journal of Process Control*, vol. 22, no. 7, pp. 1273–1286, 2012.
- [57] J. Wang, Y. Li, G. Hu, and M. Yang, "Lightweight research in engineering: a review," *Applied Sciences*, vol. 9, no. 24, p. 5322, 2019.
- [58] X. Bu, F. Yu, Z. Hou, and H. Yang, "Robust iterative learning control for nonlinear systems with measurement disturbances," *Journal of Systems Engineering and Electronics*, vol. 23, no. 6, pp. 906–913, 2012.
- [59] B. Cichy, K. Galkowski, and E. Rogers, "2D systems based robust iterative learning control using noncausal finite-time interval data," *Systems & Control Letters*, vol. 64, pp. 36–42, 2014.
- [60] L. Wang, S. Mo, H. Qu, D. Zhou, and F. Gao, " $H_\infty$  design of 2D controller for batch processes with uncertainties and interval time-varying delays," *Control Engineering Practice*, vol. 21, no. 10, pp. 1321–1333, 2013.
- [61] M. French and E. Rogers, "Non-linear iterative learning by an adaptive Lyapunov technique," *International Journal of Control*, vol. 73, no. 10, pp. 840–850, 2000.
- [62] M. Boski and W. Paszke, "Design of robust iterative learning control schemes for systems with polytopic uncertainties and sector-bounded nonlinearities," *Journal of Physics: Conference Series*, vol. 783, no. 1, p. 012032, 2017.
- [63] P. V. Pakshin, J. P. Emelianova, M. A. Emelianov, K. Galkowski, and E. Rogers, "Stochastic stability of some classes of nonlinear 2D systems," *Automation and Remote Control*, vol. 79, no. 1, pp. 89–102, 2018.
- [64] S. Boyd, *Linear Matrix Inequalities in System and Control Theory*, Siam, Philadelphia, PA, USA, 1994.
- [65] L. Xie, "Output feedback  $H_\infty$  control of systems with parameter uncertainty," *International Journal of Control*, vol. 63, no. 4, pp. 741–750, 1996.
- [66] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to  $H_\infty$  control," *International Journal of Robust & Nonlinear Control*, vol. 4, pp. 421–448, 2010.
- [67] J. Shi, F. Gao, and T.-J. Wu, "Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system," *Journal of Process Control*, vol. 15, no. 8, pp. 907–924, 2005.
- [68] D.-H. Hwang, Z. Bien, and S.-R. Oh, "Iterative learning control method for discrete-time dynamic systems," in *IEE Proceedings D Control Theory and Applications*, vol. 138, no. 2, pp. 139–144, IET, 1991.

## Research Article

# Prediction of Seepage Pressure Based on Memory Cells and Significance Analysis of Influencing Factors

Mengdie Zhao,<sup>1,2,3,4</sup> Haifeng Jiang,<sup>1,2</sup> Shoukai Chen ,<sup>1,2,3</sup> and Yajing Bie<sup>1,2,3</sup>

<sup>1</sup>School of Water Conservancy, North China University of Water Resources and Electric Power, Zhengzhou 450045, China

<sup>2</sup>Henan Key Laboratory of Water Environment Simulation and Treatment, Zhengzhou 450045, China

<sup>3</sup>Collaborative Innovation Center of Water Resources Efficient Utilization and Protection Engineering, Zhengzhou 450045, China

<sup>4</sup>Henan Key Laboratory of Water Resources Conservation and Intensive Utilization in the Yellow River Basin, Zhengzhou 450046, China

Correspondence should be addressed to Shoukai Chen; [man200177@163.com](mailto:man200177@163.com)

Received 4 February 2021; Revised 3 March 2021; Accepted 17 March 2021; Published 12 April 2021

Academic Editor: Muhammad Ahmad

Copyright © 2021 Mengdie Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Seepage analysis is always a concern in dam safety and stability research. The prediction and analysis of seepage pressure monitoring data is an effective way to ensure the safety and stability of dam seepage. With the timeliness of a change in a monitoring value and lag due to external influences, a RS-LSTM model written in Python is developed in this paper which combines rough set theory (RS) and the long- and short-term memory network model (LSTM). The model proposed calculates the prediction score of the seepage pressure of a dam experiencing multiple effects by preordering factor importance values to eliminate the interference of redundant factors. A case study shows that the water level, rainfall, temperature, and duration are all factors that affect the seepage pressure, and their importance values decrease successively. Thus, the seepage pressure of a dam can be predicted with a determination coefficient  $R^2$  of 0.96. Compared with the recurrent neural network (RNN) model and BP neural network model, the training time of the RS-LSTM model proposed is 6.37 s, and the operation efficiency is 41% and 59% higher than that of the RNN and BP models, respectively. The mean relative error is also 3.00%, which is 50% lower than that of the RNN model and 31% lower than that of the BP model. Based on these results, this model has the advantages of fast computation speed and high accuracy in prediction.

## 1. Introduction

Reservoir dams are an important infrastructure that play an important role in the development of economies and societies. However, with the recent rapid increase in the construction of reservoir dams, many safety accidents have followed [1]. From 1954 to 2006, 3498 dams broke in China alone, and more than 93% of them were earth rock dam failures [2]. Based on the material characteristics and operation mechanism of earth rock dams, the primary failure modes are overtopping, landslide instability, and seepage damage [3]. Combined with relevant data, the number of dam failures caused by seepage failure in China is approximately 39% of the total [4]. Similarly, foreign survey data demonstrate that approximately 46% of earth rock dam

failures are caused by infiltration damage, ranking only second to overtopping failure (approximately 48%) [5–8]. Therefore, the influence of seepage failure on the safety of earth rock dams is self-evident, and its significance is second to none.

Seepage failure primarily refers to the reduction of soil effective stress and seepage deformation due to the movement of soil particles inside the dam due to the action of seepage pressure, which decreases the stability of the dam or even destroys the dam. Therefore, the establishment of adequate seepage safety monitoring is an essential prerequisite to ensure seepage safety: prototype observation data are used for analysis, and a seepage monitoring model is developed to monitor and evaluate the operational conditions of water conservancy projects.



For seepage instability, influencing factors primarily include water level change, rainfall, temperature change, and duration. For example, Xu et al. [9] considered that the seepage flow of dam bodies is primarily caused by the superposition of reservoir water level and rainfall, and the seepage head and reservoir water level tend to be positively correlated. Guangjin et al. [10] and Ashraf et al. [11] studied variations in the seepage head of a soil slope due to changing rainfall intensity and duration. Testing indicated that as rainfall intensity and duration increased, the deeper the influence of rainfall factors on soil, the greater the change in the seepage head and the greater the change in seepage pressure. Xu and Chai [12] investigated the influence of changes in temperature on the permeability coefficient and calculated water heads in seepage fields. Compared to the calculated results without a change in temperature, their findings show that the former is more accurate when calculating the distribution of water heads in seepage fields. However, most studies to date have focused on the influence of changes in influencing factors on seepage under single or a combination of conditions, which only verifies that there is a significant correlation between these factors and the seepage field, namely, the measured seepage pressure value, making it difficult to distinguish the correlation or importance between the two. Based on existing monitoring results of seepage pressure, a change in the measured seepage pressure over a long time period is typically complex and variable; thus, it is difficult to directly determine how seepage pressure changes over time. There also tends to be a long-time delay in response to changes in external conditions. Therefore, it is important to determine the effect of various factors and consider the dynamic change characteristics of seepage pressure monitoring for analyzing the change law of seepage pressure value of long sequence of dam and realizing the related prediction.

Currently, there are many methods to establish mathematical models to predict seepage monitoring data. Data fitting and analysis can be achieved via multiple regression; however, the accuracy of this method is restricted by the collinearity between input factors [13]. Predictions based on neural network models have also been reported; for example, Zhang et al. [14] used a genetic algorithm (GA) to optimize the weights and thresholds of a backpropagation neural network (BPNN), thus establishing the BPNN-GA seepage prediction model. Shi et al. [15], Rankovic et al. [16], and others used a variety of different neural network models, including radial basis neural network models, to achieve accurate predictions of dam seepage. These models provide various options to predict the measured seepage pressure using a neural network model; however, such static models are primarily based on correlations between the seepage pressure and associated factors and describe the lag in seepage pressure in response to changes in the factors via the averaging method using existing data. Also, these static models cannot describe the timeliness of the seepage pressure of a dam and thus cannot describe the memory function of the historical information hidden in the data, maintain the information, and output it to the current neuron for calculation. Thus, a predictive model developed based on dynamic changes in relevant factors will be useful for practical engineering applications.

A predictive model of seepage pressure is established in this paper that considers the effects of multiple factors and dynamic changes, integrates rough set theory and dynamic neural network technology, and examines the advantages of the dynamic model based on the distribution of error and the determination coefficient  $R^2$  [17]. Model performance is characterized by comparing its structural characteristics and predictive results to those of other models based on engineering examples to provide references for the prediction of the seepage pressure value and engineering safety assessments.

## 2. Materials and Methods

**2.1. Project Overview.** A concrete-faced rockfill dam is investigated in the paper, with a project scale of Class A Large (1). The elevation of the dam crest is 413.8 m with a maximum dam height of 99.8 m, a dam crest width of 10 m, and a dam crest length of 540.46 m. An L-shaped reinforced concrete wave wall with a height of 4 m is set at the upstream side of the dam crest. The normal water level of the reservoir is 410 m, and the storage capacity is 11,076,000 m<sup>3</sup>. To monitor dam safety, a comprehensive prototype dam monitoring system that is composed of deformation monitoring, seepage monitoring, and environmental monitoring (e.g., rainfall and reservoir water level) and other equipment is installed at the reservoir. The seepage monitoring equipment uses buried and installed osmometers; four osmometers are installed in total with a data-collection interval of 1–3 d, as shown in Figure 1.

Considering osmometer PB4 as an example, changes in the data over the last 8 years from 2008 to 2015 are given in Figure 2. The seepage pressure measured fluctuates within a relatively stable range and varies with changes of water level, rainfall, temperature, and other factors; thus, the monitoring results are reasonable. The rainy season is concentrated in July and August every year and exhibits an annual cycle. The maximum rainfall during the monitoring period was 95.50 mm on July 8, 2012. The temperature change between years is relatively stable; however, the change within a given year is major, with the high value of approximately 37.8°C and the low value of -9.70°C. During the monitoring period, the change in the water level upstream and downstream is minor, and the water level upstream varies between 385 and 410 m. Compared to the periodic alterations in the rainfall and temperature, the change in the seepage pressure is more moderate and lags due to the influence of the level of seepage pressure at the front of the dam. For example, from January 2011 to March 2011, the water level of the reservoir gradually decreased by 4 to 5 m; however, the change in the seepage pressure started at the end of March 2011 and gradually decreased from approximately 0.1 to 0.3 m (see Figure 2(a)).

### 2.2. Methods

**2.2.1. Rough Set Theory.** Rough set (RS) theory is a theoretical method that can achieve the same classification ability of a knowledge base via attribute reduction to solve problems or apply classification rules [18, 19] and is typically

represented by a decision information table  $S = \langle U, C \cup D, V, f \rangle$ , where  $U$  is the domain or set of nonempty finite objects;  $C \cup D$  is the attribute set;  $C$  is the conditional attribute;  $D$  is the decision attribute, in which  $C \cap D = \emptyset$  is satisfied;  $V$  is the set of all attribute value domains; and  $f: U \times R \rightarrow V$  represents the attribute value of each object in the domain on the corresponding attribute. The reduction process of the influencing factors of the real seepage pressure is described as follows:

- (1) The seepage pressure value domain  $U$ , which is composed of the seepage pressure and the related influencing factor data, is divided into two equivalent classes by the Pawlak attribute importance method, which is  $\text{IND}(C) = \bigvee_{i,j,k,n \in N} \{\{C_i\}, \dots, \{C_j, \dots, C_k\}\}_n$  and  $\text{IND}(D) = \bigvee_{i,j,k,n \in N} \{\{D_i\}, \dots, \{D_j, \dots, D_k\}\}_n$  respectively.
- (2)  $\text{pos}_C(D)$  is calculated based on the dependence of the seepage pressure value  $D$  on the influencing factor set  $C$ , and the formula is as follows:

$$\text{pos}_C(D) = \bigcup_{X \in \text{IND}(D)} \underline{C}(X), \quad (1)$$

$$\text{pos}_C(D) = \frac{\sum_{i=1}^k |\text{pos}_C(D_i)|}{|U|}. \quad (2)$$

- (3) Removing the importance of a given influencing factor  $c_i$  from the seepage pressure  $D$  can be defined as

$$\text{pos}_{C-\{c_i\}}(D) = \frac{\sum_{i=1}^k |\text{pos}_{C-\{c_i\}}(D_i)|}{|U|}. \quad (3)$$

- (4) The attribute importance of the influencing factor  $c_i$  ( $c_i \in C$ ) is

$$\sigma_{C D}(c_i) = \text{pos}_C(D) - \text{pos}_{C-\{c_i\}}(D). \quad (4)$$

Formula (4) represents the degree to which the single factor  $c_i$  affects the classification of the seepage pressure after factor removal: the larger the value  $\sigma_{C D}(c_i)$  is, the greater the importance of the influencing factor is, and vice versa.

The seepage characteristics of dams are affected by multiple factors. However, due to the different importance of various factors regarding seepage, factors with little affect or irrelevant redundancy can be screened out by this method to facilitate the prediction of dynamic neural networks.

**2.2.2. Neural Network Dynamic Model/Long- and Short-Term Memory Network Model.** The RS method can be employed to screen the influencing factors of seepage pressure; however, it cannot obtain accurate predictions. Considering the timeliness of changes in seepage pressure and the lag between influencing factors, the long- and short-term memory network model (LSTM) is introduced [20]. The LSTM model, which is an improved form of recurrent neural networks (RNNs), is a deep learning model that can

process time series data [21]. Compared to the BP neural network model (BP model), the nodes between the hidden layers in LSTM are connected to each other; the adjacent nodes of the hidden layers are connected [15], which can describe the memory function, retain hidden information in the historical data, and output the data to the current neuron for data calculation. These nodes are also constantly updated with input data (see Figure 3). Compared to RNN, LSTM includes a memory unit to solve the problem of gradient disappearance or explosion during prediction [22] to express the hidden historical information in the data more accurately. The memory unit of the model consists of an input gate, forget gate, and output gate, which are utilized to control information transmission at different times. The input gate controls the strength of the new input heading into the memory unit and determines how many new memories will be merged with previous memories. The forget gate manages the strength of the memory unit to maintain the value at the last moment (i.e., selects or rejects historical information). If the forget gate is closed, no memory can pass; otherwise, all memories can come through. The output gate controls the strength of the output memory unit and determines the LSTM response to the outside world [14]. The descriptions in detail can be figured out in Figure 4.

The aforementioned function equation of propagation in each mechanism during LSTM training is described as follows [23–25]:

$$\begin{aligned} f_t &= g(W_{f1}C_{\text{LSTM}}^t + W_{f2}D_{\text{LSTM}}^{t-1} + b_f), \\ I_t &= g(W_{I1}C_{\text{LSTM}}^t + W_{I2}D_{\text{LSTM}}^{t-1} + b_I), \\ \tilde{S}_t &= \tan h(W_{c1}C_{\text{LSTM}}^t + W_{c2}D_{\text{LSTM}}^{t-1} + b_S), \\ S_t &= f_t \times S_{t-1} + I_t \times \tilde{S}_t, \\ O_t &= g(W_{o1}C_{\text{LSTM}}^t + W_{o2}D_{\text{LSTM}}^{t-1} + b_o), \\ h_t &= O_t \times \tan h(S_t), \end{aligned} \quad (5)$$

where  $I_t$ ,  $f_t$ , and  $O_t$  are vectors describing the input gate, forget gate, and output gate of the model at time  $t$ , respectively;  $C_{\text{LSTM}}^t$  is the set of influencing factors of the seepage pressure after RS reduction at time  $t$ ;  $D_{\text{LSTM}}^{t-1}$  is the seepage pressure at time  $t-1$ , representing the hidden historical information of LSTM;  $W_1$  is the connection weight between the input layer and the hidden layer;  $W_2$  is the connection weight between the hidden layer and the output layer;  $b_f$ ,  $b_I$ ,  $b_o$ , and  $b_S$  are bias terms that correspond to each structure, respectively;  $S_t$  is the vector of the memory unit at time  $t$  and is only used for memory units to forget old information and add new information;  $\tilde{S}_t$  is a new candidate vector value created for  $\tan h$ , which is the hyperbolic tangent function; and  $g$  is an activation function, which can map real numbers to  $[0, 1]$ , where 1 indicates that all information in the unit at the previous time is reserved and 0 represents that all information in the unit at the previous time is discarded.

**2.2.3. RS-LSTM Model Structure.** Dam seepage is complicated and typically affected by many factors. To demonstrate

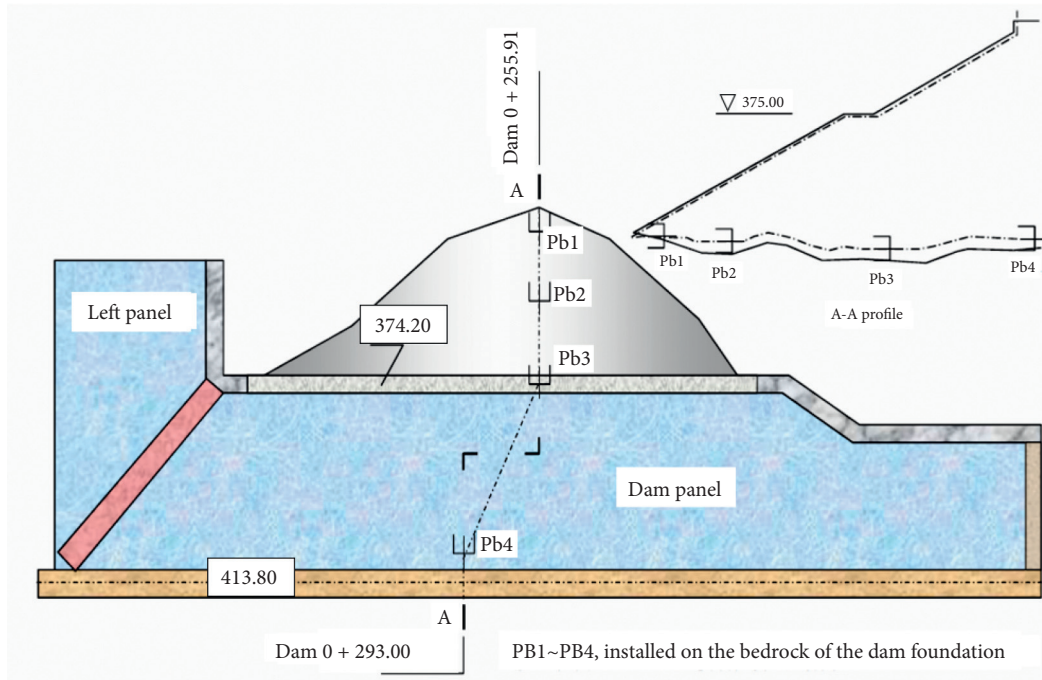


FIGURE 1: Arrangement of seepage pressure measuring instruments for dam foundations.

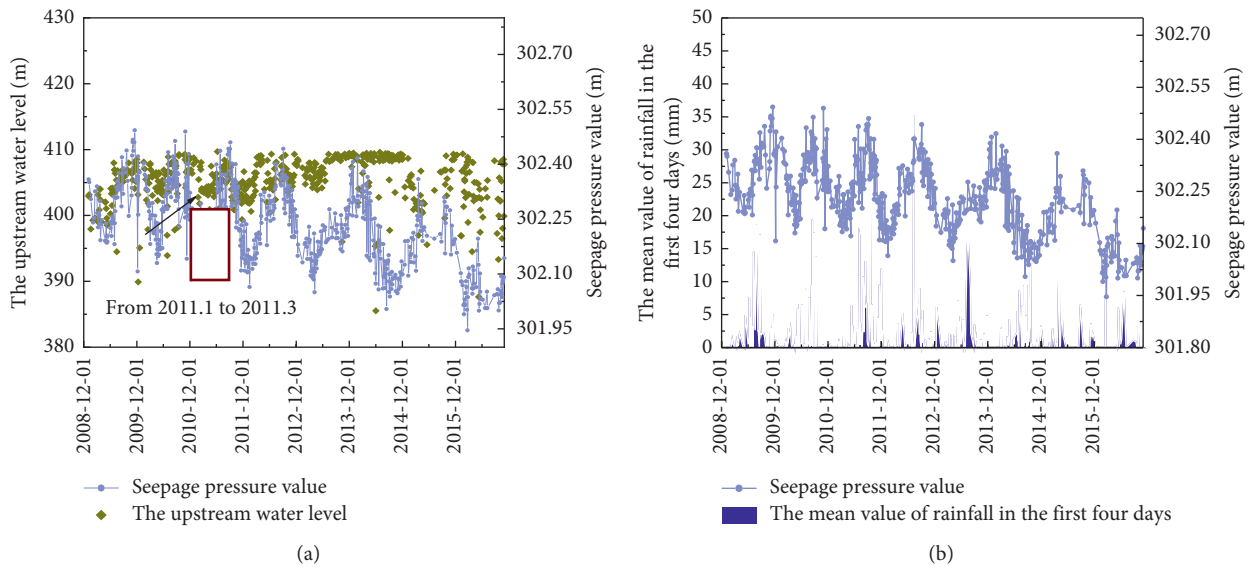


FIGURE 2: Continued.

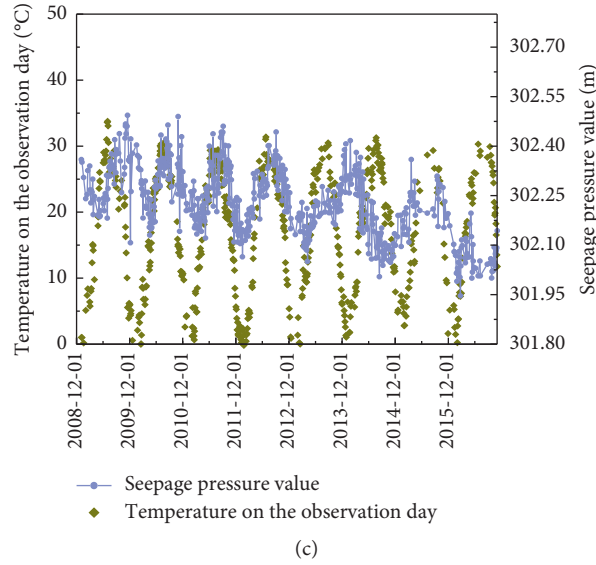


FIGURE 2: Long series change curve of permeability pressure value. (a) Relationship between the upstream water level and the seepage pressure. (b) Change in the seepage pressure due to rainfall. (c) Change in the osmotic pressure due to temperature.

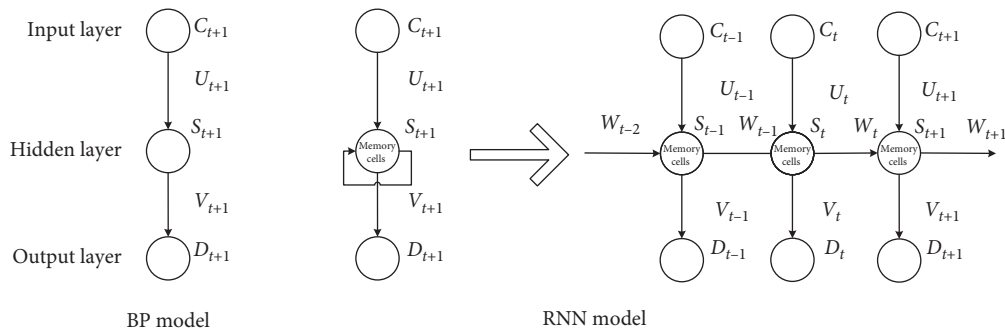


FIGURE 3: Static and dynamic model comparison.

the superior performance of the two methods proposed in data analysis, RS reduction is adopted before the LSTM prediction and analysis, and the RS-LSTM model structure is constructed as illustrated in Figure 5 to achieve the analysis and prediction of dam seepage in the operating period under the influence of multiple factors.

Based on the RS analysis results, an LSTM prediction model that considers various influencing factors is built to predict the seepage pressure. Two hidden layers are selected, and the number of hidden layer neurons is determined by the following formula:

$$l \leq \sqrt{m+n} + \alpha, \quad (6)$$

where  $l$  is the number of nodes in the hidden layers;  $m$  is the number of nodes in the output layers;  $n$  is the number of nodes in the input layers; and  $\alpha$  is the adjustment constant between 1 and 10. To determine the best prediction model structure after multiple calculations and training, training was found to be most effective when the number of nodes in

the hidden layers is 10; thus, the neural network topology structure is 6-10-1, which is displayed in Figure 6.

Considering PB4 as an example, the sample data contain water level, temperature, rainfall, duration, and monitoring values and have a total of 8,673 groups, including 1,239 groups of seepage pressure monitoring values. A total of 196 groups of data (2% of the total data set) were selected as prediction samples, 28 groups of data were selected as output samples for prediction, and the remaining were selected as training samples. The number of iterations was set to 5,000, the error range was set to 0.001, and the correction factor was set to 0.02.

### 3. Results and Discussion

**3.1. Application Results and Analysis of RS.** Based on existing literatures [9–12], seven influencing factors are considered for prediction and analysis in this study, including the upstream water level, downstream water level and temperature on the observation day, the mean rainfall in the first

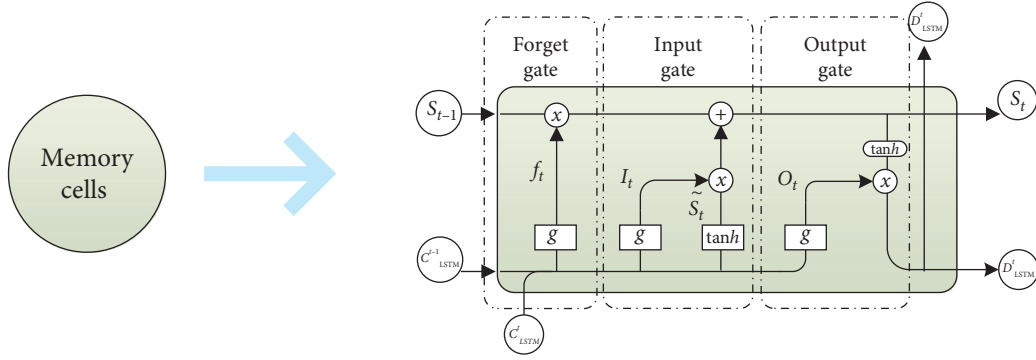


FIGURE 4: Structural model diagram of the LSTM memory unit.

four days, and the time-dependent components  $\theta$ ,  $e^{1+\theta}$ , and  $\ln(1+\theta)$ . These factors are recorded as  $X_1$ – $X_7$ , respectively, in the influencing factor set  $C$ , in which  $C_1: X_2$ – $X_7$ ;  $C_2: X_1$ ,  $X_3$ – $X_7$ ;  $C_3: X_1$ – $X_2$ ,  $X_4$ – $X_7$ ;  $C_4: X_1$ – $X_3$ ,  $X_5$ – $X_7$ ;  $C_5: X_1$ – $X_4$ ,  $X_6$ – $X_7$ ;  $C_6: X_1$ – $X_5$ ,  $X_7$ ; and  $C_7: X_1$ – $X_6$ . The seepage pressure is set as decision attribute  $D$ . After the normalization and discretization of  $C$  and  $D$ , the decision sequence is developed. The discretization is achieved by the method of equidistance; thus, after the normalization of the decision sequence value, they are discretized into five levels, i.e.,  $[0, 0.2)$ ,  $[0.2, 0.4)$ ,  $[0.4, 0.6)$ ,  $[0.6, 0.8)$ , and  $[0.8, 1]$ , which are recorded as 1, 2, 3, 4, and 5, respectively. After merging similar values, 100 groups of data to be deduced are obtained and recorded as  $U_1$ – $U_{100}$  (see Figure 7(a) for details; according to the decision attribute  $D$ , it can be divided into normal data and abnormal data). These groups of data can be reduced using formulae (1)–(3), and the results are presented in Figure 7(b), which show that the importance of  $X_6$  is 0;  $X_1$ ,  $X_2$ ,  $X_3$ ,  $X_4$ ,  $X_5$ , and  $X_7$  are all necessary attributes, and their importance order can be obtained. The importance of the water level upstream and downstream is the highest, scoring 0.31 and 0.29, respectively; the influence of the rainfall and temperature components follows closely, scoring 0.25 and 0.24; and the influence of time-dependent components  $\theta$  and  $\ln(1+\theta)$  is small, only scoring 0.12 and 0.14.

Considering  $X_1$  as an example to describe the decision reduction process, the equivalence classes of conditional attributes  $C$  and  $D$  are  $\text{IND}(D) = \{\{U_1, U_2, \dots, U_{37}\}, \{U_{38}, U_{39}, \dots, U_{100}\}\}$  and  $\text{IND}(C) = \{\{U_1\}, \{U_2\}, \dots, \{U_{100}\}\}$ , respectively, which are determined by combining collective knowledge based on  $\text{IND}(C) \subseteq \text{IND}(D)$ , where  $\text{pos}_C(D) = U$ . With  $\text{IND}(C_1)$  and setting  $\{U_{32}, U_{60}, U_{95}\}, \{U_7, U_{55}\}, \{U_{37}, U_{98}\}$ , etc., a total of 31 sets of datasets are not subsets (as shown in Figure 7(b), the horizontal axis is the collection number that does not belong to  $\text{IND}(D)$ ); therefore,  $\text{pos}_C(D) \neq \text{pos}_{C_1}(D)$ .  $X_1$  is necessary in  $C$  relative to  $D$ , and the importance of  $X_1$  to  $C$  relative to  $D$  can be determined based on equation (4), yielding a result of 0.31.

Based on this analysis, the factors that affect seepage pressure do not include  $X_6$  because its importance is 0; thus,

the relative reduction of the decision sequence can be obtained.

**3.2. Prediction Results and Analysis.** Figure 8 reveals the prediction results of the seepage pressure, and Table 1 makes a comparison of the model prediction results. From the results, the neural network method is shown to predict the seepage pressure accurately. The determination coefficient  $R^2$  of the RS-LSTM, RNN, and BP models is 0.97, 0.89, and 0.83, respectively, and the mean relative errors are 3.00%, 6.08%, and 9.85%, respectively, all of which are within 10%; thus, the prediction accuracy requirements are satisfactory. Comparing the prediction results of the three models, the RS-LSTM model is found to have the highest prediction accuracy, followed by the RNN model and the BP model. The calculation accuracy of the former is 2.03 times and 3.28 times that of the RNN and BP models, respectively, based on the mean relative error.

During the operation of the RS-LSTM model, the importance ranking of influencing factors is determined by RS calculation, which provides decision support for the selection of the input layer and weight setting of LSTM; eliminates the interference of redundant factors; and markedly improves the operation efficiency. In this case, prediction can be completed in 6.37 s and 278 iterations; the training error convergence process is shown in Figure 9, and the average operation efficiency is 41% and 59% higher than that of the RNN model and BP model, respectively. Additionally, both the LSTM and RNN models consider historical information hidden in the data, which can maintain correlations between the seepage pressure at different times and use them to describe the dynamic characteristics of the seepage pressure, which the BP model cannot describe.

The realization process considers historical seepage pressure values. The output of the hidden layer in the model has a connection weight from the previous hidden layer to the current layer. When the seepage pressures are output, the current water level of the upstream, rainfall, and other factors will affect the current output through the memory unit with the output layer status and will again affect the

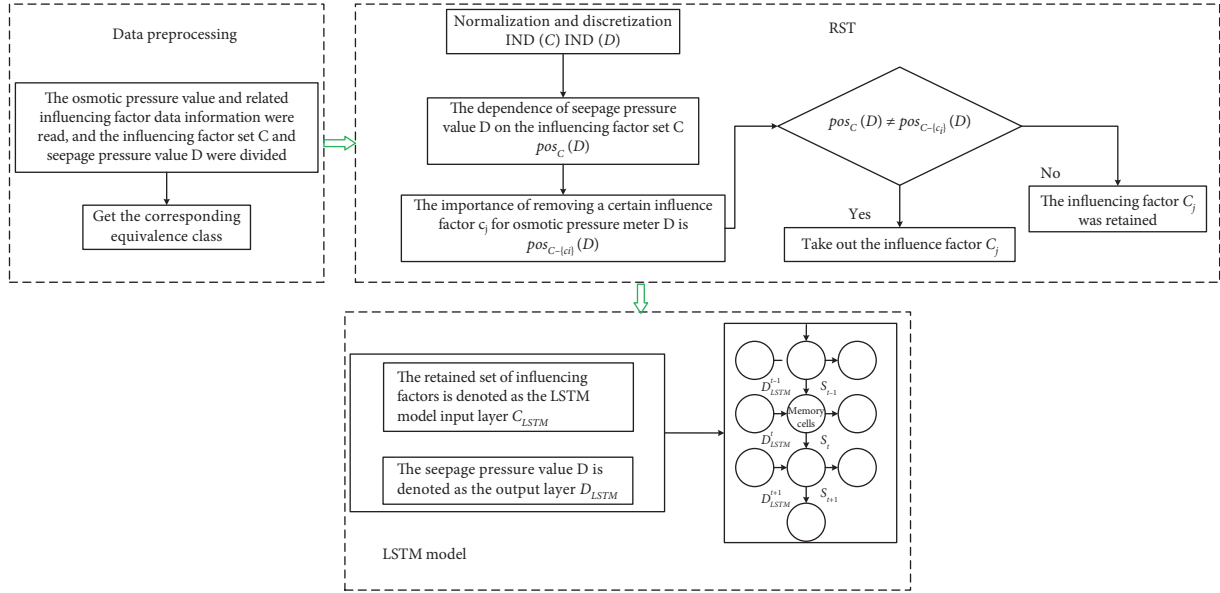


FIGURE 5: Prediction flowchart of the RS-LSTM model.

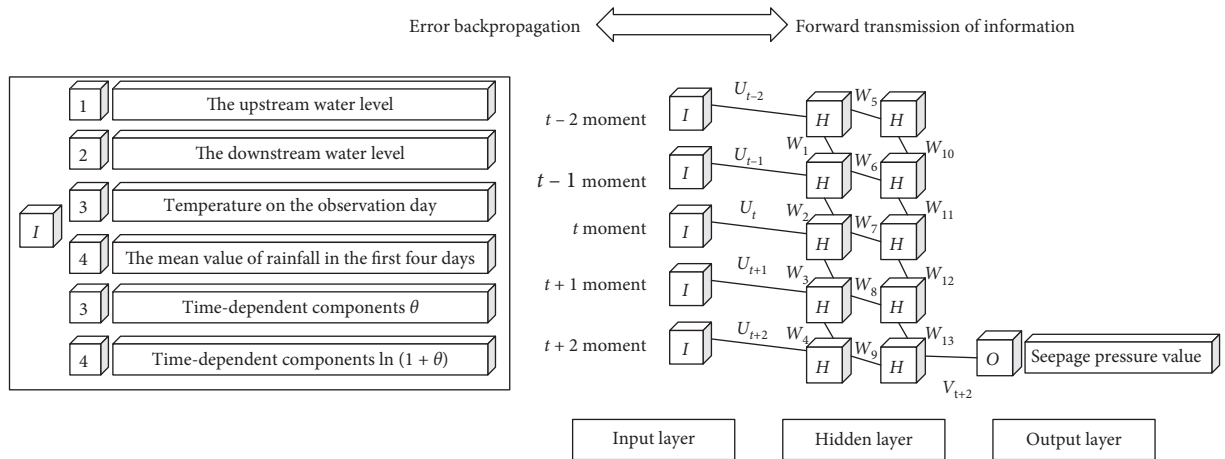


FIGURE 6: Network topology of the RS-LSTM prediction model.

output of the seepage pressure in the next time step through the memory unit with the current output layer status. This cycle then repeats.

Compared to the RNN model, the LSTM model comprises an input gate, forget gate, and output gate of the memory unit, which can select, forget, and output the current input  $x_t$  (namely, the influencing factors at the current moment), the output values of the hidden status  $h_{t-1}$  at the last moment (namely, the seepage pressure values of the hidden layer at the last moment), and the maximum reservation and output of the important hidden information, respectively, to markedly improve prediction accuracy. The specific process is described as follows. When data  $C_{LSTM}^t$  and  $D_{LSTM}^{t-1}$  pass through the forgetting gate, the information that must be discarded in the data is determined based on the setting range  $[0,1]$  of the model and is determined as the information retention level  $f_t$ . After the input gate, the data in this study consist

of two parts: the sigmoid layer is primarily used to describe the renewal of the numerical range and the tanh layer is employed to create the new candidate vector-valued  $\tilde{C}_t$ . These layers are combined with the forgotten door to generate a new vector-valued  $S_t$ . For the next time step, the memory cell seepage pressure is predicted, and the output includes  $S_t$ . The output value of the current time is then determined.

The primary advantages of the RS-LSTM include the following three points. Firstly, the RS-LSTM considers historical information. By storing, writing, or reading information through the memory unit, each unit can determine which piece of information is going to be transmitted through the door switch and when it is allowed to be read, written, or cleared and can more intelligently describe the prediction and analysis of time series data. Secondly, the RS-LSTM yields high prediction accuracies and fast convergence speeds with fewer iterations and avoids local minima.

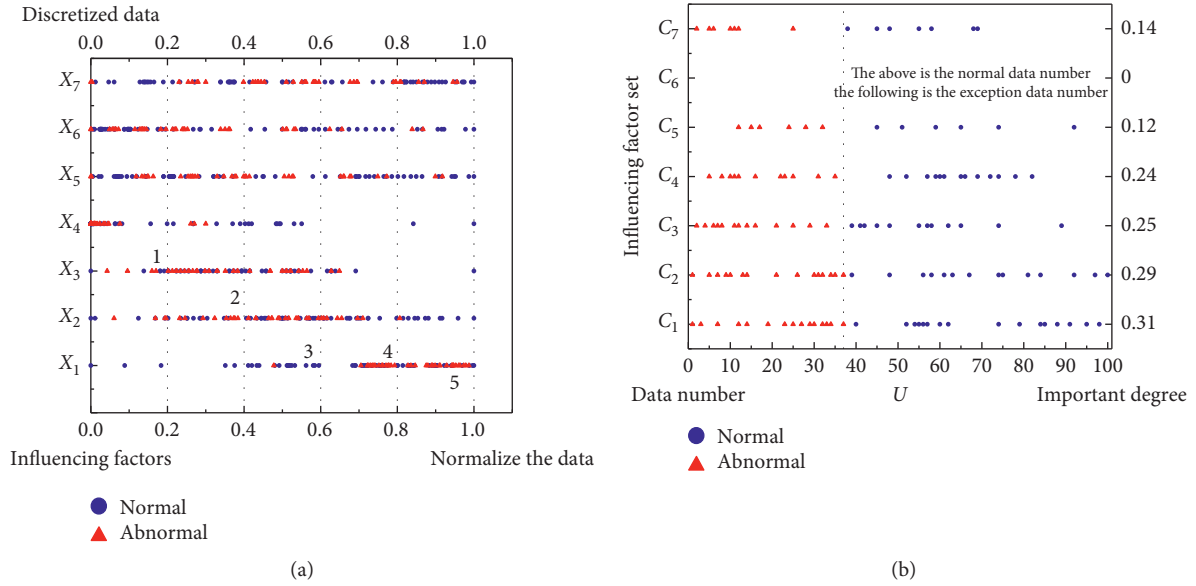


FIGURE 7: Rough set theory decision making and attribute reduction classification. (a) Decision sequence of influencing factors of the seepage pressure water level. (b) Reduction classification results of influencing factors of the seepage pressure water level.

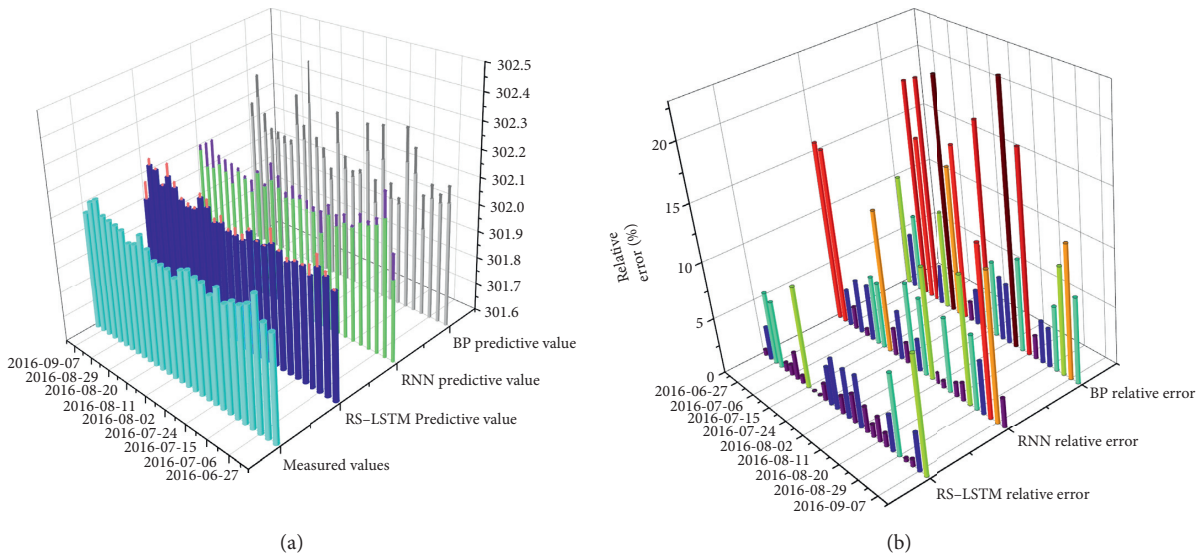


FIGURE 8: Comparison between prediction and measurement of different models. (a) Comparison diagram of the seepage pressure of each model. (b) Relative error comparison diagram.

TABLE 1: Comparison of model prediction results.

Model	Absolute error (m)				Relative error (%)				Iteration time (s)	Iteration times (times)	$R^2$
	Maximum value	Minimum value	Median	Mean value	Maximum value	Minimum value	Median	Mean value			
RNN model	0.109	0.004	0.031	0.040	15.84	0.59	4.85	6.08	10.85	775	0.89
BP model	0.156	0.011	0.051	0.064	22.55	1.67	7.83	9.85	15.55	1344	0.83
RS-LSTM model	0.054	0.000	0.015	0.020	9.54	0.03	2.34	3.00	6.37	278	0.97

Remarks: absolute error = |measured value of seepage pressure - predicted value of seepage pressure|; relative error = absolute error/water head  $\times$  100%.

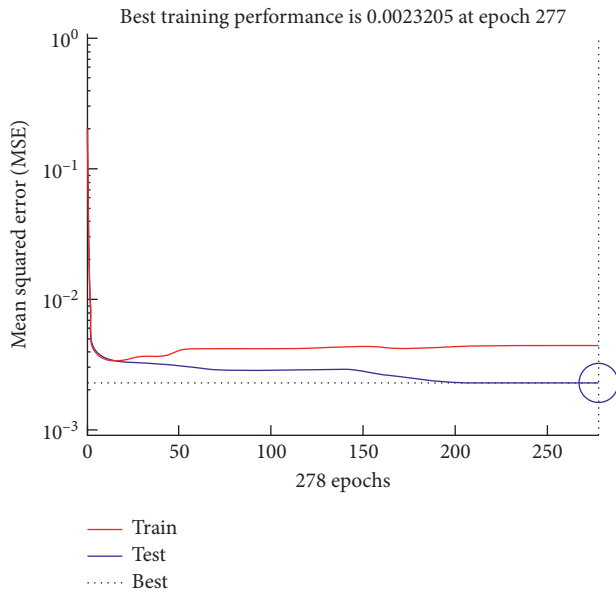


FIGURE 9: Training error convergence process diagram.

Finally, the RS-LSTM mitigates correlations between a single factor and the seepage pressure during seepage, such as the increase in the water level increasing the seepage pressure.

#### 4. Conclusion

In this paper, the RS-LSTM model structure is constructed using engineering examples, the significances of different influencing factors on the seepage pressure are investigated using rough set theory, and the influence of different models on the prediction of the seepage pressure is described by comparing average relative errors. The specific conclusions of this study are as follows:

- (1) Written in Python, the RS theory and LSTM model are integrated to develop the RS-LSTM model in this paper. Examples elucidate that this model considers changes in measured seepage pressure during seepage and their correlations with external factors and synchronously describes the importance ranking of the influencing factors of seepage. Meanwhile, this model can eliminate the influence of redundant factors and predict dam seepage monitoring accurately and effectively. The model can provide the corresponding and the theoretical support for seepage safety during the operation of dam projects in the future.
- (2) A practical example demonstrates that the primary influencing factors of seepage pressure include the water level upstream, the water level downstream, the temperature, the average rainfall in the first four days, and the time-dependent components  $\theta$  and  $\ln(1 + \theta)$ . The importance scores of these factors are 0.31, 0.29, 0.25, 0.24, 0.12, and 0.14, respectively, and the time-dependent component  $e^{1+\theta}$  is redundant.

- (3) The prediction results demonstrate that neural networks can be used to predict seepage pressures in dams, and the determination coefficient  $R^2$  of the RS-LSTM model proposed in this paper can reach 0.96. Compared to the traditional neural network model, the RS-LSTM model yields a higher prediction accuracy, and the accuracy of the RS-LSTM model is 2.03 and 3.28 times that of the RNN model and BP model, respectively.

As part of the future study, the proposed model could be utilized for datasets in other dams. To further enhance the performance of the proposed model, it can be attempted to integrate a variety of seepage pressure gauges to achieve multi-point and multi-factor pressure prediction.

#### Data Availability

The data used to support the findings of this study are included within the article.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Authors' Contributions

Jiang Haifeng developed the RS-LSTM model and prepared the manuscript. Zhao Mengdie, Shoukai Chen, and Bie Yajing contributed to conceptualization, provided idea and suggestion for the study, and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

#### Acknowledgments

The authors would like to thank the North China University of Water Resources and Electric Power. This research was funded by the National Key R&D Plan (2018YFC0406900) and the National Natural Science Foundation of China (52009045).

#### References

- [1] W. Ge, H. Sun, H. Zhang et al., "Economic risk criteria for dams considering the relative level of economy and industrial economic contribution," *Science of the Total Environment*, vol. 725, Article ID 138139, 2020.
- [2] J. Li and D.-y. Sun, "Statistics of dam failures in China and analysis on failure causations," *Water Resources and Hydro-power Engineering*, vol. 40, no. 12, pp. 124–128, 2009.
- [3] S.-h. Jiang and Z.-w. Fan, "Earth-rockfill dam safety classification and risk rate assessment on flood control," *Journal of Hydraulic Engineering*, vol. 39, no. 1, pp. 35–40, 2008.
- [4] W. Ge, X. Wang, Z. Li et al., "Interval analysis of loss of life caused by dam failure," *Journal of Water Resources Planning and Management*, vol. 147, no. 1, Article ID 04020098, 2021.
- [5] S.-s. Chen, *The Failure Mechanism of Earth-Rock DAMS and the Simulation of Dam Failure Process*, China Water & Power Press, Beijing, China, 2012.



- [6] Z. Li, Y. Zhang, J. Wang et al., "Impact evaluation of geomorphic changes caused by extreme floods on inundation area considering geomorphic variations and land use types," *Science of the Total Environment*, vol. 754, Article ID 142424, 2020.
- [7] W. Ge, Y. Qin, Z. Li et al., "An innovative methodology for establishing societal life risk criteria for dams: a case study to reservoir dam failure events in China," *International Journal of Disaster Risk Reduction*, vol. 49, Article ID 101663, 2020.
- [8] M. Zhang, R. Fell, and M. Spannagle, "The statistics of embankment dam failures and accidents," *Canadian Geotechnical Journal*, vol. 37, no. 5, pp. 1000–1024, 2000.
- [9] J. Xu, W. Wei, H. Bao et al., "Failure models of a loess stacked dam: a case study in the Ansai Area (China)," *Bulletin of Engineering Geology and the Environment*, vol. 79, no. 12, 2019.
- [10] W. Guangjin, K. Jingwen, D. Chao et al., "Study on tailings dam over-topping failure model test and break mechanism under the rainfall condition," *Tehnicki Vjesnik*, vol. 24, no. 6, pp. 1897–1904, 2017.
- [11] M. Ashraf, M. Ismail, S. M. Ng, and A. Ismail, "Parametric study of horizontal drains for slope stability measure: a case study in Putrajaya, Malaysia," *KSCE Journal of Civil Engineering*, vol. 21, pp. 2162–2167, 2017.
- [12] Z.-g. Xu and J.-r. Chai, "Numerical analysis of stable seepage field in rock mass fracture network under the influence of temperature," *Journal of Xi'an Petroleum University (Natural Science Edition)*, vol. 02, pp. 169–172, 2007.
- [13] C. Xiang, L. Qingquan, Z. Zhiwei et al., "Research on a seepage monitoring model of a high core rockfill dam based on machine learning," *Sensors*, vol. 18, no. 9, p. 2749, 2018.
- [14] X. Zhang, X. Chen, and J. Li, "Improving dam seepage prediction using back-propagation neural network and genetic algorithm," *Mathematical Problems in Engineering*, vol. 2020, no. 11, 8 pages, Article ID 1404295, 2020.
- [15] Z. Shi, C. Gu, E. Zhao, and B. Xu, "A novel seepage safety monitoring model of CFRD with slab cracks using monitoring data," *Mathematical Problems in Engineering*, vol. 2020, no. 12, pp. 1–13, Article ID 1641747, 2020.
- [16] V. Rankovic, A. Novakovic, N. Grujovic et al., "Predicting piezometric water level in dams via artificial neural networks," *Neural Computing & Applications*, vol. 24, no. 5, pp. 1115–1121, 2014.
- [17] B. Yang, K. Yin, and J. Du, "A model for predicting landslide displacement based on time series and long and short-term memory neural network," *Chinese Journal of Rock Mechanics and Engineering*, vol. 37, no. 10, pp. 2334–2343, 2018.
- [18] Z. Pawlak, "Rough set theory and its applications to data analysis," *Cybernetics and Systems*, vol. 29, no. 7, pp. 661–688, 1998.
- [19] J. Zhao, J.-m. Liang, Z.-n. Dong, D.-y. Tang, Z. Liu et al., "Accelerating information entropy-based feature selection using rough set theory with classified nested equivalence classes," *Pattern Recognition*, vol. 107, 2020.
- [20] S. Hochreite and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [21] I. Good fellow, Y. Bengio, A. Courville et al., *Deep Learning*, MIT press, Cambridge, MA, USA, 2016.
- [22] S. Zhang, Y. Wang, M. Liu et al., "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2017.
- [23] Y. Guo and X. Lai, "Water level prediction of Lake Poyang based on long short term memory neural network," *Journal of Lake Sciences*, vol. 32, no. 3, pp. 865–876, 2020.
- [24] J. Zhanga, Y. Zhub, X. Zhanga et al., "Developing a Long short-term memory (LSTM) based model for predicting water table depth in agricultural areas," *Journal of Hydrology*, vol. 561, pp. 918–929, 2018.
- [25] D. Zhang, G. Lindholm, and H. Ratnaweera, "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring," *Journal of Hydrology*, vol. 556, pp. 409–418, 2018.

## Research Article

# A Computationally Efficient User Model for Effective Content Adaptation Based on Domain-Wise Learning Style Preferences: A Web-Based Approach

Dong Pan <sup>1</sup>, Anwar Hussain <sup>2</sup>, Shah Nazir <sup>2</sup>, and Sulaiman Khan <sup>2</sup>

<sup>1</sup>Southwest Minzu University Information and Educational Technology Center, Chengdu 610041, China

<sup>2</sup>Department of Computer Science, University of Swabi, Swabi, Pakistan

Correspondence should be addressed to Dong Pan; [pandongdong1126@163.com](mailto:pandongdong1126@163.com) and Shah Nazir; [snsahnzr@gmail.com](mailto:snsahnzr@gmail.com)

Received 26 November 2020; Revised 21 December 2020; Accepted 27 February 2021; Published 1 April 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Dong Pan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the educational hypermedia domain, adaptive systems try to adapt educational materials according to the required properties of a user. The adaptability of these systems becomes more effective once the system has the knowledge about how a student can learn better. Studies suggest that, for effective personalization, one of the important features is to know precisely the learning style of a student. However, learning styles are dynamic and may vary domain-wise. To address such aspects of learning styles, we have proposed a computationally efficient solution that considers the dynamic and nondeterministic nature of learning styles, effect of the subject domain, and nonstationary aspect during the learning process. The proposed model is novel, robust, and flexible to optimize students' domain-wise learning style preferences for better content adaptation. We have developed a web-based experimental prototype for assessment and validation. The proposed model is compared with the existing available learning style-based model, and the experimental results show that personalization based on incorporating discipline-wise learning style variations becomes more effective.

## 1. Introduction

Adaptive systems provide and adapt contents according to the properties of an individual user [1]. In the educational environment, these systems filter and provide educational contents based on the preferences of a student and are known as adaptive educational hypermedia systems [1, 2]. The user model in these systems is confined and focuses on the features that are associated with the learning aspect of a student [2, 3], and learning styles are exploited in the adaptive hypermedia domain as a source of content adaptation [4–6]. By integrating learning styles in the user model, it covers the cognition aspect of human learning, such as how a user learns [2, 3]. Preferences of learning styles are exploited in a number of ways to solve and address different problems in the domain [7–9]. As a result, learning styles are mainly used in the educational domain where educational contents need to be filtered for the students.

One of the significant features of learning styles is its dynamic and nonstationary aspects that can be changed from domain to domain. Jones in his study [10] investigated and came to the conclusion that learning styles have variations domain-wise. According to his study, significant differences were identified in learning styles for various disciplines, such as Mathematics, Social studies, Science, and English. In order to enhance the adaptation of a learning system, it is important to consider the domain-wise learning style preferences too. Researchers emphasized on incorporating a precise representation of the student needs. As a result, recent models in the literature have the capability to some extent to adjust learning style preferences automatically with the system's usage [11–13]. However, the existing systems still have deficiencies in content adaptation because learning is not an easily quantifiable and continuous process. Due to these properties, most of the approaches are probabilistic for updating learning style preferences instead of deriving the exact measures [14].

A number of surveys are conducted in the adaptive educational hypermedia domain that emphasizes on the effectiveness of learning styles for adapting the educational contents [2, 3, 15]. These studies provide useful insights regarding the important variables, models developed, features exploited, and issues faced by these approaches. While studying the recent work, to our knowledge, these approaches mainly work in the content of a single domain [12, 13, 15]. In fact, learning styles are dynamic and can change with the passage of time [16, 17]. It is not necessary that a student will always have the same learning approach for learning the concept of different academic subjects. For example, it is possible that, for one subject, students prefer examples of the concept, while for the other, theories of concepts are more important. So it is necessary to provide adaptation that also depends upon the need of the subject.

To address a domain-wise aspect of learning styles, we have proposed a domain-wise learning style solution that effectively exploits a student's *discipline-specific learning style preferences* for content adaptation. The proposed model dynamically and continuously updates student's learning preferences for different academic disciplines accordingly. The proposed solution tries to address student's *learning style preferences* that are not necessarily the same for multiple disciplines. Furthermore, for assessment and validation of our model, a web-based experimental prototype is developed which adapts contents using the proposed model, and comparison is done with the existing learning style-based solutions. The experimental results show that the proposed approach exploits student's learning styles in an effective way and enhances the learning process. Following are the main contributions of the study:

- (i) To exploit domain-wise learning preferences for effective learning
- (ii) To propose a model which has the capability to adjust domain-wise learning style variations for different subject domains
- (iii) To provide a computationally efficient solution that can be easily adopted for developing learning systems in the educational hypermedia domain

The rest of the paper is divided into the following sections. Section 2 provides the background and related work to the current research. The proposed models and experimental prototype application are explained in Section 3 and Section 4, respectively. The research design for the study is given in Section 5. Results are presented in Section 6 while Section 7 is the conclusion of the study.

## 2. Background Study

With the passage of time, applications of computer increases and its scopes span in order to provide more efficient solutions for the problems in different domains. Some of the major applications of computers can be seen in the form of online shopping recommender system, expert system, image processing, artificial intelligence, and so on. In the educational domain, systems try to enhance the user learning

process by adapting contents according to the requirements of a user and are called adaptive educational hypermedia systems. Contributions of experts in the field of adaptive educational hypermedia domain have been observed, and their studies mainly focus to make the system capable of understanding the requirements of the user more precisely. The experts believe that, for better adaptation, it is more important for a system to know the user cognitive abilities along with physical capabilities [18]. The cognitive abilities include the learning process of humans and the way people think and process information. As a result, learning style becomes one of the important features to be considered in the user model of these systems. Most of the models of human cognition are exploited to enhance the ability of the system's adaptation. The following sections provide an overview of the most frequent learning style models that are exploited and the approaches used for applying these models in the adaptive educational hypermedia domain.

*2.1. Frequently Used Learning Style Models.* Experts have used a number of learning style models for content adaptation in the adaptive educational hypermedia domain. Some of them are used more frequently in the literature. The details of these models are given below.

*2.1.1. Verbal or Imager Holistic or Analytical.* This model categorizes the users into two main categories. These categories include a total of four types of users [19]. The first one is verbal users that understand well from the contents represented in written form. The second is imager users that understand the material which is represented in graphical form. The third is the holistic, who looks at the situation as a whole and uses a top-down approach for concept understanding while the fourth one is analytical which understands the material by individual parts and then combines them.

*2.1.2. Kolb's Learning Style.* This model [20] is based on experiential learning theory (ELT). It defines that how knowledge is created from experience (Kolb (1984)). According to this model, a learner goes through four modes in a cyclic fashion. These include concrete experience, reflective observation, abstract conceptualization, and active experimentation. According to this model, previous experience has a great contribution in the learning process.

*2.1.3. Field Dependent or Independent.* This model considers the user's cognitive psychological structure by which a user differentiates contextual information [21]. It mainly divides users into one of the two types such as *field dependent* and *field independent*. The *field-dependent* users are affected by the external environment. Normally, they use nonverbal gestures for conveying their idea and also differentiate things from their context with difficulty. However, the *field-independent style* users look to the things with their own perspective in mind. They are not affected by others easily, and

normally, they respond quickly inside a classroom environment.

*2.1.4. Felder and Silverman Learning Style.* This model is based on four dimensions, and each dimension has two types of users. The first dimension has active and passive learners. Active learners are more experimental, work in groups, and share their ideas while passive learners take more time on thinking before applying experiments. The second dimension has sensing and intuitive learners. The first learner relates knowledge to the world application, more practical, while the former learners are more creative and they like abstract learning content which possesses theories. The third dimension has visual and verbal learners. As their names show, visual learners understand more from the graphical form while verbal learners learn from textual information and verbal explanations. The fourth dimension includes global and sequential learners. Sequential learners tend to develop knowledge step by step from fragments of information and integrate them. On the other hand, global learners see the overall picture first and then go towards the fragments in a top-down manner.

*2.2. Modelling Approaches for Adaptive Hypermedia System.* Adaptive educational hypermedia systems are the application of adaptive systems in the educational domain [1]. In this case, the domain becomes confined in the context of the user model and contents. The user model includes those features that are associated with the students and hypermedia such as educational contents instead of general information. These systems are categorized into two types based on their adaptation mechanism [22]. One is based on the presentation of contents, called adaptive presentation system, while the other is based on the link level, known as adaptive navigation systems. The adaptive presentation systems adapt contents and present them in different forms. User model in these systems exploits learning styles to associate different forms of contents with the learning type of user. The adaptive navigation systems change the navigation pattern of the content accessibility based on the requirements of the user. These include links disabling, accessing links, and hiding these links. The summary of the number of adaptive presentation and navigation systems is given in Tables 1 and 2. In addition to these two types of systems, there are a number of hybrid approaches that take benefits from the features of both above approaches. Table 3 shows the number of modeling approaches used in the adaptive hypermedia domain.

As shown in Table 3, user models are mainly based on three approaches. These approaches are overlay, stereotype, and combination of the former two models [36]. According to the survey [2, 3], user knowledge is one of the most important features that are being considered for adaptation. Systems that are based on overlay model are HYPERFLEX [37, 38], Adaptive Hypermedia Architecture System (AHA)! [26], and ISIS-Tutor [33]. HyperTutor [39] is based on the stereotype user model. In the stereotype user model-based system, the user is assigned to a category or

type as we have seen in different learning style models. Some of the systems use diverse approaches that benefit from more than one modeling approach. These systems include ANATOM-TUTOR [24], ELM-ART [22], Interbook I, and Avanti.

Furthermore, Abyaa et al. [15] have done a detailed survey on the modeling approaches, variables exploited, and learning factors that are used in the learner model. Survey analysed and categorized the characteristic of the modeling approaches. Their analysis shows that, among different characteristics, some of the most important characteristics are exploited more frequently in the past. These include learner's knowledge and their cognitive abilities that are reflected in the system by incorporating learning style preferences in the user model.

In short, incorporating learning styles for content adaptation in the adaptive system is an important and focused area for experts [14, 15, 40]. Studies try to enhance students learning outcomes such as open learning model (OLM) [41]. Furthermore, current studies mainly focus on how to minimize the gap between user learning behavior and system adaptation level [42–44]. In order to minimize this gap, the learning styles of the user are required to be modelled in a way that should reflect the true needs of the user. One of the main properties is the domain-wise learning style variations, which has not been taken into account, and the previous system is used to update these learning styles in the context of a single domain. However, our work considers this important dimension in the adaptation process, such as domain-wise learning style variations.

### 3. Proposed Model

The domain-wise adaptive (DWA) model is inspired from [45, 46], where the proposed research has made it capable of handling discipline-specific variations in learning style. Learning style preferences are represented by real values so that they can be exploited in an environment where moderate preferences are necessary to be considered. Preferences of users are identified via a learning style questionnaire. Furthermore, these preferences are updated according to the requirements of the subject's domain. The problem definition along with the contribution of the proposed study is shown in Figure 1. The following sections show the main contribution of the proposed study.

*3.1. Learning Styles of the User.* Learning style model assigns users to one of the 16 learning style preferences. Preferences are mainly the combination of four dimensions of ILS. Every dimension categorizes users into two types (preferences). Detail of the dimensions, their purpose, and their associated preferences are shown in Table 4. Learning style is a collective measure that combines all of the four dimensions for a user and is represented by the following equation:

$$\text{LSC} = \left\{ a \in \left( \frac{\text{Act}}{\text{Ref}} \right), b \in \left( \frac{\text{Vis}}{\text{Ver}} \right), c \in \left( \frac{\text{Se}}{\text{In}} \right), d \in \left( \frac{\text{Glo}}{\text{Seq}} \right) \right\}. \quad (1)$$

TABLE 1: Several adaptive presentation systems in the adaptive hypermedia domain.

System type	Technique used	Proposed system	Background	Goals	Preferences	Knowledge	Hyperspace experience
Presentation	Adaptive link annotation/ sorting	ELM-ART [22]	No	No	No	Yes	No
	Fragments are altered	TANGOW [23]	No	Yes	Yes	No	No
	Fragments are altered	ANATOM-TUTOR [24]	Yes	No	No	Yes	Yes
	Fragments are inserted or removed or altered	WHULEHM [25]	No	No	No	Yes	No
	Fragments are inserted or removed or altered	AHA! [26]	Yes	No	No	No	No
	Fragments are inserted or removed or altered	C-book [27]	Yes	No	No	No	No
	Fragments are stored	Hypadapter [28]	No	No	Yes	Yes	No

TABLE 2: Several adaptive navigation systems in the adaptive hypermedia domain.

System type	Technique used	Proposed system	Background	Goals	Preferences	Knowledge	Hyperspace experience
Navigation	Stretchtext	Metadoc [29]	Yes	No	No	Yes	No
	Adaptive link annotation/hiding	AHA! [26]	No	No	No	Yes	No
	Direct guidance	SHIVA [30]	No	No	No	Yes	No
	Direct guidance, hiding links	HyperTutor	No	No	No	Yes	Yes
	Direct guidance, hiding links	Netcoach [31]	No	Yes	Yes	Yes	No
	Direct guidance, hiding links	Interbook [32]	No	No	No	Yes	No
	Direct guidance, link annotation, hiding links	ISIS-Tutor [33]	No	No	No	Yes	No
	Disable links, link sorting	TANGOW [23]	Yes	Yes	Yes	No	No
	Disable links, link sorting	TANGOW [23]	Yes	Yes	Yes	No	No
	Link annotation	ITEM/PG [34]	No	No	No	Yes	No
	Link hiding	Hypadapter [28]	No	No	Yes	Yes	No
	Removal of links	CHEOPS [35]	No	No	No	Yes	No

TABLE 3: Adaptive hypermedia system modeling approaches.

S. no	System proposed	Overlay-based models	Stereotype-based models	Hybrid approach
1	AHA	Yes	No	No
2	INTERBOOK	No	No	Yes
3	Avanti	No	No	Yes
4	HYPERFLEX	Yes	No	No
5	HyperTutor	No	Yes	No
6	ANATOM-TUTOR	No	No	Yes
7	AHM AND XAHM	Yes	No	No
8	ISIS-Tutor	Yes	No	No
9	ELM-ART	No	No	Yes

Based on equation (1), the following 16 learning style preferences are obtained: LSCs = {(Ref, Se, Vis, Seq), (Ref, Se, Vis, Glo), (Act, Se, Vis, Seq), (Act, Se, Vis, Glo), (Act, Se, Ver, Seq), (Act, Se, Ver, Glo), (Ref, Se, Ver, Seq), (Ref, Se, Ver, Glo), (Act, In, Vis, Seq), (Act, In, Vis, Glo), (Act, In, Ver, Seq), (Act, In, Ver, Glo), (Ref, In, Vis, Seq), (Ref, In, Vis, Glo), (Ref, In, Ver, Seq), and (Ref, In, Ver, Glo)}

**3.2. Identification of Learning Styles.** As we know, learning is not easily quantifiable to be measured; therefore, each learning style preference is shown by its probability. The total probability of each dimension is 1, while the preference inside shares

part of it. For example, if the probability of visual preference is  $x$ , then the probability of verbal preference will be  $(1-x)$ . The total probability becomes 1, such that  $x + (1-x)$ . Detail of probabilities and their relationship are given in Table 5.

Preferences' probability is calculated from the questions associated with each dimension. The ILS consists of 11 questions for each dimension, having a total of 44 questions for 4 dimensions. The students answer these questions, and if the number of answers associated with preference  $A$  is more than that associated with preference  $B$ , then the user gets learning style having preference  $A$ . The general formula for learning style preference calculation is given by

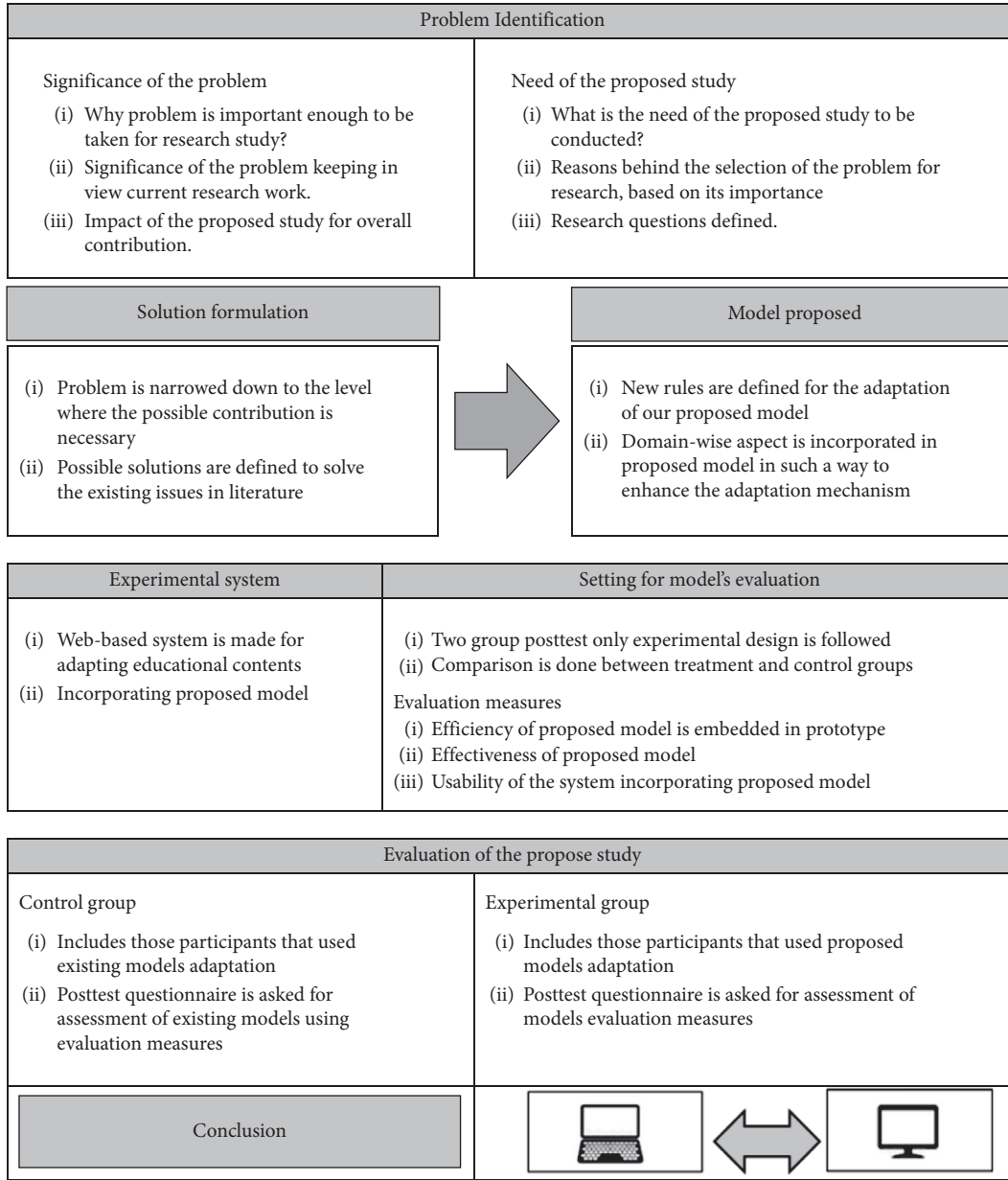


FIGURE 1: Overall approach for research.

TABLE 4: Learning style categories.

S no	Learning style dimension	User type A	User type B
A	The way information is processed	Active	Reflective
B	How contents are represented	Visual	Verbal
C	The way user perceives information	Sensing	Intuitive
D	Learner's perspective on information	Sequential	Global

$$Pr_i = \frac{A_i}{11}, \quad (2)$$

where  $Pr_i$  represents the probability for dimension  $i$ ,  $A_i$  represents the number of answers that favour learning preference  $A$  in dimension  $i$ , and 11 in the equation is constant because the ILS questionnaire assesses each dimension with the help of 11 questions. This procedure is repeated for all the remaining 3 dimensions.

After calculating the preference for all the four dimensions, preferences are initially stored the same for all the subjects as shown in Table 6. When a student uses the system, then the user model updates these preferences based on the difficulty a student faces in the subject. As learning style variations exist for different subjects, therefore, a student may face more difficulty in a subject as compared to the other. In such a case, the proposed model will update these preferences for each subject accordingly.

TABLE 5: Representation of preferences' probabilities.

Dimension	1st preference	2nd preference	Dimension total probability = 1
a	$\text{Prob}_{(\text{Act})} = w$	$\text{Prob}_{(\text{Ref})} = 1 - w$	$\text{Prob}_{(\text{Act})} + \text{Prob}_{(\text{Ref})} = 1$
b	$\text{Prob}_{(\text{Se})} = x$	$\text{Prob}_{(\text{In})} = 1 - x$	$\text{Prob}_{(\text{Se})} + \text{Prob}_{(\text{In})} = 1$
c	$\text{Prob}_{(\text{Vis})} = y$	$\text{Prob}_{(\text{Ver})} = 1 - y$	$\text{Prob}_{(\text{Vis})} + \text{Prob}_{(\text{Ver})} = 1$
D	$\text{Prob}_{(\text{Seq})} = z$	$\text{Prob}_{(\text{Glo})} = 1 - z$	$\text{Prob}_{(\text{Seq0})} + \text{Prob}_{(\text{Glo})} = 1$

TABLE 6: Preference values before.

S. no	Subject	Prob (Act)	Prob (Ref)	Prob (Se)	Prob (In)	Prob (Vis)	Prob (Ver)	Prob (Glo)	Prob (Seq)
1	Subject 1	0.4	0.6	0.30	0.70	0.20	0.80	0.35	0.65
2	Subject 2	0.4	0.6	0.30	0.70	0.20	0.80	0.35	0.65

For the same user, learning style preferences are adjusted according to the requirement of the subject domain. Table 7 shows the learning style preferences for subject 1 and subject 2 for the same user. In order to identify that how much a student faces difficulty in a particular subject, this decision depends upon the performance of the student in the tests just like taking quizzes or exams [13, 14, 40, 47]. In the literature, the main criteria for updating the learning style preferences are based on the assessment of student's performance.

In the proposed model, the assessment criteria for updating learning preferences are based on the student's performance. User preferences are updated when a learning difficulty arises. For example, if a user has good performance in a subject and passes the test, then learning style preferences will not be updated. It will be assumed that the current learning style is suitable and the student is performing well. On the other hand, if the performance is not satisfactory, then the model will be updated. Our model will update learning style preferences only for that particular subject. For achieving this phenomenon, we have proposed some adaptation rules to be incorporated in the user model.

**3.3. Proposed Rules for Adaptation.** For the system's adaptation and updating learning style preferences, we have proposed some rules, which are mathematically represented and shown in Algorithm 1. The key element for decision-making in our approach is the performance of the student. Measuring the performance of the student is represented by "m" which is the desired performance. The brief description is as follows: if a student performance  $\text{Perf}[\text{sub}_j]$  is less than the desired performance  $P_{\text{measure}}$  in a subject  $j$ , then the model has to be updated subtracting the current learning style preferences by some factor  $R[\text{sub}_j]$  and add this  $R[\text{sub}_j]$  to the learning style preference which is missing.  $R[\text{sub}_j]$  is called the reinforcement value.

In Algorithm 1, the description of each element is as follows:

$P_{\text{measure}}$ : performance measure, i.e.,  $P_{\text{measure}} = 60$

$i$ : learning style categories or dimension, which varies in range  $i = 1 \dots 4$

$j$ : numbers given to the subject; if we have  $n$  number of subjects, then  $j$  ranges from  $[1-n]$

$\text{Perf}[\text{sub}_j]$ : performance of the student in  $j^{\text{th}}$  subject;  $\text{Perf}[\text{sub}_j]$  values vary in the range  $[0-100]$

$\text{LSC}[\text{dim}_i][\text{sub}_j]$ : learning style currently used for  $i^{\text{th}}$  dimension and  $j^{\text{th}}$  subject

$\text{SM}[\text{dim}_i][\text{sub}_j]_A$ : learning style for preference  $A$  for  $i^{\text{th}}$  dimension and  $j^{\text{th}}$  subject

$R[\text{sub}_j]$ : the reinforcement to be made for eliminating difficulty for  $j^{\text{th}}$  subject;  $R[\text{sub}_j]$  varies in the range  $[0-1]$  and may not be the same as  $R[\text{sub}_{j+1}]$  or  $R[\text{sub}_{j-1}]$

It is essential to note that the user model will be updated whenever a student does not perform well. Reinforcement  $R[s_j]$ , as we can see in equation (3), has an inverse relationship with student performance as well as the distance between learning style preferences  $\text{DLS}[s_j]$ . Inverse relation is due to the fact that if student performance is too low, then it means that current learning style preference is not a true representation of the user learning style and needs to be changed by a higher factor. In the case of the distance between learning style preferences, inverse relationship makes sense because when this difference is little then learning style preferences are not considered on either side significant.

$$R[\text{sub}_j] = \frac{1}{\text{Perf}[\text{sub}_j] * \text{DLS}[\text{sub}_j]} \quad (3)$$

$\text{DLS}$  can be calculated from equation (4). The value of  $R[\text{sub}_j]$  is restricted not to be too high or too low. In the case of high value, the change in learning style preference will not be gradual. On other hand, in the case of low value, the system will take long time to update the learning style preferences to meet the desire learning style of the user. Based on the experimentation in [45, 46], the maximum value for  $R[\text{sub}_j]$  is suggested to be 0.05 and the minimum value for  $R[\text{sub}_j]$  is 0.02. The proposed model has restricted the higher and lower values for  $R[\text{sub}_j]_{\text{Max}}$  and  $R[\text{sub}_j]_{\text{Min}}$  to 0.05 and 0.02, respectively.

$$\text{DLS}[\text{sub}_j] = \left| \text{SM}[\text{dim}_i][\text{sub}_j]_A - \text{SM}[\text{dim}_i][\text{sub}_j]_B \right| \quad (4)$$

TABLE 7: Preference values after.

S. no	Subject	Prob (Act)	Prob (Ref)	Prob (Se)	Prob (In)	Prob (Vis)	Prob (Ver)	Prob (Glo)	Prob (Seq)
1	Subject 1	0.35	0.65	0.10	0.90	0.60	0.40	0.45	0.55
2	Subject 2	0.70	0.30	0.35	0.65	0.33	0.67	0.25	0.75

```

Result: optimized domain-wise learning style preferences
while (Perf[subj] > Pmeasure) do
  Rule A:
  if (Perf[subj] < Pmeasure) AND
  (LSC[dimi][subj] = "A") then
    SM[dimi][subj]A = SM[dimi][subj]A - R[subj]
    SM[dimi][subj]B = SM[dimi][subj]V + R[subj]
  Rule B:
  if (Perf[subj] < Pmeasure) AND
  (LSC[dimi][subj] = "B") then
    SM[dimi][subj]A = SM[dimi][subj]A + R[subj]
    SM[dimi][subj]B = SM[dimi][subj]V - R[subj]

```

ALGORITHM 1: Algorithm for updating domain-wise learning style preferences.

## 4. Development of Experimental Prototype Application

An experimental prototype is developed to assess the effectiveness of the proposed model. The prototype is developed by using web programming constructs such as PHP, MySQL, JavaScript, and CSS. Details of the prototype are given in the following sections.

**4.1. Selection of Learning Style Model.** For capturing students' learning preferences and profile generation, the ILS questionnaire (proposed by Felder and Silverman) is used. The reason behind its selection is that learning style categories and types of users are more suitable for the coursework hypermedia [11, 13, 48, 49]. ILS model has also the suitability to the user's categories as well as the nature of educational contents. Detail of its categories and users' preferences is given in Table 4. Identification of user preferences, content adaptation, and updating preferences has been done using the procedures and rules discussed in Section 3.

**4.2. Dataset.** For the developed system, the dataset is developed from the educational contents of the Virtual University of Pakistan. The dataset includes text documents, PowerPoint slides, and videos. Educational contents from the selected source are more reliable as compared to the other Internet sources because these are prepared by subject experts, specialists, and professors. For the assessment of our proposed model, videos and textual document are used for subjects' lectures. PowerPoint presentations are not used as a learning material because they did not have much explanation regarding the concept. In addition, for the inclusion/exclusion of the educational contents in our prototype, suggestions from the experts were also taken into account. The experimental prototype setup is made only for two

subjects, that is, database and programming. Furthermore, the prototype is capable of any extension to include more subjects. Videos and lectures are further refined to the topic level, and each topic has two forms of representation because topics can provide the basis for successful personalization [50]. The hierarchical structure of these contents is shown in Figure 2.

**4.3. Prototype Architecture.** The main architecture of the prototype consists of the main four components as shown in Figure 3. A brief description of each of the components is given as follows.

**4.3.1. User Model.** It is the most significant module of the experimental prototype. The user model is responsible for capturing learning style preferences of the user. The adaptation module uses these learning style preferences to select educational contents. These preferences are updated time to time.

**4.3.2. Database of Contents.** This component acts as a repository for the system. Different forms of educational content are kept in this module. After identifying learning style preferences from the user model, the adaptation module then filters contents from the database to provide educational material according to the user's preferences.

**4.3.3. Interface Module.** It provides an interface for students to interact with the system. It is responsible to capture information through the ISL questionnaire. The captured information is then transformed into the user model. Contents filtered by the adaptation model are represented by interface module. The responsibility of the interface module is to present the adapted contents to the user, to capture user



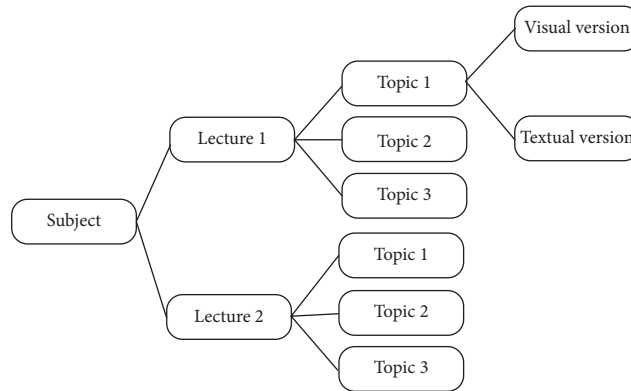


FIGURE 2: Hierarchical representation of course contents.

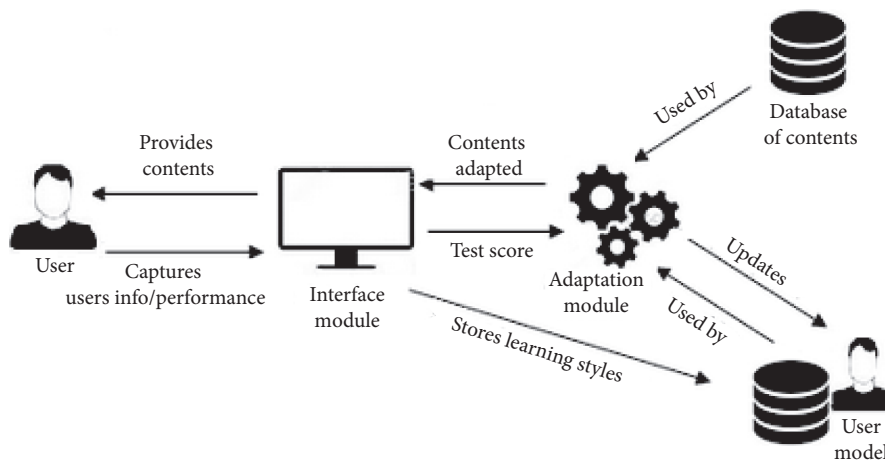


FIGURE 3: Architecture of prototype.

learning style preferences, and to provide test sessions to the user. The interface layout of our application is shown in Figure 4.

**4.3.4. Adaptation Module.** The adaptation module consists of rules which filter and select contents from the database. It is the decision-making component of the system. It exploits information of the user model, database of contents, and information coming from the interface model. Using the information of different components, it decides the suitability of contents and updates learning style preferences in order to provide more accurate system adaptation.

**4.4. Adaptation Mechanism.** The prototype has main four components which are discussed above, and these components are working together around the following four steps.

**4.4.1. User Profile Generation.** In the first step, a new user is asked to register himself and create his profile, a questionnaire is provided by the interface module, and the user provides information about learning style preferences. Based on the user information, a category is assigned which is then used by the adaptation module.

**4.4.2. Providing Adaptive Contents.** After capturing user preferences, an initial model for the user is generated. Contents are selected/filtered and provided to the user. Our model is capable of handling subject-wise learning style preferences, so different contents can be selected for different subjects based on the stored learning style preferences.

**4.4.3. Assessment of User Learning.** One of the most significant tasks is to know the effectiveness of the system. For this purpose, at the end of each session, the user is asked to perform a test. If user performance is satisfactory, then the next lecture is unlocked and learning styles are not updated. It is assumed that current system adaptation is effective for his learning as his performance is satisfactory. In contrast, if the student failed the test, then its learning style preferences are updated in order to optimize the suitability of system adaptation with user learning style. After updating learning styles, the same lecture is provided with more filtered parameters. The user is asked to study the lecture again.

**4.4.4. Updating User Preferences Domain-Wise.** The proposed model updates user preferences according to the requirement of the subject domain. In our case, when a

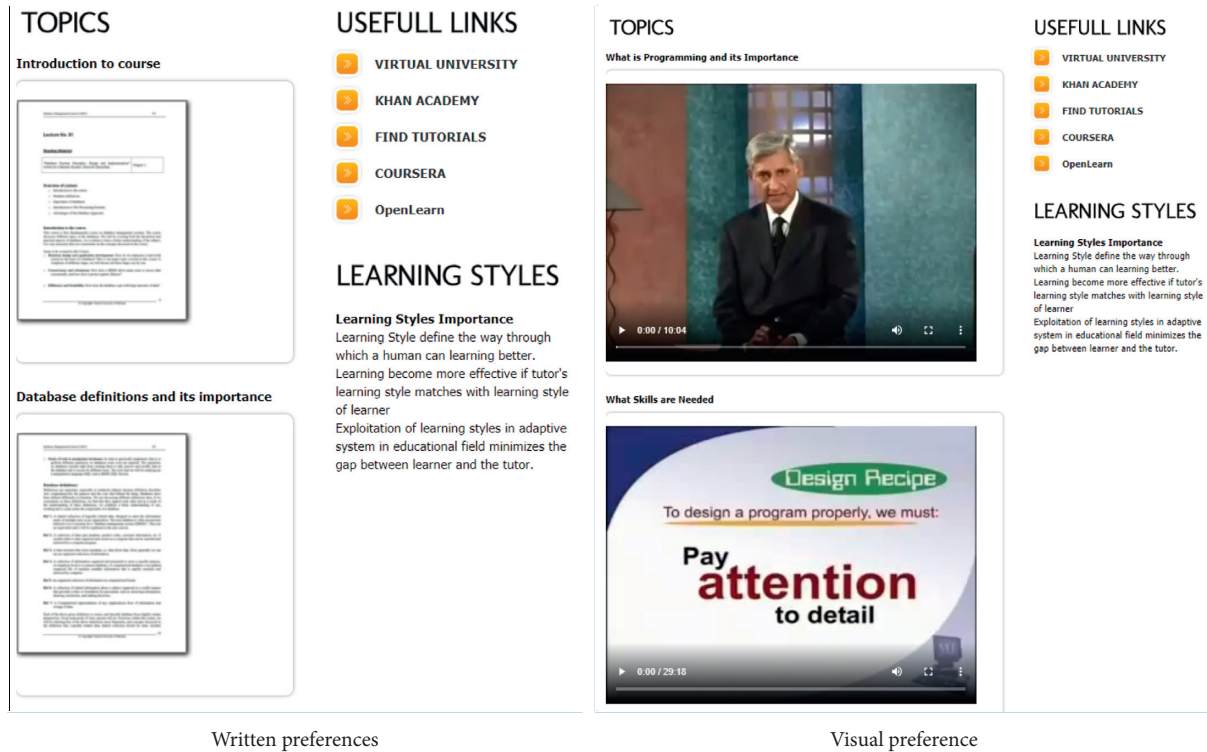


FIGURE 4: Interface for visual and written preferences.

student does not perform well in a subject, then preferences are updated only for that subject. This approach updates preferences in a more optimized way and reflects the true needs of learners.

## 5. Research Design

The proposed model is evaluated by comparing with the existing learning style model using two-group posttest only randomized experimental design. Among 43 students, finally, 30 participants were chosen randomly for the study. These participants are undergraduate students from the computer discipline. Questionnaire (ILS) is used for the identification of participants' learning style preferences. For identifying 30 students for the control and experimental groups with unbiased learning style preferences, a pretest is taken from 43 students. After achieving the required number of 15 visual and 15 verbal learning preferences for students, these participants are randomly assigned to the control and experimental groups. At the end, each group has half participants of each learning preference in order to avoid group difference formation. To see the distribution of the control and experimental groups, independent-samples  $T$ -test is applied. For 95% confidence interval and 0.05 value of alpha, the  $p$  value is obtained 0.860 which shows that the difference among groups is not significant. This shows that the group's formation is unbiased which ensures the validity of the evaluation process. Figure 5 shows the overview of the research design followed for the evaluation of the proposed model.

**5.1. Overview of Research Design.** Participants of the control group have used application based on the existing models, for one week. On the other hand, the experimental group used our proposed experimental prototype application for the lectures. After a week, postsurvey was done from both groups' participants to assess that whether the proposed system is effective in enhancing the learning of the students. Postsurvey includes qualitative as well as quantitative measure for the assessment of the proposed model. In addition, the performance history is also kept into account to know the level of difficulty the users face in these subjects.

**5.1.1. Evaluation Criteria for the Proposed Model.** The usefulness of the proposed model is measured by using three aspects. These aspects are efficiency, effectiveness, and user satisfaction [2, 3]. A total of 13 questions were used for these aspects. Efficiency is measured using task-based measures, six questions were aimed to address the effectiveness, and the remaining four questions were for the assessment of user satisfaction. The detail of the evaluation measures is shown in Table 8.

## 6. Results and Discussions

The proposed model is compared with the existing models in the literature. For this purpose, students were divided into two groups. One group is the control group and the other is the experimental group. The control group was assigned the task to learn in one-week duration using existing adaptation, while our developed experimental prototype is given to the

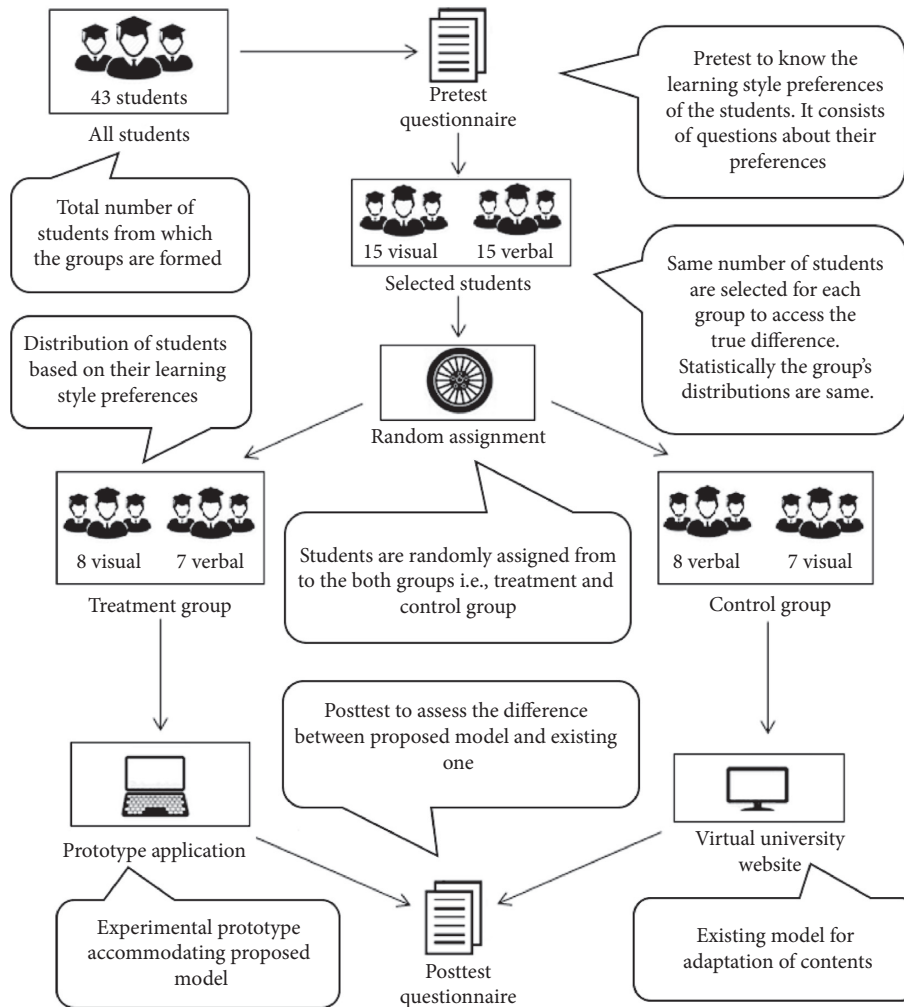


FIGURE 5: Overview of research design.

TABLE 8: Evaluation measures.

S.no	Evaluation criteria	Purpose
1	Performance efficiency	Q2: time to be taken for the identification of lecture topic Q8: navigation efficiency Q5: overall effect
2	Satisfaction of the user	Q6: the way contents are presented Q11: scalability towards other disciplines Q12: future scope
3	Systems effectiveness	Q1: effect of students motivation level Q3: fairness of performance assessment Q4: usefulness of the system during usage Q7: system capability to pinpoint weaknesses Q9: system guidance for problem-solving Q10: reducing memory load upon user
4	Comments	Q13: suggestions to improve the application in future

experimental group to do the same task. At the end of one week, the performance of both groups along with postsurvey was evaluated to assess the effectiveness of the proposed model for enhancing learning. The model is evaluated based on the aspects as given in Table 8.

For the assessment of each evaluation measure, we have performed statistical tests to know its effectiveness. To

achieve that, we have used the IBM SPSS tool for performing these tests. Based on the nature of data and scale of measurement taken, *independent-samples Mann-Whitney U test* is applied in our study's setting. Null hypothesis is considered for each variable. In the case of significant effect size, the test rejects the null hypothesis for the observed variable. Detail of these tests is discussed in subsequent sections.

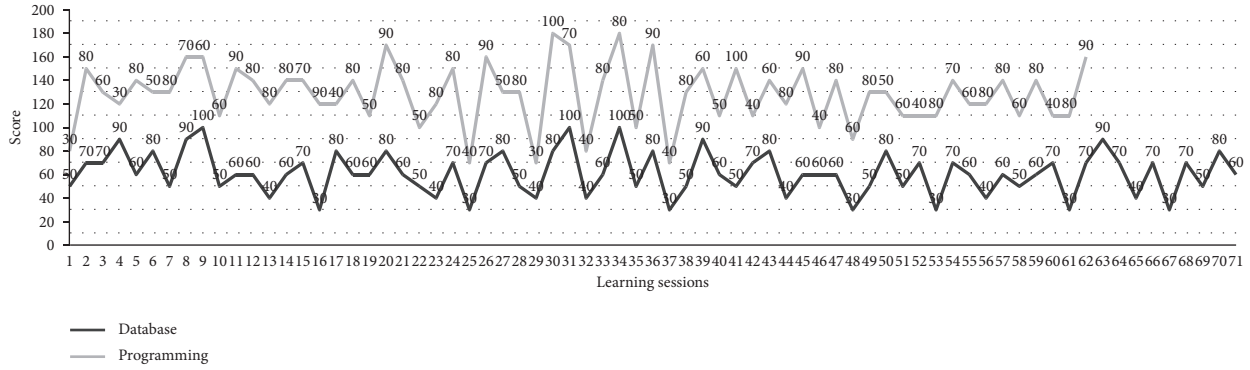


FIGURE 6: Domain-wise performance.

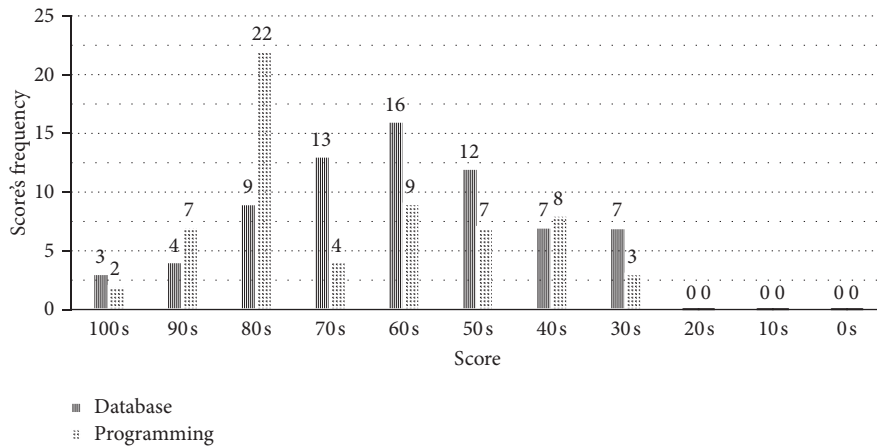


FIGURE 7: Student performance score.

6.1. *Domain-Wise Student Performance.* As we have discussed that learning styles vary domain-wise. For this purpose, the history of the student learning session and its performance were recorded for each learning session. After each learning session, student performance was assessed through the test. If a student did not score (passing criteria), then the next lecture is not available for the user and was instructed to learn the same lecture again and pass the test. In case of success, the next lecture is provided. Using these criteria, students were instructed to complete the lessons as shown in Figure 6. Furthermore, detail of students' performance is shown in Figure 7.

At the end, we have compared the performances and learning sessions required for both subjects to know whether our claim for domain-wise learning style variation is true or not. From the analysis, it has been identified that there are significant performance differences in these subjects for students as shown in Figure 6. We can see that, in *Database* subject, student faces more difficulty and learning styles have been changed time to time while on other hand, the students face less difficulty and complete the lectures in a less number of sessions.

The difference is assessed via a statistical test for the confidence interval of 95%. Table 9 shows the detail of the test, the null hypothesis for each variable, and the results of the test. From Table 9, we can see that the null hypothesis is

rejected for students' performance score which shows that difference is significant and students do face different levels of difficulty for the subject domain. It proves that learning styles need to be adjusted for each subject domain accordingly to enhance system effectiveness.

6.2. *Performance Efficiency.* Performance efficiency includes those aspects that define how much the proposed model enhances the efficiency of the system. These aspects include navigation efficiency, which is assessed that how much time a user takes to identify all individual topics of a lecture. Another measure for navigation efficiency is that how quickly a user navigates among different topics for understanding a lesson. These assessments were task-based, and time has been recorded for each user activity for both groups. Based on the evaluation, the proposed model significantly enhances the navigation efficiency of the system as shown in Figure 8.

Details of the test for efficiency are given in Table 10. We have performed *independent-samples Mann-Whitney U test* on the SPSS software for confidence interval with 95%. The test rejects the null hypothesis for  $p$  value less than 0.05 for both factors as shown in Table 10 which means that the difference is statistically significant and our models enhance the navigation efficiency of the system.

TABLE 9: Students' performance analysis.

S. no	Null hypothesis for the measure	Test conducted	Sig.	Result
1	User performance is the same for both groups Level of significance is 0.05	Independent-samples Mann-Whitney $U$ test	.043	Null hypothesis is rejected

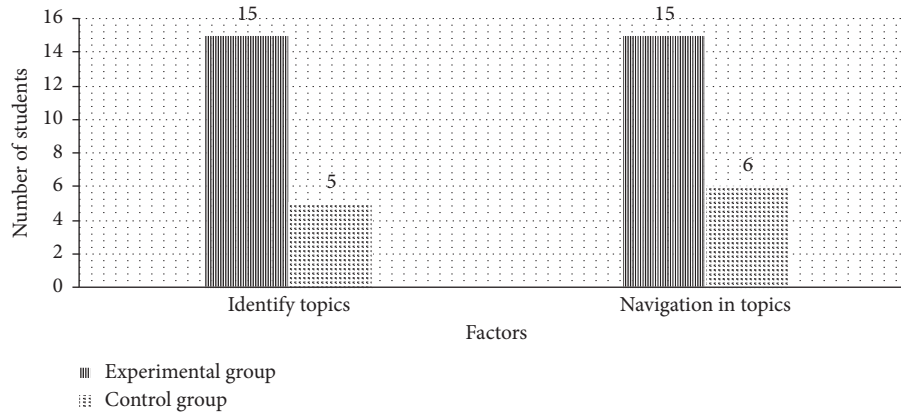


FIGURE 8: Efficiency of the system.

TABLE 10: Efficiency analysis.

S.no	Null hypothesis for the measure	Test conducted	Sig.	Result
1	Time required for identifying topics is the same for both groups	Independent-samples Mann-Whitney $U$ test	.000 <sup>1</sup>	Null hypothesis is rejected
2	The distribution of navigation is the same across categories of group type	Independent-samples Mann-Whitney $U$ test	.000 <sup>1</sup>	Null hypothesis is rejected

<sup>1</sup>The significance level is 0.05.

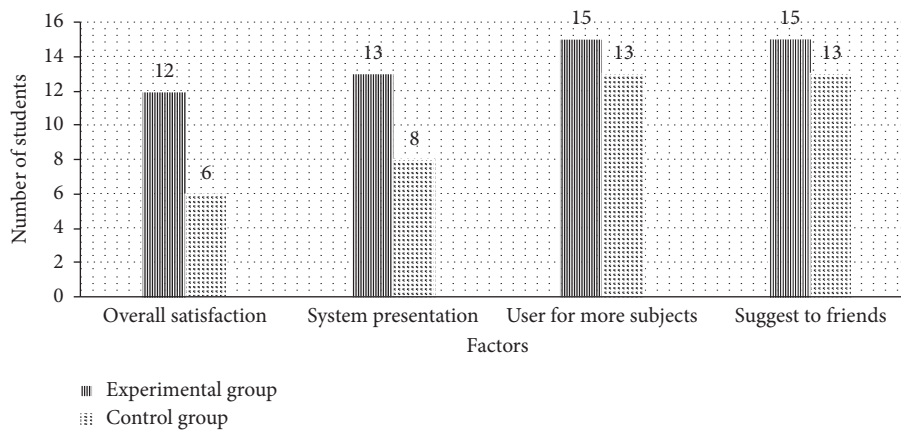


FIGURE 9: User satisfaction.

**6.3. User Satisfaction.** For user satisfaction, responses have been taken from both groups to know how much our experimental prototype fulfills the requirements of a user. Based on the statistics of responses, contents are better presented to the user, and the users suggest that the system will be better to incorporate other subjects too. We can see that measures taken for user satisfaction are enhanced for the experimental group as shown in Figure 9.

For ensuring that improvement is enhanced, we have applied independent-samples Mann-Whitney  $U$  test for all measures associated with user satisfaction for 95%

confidence interval. The null hypothesis is rejected as the value of  $p$  is less than 0.05, which is shown in Table 11. It shows that the proposed prototype provides a better user experience.

**6.4. Model Effectiveness.** This is one of the most important aspects of the evaluation of the proposed model. Six measures were taken for assessment of the effectiveness. These measures include motivation, fairness of marking, usefulness, pinpoint weakness, memory load, and systems

TABLE 11: Analysis of user satisfaction.

S.no	Null hypothesis for the measure	Test conducted	Sig.	Result
1	User satisfaction is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.033 <sup>1</sup>	Null hypothesis is rejected
2	Level of the proposed system scalability is similar for both groups.	Independent-samples Mann–Whitney <i>U</i> test	0.033 <sup>1</sup>	Null hypothesis is rejected
3	Content presentation is the same for both groups.	Independent-samples Mann–Whitney <i>U</i> test	0.045 <sup>1</sup>	Null hypothesis is rejected

<sup>1</sup>The significance level is 0.05.

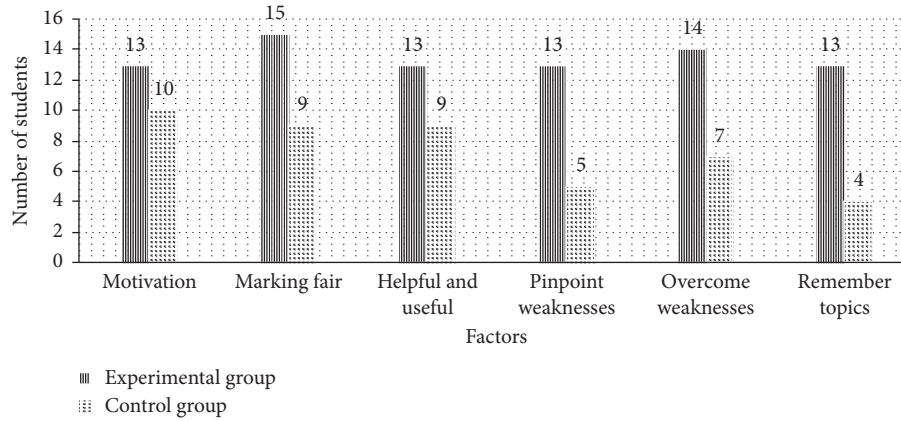


FIGURE 10: Effectiveness.

TABLE 12: Analysis of model's effectiveness.

S. no	Null hypothesis for the measure	Test conducted	Sig.	Result
1	Motivation is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.285 <sup>1</sup>	Null hypothesis is not rejected
2	Marking fairness is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.003 <sup>1</sup>	Null hypothesis is rejected
3	Level of usefulness is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.137 <sup>1</sup>	Null hypothesis is not rejected
4	Level of pinpointing weakness by the system is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.019 <sup>1</sup>	Null hypothesis is rejected
5	Overcoming the weakness from the system is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.037 <sup>1</sup>	Null hypothesis is rejected
6	Reducing memory load is the same for both groups	Independent-samples Mann–Whitney <i>U</i> test	0.000 <sup>1</sup>	Null hypothesis is rejected

<sup>1</sup>The significance level is 0.05.

suggestions to overcome weakness. Figure 10 shows the comparisons for both the experimental and control groups.

However, to know that the improvement is meaningful, we have performed independent-samples Mann–Whitney *U* test for its relative difference. The overall assessment is shown in Table 12, where we can see that the null hypothesis is rejected for measures such as fairness of assessment, pinpoint weakness, overcoming the weakness, and memory load, which shows that these factors are significantly enhanced during the experimental group's learning process. On the other hand, measures such as motivation and usefulness are enhanced, but statistically, their difference is not relatively significant which means that our prototype performs relatively better than the existing one.

## 7. Conclusion

Understanding student's learning preferences help us to know the way by which learner can learn easily and effectively. For the effectiveness of learning systems in the educational domain such as adaptive hypermedia, learning styles are used for content adaptation according to the user learning style. But learning styles have domain-wise variations and approaches based on learning styles mainly work in the context of single domain. We have provided a more effective means by incorporating its changeable aspects such as domain-wise learning style preferences in the user model.

To achieve the objective, we have proposed a robust and flexible model to update student learning style preferences

according to their discipline. A web-based experimental prototype is developed for the assessment and validation of the proposed model. The proposed model is compared, and the experimental results show that personalization based on discipline-wise learning style variations becomes more effective. The results also show that adaptation based on domain-wise learning style variations enhances the learning outcome of these systems. The proposed research can be used in making the learning systems more intelligent for adapting contents. Furthermore, the proposed model also provides a foundation for upcoming learning-based solutions which required more refined adaptation rules as well as user models to incorporate features that are the true representations of their needs.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding this paper.

### References

- [1] C. Froschl, "User modeling and user profiling in adaptive E-learning systems," Master Thesis, Springer, Graz, Austria, 2005.
- [2] E. Knutov, P. De Bra, and M. Pechenizkiy, "AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques," *New Review of Hypermedia and Multimedia*, vol. 15, no. 1, pp. 5–38, 2009.
- [3] C. Mulwa, S. Lawless, M. Sharp, I. Arnedillo-Sanchez, and V. Wade, "Adaptive educational hypermedia systems in technology enhanced learning: a literature review," in *Proceedings of the 2010 ACM Conference on Information Technology Education*, pp. 73–84, ACM, New York, NY, USA, October 2010.
- [4] E. Kurilovas, I. Zilinskiene, and V. Dagiene, "Recommending suitable learning scenarios according to learners' preferences: an improved swarm based approach," *Computers in Human Behavior*, vol. 30, pp. 550–557, 2014.
- [5] V. Štūkys, "Context-aware adaptation of smart los," in *Proceedings of the Smart Learning Objects For Smart Education In Computer Science*, pp. 161–181, Springer, Berlin, Germany, November 2015.
- [6] E. Kurilovas, I. Zilinskiene, and V. Dagiene, "Recommending suitable learning paths according to learners' preferences: experimental research results," *Computers in Human Behavior*, vol. 51, pp. 945–951, 2015.
- [7] E. Kurilovas, S. Kubilinskiene, and V. Dagiene, "Web 3.0-based personalisation of learning objects in virtual learning environments," *Computers in Human Behavior*, vol. 30, pp. 654–662, 2014.
- [8] A. Jeghal, L. Oughdir, H. Tairi, and A. Radouane, "Approach for using learner satisfaction to evaluate the learning adaptation policy," *International Journal of Distance Education Technologies*, vol. 14, no. 4, pp. 1–12, 2016.
- [9] A. Alam, S. Ullah, and N. Ali, "The effect of learning-based adaptivity on students' performance in 3D-virtual learning environments," *IEEE Access*, vol. 6, pp. 3400–3407, 2018.
- [10] C. Jones, C. Reichard, and K. Mokhtari, "Are students' learning styles discipline specific?" *Community College Journal of Research and Practice*, vol. 27, no. 5, pp. 363–375, 2003.
- [11] H. M. El-Bakry, A. A. Saleh, T. T. Asfour, and N. Mastorakis, "A new adaptive e-learning model based on learner's styles, in," in *Proceedings of the 13th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering (MACMESE'11)*, pp. 440–448, Catania, Sicily, Italy, November 2011.
- [12] C. Martins, L. Faria, C. V. De Carvalho, and E. Carrapatoso, "User modeling in adaptive hypermedia educational systems," *Educational Technology & Society*, vol. 11, no. 1, pp. 194–207, 2008.
- [13] H. M. El-Bakry and A. A. Saleh, "Adaptive e-learning based on learner's styles," *Bulletin of Electrical Engineering and Informatics*, vol. 2, no. 4, pp. 240–251, 2013.
- [14] M. A. Fernandes, C. R. Lopes, F. A. Dorca, and L. V. Lima, "A stochastic approach for automatic and dynamic modeling of students' learning styles in adaptive educational systems," *Informatics in Education-An International Journal*, vol. 112, pp. 191–212, 2012.
- [15] A. Abyaa, M. K. Idrissi, and S. Bennani, "Learner modelling: systematic review of the literature from the last 5 years," *Educational Technology Research and Development*, vol. 67, pp. 1–39, 2019.
- [16] A. C. Wintergerst, A. DeCapua, and M. Ann Verna, "Conceptualizing learning style modalities for esl/efl students," *System*, vol. 31, no. 1, pp. 85–106, 2003.
- [17] D. W. Salter, N. J. Evans, and D. S. Forney, "A longitudinal study of learning style preferences on the myers-briggs type indicator and learning style inventory," *Journal of College Student Development*, vol. 47, no. 2, pp. 173–184, 2006.
- [18] E. P. Rozanski and A. R. Haake, "The many facets of hci," in *Proceedings of the 4th Conference on Information Technology Curriculum*, pp. 180–185, ACM, New York, NY, USA, October 2003.
- [19] E. Sadler-Smith and R. Riding, "Cognitive style and instructional preferences," *Instructional Science*, vol. 27, no. 5, pp. 355–371, 1999.
- [20] A. Y. Kolb, *The kolb learning style inventory-version 3.1 2005 technical specifications*, Hay Resource Direct, Boston, MA, USA, 2005.
- [21] H. A. Witkin, *A Manual for the Embedded Figures Tests*, Consulting Psychologists Press, Palo Alto, CA, USA, 1971.
- [22] P. Brusilovsky, E. Schwarz, and G. Weber, *ELM-art: An Intelligent Tutoring System on World Wide Web*, in *Intelligent Tutoring Systems*, Springer, Berlin, Germany, 1996.
- [23] R. M. Carro, E. Pulido, and P. Rodriguez, *Tangow: Task-Based Adaptive Learner Guidance on the Www*, *Computer Science Report*, Eindhoven University of Technology, Eindhoven, Netherlands, 1999.
- [24] I. H. Beaumont, "User modelling in the interactive anatomy tutoring system anatom-tutor," *User Modelling and User-Adapted Interaction*, vol. 4, no. 1, pp. 21–45, 1994.
- [25] M. R. Zakaria, A. Moore, H. Ashman, C. Stewart, and T. Brailsford, *The Hybrid Model for Adaptive Educational Hypermedia*, in *Adaptive Hypermedia and Adaptive Web-based Systems*, Springer, Berlin, Germany, 2002.

- [26] P. D. Bra and L. Calvi, "Aha! an open adaptive hypermedia architecture," *New Review of Hypermedia and Multimedia*, vol. 4, no. 1, pp. 115–139, 1998.
- [27] J. Kay and B. Kummerfeld, "Adaptive hypertext for individualised instruction," in *Proceedings of the Workshop On Adaptive Hypertext And Hypermedia At Fourth International Conference On User Modeling*, Hyannis, MA, USA, August 1994.
- [28] H. Hohl, H.-D. Boecker, and R. Gunzenhaeuser, *Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming*, in *Adaptive Hypertext And Hypermedia*, Springer, Berlin, Germany, 1998.
- [29] C. Boyle and A. O. Encarnacion, *Metadoc: An Adaptive Hypertext Reading System*, in *Adaptive Hypertext And Hypermedia*, Springer, Berlin, Germany, 1998.
- [30] R. Zeiliger, *Adaptive Testing: Contribution of the Shiva Model*, In *Item Banking: Interactive Testing and Self-Assessment*, Springer, Berlin, Heidelberg, Germany, 1993.
- [31] G. Weber, H.-C. Kuhl, and S. Weibelzahl, *Developing Adaptive Internet Based Courses with the Authoring System Netcoach*, In *Hypermedia: Openness, Structural Awareness, and Adaptivity*, Springer, Berlin, Heidelberg, Germany, 2002.
- [32] P. Brusilovsky, E. Schwarz, and G. Weber, "A tool for developing adaptive electronic textbooks on www," *WebNet*, vol. 96, pp. 64–69, 1996.
- [33] P. Brusilovsky and L. Pesin, "Isis-tutor: an adaptive hypertext learning environment," in *Proceedings of the JCKBSE'94, Japanese-CIS Symposium on Knowledge-Based Software Engineering*, vol. 94, pp. 10–13, Pereslavl-Zalesski, Russia, May 1994.
- [34] P. Brusilovsky, L. Pesin, and M. Zyryanov, *Towards an Adaptive Hypermedia Component for an Intelligent Learning Environment in Human-computer Interaction*, pp. 348–358, Springer, Berlin, Germany, 1993.
- [35] S. Ferrandino, A. Negro, V. Scarano, and Cheops, "Adaptive hypermedia on world wide web," In *Interactive Distributed Multimedia Systems And Telecommunication Services*, Springer, Berlin, Heidelberg, Germany, 1997.
- [36] E. J. Brown, T. J. Brailsford, T. Fisher, and A. Moore, "Evaluating learning style personalization in adaptive systems: quantitative methods and approaches," *IEEE Transactions on Learning Technologies*, vol. 2, no. 1, pp. 10–22, 2009.
- [37] P. Brusilovsky, "Methods and techniques of adaptive hypermedia," *User Modeling and User-Adapted Interaction*, vol. 6, no. 2-3, pp. 87–129, 1996.
- [38] B. Peter, *Adaptive Hypermedia: an Attempt to Analyze and Generalize*, in *Multimedia, Hypermedia, and Virtual Reality Models, Systems, and Applications*, Springer, Berlin, Heidelberg, Germany, 1996.
- [39] A. Kavcic, "The role of user models in adaptive hypermedia systems," vol. 1, pp. 119–122, in *Proceedings of the MELECON 2000 10th Mediterranean*, vol. 1, pp. 119–122, IEEE, Lemesos, Cyprus, May 2000.
- [40] V. Esichaikul, S. Lamnoi, and C. Bechter, "Student modelling in adaptive e-learning systems," *Knowledge Management & E-Learning: An International Journal (KM&EL)*, vol. 3, no. 3, pp. 342–355, 2011.
- [41] Y. Long and V. Aleven, "Enhancing learning outcomes through self-regulated learning support with an open learner model," *User Modeling and User-Adapted Interaction*, vol. 27, no. 1, pp. 55–88, 2017.
- [42] E. Kurilovas, J. Kurilova, and T. Andruskevici, *On Suitability Index to Create Optimal Personalised Learning Packages*, In *International Conference On Information And Software Technologies*, Springer, Berlin, Germany, 2016.
- [43] R. Z. Cabada, M. L. B. Estrada, F. G. Hernández, R. O. Bustillos, and C. A. Reyes-García, "An affective and web 3.0-based learning environment for a programming language," *Telematics and Informatics*, vol. 35, no. 3, pp. 611–628, 2018.
- [44] J. D. Guerra Hollstein, *Open learner models for self-regulated learning: exploring the effects of social comparison and granularity*, Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA, USA, 2018.
- [45] F. A. Dorça, L. V. Lima, M. A. Fernandes, and C. R. Lopes, "A new approach to discover students learning styles in adaptive educational systems," *Revista Brasileira de Informática na Educação*, vol. 21, no. 1, p. 76, 2013.
- [46] F. A. Dorça, L. V. Lima, M. A. Fernandes, and C. R. Lopes, "Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: an experimental analysis," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2092–2101, 2013.
- [47] P. Anthony, N. E. Joseph, and C. Ligadu, "Learning how to program in c using adaptive hypermedia system," *International Journal of Information and Education Technology*, vol. 3, no. 2, pp. 151–155, 2013.
- [48] S. Graf, S. R. Viola, and T. L. Kinshuk, "Representative characteristics of felder-silverman learning styles: an empirical model," in *Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2006)*, pp. 235–242, Barcelona, Spain, June 2006.
- [49] C. A. Carver, R. A. Howard, and W. D. Lane, "Enhancing student learning through hypermedia courseware and incorporation of student learning styles," *IEEE Transactions on Education*, vol. 42, no. 1, pp. 33–38, 1999.
- [50] S. Sosnovsky and P. Brusilovsky, "Evaluation of topic-based adaptation and student modeling in quizguide," *User Modeling and User-Adapted Interaction*, vol. 25, no. 4, pp. 371–424, 2015.



## Research Article

# Isolated Handwritten Pashto Character Recognition Using a $K$ -NN Classification Tool based on Zoning and HOG Feature Extraction Techniques

Juanjuan Huang <sup>1</sup>, Ihtisham Ul Haq,<sup>2</sup> Chaolan Dai,<sup>1</sup> Sulaiman Khan <sup>2</sup>, Shah Nazir <sup>2</sup>, and Muhammad Imtiaz<sup>2</sup>

<sup>1</sup>Hunan Police Academy, Changsha 410138, China

<sup>2</sup>Department of Computer Science, University of Swabi, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Juanjuan Huang; [huangjj@hnpa.edu.cn](mailto:huangjj@hnpa.edu.cn) and Sulaiman Khan; [enr.sulaiman88@gmail.com](mailto:enr.sulaiman88@gmail.com)

Received 3 February 2021; Revised 22 February 2021; Accepted 16 March 2021; Published 24 March 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Juanjuan Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Handwritten text recognition is considered as the most challenging task for the research community due to slight change in different characters' shape in handwritten documents. The unavailability of a standard dataset makes it vaguer in nature for the researchers to work on. To address these problems, this paper presents an optical character recognition system for the recognition of offline Pashto characters. The problem of the unavailability of a standard handwritten Pashto characters database is addressed by developing a medium-sized database of offline Pashto characters. This database consists of 11352 character images (258 samples for each 44 characters in a Pashto script). Enriched feature extraction techniques of histogram of oriented gradients and zoning-based density features are used for feature extraction of carved Pashto characters.  $K$ -nearest neighbors is considered as a classification tool for the proposed algorithm based on the proposed feature sets. A resultant accuracy of 80.34% is calculated for the histogram of oriented gradients, while for zoning-based density features, 76.42% is achieved using 10-fold cross validation.

## 1. Introduction

In this modern digital age of ever-growing computer technology, the machine learning algorithms play a key role in all fields of life, especially in the areas of text recognition [1], network security [2, 3], privacy [4], traffic flow predictions [5], object detection [6], and many others. One of the major applications of machine learning algorithm is Optical Character Recognition (OCR) system development. The OCR system reads the text from an image and converts it into a computer-readable form. Several research works have been addressed on the automatic recognition of multiple languages such as Arabic, English, Persian, Chinese, and Urdu [7, 8]. The main problems associated with these languages are the cursive writing styles, writer's handwriting habits, and secondary components (diacritics). The Pashto language has incorporated most of the Arabic, Urdu, and

Persian letters with some minor modifications. Due to this reason of incorporation of letters, the Pashto language is cursive in nature. The Pashto language consists of a large character set (44 characters) greater than Urdu (38 characters), Arabic (28 characters), and Persian (32 characters). This large character script and minor change in character shape make the recognition process more complex for Pashto script.

Pashto is the maternal language of a large community of residents in Northern areas of Pakistan and official language of Afghanistan. Ahmad et al. [9] used  $k$ -nearest neighbors ( $k$ -NNs) as a classification tool for printed Pashto character recognition by using high-level feature extraction techniques. Boulid et al. [7] suggested the use of a neural network with spatial distribution of pixels (SDPs) and local binary patterns (LBPs) for the recognition of handwritten Arabic characters. Boufenar et al. [10, 11] presented an artificial

immune recognition (AIR) system based on both statistical and structural features for handwritten Arabic letters recognition. Askari et al. [12] introduced the derivative projection profile (DPP) as a feature extractor technique and neural network as a classification tool for isolated Arabic character recognition. El-Sawy presented an Arabic OCR system by using convolutional neural networks [13].

Boufenar and Batouche proposed a deep learning convolutional neural network (DCNN) for Arabic letters recognition [14]. The performance of the suggested system is dependent on hyperparameter tuning and the size of the dataset in use. Naz et al. [8] suggested an approach of the convolution neural network and recursive neural network for Urdu Nastaliq text recognition. They tested the system on the famous Urdu printed text-line image (UPTI) dataset. Sarvaramini et al. [15] suggested the use of the convolutional neural network (CNN) for offline Persian character recognition.

This paper presents an OCR system for offline Pashto characters. A medium-sized database of handwritten Pashto characters is developed for the proposed research work. Two enriched feature extraction techniques of Histogram of Oriented Gradients (HOGs) and zoning-based density features are used for feature set generation. A 10-fold cross validation using  $k$ -nearest neighbors ( $k$ -NNs) is considered as a classification tool based on the feature set calculated using the proposed feature extraction algorithms to evaluate the proposed system performance. Objectives (significance) of the proposed work are listed below:

- (i) To develop a medium-sized database of 11352 (258 samples for each character) for Pashto handwritten characters
- (ii) To provide a benchmark for the identification of handwritten Pashto characters using histogram of oriented gradients and zoning features and the  $k$ -NN classifier
- (iii) To evaluate system performance and provide results that would help open the gate towards handwritten Pashto character recognition

This paper is organized in sections as follows: Section 2 gives a brief detail of the related research work. Section 3 explains the proposed methodology for image extraction, feature set generation, and classification. The results are discussed in Section 4 followed by conclusions in Section 5.

## 2. Literature Review

Recent research shows prominent improvements in OCRs for many languages, especially which are cursive in nature. These include Arabic, Urdu, Persian, and others that possess same cursive nature script. Optical character recognition (OCR) systems convert images of text into a computer-readable form. The recognition rate of cursive scripts is low as compared to noncursive scripts such as English. This is due to the ambiguity in writing styles. Recent work on OCR for cursive-text-based languages achieved significant results, which are discussed below.

Khan et al. [16] provided a baseline study for handwritten Pashto character recognition using zoning features and three different classifier models. The proposed model showed an accuracy of 56% for support vector machine, 78% for an artificial neural network, and 80.7% for a convolution neural network. A dataset of 4488 characters was used for simulation purposes in their research work. Bhuiyan and Alsaade [17] suggested a hybrid neural network model for Arabic character recognition. They used a hybrid neural network by combining a bidirectional associate memory (BAM) and a multilayer perceptron (MLP). Tavoli et al. [18] proposed a new feature extractor for the recognition of Arabic and Persian words, namely, the statistical geometric components of straight lines (SGCSLs) technique. Oujaoura et al. [19] suggested a method for offline Arabic letters identification using three feature extraction techniques including Zernike moments in conjunction with neural networks. Zernike moments surpass rest of the two in recognition rate.

Boufenar et al. [10] have conducted a study for handwritten Arabic character recognition on the famous Offline Isolated Handwritten Arabic Character (OIHACDB) and Arabic Handwritten Character Database (AHCD) datasets using Deep Convolutional Neural Networks (DCNN) and showed state-of-the-art accuracy using this method. Younis presented a DCNN for handwritten Arabic character recognition [20]. He also performed batch normalization to prevent overfitting. The model was tested on AIA9K and AHCD datasets.

Jebri et al. [21] used histogram of oriented gradients (HOGs) as features and support vector machines as classifier on a self-made database. Althobaiti and Lu suggested a novel approach to feature extraction using an encoded freeman chain code and change of tangent for isolated handwritten Arabic character recognition [22, 23]. Jehangir et al. [24] proposed Zernike moments for feature extraction purposes and linear discriminant analysis for the automatic recognition of the handwritten Pashto text.

Naz et al. [25] presented the use of 2-dimensional long short term memory (2DMLSTM) networks for Urdu script recognition based on zoning features. The referenced model is tested on the Urdu Printed Text line Images (UPTI) dataset. Ahmed et al. [26] presented an algorithm for Urdu character recognition using bidirectional long short-term memory (BLSTM) on the Urdu nasta'liq handwritten dataset (UNHD). Jameel and Kumar proposed basis spline (B-spline) curves for Urdu character recognition [27]. Nawaz et al. [28] compared siamese and triplet networks and showed performance improvement when combined with a CNN for handwritten Urdu character recognition.

This work is based on offline Pashto character recognition using  $k$ -nearest neighbors ( $k$ -NNs) as a classification tool. The histogram of oriented gradients (HOGs) and zoning-based density-based feature extraction techniques is followed as a features extraction tools in the proposed research work. This work proposes a character database of 11352 character images (258 samples for each 44 characters in the Pashto language). This work also uses both HoGs and the zoning technique for feature extraction purposes. The

performance capabilities of the proposed OCR system are tested using 10-fold cross validation.

### 3. Proposed Methodology

The proposed handwritten Pashto character recognition system consists of 4 main phases as depicted in Figure 1. The data collection and accumulation phase, the data processing and character database development phase, the feature extraction and feature map development phase, and at last, the recognition and identification phase. The data collection and accumulation phase is completed by collecting handwritten Pashto samples from different people, while the pre-processing steps include scanning and correction steps. This phase aims to prepare data for the feature extraction purposes as proper characters results in achieving high and accurate feature values that ultimately results in high recognition rates of the handwritten characters. For the feature extraction purposes, we have proposed HoGs and zoning techniques. These techniques grab the astute numerical values of the characters. The classification and recognition phase is completed using a  $k$ -NN classifier based on the accumulated feature map using HoG and zoning techniques.

**3.1. Data Collection.** Since there is no standard database available for handwritten Pashto characters, a medium-sized database is developed by collecting data from different people. Most of the samples are collected from the Department of Pashto, University of Swabi, KP (Khyber Pakhtunkhwa), Pakistan, from multiple students and teachers varying in age, gender, and educational backgrounds.

Individual characters are extracted from the scanned images so as to make a database. The database, thus, formed contained 258 samples for each of 44 characters and a total of 11,352 characters ( $258 \times 44 = 11352$  characters). The final database contained character images with nonuniform dimensions and decentralized characters (appearing either at the top, bottom, right, or left), as shown in Figure 2. These sliced images were preprocessed to form normalized and centralized character images.

**3.2. Preprocessing.** Preprocessing is a preliminary step necessary to achieve better classification accuracy. Preprocessing steps are applied here to form normalized and centralized character images. Preprocessing greatly improves OCR accuracy. We applied the following preprocessing steps.

**3.2.1. Size Normalization.** To achieve best classification results, it is necessary that the sliced images are normalized and centralized. By normalizing, image size is scaled to a fixed size. All the images here are normalized to the size of  $64 \times 64$  and are converted to the color map of grayscale. Figure 3 shows the size normalization to the dimensions of  $64 \times 64$ .

**3.2.2. Centralization.** Some of the images contained characters that occurred at different positions (top, bottom, right, and left). Firstly, the centroid of the character and image are calculated separately to fix all the characters at the central point so as to calculate accurate features of each handwritten Pashto character. In our case, the character is of  $64 \times 64$  dimensional size, so the central point of the character is  $32 \times 32$  in our case. Then, the character centroid is shifted to the centroid of the image to produce a centralized image. Figure 4 shows the centralization of characters “alif” and “twe.”

**3.3. Feature Extraction.** Feature extraction is a pivotal stage of an OCR. Features are used to describe the image in terms of numerical values. There are two types; statistical features are calculated via mathematical computations whereas structural features are derived from the structure of the image. A good feature extractor should have the ability to discriminate while retaining similarity for similar character images. We applied two techniques, namely, histogram of oriented gradients (HOGs) and zoning-based density features, to our database and compared their results.

**3.3.1. Zoning Features.** Zoning-based features are efficient for reading and extracting accurate image patterns. Due to its high feature extraction capabilities, this technique is frequently used in many text recognition problems. This technique divides the image into  $8 \times 8$  zones and then calculates the image pixel densities in each zone that forms the feature vector. It gives a feature vector of 64 features. Figure 5 shows the character “bhe” divided into 64 zones.

**3.3.2. Histograms of Oriented Gradients.** The histogram of oriented gradients (HOGs) was firstly introduced by Dalal and Triggs [29]. The primary purpose was human detection. Nowadays, this technique is used for character recognition [21, 30, 31], pedestrian detection [32], face recognition [33], and many other problems of interest. We generated HOG features using cell size  $16 \times 16$  pixels, block size  $2 \times 2$  cells, and 9 bins. HOG visualization over the Pashto character “ye” is shown in Figures 6(a) and 6(b).

**3.4. Classification.** Classification acts as a kernel in any recognition problem. Classification is the stage to classify unpredicted input data into given classes. This paper presents the use of  $k$ -nearest neighbor ( $k$ -NN) as a classification model for offline Pashto character recognition.  $K$ -NN is a supervised learning model.  $K$ -NN works on the nearest-neighbor rule which classifies data by measuring the distance between the input instance and training data and chooses the class for unpredicted instance based on nearest instance in training data.

There are four-distance functions that can be used in  $k$ -NN, Euclidean, Manhattan, Chebyshev, and Minkowski distances. Here, we used the Euclidean distance function and evaluated its performance with respect to different values of  $k$ . The distance function is shown in the following equation:

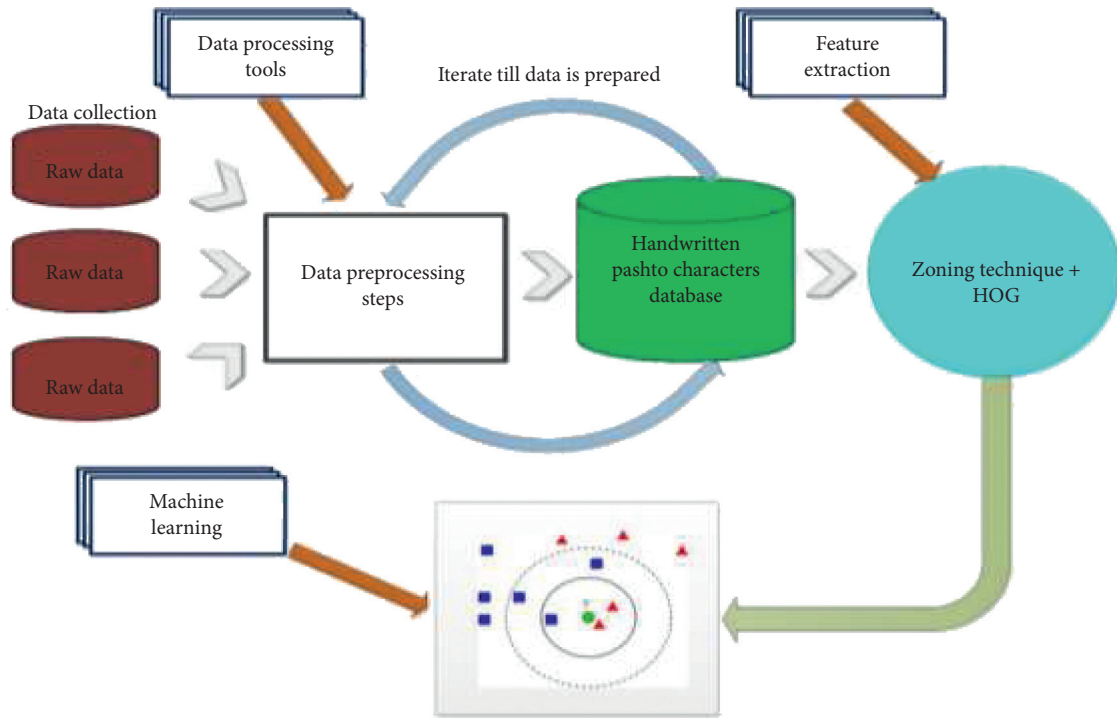


FIGURE 1: Proposed methodology flow diagram.

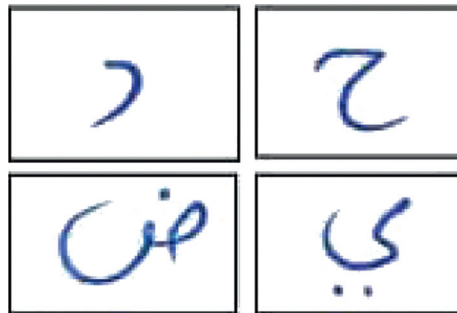


FIGURE 2: Character images before preprocessing steps.



FIGURE 3: Size normalization to the dimensions of  $64 \times 64$ .

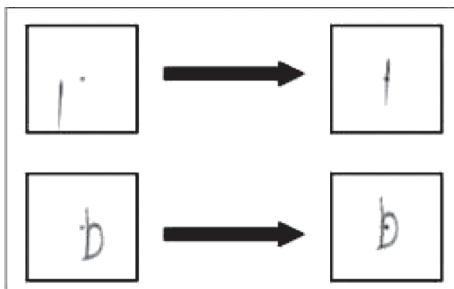


FIGURE 4: Centralization of characters “alif” and “twe.”

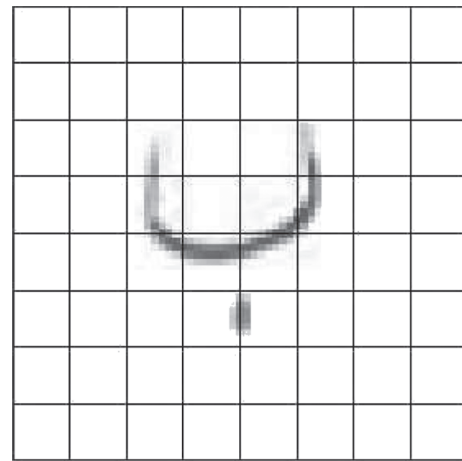


FIGURE 5: Character ‘bhe’ divided into 64 zones of  $8 \times 8$  pixels.

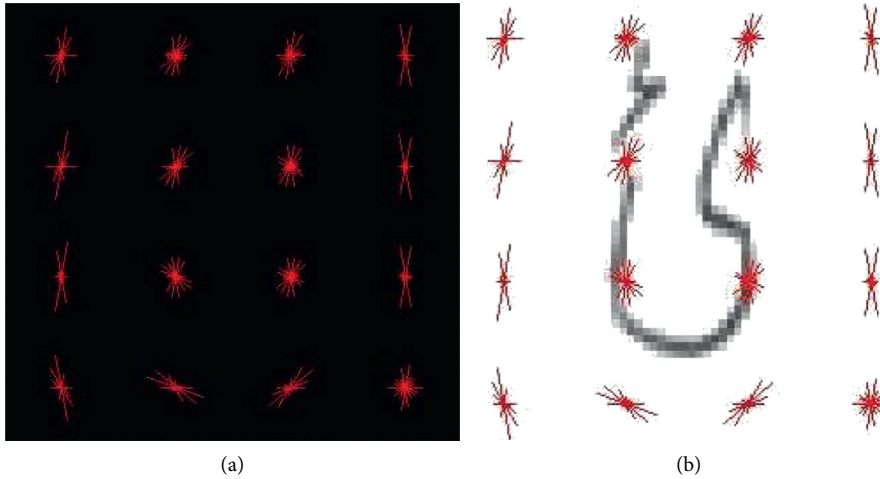


FIGURE 6: (a) HOG feature of the letter “ye.” (b) HOG feature of letter “ye” over it.

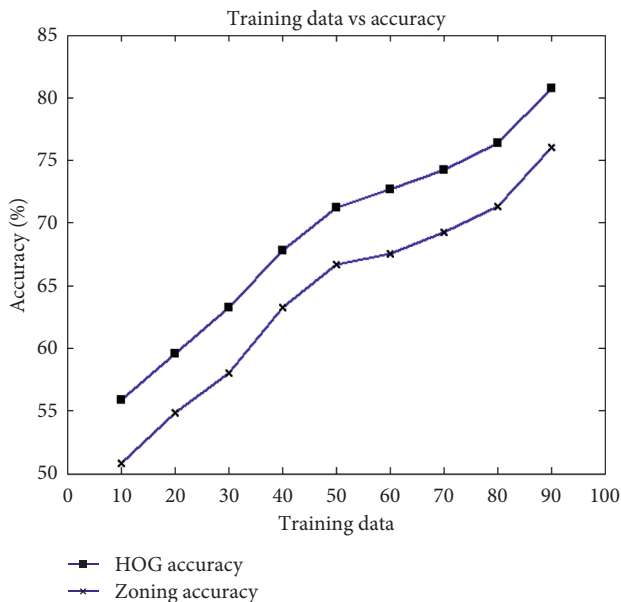


FIGURE 7: Training data vs. accuracy.

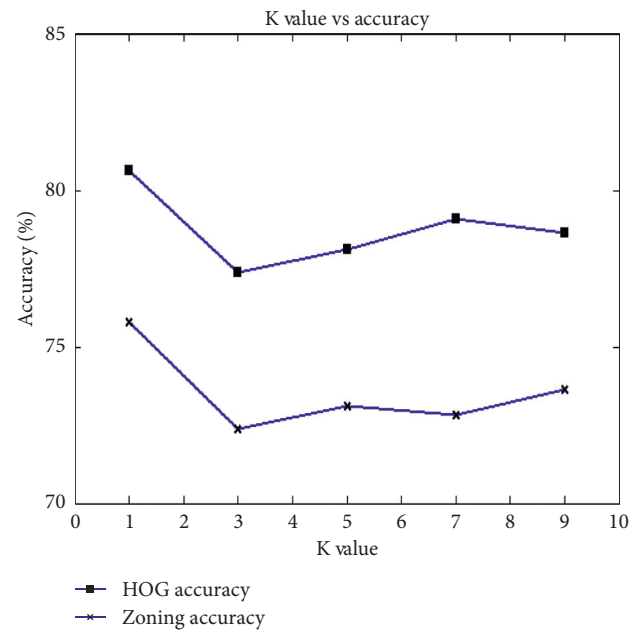


FIGURE 8:  $k$  value vs. accuracy.

$$D(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}. \quad (1)$$

## 4. Results

Results are calculated for the proposed system based on a zoning-based density feature set and Histogram of Oriented Gradients (HOGs) feature set. For each of the feature extractor technique, the results are drawn based on the  $k$ -nearest neighbors ( $k$ -NN) classification model using 10-fold cross validation.

Ten-fold cross validation using the 1-NN classifier and Histograms of Oriented Gradients (HOGs) achieved an accuracy of 80.34%, while for zoning features, a relatively lower accuracy of 76.42% is achieved. The results are compared, and a graph is generated using these data as

TABLE 1:  $k$  value vs. accuracy.

Technique	$k=1$ (%)	$k=3$ (%)	$k=5$ (%)	$k=7$ (%)	$k=9$ (%)
HOG	80.63	77.36	78.09	79.09	78.63
Zoning	75.81	72.36	73.09	72.81	73.63

shown in the graph in Figure 7. Accuracy tends to increase as training data increase because the classifier learns more accurately and produces better results.

The  $k$ -NN parameter  $k$  value was tuned, and the best score for  $k=1$  was calculated using the Euclidean distance as depicted by a graph in Figure 8.

Figure 8 shows the  $k$  value vs. accuracy.

The accuracy vs.  $k$  value table is generated as shown in Table 1. Different accuracies were calculated for different values of  $k$ .

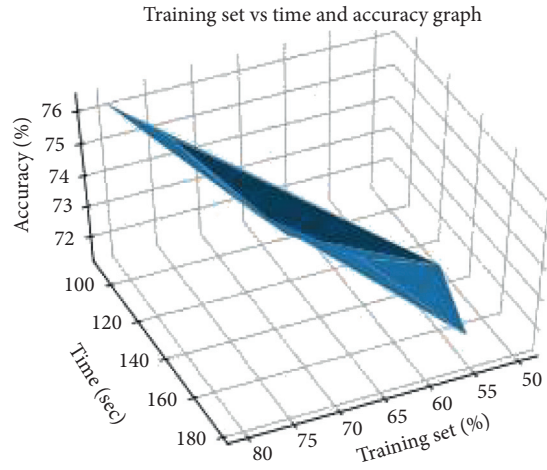


FIGURE 9: Recognition rates using an HOG-based feature set.

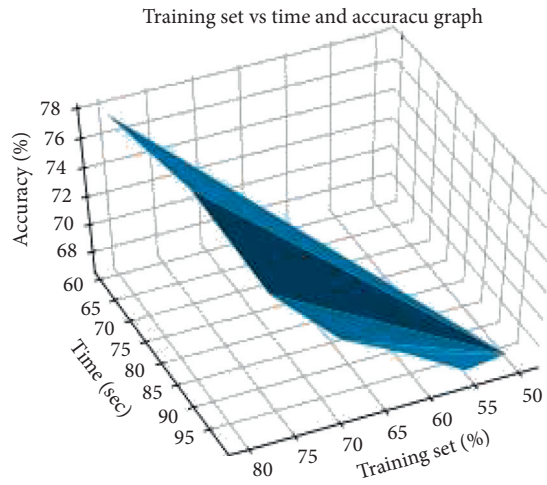


FIGURE 10: Recognition rates using a zoning-based feature set.

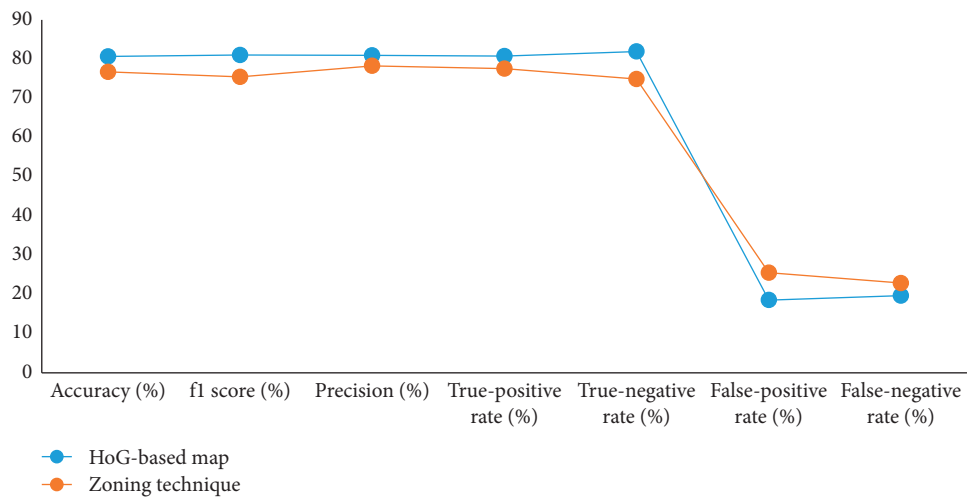


FIGURE 11: OCR results based on different performance metrics.

Figures 9 and 10 show the plot of training data vs. time vs. accuracy for HOGs and zoning-based density features. From Figures 9 and 10, it is evident that when the training set increases, the overall recognition rate of the classifier increases, but ultimately, the time consumption also increases.

Applicability of the system is also validated by using other performance metrics such as precision, false-positive rate, false-negative rates, true-positive rates, true-negative rates, f1 score, and accuracy based on both HoG-based and zoning-based feature maps. Experimental results based on these performance metrics are depicted in Figure 11.

## 5. Conclusions

Handwritten text recognition is followed as the most daunting step in the research work. During the last two decades, cursive text recognition gained a significant interest in the research community to explore. However, the unavailability of a standard database makes it more challenging. To address these problems, an OCR system for handwritten Pashto character recognition is presented in this paper. A medium-sized database containing 11352 character samples (44 characters  $\times$  258 samples) was developed for the analysis and experimental work. Histogram of oriented gradients and zoning techniques are used for the feature accumulation purposes. This feature map is used for the identification and recognition of the handwritten Pashto characters using the  $k$ -NN classification tool. Based on the calculated feature map, histograms of oriented gradients gives an accuracy rate of 80.34% while zoning-based density features give an accuracy of 76.42%. Ten-fold cross validation was applied for evaluating system results.

## Data Availability

No data were used in this work.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] S. Khan, A. Hafeez, H. Ali, S. Nazir, and A. Hussain, "Pioneer dataset and recognition of handwritten Pashto characters using convolution neural networks," *Measurement and Control*, vol. 53, no. 2, p. 20294020964826, 2020.
- [2] S. Wang, S. Khan, C. Xu, S. Nazir, and A. Hafeez, "Deep learning-based efficient model development for phishing detection using random forest and BLSTM classifiers," *Complexity*, vol. 2020, Article ID 8694796, 7 pages, 2020.
- [3] J. Jurasz, F. A. Canales, A. Kies, M. Guezgouz, and A. Beluco, "A review on the complementarity of renewable energy sources: concept, metrics, application and future research directions," *Solar Energy*, vol. 195, pp. 703–724, 2020.
- [4] Y. He, S. Nazir, B. Nie, S. Khan, and J. Zhang, "Developing an efficient deep learning-based trusted model for pervasive computing using an LSTM-based classification model," *Complexity*, vol. 2020, Article ID 4579495, 6 pages, 2020.
- [5] S. Khan, S. Nazir, I. García-Magariño, and A. Hussain, "Deep learning-based urban big data fusion in smart cities: towards traffic monitoring and flow-preserving fusion," *Computers & Electrical Engineering*, vol. 89, Article ID 106906, 2021.
- [6] S. Khan, S. Nazir, and H. Ullah Khan, "Smart object detection and home appliances control system in smart cities," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 895–915, 2021.
- [7] Y. Bouldid, A. Souhar, and M. Y. Elkettani, "Handwritten character recognition based on the specificity and the singularity of the Arabic language," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 4, 2017.
- [8] S. Naz, A. I. Umar, R. Ahmad et al., "Urdu Nastaliq recognition using convolutional-recursive deep learning," *Neuro-computing*, vol. 243, pp. 80–87, 2017.
- [9] N. Ahmad, A. A. Khan, S. A. R. Abid, and M. Yasir, "Pashto isolated character recognition using K-nn classifier," *Sindh University Research Journal-SURJ*, vol. 45, 2013.
- [10] C. Boufenar, M. Batouche, and M. Schoenauer, "An artificial immune system for offline isolated handwritten Arabic character recognition," *Evolving Systems*, vol. 9, no. 1, pp. 25–41, 2018.
- [11] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cognitive Systems Research*, vol. 50, pp. 180–195, 2018.
- [12] M. Askari, M. Asadi, A. Asilian Bidgoli, and H. Ebrahimpour, "Isolated Persian/Arabic handwriting characters: derivative projection profile features, implemented on GPUs," *Journal of AI and Data Mining*, vol. 4, pp. 9–17, 2016.
- [13] A. El-Sawy, M. Loey, and E. Hazem, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research*, vol. 5, pp. 11–19, 2017.
- [14] C. Boufenar and M. Batouche, "Investigation on deep learning for off-line handwritten arabic character recognition using theano research platform," in *Proceedings of the 2017 Intelligent Systems And Computer Vision (ISCV)*, pp. 1–6, Fez, Morocco, April 2017.
- [15] F. Sarvaramini, A. Nasrollahzadeh, and M. Soryani, "Persian handwritten character recognition using convolutional neural network," in *Proceedings of the Iranian Conference on Electrical Engineering (ICEE)*, pp. 1676–1680, Mashhad, Iran, May 2018.
- [16] S. Khan, H. Ali, Z. Ullah, N. Minallah, S. Maqsood, and A. Hafeez, "KNN and ANN-based recognition of handwritten pashto letters using zoning features," 2019, <https://arxiv.org/abs/1903.10921>.
- [17] M. A.-A. Bhuiyan and F. W. Alsaade, "On Arabic character recognition employing hybrid neural network," *International Journal of Advanced Computer Science and Applications*, vol. 8, pp. 96–101, 2017.
- [18] R. Tavoli, M. Keyvanpour, and S. Mozaffari, "Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic/Persian handwritten words recognition," *IET Image Processing*, vol. 12, no. 9, pp. 1606–1616, 2018.
- [19] M. Oujaoura, R. El Ayachi, M. Fakir, B. Bouikhalene, and B. Minaoui, "Zernike moments and neural networks for recognition of isolated Arabic characters," *International Journal of Computer Engineering Science (IJCES)*, vol. 2, pp. 17–25, 2012.
- [20] K. S. Younis, "Arabic handwritten character recognition based on deep convolutional neural networks," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 3, 2017.
- [21] N. A. Jebiril, H. R. Al-Zoubi, and Q. Abu Al-Haija, "Recognition of handwritten Arabic characters using histograms of

- oriented gradient (HOG),” *Pattern Recognition and Image Analysis*, vol. 28, no. 2, pp. 321–345, 2018.
- [22] H. Althobaiti and C. Lu, “A survey on Arabic optical character recognition and an isolated handwritten Arabic character recognition algorithm using encoded freeman chain code,” in *Proceedings of the In 51st Annual Conference On Information Sciences And Systems (CISS)*, pp. 1–6, Baltimore, MD, USA, March 2017.
- [23] H. Althobaiti, K. Shah, and C. Lu, “Isolated handwritten Arabic character recognition using freeman chain code and tangent line,” in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pp. 79–84, Krakow, Poland, April 2017.
- [24] S. Jehangir, S. Khan, S. Khan, S. Nazir, and A. Hussain, “Zernike moments based handwritten Pashto character recognition using linear discriminant analysis,” *Mehran University Research Journal of Engineering and Technology*, vol. 40, pp. 152–159, 2021.
- [25] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, “Zoning features and 2DLSTM for Urdu text-line recognition,” *Procedia Computer Science*, vol. 96, pp. 16–22, 2016.
- [26] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, “Handwritten Urdu character recognition using one-dimensional BLSTM classifier,” *Neural Computing and Applications*, vol. 31, pp. 1–9, 2017.
- [27] M. Jameel and S. Kumar, “Offline recognition of handwritten Urdu characters using B spline curves: a survey,” *International Journal of Computer Applications*, vol. 157, 2017.
- [28] S. Nawaz, A. Calefati, N. Ahmed, and I. Gallo, “Hand Written Characters Recognition via Deep Metric Learning,” in *Proceedings of the 2018 13th IAPR International Workshop On Document Analysis Systems (DAS)*, pp. 417–422, Vienna, Austria, July 2018.
- [29] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision And Pattern Recognition. CVPR 2005*, pp. 886–893, San Diego, CA, USA, July 2005.
- [30] A. J. Newell and L. D. Griffin, “Multiscale histogram of oriented gradient descriptors for robust character recognition,” in *Proceedings of the 2011 International Conference On Document Analysis And Recognition (ICDAR)*, pp. 1085–1089, Peking, China, September 2011.
- [31] S. B. Khan, G. Ahmad, F. Ali, F. Faisal, I. Ahmed, and S. Elahi, “Classification performance of linear binary pattern and histogram oriented features for arabic characters images: A,” *Applied Physics of Condensed Matter (APCOM 2019)*, vol. 5, 2018.
- [32] T. Watanabe, S. Ito, and K. Yokoi, “Co-occurrence histograms of oriented gradients for pedestrian detection,” in *Proceedings of the Pacific-Rim Symposium on Image and Video Technology*, pp. 37–47, Tokyo, Japan, January 2009.
- [33] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, “Face recognition using histograms of oriented gradients,” *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1598–1603, 2011.



## Research Article

# Features to Text: A Comprehensive Survey of Deep Learning on Semantic Segmentation and Image Captioning

Ariyo Oluwasammi <sup>1</sup>, Muhammad Umar Aftab <sup>2</sup>, Zhiguang Qin <sup>1</sup>, Son Tung Ngo <sup>3</sup>,  
Thang Van Doan <sup>3</sup>, Son Ba Nguyen <sup>3</sup>, Son Hoang Nguyen <sup>3</sup>,  
and Giang Hoang Nguyen <sup>3</sup>

<sup>1</sup>School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

<sup>2</sup>Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan

<sup>3</sup>ICT Department, FPT University, Hanoi 10000, Vietnam

Correspondence should be addressed to Zhiguang Qin; qinz@uestc.edu.cn

Received 8 January 2021; Revised 31 January 2021; Accepted 6 March 2021; Published 23 March 2021

Academic Editor: Dan Selisteanu

Copyright © 2021 Ariyo Oluwasammi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the emergence of deep learning, computer vision has witnessed extensive advancement and has seen immense applications in multiple domains. Specifically, image captioning has become an attractive focal direction for most machine learning experts, which includes the prerequisite of object identification, location, and semantic understanding. In this paper, semantic segmentation and image captioning are comprehensively investigated based on traditional and state-of-the-art methodologies. In this survey, we deliberate on the use of deep learning techniques on the segmentation analysis of both 2D and 3D images using a fully convolutional network and other high-level hierarchical feature extraction methods. First, each domain's preliminaries and concept are described, and then semantic segmentation is discussed alongside its relevant features, available datasets, and evaluation criteria. Also, the semantic information capturing of objects and their attributes is presented in relation to their annotation generation. Finally, analysis of the existing methods, their contributions, and relevance are highlighted, informing the importance of these methods and illuminating a possible research continuation for the application of semantic image segmentation and image captioning approaches.

## 1. Introduction

The data of optical perception are becoming increasingly available in large volume nowadays, creating a crucial use in several real-world applications such as quality assurance, medical analysis, surveillance, autonomous vehicles, face recognition, forensic and biometrics, and 3D reconstruction [1–4]. This upsurge in the bulk of digital images and video has directed the creation of computer vision (CV), a branch of computer science (CS). From a general overview, computer vision relates to the use of the computer to gain a high-level understanding of images and videos [5]. Rather than manual operations, it encompasses the automatic acquisition, processing, and analyzing of large data for the sole

purpose of extracting patterns and intuition. In most cases, computer vision seeks to apply artificial intelligence (AI) theory, equations, tools, frameworks, and algorithms for accomplishing the task of helping computers to see and also understand the content of both digital and analog world through the mimicking of the human visual system [6]. Although seeing and understanding seems a trivial task or very easy for humans, it is nevertheless a complex problem for computers partly because of our limited understanding of how the human brain works and how it processes things [7]. However, through years of research and technological advancement, some feats have been achieved, and computer vision has extensively evolved [8–11]. Today, semantic segmentation remains a huge challenge in the scope of image

and video understanding alongside image captioning which combines computer vision with another branch of artificial intelligence called natural language processing (NLP) to derive sentence description of an image [12]. Notwithstanding, as with all other AI-related tasks, a modern subset of machine learning (ML), namely, deep learning (DL), has been the evolution of machine learning, producing state-of-the-art results in almost all of the tasks compared to other traditional algorithms such as decision trees, naive Bayes, support vector machines (SVMs), ensembles, and clustering algorithms [13–16].

Deep learning, as a branch of machine learning, uses layers of artificial neural networks to imitate the human neural networks in decoding intuition from a large amount of data automatically [17] and is unlike other machine learning algorithms which rely heavily on feature engineering, utilizing domain knowledge in the creation of feature extractors [18]. The stacked layer of neural networks represents feature hierarchy as simple features at the initial layers are reconstructed from one layer to another in forming complex features [19]. As a result, the deeper networks are computationally intensive to model and train, leading to the manufacture of more advanced computational chips, including Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs) [20, 21]. Presently, several deep learning models exist, and some of the most popular ones include recurrent neural networks (RNNs), autoencoders, convolutional neural network (CNN), deep belief networks (DBNs), and deep Boltzmann machine (DBM) [22–25]. Among the most common deep learning algorithms, the convolutional neural network is the most suitable for analyzing visual imagery because of its shift and space invariant characteristics, taking advantage of hierarchical learning in combining simpler patterns to form complex patterns and structures [26]. Using the shared weights architectural pattern of filters [27–29], each filter represents different features of the input data which when summed can yield more complex structures [30–32].

In this paper, our prime motivation is fixating on the recent deep learning segmentation techniques of both 2D and 3D images using fully convolutional network and other high-level hierarchical feature extraction methods as an integral component of computer vision. This is further expanded into the generation of captions for images, emerging as a subset of artificial intelligence’s natural language processing. Furthermore, we review the discussed models’ accomplishments by comparing their evaluation, which indicates the most effective and efficient approaches for different tasks and challenges encountered. This, we believe, is enlightening as it provides insight for the further evolution of practical model design.

This paper is organized as follows: it introduces segmentation, popular segmentation algorithms, characteristics, datasets, and evaluation in Section 2. An introduction of captioning and its various models are followed in Section 3 alongside available datasets, evaluation metrics, and a comparative discussion of the models. Finally, Section 4 concludes the paper with the overall summary of the typical

problems, solutions, and possible directions in semantic segmentation and image captioning.

## 2. Semantic Segmentation

Semantic segmentation relates to the process of pixel-level classification of images such that each pixel in an image is classified into a distinguished class cluster [33]. Since the inception of deep learning, semantic segmentation has been a pivotal area of image processing and computer vision which has seen major research and application in several domains [34]. Image segmentation recognizes boundaries between objects in an image by using line and curve segments to categorize such objects, while instance segmentation, however, classifies instances of all the available classes in an image such that all objects are identified as a separate entity. All the same, semantic segmentation differs from ordinary segmentation which, on the one hand, only expresses the partitioning of an image into clusters without a tangible intuitive attempt at understanding the partitioned clusters or relating them with one another [35]. Semantic segmentation, on the other hand, as the name implies, tries to describe semantically meaningful objects in an image based on their well-defined association and understanding [36]; these differences are well depicted in Figure 1.

### 2.1. Methods and Approaches

*2.1.1. Traditional Methods.* During the pre-ANN era, most segmentation and semantic segmentation approaches were predominantly thresholding and clustering algorithms which are largely unsupervised methods. In most cases, traditional semantic segmentation methods consume less time for model computation. Also, most of the approaches require less data than the modern-day era of artificial neural networks and deep learning. The simplest, by all means, is the thresholding techniques which apply pixel intensity as the criteria for distribution. For binary segmentation, a single threshold value is required, and pixels on both sides of the threshold are classified separately into two distinct classes. There are also advance forms of thresholding involving more classes, and they are often grouped as histogram shape-based, entropy-based, object attribute-based, and spatial-based [37].

*K*-means clustering uses a predefined number of centroids to determine the number of clusters in which objects are to be categorized. The centroids are randomly selected at the beginning and then are iteratively adjusted by computing the distance apart from other points in the dataset, assigning each point to the closest centroid [38]. Fuzzy C-means (FCM), a technically advanced form of *K*-means, allows classification of data points into many label classes based on the level of membership [39]. This is of advantage in situations where dataset texture overlaps or does not have a well-defined cluster [40]. Gaussian mixture model (GMM) is also often used for both hard clustering and soft clustering by assigning the pixel to the component having the highest posterior probability [41]. GMM assumes that the data’s Gaussian distributions represent the number of clusters

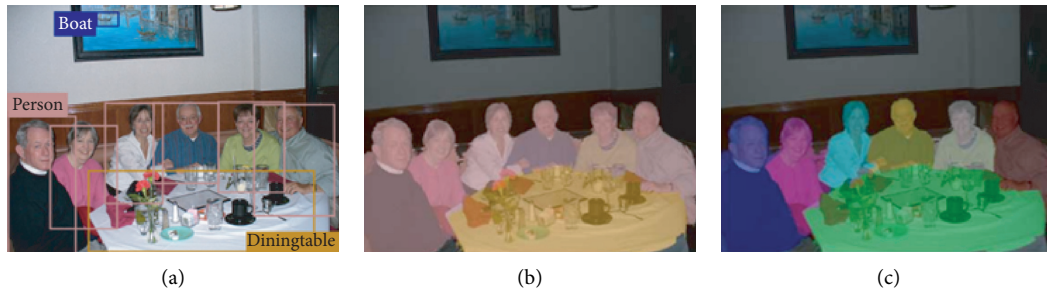


FIGURE 1: Illustration of differences in segmentation: (a) object detection, (b) semantic segmentation, and (c) instance segmentation [36].

available in the data, and it uses the expectation-maximization (EM) algorithm to determine missing latent variables [42]. Random forest [43], naive Bayes [44], ensemble modeling [45], Markov random network (MRF) [46], and support vector machines (SVMs) [47] are also techniques that are useful for several tasks, especially classification and regression [48].

**2.1.2. Region-Based Models.** In the region-based semantic segmentation design, regions are first extracted in an image and described based on their constituent features [49]. Then, a region classifier that has been trained is used to label pixels per region with which it has the highest occurrence. The region-based approaches use the divide and conquer method such that many features are captured using multiscale features and then combined to form a whole. In cases where objects overlap on several regions, the classifier either determines the most suitable region or the model is set to select the region with the maximum value [50]. This often causes pixel mismatch, but a postprocessing operation is mostly used to reduce the effects [51].

Region CNN (R-CNN) uses a bounding box to identify and classify objects in an image by proposing several boxes convolving an image and identifying if they correspond to an object [52, 53]. The process of selective search is used in creating boundary boxes of varying window sizes for region proposal, and each of the boxes classifies objects based on different properties, making the algorithm quite impressive but slow [54]. To overcome the drawbacks of the R-CNN, Fast R-CNN [55] was proposed which eliminates the redundancy in the proposed region, thereby lessening the computational requirements. The R-CNN model was replaced with a single CNN per image whose computation would be shared among the proposals, using the region of interest pooling technique and training all the models including the use of convolutional neural network to classify the images, and bounding boxes regressor as a single entity. The Faster R-CNN [56] uses a region proposal network (RPN) to obtain a proposal from the network instead, while the Mask R-CNN [57] was extended to include a pixel-level segmentation. Technically, the Mask R-CNN replaces the region of interest pooling module in the Faster R-CNN with another which has an accurate alignment module. Also, it includes an additional parallel branch for segmentation mask prediction [58].

**2.1.3. Fully Convolutional Network-Based Models.** Fully convolutional network (FCN) models do not have dense layers, such as in other traditional CNNs; they are composed of  $1 \times 1$  convolutions that achieve the task of dense layers or fully connected layers. Also, fully convolutional network (FCN), as displayed in Figure 2, takes images of arbitrary sizes as the input and returns outputs of corresponding spatial dimensions. This model principally builds on the encoder-decoder model to classify pixels in an image into predefined classes by using a convolution network in the encoder to extract features, thereby reducing the feature maps' dimensionality before being upsampled by the decoder (SegNet) [60]. During convolutional neural network computation, input images are downsized, resulting in a smaller output with reduced spatial features. This problem is solved via the upsample technique, which transposes the downsized images to a larger size, making pixelwise comparison efficient and effective. Some upsampling methods such as transpose convolutions are learned, thus increasing model complexity and computation, while several others exclude learning including nearest neighbor, the bed of nails, and max unpooling [61]. The fully convolutional network is majorly trained as an end-to-end model to compute pixelwise loss and trained using the backpropagation approach. The FCN was firstly inspired by Long et al. [59] using the popular AlexNet CNN architecture in the encoder and transpose convolution layers in the decoder to upsample the feature to the desired dimension. A variant FCN having skip connections from previous layers in the network was proposed named UNet [62]. UNet intends to compliment the learned features with fine-grain details from contracting paths to capture the context and enhance classification accuracy.

Residual blocks were introduced with shorter skip connections between the encoder and decoder, granting faster convergence of deeper models during training [63]. Multiscale induction in the dense blocks conveys low-level features across layers to ones with high-level features, resulting in feature reuse [64]. This makes the model easier to train and improve the performance as well. An architecture having two-stream CNN uses its gates to process the image shape in different branches, then connected and fused at a later stage. The model also proposed a new loss function that aligns segmentation prediction with the ground truth boundaries [65]. An end-to-end single-pass model applying dense decoder shortcut connections extracts semantics from

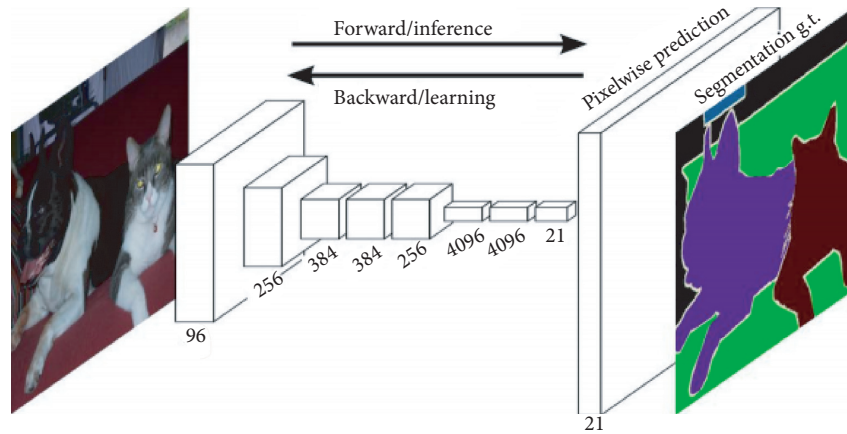


FIGURE 2: Illustration of fully convolutional networks for semantic segmentation [59].

high-level features such that propagation of information from one block to another combines multiple-level features [66]. The model designed based on the ResNeXt’s residual building blocks helps it to aggregate blocks of feature captures which are fused as output resolutions [67].

ExFuse aims to connect the gap between low- and high-level features in convolutional networks by introducing semantic information at the lower-level features as well as high resolutions into the high-level features. This was achieved by proposing two fusion methods named explicit channel resolution embedding and densely adjacent prediction [68]. Contrary to most models, a balance between model accuracy and speed was achieved in ICNet which consolidates several multiresolution branches by introducing an image cascade network that allows real-time inference. The network cascade image inputs into different segments as low, medium, and high resolution before being trained with this label guidance [69].

**2.1.4. Refinement Network.** Because of the resolution reduction caused by the encoder models in the typical FCN-based model, the decoder has inherent problems of producing fine-grained segmentation, especially at the boundaries and edges [70]. Though this problem has been tackled by incorporating skip connections, adding global information, and others means, the problem is by no means solved, and some algorithms have involved several features or, in some cases, certain postprocessing functions to find alternative solutions [71]. DeepLab1 [72] combines ideas from the deep convolutional neural network and probabilistic graphical models to achieve pixel-level classification. The localization problem of the neural network output layer was remedied using a fully connected conditional random field (CRF) as a means of performing segmentation with controlled signal extermination. DeepLab1 applied atrous convolutions instead of the regular convolutions which accomplish the learning of aggregate multiscale contextual features. Visible in Figure 3, DeepLab1 allows the expansion of kernel window sizes without increasing the number of weights. The multiscale atrous convolutions help to overcome the problem of insensitivity to fine details by other

models and decrease output blurriness. This could result in additional complexity in computation and time depending on the postprocessing network’s computational processes.

The ResNet deep convolutional network architecture was applied in DeepLab2 which enables the training of various distinct layers while preserving the performance [73]. Besides, DeepLab2 uses atrous spatial pyramid pooling (ASPP) to capture long-range context. The existence of objects at different scales and the reduced feature resolution problems of semantic segmentation are tackled by designing a cascade of atrous convolutions which could run in parallel to capture various scales of context information alongside global average pooling (GAP) to embed context information features [74]. FastFCN implements joint pyramid upsampling which substitutes atrous convolutions to free up memory and lessen computations. Using a fully connected network framework, the joint pyramid upsampling technique extracts feature maps of high resolution into a joint upsampling problem. The models used atrous spatial pyramid pooling to extract the last three-layer output features and a global context module to map out the final predictions [75]. The atrous spatial pyramid pooling limitation of lack of dense feature resolution scale is attempted by concatenating multiple branches of atrous-convolved features at different rates which are later fused into a final representation, resulting in dense multiscale features [76].

**2.1.5. Weakly Supervised and Semisupervised Approaches.** Though most models depend on a large number of images and their annotated label, the process of manually annotating labels is quite daunting and time-consuming, so semantic segmentation models have been attempted with weakly supervised approaches. Given weakly annotated data at the image level, the model was trained to assign higher weights to pixels corresponding with the class label. Trained on a subset of the ImageNet dataset, during training, the networks focus on recognizing important pixels relating to a prior labeled single-class object and matching them to the class through inference [77]. Bounding box annotation is used to train semantic labeling of image segmentation which accomplishes 95% quality of fully supervised models. The

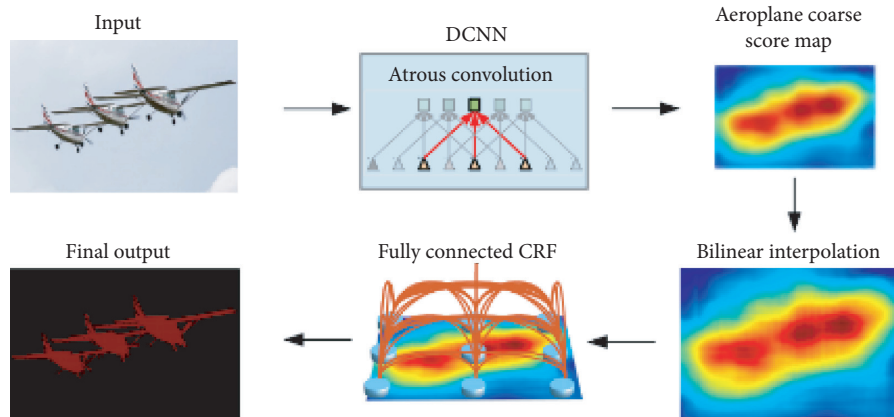


FIGURE 3: DeepLab framework with fully connected CRFs [72].

bounding box and information of the constituent object were used prior to training [78]. A model combining both labeled and weakly annotated images with a clue of the presence or absence of a semantic class was developed using the deep convolutional neural network and expectation-maximization (EM) algorithm [79].

BoxSup iteratively generates automatic region proposals while training convolutional networks to obtain segmentation masks and as well as improve the model's ability to classify objects in an image. The network uses bounding box annotations as a substitute for supervised learning such that regions are proposed during training to determine candidate masks, overtime improving the confidence of the segmentation masks [80]. A variant of generative adversarial learning approach which constitutes a generator and discriminator was used to design a semisupervised semantic segmentation model. The model was first trained using full labeled data which enable the model's generator to learn the domain sample space of the dataset which is leveraged to supervise unlabeled data. Alongside the cross-entropy loss, an adversarial loss was proposed to optimize the objective function of tutoring the generator to generate images as close as possible to the image labels [81].

**2.2. Datasets.** Deep learning requires an extensive amount of training data to comprehensively learn patterns and fine-tune the number of parameters needed for its gradient convergence. Accordingly, there are several available datasets specifically designed for the task of semantic segmentation which are as follows:

**PASCAL VOC:** PASCAL Visual Object Classes (VOC) [82] is arguably the most popular semantic segmentation dataset with 21 classes of predefined object labels, background included. The dataset contains images and annotations which could be used for detection, classification, action classification, person layout, and segmentation tasks. The dataset's training, validation, and test set has 1464, 1449, and 1456 images, respectively. Yearly, the dataset has been used for public competitions since 2005.

**MS COCO:** Microsoft Common Objects in Context [83] was created to push the computer vision state of the art with

standard images, annotation, and evaluation. Its object detection task dataset uses either instance/object annotated features or a bounding box. In total, it has 80 object categories with over 800,000 available images for its training, test, and validation sets, as well as over 500,000 object instances that are segmented.

**Cityscapes:** Cityscapes dataset [84] has a huge amount of images taken from 50 cities during different seasons and times of the year. It was initially a video recording, and the frames were extracted as images. It has 30 label classes in about 5000 densely annotated images and 20,000 weakly annotated images which have been categorized into 8 groups of humans, vehicles, flat surfaces, constructions, objects, void, nature, and sky. It was primarily designed for urban scene segmentation and understanding.

**ADE20K:** ADE20K dataset [85] has 20,210 training images, 2000 validation images, and 3000 test images which are well suited for scene parsing and object detection. Alongside the 3-channel images, the dataset contains segmentation masks, part segmentation masks, and a text file that contains information about the object classes, identification of instances of the same class, and the description of each image's content.

**CamVid:** CamVid [86] is also a video sequence of scenes which have been extracted into images of high resolution for segmentation tasks. It consists of 101 images of  $960 * 720$  dimension and their annotations which have been classified into 32 object classes including void, indicating areas which do not belong to a proper object class. The dataset RGB class values are also available, ranging from 0 to 255.

**KITTI:** KITTI [87] is popularly used for robotics and autonomous car training, focusing extensively on 3D tracking, stereo, optical flow, 3D object detection, and visual odometry. The images were obtained through two high-resolution cameras attached to a car driving around the city of Karlsruhe, Germany, while their annotations were done by a Velodyne laser scanner. The data aim to reduce bias in existing benchmarks with a standard evaluation metric and website.

**SYNTHIA:** SYNTHetic [88] Collection of Imagery and Annotations (SYNTHIA) is a compilation of imaginary images from a virtual city that has a high pixel-level

resolution. The dataset has 13 predefined label classes consisting of road, sidewalk, fence, sky, building, sign, pedestrian, vegetation, pole, and car. It has a total of 13,407 training images.

**2.3. Evaluation Metrics.** The performance of segmentation models is computed mostly in the supervised scope whereby the model's prediction is compared with the ground truth at the pixel level. The common evaluation metrics are pixel accuracy (PA) and intersection over union (IoU). Pixel accuracy refers to the ratio of all the pixels classified in their correct classes to the total amount of pixels in the image. Pixel accuracy is trivial and suffers from class imbalance such that certain classes immensely dominate other classes.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (1)$$

$$\text{PA} = \frac{\sum_i n_{ii}}{\sum_i t_i},$$

where  $n_c$  is represented as the number of classes and  $n_{ii}$  is also represented as the number of pixels of class  $i$  which are predicted to class  $i$ , while  $n_{ij}$  represents the number of pixels of class  $i$  which are predicted as class  $j$  with the total number of pixels of a particular class  $i$  represented as  $t_i = \sum_j n_{ij}$ .

Mean pixel accuracy (mPA) improves pixel accuracy slightly; it computes the accuracy of the images per class instead of a global computation of all the classes. The mean of the class accuracies is then computed to the overall number of classes.

$$\text{mPA} = \frac{1}{n_c} \sum_i \frac{n_{ii}}{t_i}. \quad (2)$$

Intersection over union (IoU) metric, which is also referred to as the Jaccard index, measures the percentage overlap of the ground truth to the model prediction at the pixel level, thereby computing the amount of pixels common with the ground truth label and mask prediction [89].

$$\text{mIoU} = \frac{1}{n_c} \frac{\sum_i n_{ii}}{\sum_j n_{ij} + \sum_j n_{ji} - n_{ii}}. \quad (3)$$

**2.4. Discussion.** Different machine learning and deep learning algorithms and backbones yield different results based on the models' ability to learn mappings from input images to the label. In tasks involving images, CNN-based approaches are by far the most expedient. Although they can be computationally expensive compared to other simpler models, such models occupy a bulk of the present state of the art. Traditional machine learning algorithms such as random forest, naive Bayes, ensemble modeling, Markov random field (MRF), and support vector machines (SVMs) are too simple and rely heavily on domain feature understanding or handcrafted feature engineering, and in some cases, they are not easy to fine-tune. Also, clustering algorithms such as  $K$ -means and fuzzy C-means mostly require that the number of

clusters is specified beforehand, and they are not very effective with multiple boundaries.

Because of the CNN's invariant property, it is very effective for spatial data and object detection and localization. Besides, the modern backbone of the fully convolutional network has informed several methods of improving segmentation localization. First, the decoder uses upsampling techniques to increase the features' resolution, and then skip connections are added to achieve the transfer of fine-grain features to the other layers. Furthermore, postprocessing operations as well as the context and attention networks have been exploited.

The supervised learning approach still remains the dormant technique as there have been many options for generating datasets as displayed in Table 1. Data augmentation involving operations such as scaling, flipping, rotating, scaling, cropping, and translating has made multiplication of data possible. Also, the application of the generative adversarial network (GAN) has played a major role in the replication of images and annotations.

### 3. Image Captioning

Image captioning relates to the general idea of automatically generating the textual description of an image, and it is often also referred to as image annotation. It involves the application of both computer vision and natural language processing tools to achieve the transformation of imagery depiction into a textual composition [111]. Tasks such as captioning were almost impossible prior to the advent of deep learning, and with advances in sophisticated algorithms, multimodal techniques, efficient hardware, and a large bulk of datasets, such tasks are becoming easy to accomplish [112]. Image captioning has several applications to solving some real-world problems including providing aid to the blind, autonomous cars, academic bot, and military purposes. To a large extent, the majority of image captioning success so far has been from the supervised domain whereby huge amounts of data consisting of images and about two to five label captions describing the actions of the images are provided [113]. This way, the network is tasked with learning the images' feature presentation and mapping them to a language model such that the end goal of a captioning network is to generate a textual representation of an image's depiction [114].

Though characterizing an image in the form of text seems trivial and straightforward for humans, it is by no means simple to be replicated in an artificial system and requires advanced techniques to extract the features from the images as well as map the features to the corresponding language model. Generally, a convolutional neural network (CNN) is tasked with feature extraction, while a recurrent neural network (RNN) relies upon to translate the training annotations with the image features [115]. Aside from determining and extracting salient and intricate details in an image, it is equally important to extract the interactions and semantic relationship between such objects and how to illustrate them in the right manner using appropriate tenses and sentence structures [116]. Also, because the training

TABLE 1: Class pixel label distribution in the CamVid dataset.

Dataset	Method	mIoU
CamVid	ApesNet [90]	48.0
	ENet [91]	51.3
	SegNet [60]	55.6
	LinkNet [92]	55.8
	FCN8 [59]	57.0
	AttentionM [93]	60.1
	DeepLab-LFOV [72]	61.6
	Dilation8 [66]	65.3
	BiseNet [94]	68.7
	PSPNet [60]	69.1
	DenseDecoder [67]	70.9
	AGNet [95]	75.2
	PASCAL VOC	Wails [96]
FCN8 [59]		62.2
PSP-CRF [97]		65.4
Zoom Out [98]		69.6
DCU [99]		71.7
DeepLab1 [72]		71.6
DeConvNet [61]		72.5
GCRF [100]		73.2
DPN [101]		74.1
Piecewise [102]		75.3
Cityscapes	FCN8 [59]	65.3
	DPN [101]	66.8
	Dilation10 [103]	67.1
	LRR [104]	69.7
	DeepLab2 [73]	70.4
	FRRN [105]	71.8
	RefineNet [106]	73.6
	GEM [107]	73.69
	PEARL [108]	75.4
	TuSimple [109]	77.6
PSPNet [110]	78.4	
SPP-DCU [99]	78.9	

labels which are texts are different from the features obtained from the images, language model techniques are required to analyze the form, meaning, and context of a sequence of words. This becomes even more complex as keywords are required to be identified for emphasizing the action or scene being described [117].

Visual features: deep convolutional neural network (DCNN) is often used as the feature extractor for images and videos because of the CNN’s invariance property [118] such that it is able to recognize objects regardless of variation in appearances such as size, illumination, translation, or rotation as displayed in Figure 4. The distortion in pixel arrangement has less impact on the architecture’s ability to learn essential patterns in the identification and localization of the crucial features. Essential feature extraction is paramount, and this is easily achieved via the CNN’s operation of convolving filters over images, subsequently generating feature maps from the receptive fields from which the filters are applied. Using backpropagation techniques, the filter weights are updated to minimize the loss of the model’s prediction compared to the ground truth [119]. There have been several evolutions over the years, and this has ushered considerable architectural development in the extraction

methodology. Recently, the use of a pretrained model has been explored with the advantage of reducing time and computational cost while preserving efficiency. These extracted features are passed along to other models such as the language decoder in the visual space-based methods or to a shared model as in the multimodal space for image captioning training [120].

Captioning: image caption or annotation is an independent scope of artificial intelligence, and it mostly combines two models consisting of a feature extractor as the encoder and a recurrent decoder model. While the extractor obtains salient features in the images, the decoder model which is similar in pattern to the language model utilizes a recurrent neural network to learn sequential information [121]. Most captioning tasks are undertaken in a supervised manner whereby the image features act as the input which are learned and mapped to a textual label [122]. The label captions are first transformed into a word vector and are combined with the feature vector to generate a new textual description. Most captioning architectures follow the partial caption technique whereby part of the label vector is combined with the image vector to predict the next word in the sequence [123]. Then, the prior words are all combined to predict the next word and continued till an end token is reached. In most cases, to infuse semantics and intuitive representation into the label vector, a pretrained word embedding is used to map the dimensional representation of the embeddings into the word vector, enriching its content and generalization [124].

### 3.1. Image Captioning Techniques

*3.1.1. Retrieval-Based Captioning.* Early works in image captioning were based on caption retrieval. Using this technique, the caption to a target image is generated by retrieving the descriptive sentence of such an image from a set of predefined caption databases. In some cases, the newly generated caption would be one of the existing retrieved sentences or, in other cases, could be made up of several existing retrieved sentences [125]. Initially, the features of an image are compared to the available candidate captions or achieved by tagging the image property in a query. Certain properties such as color, texture, shape, and size were used for similarity computation between a target image and predefined images [126]. Captioning via the retrieval method can be retrieved through the visual and multimodal space, and these approaches produce good results generally but are overdependent on the predefined captions [127].

Specific details such as the object in an image or the action or scene depicted were used to connect images to the corresponding captions. This was computed by finding the ratio of similarity between such information to the available sentences [128]. Using the kernel canonical correlation analysis technique, images and related sentences were ranked based on their cosine similarities after which the most similar ones were selected as the suitable labels [129], while the image features were used for reranking the ratio of image-text correlation [130]. The edges and contours of images were utilized to obtain

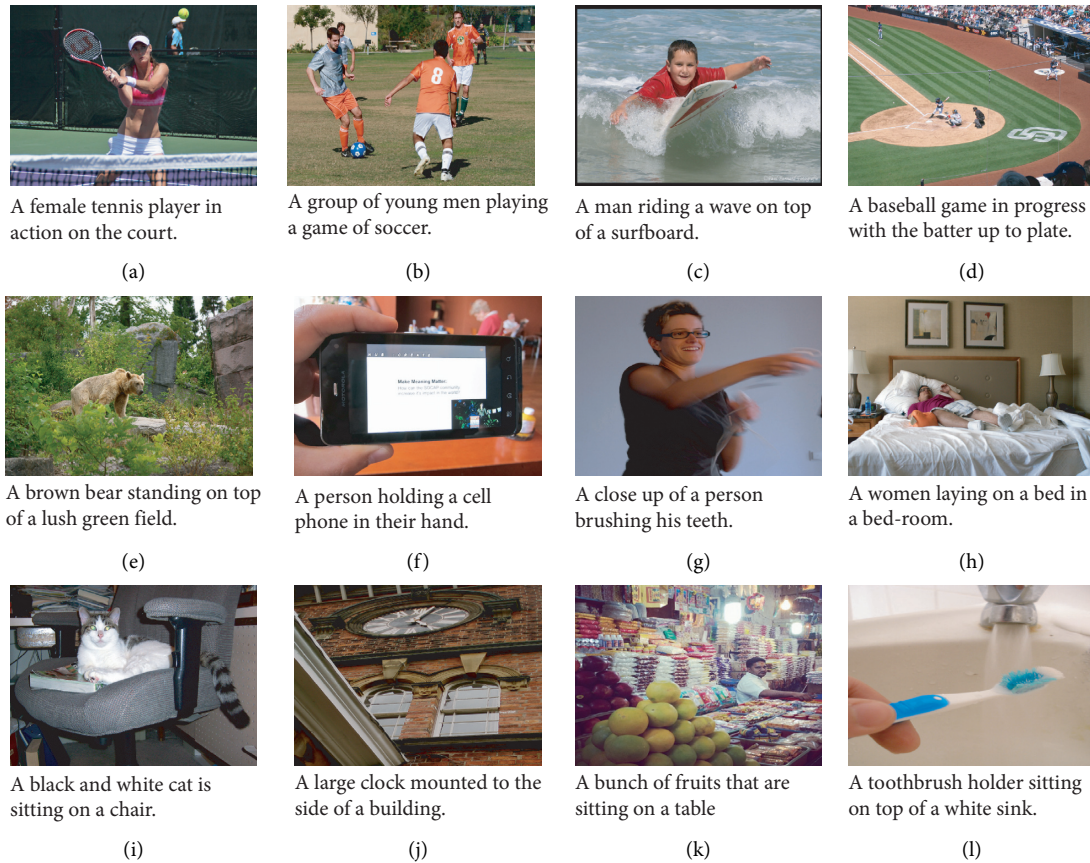


FIGURE 4: Sample images and their corresponding captions [112].

pattern and sketches such that they were used as a query for image retrieval, whereby the generated sketches and the original images were structured into a database [131]. Building on the logic of the original image and its sketch, more images were dynamically included alongside their sketches to enhance learning [132]. Furthermore, deep learning models were applied to retrieval-based captioning by using convolutional neural networks (CNNs) to extract features from different regions in an image [133].

**3.1.2. Template-Based Captioning.** Another common approach of image annotation is template-based which involves the identification of certain attributes such as object type, shape, actions, and scenes in an image, which are then used in forming sentences in a prestructured template [134]. In this method, the predetermined template has a constant number of slots such that all the detected attributes are then positioned in the slots to make up a sentence. In this case, the words representing the detected features make up the caption and are arranged such that they are syntactically and semantically related, thus generating grammatically correct representations [135]. The fixed template problem of the template-based approach was overcome by incorporating a parsed language model [136], giving the network higher flexibility and ability to generate better captions.

The underlying nouns, scenes, verbs, and prepositions determining the main idea of a sentence were explored and

trained using a language model to obtain the probability distribution of such parameters [137]. Certain human postures and orientation which do not involve the movement of the hands such as walking, standing, and seeing and the position of the head were used to generate captions of an image [138]. Furthermore, the postures were extended to describe human behavior and interactions by incorporating motion features and associating them with the corresponding action [139]. Each body part and the action it is undergoing are identified, and then this is compiled and integrated to form a description of the complete human body. Human posture, position, and direction of the head and position of the hands were selected as geometric information for network modeling.

**3.1.3. Neural Network-Based Captioning.** Compared to other machine learning algorithms or preexisting approaches, deep learning has achieved astonishing heights in image captioning, setting new benchmarks with almost all of the datasets in all of the evaluation metrics. These deep learning approaches are mostly in the supervised setting which requires a huge amount of training data including both images and their corresponding caption labels. Several models have been applied such as artificial neural network (ANN), convolutional neural network (CNN), recurrent neural network (RNN), autoencoder, generative adversarial network (GAN), and even a combination of one or more of them.



**Dense captioning:** dense captioning emerges as a branch of computer vision whereby pictorial features are densely annotated depending on the object, object's motion, and its interaction. The concept depends on the identification of features as well as their localization and finally expressing such features with short descriptive phrases [140]. This idea is drawn from the understanding that providing a single description for a whole picture can be complex or sometimes bias; therefore, a couple of annotations are generated relating to different recognized salient features of the image. The training data of a dense caption in comparison to a global caption are different in that various labels are given for individual features identified by bounding boxes, whereby a general description is given in the global captioning without a need for placing bounding boxes on the images [141]. Visible in Figure 5, a phrase is generated from each region in the image, and these regions could be compiled to form a complete caption of the image. Generally, dense captioning models face a huge challenge as most of the target regions in the images overlap which makes accurate localization challenging and daunting [143].

The intermodal alignment of the text and images was investigated on region-level annotations which pioneers a new approach for captioning, leveraging the alignment between the feature embedding and the word vector semantic embedding [144]. A fully convolutional localization network (FCLN) was developed to determine important regions of interest in an image. The model combines a recurrent neural network language model and a convolutional neural network to enforce the logic of object detection and image description. The designed network uses dense localization layer and convolution anchors built on the Faster R-CNN technique to predict region proposal from the input features [142]. A contextual information model that combines previous and future features of a video spanning up to two minutes achieved dense captioning by transforming the video input into slices of frames. With this, an event proposal module helps to extract the context from the frames which are fed into a long short-term memory (LSTM) unit, enabling it to generate different proposals at different time steps [145]. Also, a framework having separate detection network and localization captioning network accomplished improved dense captioning with faster speed by directly producing detected features rather than via the common use of the region proposal network (RPN) [146].

**Encoder-decoder framework:** most image captioning tasks are built on the encoder-decoder structure whereby the images and texts are managed as separate entities by different models. In most cases, a convolutional neural network is presented as the encoder which acts as a feature extractor, while a recurrent neural network is presented as the decoder which serves as a language model to process the extracted features in parallel with the text label, consequently generating predicted captions for the input images [147]. CNN helps to identify and localize objects and their interaction, and then this insight is combined with long-term dependencies from a recurrent network cell to predict a word at a time, depending on the image context vector and previously generated words [148]. Multiple CNN-based encoders were

proposed to provide a more comprehensive and robust capturing of objects and their interaction from images. The idea of applying multiple encoders is suggested to complement each unit of the encoder to obtain better feature extraction. These interactions are then translated to a novel recurrent fusion network (RFNet) which could fuse and embed the semantics from the multiple encoders to generate meaningful textual representations and descriptions [149].

Laid out in Figure 6, a combination of two CNN models as both encoder and decoder was explored to speed up computational time for image captioning tasks. Because RNN's long-range information is computed step by step, this causes very expensive computation and is solved by stacking layers of convolution to mimic tree structure learning of the sentences [150]. Three distinct levels of features which are regional, visual, and semantic features were encoded in a model to represent different analyses of the input image, and then this is fed into a two-layer LSTM decoder for generating well-defined captions [151]. A concept-based sentence reranking technique was incorporated into the CNN-LSTM model such that concept detectors are added to the underlying sentence generation model for better image description with minimal manual annotation [152]. Furthermore, the generative adversarial network (GAN) was conditioned on a binary vector for captioning. The binary vector represented some form of sentiment which the image portrays and then was used to train the adversarial model. The model took both images and an adjective or adjective-noun pair as the input to determine if the network could generate a caption describing the intended sentimental stance [153].

**Attention-guided captioning:** attention has become increasingly paramount, and it has driven better benchmarks in several tasks such as machine translation, language modeling, and other natural language processing tasks, as well as computer vision tasks. In fact, attention has proven to correlate the meaning between features, and this helps to understand how such a feature relates to one another [154]. Incorporating this into a neural network, it encourages the model to focus on salient and relevant features and pay less consideration to other noisy aspects of the data space distribution [155]. To estimate the concept of attention in image annotation, a model is trained to concentrate its computation on the identified salient regions while generating captions using both soft and hard attention [156]. The deterministic soft attention which is trainable via standard backpropagation is learned by weighting the annotated vector of the image features, while the stochastic hard attention is trained via maximizing a variational lower bound, setting it to 1 when the feature is salient [157].

Following the where and what analysis of what the model should concentrate on, adaptive attention used a hierarchical structure to fuse both high-level semantic information and visual information from an image to form intuitive representation [120]. The top-down and bottom-up approaches are fused using semantic attention which first defines attribute detectors that dynamically enable it to switch between concepts. This empowers the detectors to determine suitable candidates for attention computation based on the specified

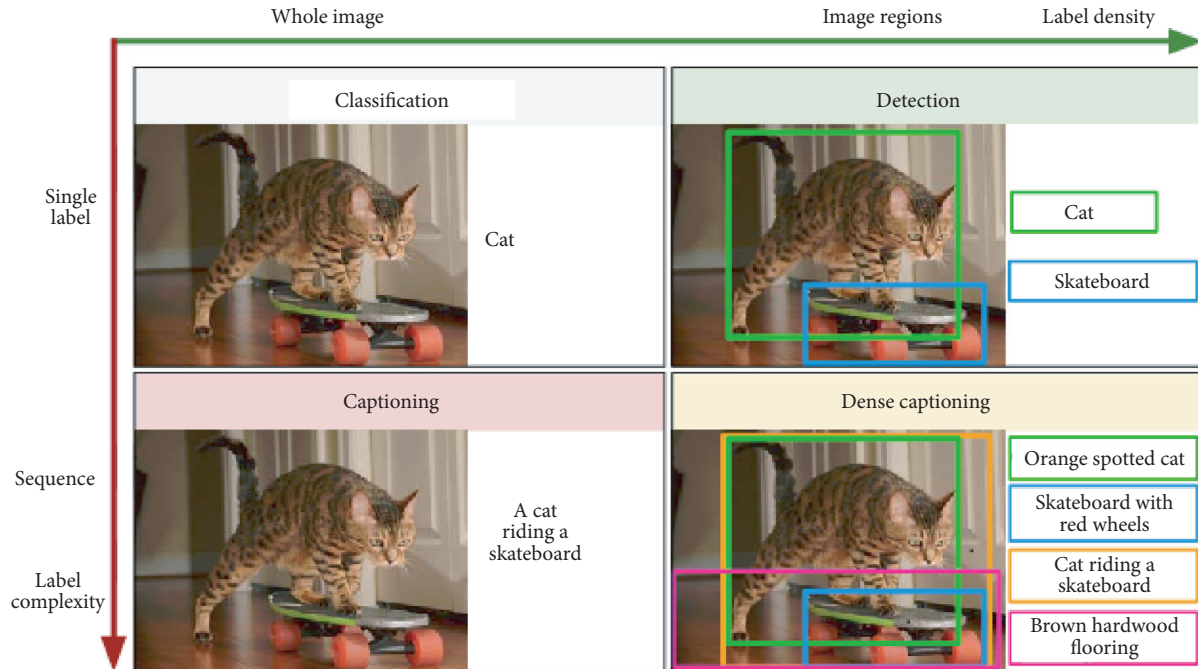


FIGURE 5: Dense captioning illustrating multiple annotations with a single forward pass [142].

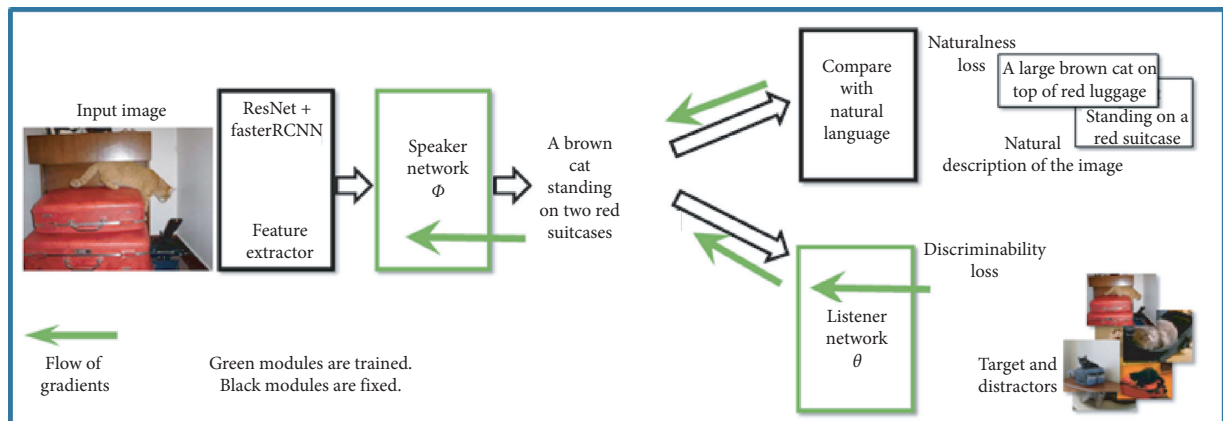


FIGURE 6: Sample architecture of a multimodal image captioning network [111].

inputs [158]. The limitation of long-distance dependency and inference speed in medical image captioning was tackled using a hierarchical transformer. The model includes an image encoder that extracts features with the support of a bottom-up attention mechanism to capture and extract top-down visual features, as well as a nonrecurrent transformer captioning decoder which helps to compile the generated medical illustration [159]. Salient regions in an image are also extracted using a convolutional model as region features were represented as pooled feature vector. These in-train image region vectors are appropriately attended to obtain suitable weights describing their influence before they are fed into a recurrent model that learns their semantic correlation. The corresponding sequence of the correlated features is transformed into representations which are illustrated as sentences describing the features' interactions [160].

**3.1.4. Unsupervised or Semisupervised Captioning.** Supervised captioning has so far been productive and successful partly due to the availability of sophisticated deep learning algorithms and an increasing outpour of data. Through the supervised deep learning techniques, a combination of models and frameworks which learn the joint distribution of images and labels has displayed a very intuitive and meaningful illustration of images even similar to humans. However, despite the achievement, the process of completely creating a captioning training set is quite daunting, and the manual effort required to annotate the myriad of images is very challenging. As a result, other means which are free of excessive training data are explored. An unsupervised captioning approach that combines two steps of query and retrieval was researched in [161]. First, several target images are obtained from the internet as well

as a huge database of words describing such images. For any chosen image, words representing its visual display are used to query captions from a reference dataset of sentences. This strategy helps to eliminate manual annotation and also uses multimodal textual-visual information to reduce the effect of noisy words in the vocabulary dataset.

Transfer learning which has seen increasing application in other deep learning domains, especially in computer vision, was applied to image captioning. First, the model is trained on a standard dataset in a supervised manner, and then the knowledge from the supervised model is transferred and applied on a different dataset whose sentences and images are not paired. For this purpose, two autoencoders were designed to train on the textual and visual dataset, respectively, using the distribution of the learned supervised embedding space to infer the unstructured dataset [162]. Also, the process of manual annotation of the training set was semiautomated by evaluating an image into several feature spaces which are individually estimated by an unsupervised clustering algorithm. The centers of the clustered groups are then manually labeled and compiled into a sentence through a voting scheme which compiles all the opinions suggested from each cluster [163]. A set of naïve Bayes model with AdaBoost was used for automatic image annotation by first using a Bayesian classifier to identify unlabeled images and then labeled by a succeeding classifier based on the confidence measurement of the prior classifier [164]. A combination of keywords which have been associated with both labeled and unlabeled images was trained using a graph model. The semantic consistency of the unlabeled images is computed and compared to the labeled images. This is continued until all the unlabeled images are successfully annotated [165].

*3.1.5. Difference Captioning.* As presented in Figure 7, a spot-the-difference task which describes the differences between two similar images using advance deep learning technique was first investigated in [166]. Their model used a latent variable to capture visual salience in an image pair by aligning pixels which differ in both images. Their work included different model designs such as nearest neighbor matching scheme, captioning masked model, and Difference Description with Latent Alignment uniform for obtaining difference captioning. The Difference Description with Latent Alignment (DDLA) compares both input images at a pixel level via a masked L2 distance function.

Furthermore, the Siamese Difference Captioning Model (SDCM) also combined techniques from deep Siamese convolutional neural network, soft attention mechanism, word embedding, and bidirectional long short-term memory [167]. The features in each image input are computed using the Siamese network, and their differences are obtained using a weighted L1 distance function. Different features are then recursively translated into text using a recurrent neural network and an attention network which focuses on the relevant region on the images. The idea of the Siamese Difference Captioning Model was extended by converting the Siamese encoder into a Fully Convolutional

CaptionNet (FCC) through a fully convolutional network [168]. This helps to transform the extracted features into a larger dimension of the input images which makes difference computation more efficient. Also, a word embedding pre-trained model was used to embed semantics into the text dataset and beam search technique to ensure multiple options for robustness.

*3.2. Datasets.* There are several publicly available datasets which are useful for training image captioning tasks. The most popular datasets include Flickr8k [169], Flickr30k [170], MS COCO dataset [83], Visual Genome dataset [171], Instagram dataset [172], and MIT-Adobe FiveK dataset [173].

Flickr30K dataset: it has about 30,000 images from Flickr and about 158,000 captions describing the content of the images. Because of the huge volume of the data, users are able to determine their preferred split size for using the data.

Flickr8K dataset: it has a total of 8,000 images which are divided as 6,000, 1,000, and 1,000 for the training, test, and validation set, respectively. All the images have 5 label captions which are used as a supervised setting for training the images.

Microsoft COCO dataset: it is perhaps the largest captioning dataset, and it also includes training data for object recognition and image segmentation tasks, respectively. The dataset contains around 300,000 images with 5 captions for each image.

*3.3. Evaluation Metrics.* The automatically generated captions are evaluated to confirm their correctness in describing the given image. In machine learning, some of the common image captioning evaluation measures are as follows.

BLEU (BiLingual Evaluation Understudy) [174]: as a metric, it counts the number of matching  $n$ -grams in the model's prediction compared to the ground truth. With this, precision is calculated based on the mean  $n$ -grams computed, and the recall is computed via the introduction of a brevity penalty in the caption label.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [175]: it is useful for summary evaluation and is calculated as the overlap of either 1-gram or bigrams between the referenced caption and the predicted sequence. Using the longest sequence available, the co-occurrence  $F$ -score mean of the predicted sequence's recall and prediction is obtained.

METEOR (Metric for Evaluation of Translation with Explicit Ordering) [176]: it addresses the drawback of BLEU, and it is based on a weighted  $F$ -score computation as well as a penalty function meant to check the order of the candidate sequence. It adopts synonyms matching in the detection of similarity between sentences.

CIDEr (Consensus-based Image Description Evaluation) [177]: it determines the consensus between a reference sequence and a predicted sequence via cosine similarity, stemming, and TF-IDF weighting. The predicted sequence is compared to the combination of all available reference sequences.



FIGURE 7: Image pair difference annotations of the Spot-the-Diff dataset [166]. (a) The blue truck is no longer there. (b) A car is approaching the parking lot from the right.

TABLE 2: Class pixel label distribution in the CamVid dataset.

Dataset	Method	B-1	B-2	B-3	B-4	M	C
MS COCO	LSTM-A-2 [179]	0.734	0.567	0.430	0.326	0.254	1.00
	Att-Reg [180]	0.740	0.560	0.420	0.310	0.260	—
	Attend-tell [156]	0.707	0.492	0.344	0.243	0.239	—
	SGC [181]	67.1	48.8	34.3	23.9	21.8	73.3
	phi-LSTM [182]	66.6	48.9	35.5	25.8	23.1	82.1
	COMIC [183]	70.6	53.4	39.5	29.2	23.7	88.1
	TBVA [184]	69.5	52.1	38.6	28.7	24.1	91.9
	SCN [185]	0.741	0.578	0.444	0.341	0.261	1.041
	CLGRU [186]	0.720	0.550	0.410	0.300	0.240	0.960
	A-Penalty [187]	72.1	55.1	41.5	31.4	24.7	95.6
	VD-SAN [188]	73.4	56.6	42.8	32.2	25.4	99.9
	ATT-CNN [189]	73.9	57.1	43.3	33	26	101.6
	RTAN [190]	73.5	56.9	43.3	32.9	25.4	103.3
	Adaptive [191]	0.742	0.580	0.439	0.332	0.266	1.085
Full-SL [192]	0.713	0.539	0.403	0.304	0.251	0.937	
Flickr30K	hLSTMat [193]	73.8	55.1	40.3	29.4	23	66.6
	SGC [181]	61.5	42.1	28.6	19.3	18.2	39.9
	RA + SF [194]	0.649	0.462	0.324	0.224	0.194	0.472
	gLSTM [195]	0.646	0.446	0.305	0.206	0.179	—
	Multi-Mod [196]	0.600	0.380	0.254	0.171	0.169	—
	TBVA [184]	66.6	48.4	34.6	24.7	20.2	52.4
	Attend-tell [156]	0.669	0.439	0.296	0.199	0.185	—
	ATT-FCN [158]	0.647	0.460	0.324	0.230	0.189	—
	VQA [197]	0.730	0.550	0.400	0.280	—	—
	Align-Mod [144]	0.573	0.369	0.240	0.157	—	—
	m-RNN [198]	0.600	0.410	0.280	0.190	—	—
	LRCN [112]	0.587	0.391	0.251	0.165	—	—
	NIC [141]	0.670	0.450	0.300	—	—	—
	RTAN [190]	67.1	48.7	34.9	23.9	20.1	53.3
	3-gated [199]	69.4	45.7	33.2	22.6	23	—
	VD-SAN [188]	65.2	47.1	33.6	23.9	19.9	—
ATT-CNN [189]	66.1	47.2	33.4	23.2	19.4	—	

SPICE (Semantic Propositional Image Caption Evaluation) [178]: it is a relatively new caption metric which relates with the semantic interrelationship between the generated and referenced sequence. Its graph-based methodology uses a scene graph of semantic representations to indicate details of objects and their interaction to describe their textual illustrations.

3.4. Discussion. With an increase in the generation of data, production of sophisticated computing hardware, and

complex machine learning algorithms, a lot of achievements have been accomplished in the field of image captioning. Though there have been several implementations, the best results in almost all of the metrics have been recorded through the use of deep learning models. In most cases, the common implementation has been the encoder-decoder architecture which has a feature extractor as the encoder and a language model as the decoder.

Compared to other approaches, this has proven useful as it has become the backbone for more recent designs. To achieve better feature computation, attention mechanism

concepts have been applied to help in focusing on the salient section of images and their features, thereby improving feature-text capturing and translation. In the same manner, other approaches such as generative adversarial network and autoencoders have been thriving in achieving concise image annotation, and to this end, such idea has been incorporated with other unsupervised concepts for captioning purposes as well. For example, reinforced learning technique also generated sequences which are able to succinctly describe images in a timely manner. Furthermore, analyses of several model designs and their results are displayed in Table 2, depicting their efficiency and effectiveness in the BLEU, METEOR, ROUGE-L, CIDEr, and SPICE metrics.

#### 4. Conclusion

In this survey, the state-of-the-art advances in semantic segmentation and image captioning have been discussed. The characteristics and effectiveness of the important techniques have been considered, as well as their process of achieving both tasks. Some of the methods which have accomplished outstanding results have been illustrated including the extraction, identification, and localization of objects in semantic segmentation. Also, the process of feature extraction and transformation into a language model has been studied in the image captioning section. In our estimation, we believe that because of the daunting task of manually segmenting images into semantic classes, as well as the human annotation of images involved in segmentation and captioning, future research would move in the direction of an unsupervised setting of accomplishing this task. This would ensure more energy, and focus is invested solely in the development of complex machine learning algorithms and mathematical models which could improve the present state of the art.

#### Data Availability

The dataset used for the evaluation of the models presented in this study have been discussed in the manuscript as well as their respective references and publications, such as PASCAL VOC: PASCAL Visual Object Classes (VOC) [82], MSCOCO: Microsoft Common Objects in Context [83], Cityscapes: Cityscapes dataset [84], ADE20K: ADE20K dataset [85], CamVid [86], Flickr8k [168], Flickr30k [169], MS COCO Dataset [83], VisualGenome Dataset [170], Instagram Dataset [171], and MIT-Adobe FiveK dataset [172].

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Acknowledgments

This work was supported in part by the NSFC-Guangdong Joint Fund (Grant no. U1401257), the National Natural Science Foundation of China (Grant nos. 61300090, 61133016, and 61272527), the Science and Technology Plan Projects in Sichuan Province (Grant no. 2014JY0172), and the Opening Project of Guangdong Provincial Key

Laboratory of Electronic Information Products Reliability Technology (Grant no. 2013A061401003).

#### References

- [1] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, "Computer vision for assistive technologies," *Computer Vision and Image Understanding*, vol. 154, pp. 1–15, 2017.
- [2] A. Kovashka, O. Russakovsky, L. Fei-Fei, and K. Grauman, "Crowdsourcing in computer vision," *Foundations and Trends in Computer Graphics and Vision*, vol. 10, no. 3, pp. 177–243, 2016.
- [3] H. Ayaz, M. Ahmad, M. Mazzara, and A. Sohaib, "Hyper-spectral imaging for minced meat classification using non-linear deep features," *Applied Sciences*, vol. 10, no. 21, p. 7783, 2020.
- [4] A. Oluwasanmi, M. U. Aftab, A. Shokanbi, J. Jackson, B. Kumeda, and Z. Qin, "Attentively conditioned generative adversarial network for semantic segmentation," *IEEE Access*, vol. 8, pp. 31733–31741, 2020.
- [5] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proceedings of the 2017 Conference on Neural Information Processing Systems*, Long Beach, CA, USA, December 2017.
- [6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.
- [7] B. G. Weinstein, "A computer vision for animal ecology," *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545, 2018.
- [8] A. Oluwasanmi, M. U. Aftab, Z. Qin et al., "Transfer learning and semisupervised adversarial detection and classification of COVID-19 in CT images," *Complexity*, vol. 2021, Article ID 6680455, 11 pages, 2021.
- [9] M. Ahmad, I. Haq, Q. Mushtaq, and M. Sohaib, "A new statistical approach for band clustering and band selection using k-means clustering," *International Journal of Engineering and Technology*, vol. 3, 2011.
- [10] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 975–984, Venice, Italy, October 2017.
- [11] M. Leo, A. Furnari, G. G. Medioni, M. M. Trivedi, and G. M. Farinella, "Deep learning for assistive computer vision," in *Proceedings of the European Conference on Computer Vision Workshops*, Munich, Germany, September 2018.
- [12] H. Fang, S. Gupta, F. N. Iandola et al., "From captions to visual concepts and back," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1473–1482, Boston, MA, USA, June 2015.
- [13] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.
- [14] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, Cambridge, UK, 2014.
- [15] E. Alpaydin, "Introduction to machine learning," *Adaptive Computation And Machine Learning*, MIT Press, Cambridge, MA, USA, 2004.
- [16] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in

- Proceedings of the Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA, August 2012.
- [17] I. G. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning*. Nature, vol. 521, pp. 436–444, 2015.
- [18] F. Hutter, L. Kothhoff, and J. Vanschoren, “Automated machine learning.: methods, systems, challenges,” *Automated Machine Learning*, MIT Press, Cambridge, MA, USA, 2019.
- [19] R. Singh, A. Sonawane, and R. Srivastava, “Recent evolution of modern datasets for human activity recognition: a deep survey,” *Multimedia Systems*, vol. 26, 2020.
- [20] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: a large video database for human motion recognition,” in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2556–2563, Barcelona, Spain, November 2011.
- [21] R. Poppe, “A survey on vision-based human action recognition,” *Image Vision Comput*, vol. 28, pp. 976–990, 2011.
- [22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] R. Dey and F. M. Salem, “Gate-variants of gated recurrent unit (GRU) neural networks,” in *Proceedings of the IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, Boston, MA, USA, August 2017.
- [24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, <http://arxiv.org/abs/1409.1556>.
- [25] O. Ivanov, M. Figurnov, and D. P. Vetrov, “Variational autoencoder with arbitrary conditioning,” in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.
- [26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [28] K. Khan, M. S. Siddique, M. Ahmad, and M. Mazzara, “A hybrid unsupervised approach for retinal vessel segmentation,” *BioMed Research International*, vol. 2020, Article ID 8365783, 20 pages, 2020.
- [29] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.
- [30] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, July 2017.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [32] H. Hassan, A. Bashir, M. Ahmad et al., “Real-time image dehazing by superpixels segmentation and guidance filter,” *Journal of Real-Time Image Processing*, 2020.
- [33] C. Liu, L. Chen, F. Schroff et al., “Auto-DeepLab: hierarchical neural architecture search for semantic image segmentation,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2019.
- [34] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, “Panoptic segmentation,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2019.
- [35] H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. Ben Ayed, “Constrained-CNN losses for weakly supervised segmentation,” *Medical Image Analysis*, vol. 54, pp. 88–99, 2019.
- [36] A. Arnab, S. Zheng, S. Jayasumana et al., “Conditional random fields meet deep neural networks for semantic segmentation: combining probabilistic graphical models with deep learning for structured prediction,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 37–52, 2018.
- [37] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, pp. 146–168, 2004.
- [38] A. Oluwasanmi, Z. Qin, and T. Lan, “Brain MR segmentation using a fusion of K-means and spatial Fuzzy C-means,” in *Proceeding of International Conference on Computer Science and Application Engineering*, Wuhan, China, July 2017.
- [39] A. Oluwasanmi, Z. Qin, T. Lan, and Y. Ding, “Brain tissue segmentation in MR images with FGM,” in *Proceeding of the International Conference on Artificial Intelligence and Computer Science*, Guilin, China, December 2016.
- [40] J. Chen, C. Yang, G. Xu, and L. Ning, “Image segmentation method using Fuzzy C mean clustering based on multi-objective optimization,” *Journal of Physics: Conference Series*, vol. 1004, pp. 12–35, 2018.
- [41] A. Oluwasanmi, Z. Qin, and T. Lan, “Fusion of Gaussian mixture model and spatial Fuzzy C-means for brain MR image segmentation,” in *Proceedings of International Conference on Computer Science and Application Engineering*, Wuhan, China, July 2017.
- [42] B. J. Liu and L. Cao, “Superpixel segmentation using Gaussian mixture model,” *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4105–4117, 2018.
- [43] B. Kang and T. Q. Nguyen, “Random forest with learned representations for semantic segmentation,” *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3542–3555, 2019.
- [44] Z. Zhao and X. Wang, “Multi-egments Naïve Bayes classifier in likelihood space,” *IET Computer Vision*, vol. 12, no. 6, pp. 882–891, 2018.
- [45] T. Y. Tan, L. Zhang, C. P. Lim, B. Fielding, Y. Yu, and E. Anderson, “Evolving ensemble models for image segmentation using enhanced particle swarm optimization,” *IEEE Access*, vol. 7, pp. 34004–34019, 2019.
- [46] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, “Hyperspectral image classification with Markov random fields and a convolutional neural network,” *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2354–2367, 2018.
- [47] S. Mohapatra, “Segmentation using support vector machines,” in *Proceedings of the Second International Conference on Advanced Computational and Communication Paradigms (ICACCP 2019)*, pp. 1–4, Gangtok, India, November 2019.
- [48] Ç. Kaymak and A. Uçar, “A brief survey and an application of semantic image segmentation for autonomous driving,” *Handbook of Deep Learning Applications*, Springer, Berlin, Germany, 2018.
- [49] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik, “Simultaneous detection and segmentation,” in *Proceedings of the European Conference on Computer Vision*, Zurich, Switzerland, September 2014.
- [50] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

- [51] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4438–4446, Honolulu, HI, USA, July 2017.
- [52] H. Caesar, J. Uijlings, and V. Ferrari, "Region-based semantic segmentation with end-to-end training," in *Proceedings of the European Conference on Computer Vision*, pp. 381–397, Amsterdam, The Netherlands, October 2016.
- [53] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.
- [54] N. Wang, S. Li, A. Gupta, and D. Yeung, "Transferring rich feature hierarchies for robust visual tracking," 2015, <http://arxiv.org/abs/1501.04587>.
- [55] R. B. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.
- [56] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.
- [57] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.
- [58] A. Salvador, X. Giró, F. Marqués, and S. Satoh, "Faster R-CNN features for instance search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2016)*, pp. 394–401, Las Vegas, NV, USA, June 2016.
- [59] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [60] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [61] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, Santiago, Chile, December 2015.
- [62] O. Ronneberger, P. Fischer, T. Brox, and U-Net, *Convolutional Networks for Biomedical Image Segmentation*, MIC-CAI, Munich, Germany, 2015.
- [63] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. J. Pal, "The importance of skip connections in biomedical image segmentation," 2016, <http://arxiv.org/abs/1608.04117>.
- [64] S. Jégou, M. Drozdal, D. Vázquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: fully convolutional DenseNets for semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, July 2017.
- [65] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: gated shape CNNs for semantic segmentation," 2019, <http://arxiv.org/abs/1907.05740>.
- [66] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, <http://arxiv.org/abs/1511.07122>.
- [67] P. Bilinski and V. Prisacariu, "Dense decoder shortcut connections for single-pass semantic segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6596–6605, Salt Lake City, UT, USA, June 2018.
- [68] Z. Zhang, X. Zhang, C. Peng, D. Cheng, and J. Sun, "ExFuse: enhancing feature fusion for semantic segmentation," in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.
- [69] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision*, Kolding, Denmark, June 2017.
- [70] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: deep feature aggregation for real-time semantic segmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2019.
- [71] W. Xiang, H. Mao, and V. Athitsos, "ThunderNet: a turbo unified network for real-time semantic segmentation," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1789–1796, Village, HI, USA, January 2019.
- [72] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proceedings of the International Conference on Learning Representations*, pp. 11–25, San Diego, CA, USA, May 2015.
- [73] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2016.
- [74] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, <http://arxiv.org/abs/1706.05587>.
- [75] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: rethinking dilated convolution in the backbone for semantic segmentation," 2019, <http://arxiv.org/abs/1903.11816>.
- [76] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3684–3692, Salt Lake City, UT, USA, June 2018.
- [77] P. H. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with Convolutional Networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1713–1721, Boston, MA, USA, June 2015.
- [78] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele, "Simple does it: weakly supervised instance and semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1665–1674, Honolulu, HI, USA, July 2017.
- [79] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1742–1750, Santiago, Chile, December 2015.
- [80] J. Dai, K. He, and J. Sun, "BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1635–1643, Santiago, Chile, December 2015.

- [81] W. Hung, Y. Tsai, Y. Liou, Y. Lin, and M. Yang, "Adversarial learning for semi-supervised semantic segmentation," in *Proceedings of the British Machine Vision Conference*, Newcastle, UK, September 2018.
- [82] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2009.
- [83] T. Lin, M. Maire, S. J. Belongie et al., "Microsoft COCO: common objects in context," *ECCV*, pp. 740–755, Springer, Berlin, Germany, 2014.
- [84] M. Cordts, M. Omran, S. Ramos et al., "The Cityscapes dataset," in *Proceedings of the CVPR Workshop on the Future of Datasets in Vision*, Boston, MA, USA, June 2015.
- [85] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, Honolulu, HI, USA, July 2017.
- [86] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [87] G. Ros, L. Sellart, J. Materzynska, D. Vázquez, and A. M. López, "The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3234–3243, Las Vegas, NV, USA, June 2016.
- [88] S. Nowozin, "Optimal decisions from probabilistic models: the intersection-over-union case," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 548–555, Columbus, OH, USA, June 2014.
- [89] C. Wu, H. Cheng, S. Li, H. Li, and Y. Chen, "ApesNet: a pixel-wise efficient segmentation network for embedded devices," in *Proceedings of the 14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia*, pp. 1–7, Pittsburgh, PA, USA, October 2016.
- [90] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: a deep neural network architecture for real-time semantic segmentation," 2016, <http://arxiv.org/abs/1606.02147>.
- [91] A. Chaurasia and E. Culurciello, "LinkNet: exploiting encoder representations for efficient semantic segmentation," in *Proceedings of the IEEE Visual Communications and Image Processing*, pp. 1–4, St. Petersburg, FL, USA, December 2017.
- [92] L. Fan, W. Wang, F. Zha, and J. Yan, "Exploring new backbone and attention module for semantic segmentation in street scenes," *IEEE Access*, vol. 6, 2018.
- [93] C. Yu, J. Wang, G. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.
- [94] J. Li, Y. Zhao, J. Fu, J. Wu, and J. Liu, "Attention-guided network for semantic video segmentation," *IEEE Access*, vol. 7, pp. 140680–140689, 2019.
- [95] H. Zhou, K. Song, X. Zhang, W. Gui, and Q. Qian, "WAILS: watershed algorithm with image-level supervision for weakly supervised semantic segmentation," *IEEE Access*, vol. 7, pp. 42745–42756, 2019.
- [96] L. Zhang, P. Shen, G. Zhu et al., "Improving semantic image segmentation with a probabilistic superpixel-based dense conditional random field," *IEEE Access*, vol. 6, pp. 15297–15310, 2018.
- [97] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3376–3385, Boston, MA, USA, June 2015.
- [98] C. Han, Y. Duan, X. Tao, and J. Lu, "Dense convolutional networks for semantic segmentation," *IEEE Access*, vol. 7, pp. 43369–43382, 2019.
- [99] R. Vemulapalli, O. Tuzel, M. Y. Liu, and R. Chellapa, "Gaussian conditional random field network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3224–3233, Las Vegas, NV, USA, June 2016.
- [100] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1377–1385, Santiago, Chile, December 2015.
- [101] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3194–3203, Las Vegas, NV, USA, July 2016.
- [102] C. X. Peng, X. Zhang, K. Jia, G. Yu, and J. Sun, "MegDet: a large mini-batch object detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6181–6189, Salt Lake City, UT, USA, June 2008.
- [103] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proceedings of the European Conference on Computer Vision*, pp. 519–534, Amsterdam, The Netherlands, October 2016.
- [104] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4151–4160, Honolulu, HI, USA, July 2017.
- [105] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5168–5177, Honolulu, HI, USA, July 2017.
- [106] H.-H. Han and L. Fan, "A new semantic segmentation model for supplementing more spatial information," *IEEE Access*, vol. 7, pp. 86979–86988, 2019.
- [107] X. Jin, X. Li, H. Xiao et al., "Video scene parsing with predictive feature learning," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5581–5589, Venice, Italy, October 2017.
- [108] P. Wang, P. Chen, Y. Yuan et al., "Understanding convolution for semantic segmentation," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1451–1460, Lake Tahoe, NV, USA, March 2018.
- [109] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, Honolulu, HI, USA, June 2017.
- [110] G. Vered, G. Oren, Y. Atzmon, and G. Chechik, "Cooperative image captioning," 2019, <http://arxiv.org/abs/1907.11565>.
- [111] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, and S. Venugopalan, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634, Boston, MA, USA, June 2015.
- [112] Z. Fan, Z. Wei, S. Wang, and X. Huang, *Bridging by Word: Image Grounded Vocabulary Construction for Visual Captioning*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019.



- [113] Y. Zhou, Y. Sun, and V. Honavar, "Improving image captioning by leveraging knowledge graphs," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 283–293, Waikoloa Village, HI, USA, January 2019.
- [114] X. Li and S. Jiang, "Know more say less: image captioning based on scene graphs," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2117–2130, 2019.
- [115] Q. Wang and A. B. Chan, "Describing like humans: on diversity in image captioning," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2019.
- [116] J. Gu, S. R. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," 2019, <http://arxiv.org/abs/1903.10658>.
- [117] X. Zhang, Q. Wang, S. Chen, and X. Li, "Multi-scale cropping mechanism for remote sensing image captioning," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 10039–10042, Yokohama, Japan, August 2019.
- [118] S. Wang, L. Lan, X. Zhang, G. Dong, and Z. Luo, "Cascade semantic fusion for image captioning," *IEEE Access*, vol. 7, pp. 66680–66688, 2019.
- [119] Y. Su, Y. Li, N. Xu, and A. Liu, "Hierarchical deep neural network for image captioning," *Neural Processing Letters*, vol. 52, pp. 1–11, 2019.
- [120] M. Yang, W. Zhao, W. Xu et al., "Multitask learning for cross-domain image captioning," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1047–1061, 2019.
- [121] Z. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [122] S. Sheng, K. Laenen, and M. Moens, "Can image captioning help passage retrieval in multimodal question answering?" in *Proceedings of European Conference on Information Retrieval (ECIR)*, pp. 94–101, Springer, Cologne, Germany, April 2019.
- [123] N. Yu, X. Hu, B. Song, J. Yang, and J. Zhang, "Topic-oriented image captioning based on order-embedding," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2743–2754, 2019.
- [124] S. Sreedevi and S. Sebastian, "Content based image retrieval based on Database revision," in *Proceedings of the International Conference on Machine Vision and Image Processing*, pp. 29–32, Taipei, Taiwan, December 2012.
- [125] E. R. Vimina and J. K. Poulouse, "Image retrieval using colour and texture features of Regions of Interest," in *Proceedings of the International Conference on Information Retrieval and Knowledge Management*, pp. 240–243, Kuala Lumpur, Malaysia, December 2012.
- [126] V. Ordonez, G. Kulkarni, and T. R. Berg, "Im2text: describing images using 1 million captioned photographs," in *Advances in Neural Information Processing Systems*, pp. 1143–1151, Springer, Berlin, Germany, 2011.
- [127] J. R. Curran, S. Clark, and J. Bos, "Linguistically motivated large-scale NLP with C and C and boxer," in *Proceedings of the Forty Fifth Annual Meeting of the ACL on Inter-Active Poster and Demonstration Sessions*, pp. 33–36, Prague, Czech, June 2007.
- [128] D. R. Hardoon, S. R. Szedmak, J. R. Shawe-Taylor, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [129] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.
- [130] J. Wang, Y. Zhao, Q. Qi et al., "MindCamera: interactive sketch-based image retrieval and synthesis," *IEEE Access*, vol. 6, pp. 3765–3773, 2018.
- [131] D. Xu, X. Alameda-Pineda, J. Song, E. Ricci, and N. Sebe, "Cross-paced representation learning with partial curricula for sketch-based image retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4410–4421, 2018.
- [132] K. Song, F. Li, F. Long, J. Wang, J. Wang, and Q. Ling, "Discriminative deep feature learning for semantic-based image retrieval," *IEEE Access*, vol. 6, pp. 44268–44280, 2018.
- [133] P. Kuznetsova, V. Ordonez, A. C. Berg, T. Berg, and Y. Choi, "Collective generation of natural image descriptions," *Association for Computational Linguistics*, vol. 1, pp. 359–368, 2012.
- [134] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "TREETALK: composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, no. 10, pp. 351–362, 2014.
- [135] M. Mitchell, "Midge: generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 747–756, Avignon, France, April 2012.
- [136] Y. Yang, C. L. Teo, H. Daumé, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 444–454, Portland, OR, USA, June 2011.
- [137] A. Kojima, M. Izumi, T. Tamura, and K. Fukunaga, "Generating natural language description of human behavior from video images," in *Proceedings of the ICPR 2000*, vol. 4, pp. 728–731, Barcelona, Spain, September 2000.
- [138] A. Kojima, T. Tamura, and K. Fukunaga, "natural language description of human activities from video images based on concept hierarchy of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 171–184, 2002.
- [139] A. Tariq and H. Foroosh, "A context-driven extractive framework for generating realistic image descriptions," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 619–632, 2017.
- [140] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: a neural image caption generator," 2014, <http://arxiv.org/abs/1411.4555>.
- [141] J. M. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: fully convolutional localization networks for dense captioning," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4565–4574, Las Vegas, NV, US, June 2016.
- [142] X. Li, W. Lan, J. Dong, and H. Liu, "Adding Chinese captions to images," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 271–275, New York, NY, USA, June 2016.
- [143] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," 2014, <http://arxiv.org/abs/1412.2306>.
- [144] R. Krishna, K. Hata, F. Ren, F. Li, and J. C. Niebles, "Dense-captioning events in videos," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, Venice, Italy, October 2017.
- [145] L. Yang, K. D. Tang, J. Yang, and L. Li, "Dense captioning with joint inference and visual context," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1978–1987, Honolulu, HI, USA, July 2017.

- [146] G. Srivastava and R. Srivastava, *A Survey on Automatic Image Captioning. International Conference on Mathematics and Computing*, Springer, Berlin, Germany, 2018.
- [147] X. Li, X. Song, L. Herranz, Y. Zhu, and S. Jiang, "Image captioning with both object and scene information," in *ACM Multimedia*, Springer, Berlin, Germany, 2016.
- [148] W. Jiang, L. Ma, Y. Jiang, W. Liu, and T. Zhang, "Recurrent fusion network for image captioning," in *Proceedings of the ECCV*, Munich, Germany, September 2018.
- [149] Q. Wang and A. B. Chan, "CNN+CNN: convolutional decoders for image captioning," 2018, <http://arxiv.org/abs/1805.09019>.
- [150] K. Zheng, C. Zhu, S. Lu, and Y. Liu, *Multiple-Level Feature-Based Network for Image Captioning*, Springer, Berlin, Germany, 2018.
- [151] X. Li and Q. Jin, *Improving Image Captioning by Concept-Based Sentence Reranking*, Springer, Berlin, Germany, 2016.
- [152] T. Karayil, A. Irfan, F. Raue, J. Hees, and A. Dengel, *Conditional GANs for Image Captioning with Sentiments*, ICANN, Los Angeles, CA, USA, 2019.
- [153] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the NIPS*, Long Beach, CA, USA, December 2017.
- [154] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, January 2016.
- [155] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning*, Lille, France, July 2015.
- [156] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015, <http://arxiv.org/abs/1409.0473>.
- [157] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4651–4659, Las Vegas, NV, USA, June 2016.
- [158] Y. Xiong, B. Du, and P. Yan, "Reinforced transformer for medical image captioning," in *Proceedings of the MLMI@MICCAI*, Shenzhen, China, October 2019.
- [159] S. Wang, H. Mo, Y. Xu, W. Wu, and Z. Zhou, "Intra-image region context for image captioning," *PCM*, Springer, Berlin, Germany, 2018.
- [160] L. Pellegrin, J. A. Vanegas, J. Arevalo et al., "A two-step retrieval method for image captioning," *Lecture Notes in Computer Science*, pp. 150–161, Springer, Berlin, Germany, 2016.
- [161] A. Carraggi, M. Cornia, L. Baraldi, and R. Cucchiara, "Visual-semantic alignment across domains using a semi-supervised approach," in *Proceedings of the European Conference on Computer Vision Workshops*, pp. 625–640, Munich, Germany, September 2018.
- [162] S. Vajda, D. You, S. Antani, and G. Thoma, "Large image modality labeling initiative using semi-supervised and optimized clustering," *International Journal of Multimedia Information Retrieval*, vol. 4, no. 2, pp. 143–151, 2015.
- [163] H. M. Castro, L. E. Sucar, and E. F. Morales, "Automatic image annotation using a semi-supervised ensemble of classifiers," *Lecture Notes in Computer Science*, vol. 4756, pp. 487–495, Springer, Berlin, Germany, 2007.
- [164] Y. Xiao, Z. Zhu, N. Liu, and Y. Zhao, "An interactive semi-supervised approach for automatic image annotation," in *Proceedings of the Pacific-Rim Conference on Multimedia*, pp. 748–758, Singapore, December 2012.
- [165] H. Jhamtani and T. Berg-Kirkpatrick, "Learning to describe differences between pairs of similar images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, October 2018.
- [166] A. Oluwasanmi, M. U. Aftab, E. Alabdulkreem, B. Kumeda, E. Y. Baagyere, and Z. Qin, "CaptionNet: automatic end-to-end siamese difference captioning model with attention," *IEEE Access*, vol. 7, pp. 106773–106783, 2019.
- [167] A. Oluwasanmi, E. Frimpong, M. U. Aftab, E. Y. Baagyere, Z. Qin, and K. Ullah, "Fully convolutional CaptionNet: siamese difference captioning attention model," *IEEE Access*, vol. 7, pp. 175929–175939, 2019.
- [168] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.
- [169] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2641–2649, Santiago, Chile, December 2015.
- [170] R. Krishna, Y. Zhu, O. Groth et al., "Visual genome: connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.
- [171] K. Tran, X. He, L. Zhang, and J. Sun, "Rich image captioning in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 434–441, Las Vegas, NA, USA, July 2016.
- [172] V. Bychkovskiy, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," *Computer Vision and Pattern Recognition (CVPR)*, vol. 97, 2011.
- [173] K. Papineni, S. Roukos, T. Ward, and W. Zhu, *Bleu: A Method For Automatic Evaluation Of Machine Translation*, pp. 311–318, Association for Computational Linguistics, Stroudsburg, PA, USA, 2001.
- [174] C. Lin, *ROUGE: A Package For Automatic Evaluation Of Summaries*, pp. 74–81, Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, 2004.
- [175] S. Banerjee and A. Lavie, "METEOR: an automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings Of The Meeting Of The Association For Computational Linguistics*, pp. 65–72, Ann Arbor, MI, USA, June 2005.
- [176] R. Vedantam, C. Zitnick, and D. Parikh, "CIDER: consensus-based image description evaluation," in *Proceedings Of The Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, Boston, MA, USA, June 2015.
- [177] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: semantic propositional image caption evaluation," in *Proceedings Of The European Conference on Computer Vision*, pp. 382–398, Springer, Amsterdam, The Netherlands, October 2016.
- [178] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings Of The IEEE International Conference on Computer Vision (ICCV)*, pp. 4904–4912, Venice, Italy, October 2017.

- [179] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367–1381, 2018.
- [180] N. Xu, A.-A. Liu, J. Liu et al., "Scene graph captioner: image captioning based on structural visual representation," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 477–485, 2019.
- [181] Y. H. Tan, C. S. Chan, and C. S. Chan, "Phrase-based image caption generator with hierarchical LSTM network," *Neurocomputing*, vol. 333, pp. 86–100, 2019.
- [182] J. H. Tan, C. S. Chan, and J. H. Chuah, "COMIC: towards a compact image captioning model with attention," *IEEE Transactions on Multimedia*, vol. 99, 2019.
- [183] C. He and H. Hu, "Image captioning with text-based visual attention," *Neural Processing Letters*, vol. 49, no. 1, pp. 177–185, 2019.
- [184] Z. Gan, C. Gan, X. He et al., "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, June 2017.
- [185] J. Gu, G. Wang, J. Cai, and T. Chen, "An empirical study of language cnn for image captioning," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.
- [186] J. Li, M. K. Ebrahimpour, A. Moghtaderi, and Y.-Y. Yu, "Image captioning with weakly-supervised attention penalty," 2019, <http://arxiv.org/abs/1903.02507>.
- [187] X. He, B. Yang, X. Bai, and X. Bai, "VD-SAN: visual-densely semantic attention network for image caption generation," *Neurocomputing*, vol. 328, pp. 48–55, 2019.
- [188] D. Zhao, Z. Chang, S. Guo, Z. Chang, and S. Guo, "A multimodal fusion approach for image captioning," *Neurocomputing*, vol. 329, pp. 476–485, 2019.
- [189] W. Wang and H. Hu, "Image captioning using region-based attention joint with time-varying attention," *Neural Processing Letters*, vol. 13, 2019.
- [190] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: adaptive attention via A visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [191] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [192] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical LSTMs with adaptive attention for visual captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2019.
- [193] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang, "Aligning where to see and what to tell: image captioning with region-based attention and scene-specific contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 2016.
- [194] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2407–2415, Santiago, Chile, December 2015.
- [195] R. Kiros, R. Zemel, and R. Salakhutdinov, "Multimodal neural language models," in *Proceedings of the International Conference on Machine Learning*, Beijing, China, June 2014.
- [196] Q. Wu, C. Shen, P. Wang, A. Dick, and A. V. Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1367–1381, 2016.
- [197] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks," in *Proceedings of the International Conference on Learning Representation*, San Diego, CA, USA, May 2015.
- [198] A. Yuan, X. Li, and X. Lu, "3G structure for image caption generation," *Neurocomputing*, vol. 330, pp. 17–28, 2019.
- [199] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proceedings of the ECCV*, pp. 44–57, Marseille, France, October 2008.

## Research Article

# Transfer Learning and Semisupervised Adversarial Detection and Classification of COVID-19 in CT Images

Ariyo Oluwasanmi <sup>1</sup>, Muhammad Umar Aftab <sup>1,2</sup>, Zhiguang Qin <sup>1</sup>, Son Tung Ngo <sup>3</sup>,  
Thang Van Doan <sup>3</sup>, Son Ba Nguyen <sup>3</sup> and Son Hoang Nguyen <sup>3</sup>

<sup>1</sup>School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

<sup>2</sup>Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan

<sup>3</sup>ICT Department, FPT University, Hanoi 10000, Vietnam

Correspondence should be addressed to Zhiguang Qin; qinz@uestc.edu.cn

Received 22 December 2020; Revised 8 January 2021; Accepted 1 February 2021; Published 16 February 2021

Academic Editor: Dan Selisteanu

Copyright © 2021 Ariyo Oluwasanmi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The ongoing coronavirus 2019 (COVID-19) pandemic caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has resulted in a severe ramification on the global healthcare system, principally because of its easy transmission and the extended period of the virus survival on contaminated surfaces. With the advances in computer-aided diagnosis and artificial intelligence, this paper presents the application of deep learning and adversarial network for the automatic identification of COVID-19 pneumonia in computed tomography (CT) scans of the lungs. The complexity and time limitation of the reverse transcription-polymerase chain reaction (RT-PCR) swab test makes it disadvantageous to depend solely on as COVID-19's central diagnostic mechanism. Since CT imaging systems are of low cost and widely available, we demonstrate that the drawback of the RT-PCR can be alleviated with a faster, automated, and reduced contact diagnostic process via the use of a neural network model for the classification of infected and noninfected CT scans. In our proposed model, we explore the benefit of transfer learning as a means of resolving the problem of inadequate dataset and the importance of semisupervised generative adversarial network for the extraction of well-mapped features and generation of image data. Our experimental evaluation indicates that the proposed semisupervised model achieves reliable classification, taking advantage of the reflective loss distance between the real data sample space and the generated data.

## 1. Introduction

Computer-aided diagnosis (CAD) has become an integral part of radiology and clinical diagnosis since the past decades when the emergence of sophisticated imaging techniques, like X-ray, ultrasound, and MRI, became evident. As such, valuable information is made available to experts, which requires professional evaluation and analysis for detection of abnormalities and classification of pathological traits [1].

Traditionally, the process of medical diagnosis requires manual observations based on domain knowledge; however, owing to technological advancement, newer approaches that

employ computer-aided practices have been incorporated with the use of artificial intelligence and computer vision techniques significantly gaining grounds [3]. Today, the need for integrating computer algorithms and models with medical diagnosis is made all the more necessary with the global pandemic caused by the novel coronavirus SARS-CoV-2 or coronavirus disease 2019 (COVID-19), as named by the World Health Organization (WHO) [4]. Since December 2019, when it was first discovered, there have been a total of 75,110,651 confirmed cases, including around 1,680,395 confirmed deaths as reported by the WHO as on the 21st of December 2020 [5].

Owing to the nature of the disease, making it easily transmissible from patients to medical health givers, and also the unavailability of supplies such as personal protective equipment (PPE) as a result of mandatory lockdown forcing the closure of majority manufacturing factories, it is imperative to find an effective solution to handling the virus with minimal human interaction and the spread of the pathogens. COVID-19 is a disease of the lungs majorly caused by the acute respiratory syndrome, and its symptoms include excessive cough, fever, and shortness of breath leading to pneumonia in patients [6]. As such, computed tomography (CT) imaging has been used extensively to detect progression, infectious lesions, and the severity of pulmonary pneumonia in patients' lungs in many countries. As in Figure 1, at the preliminary stages of COVID-19 pneumonia, it is noticed that chest CT images present small, subpleural, and peripheral GGO, which is rather daunting to detect [7]. Because of the complex appearances of the lesions and the complex manual task involved in the disease detection, more focus has been shifted to other automatic alternatives of analyzing the lung CT scans. Adding to these, visual fatigue increases the risks of experts missing out on necessary diagnostic details of tiny lesions.

In the past, artificial neural networks (ANNs) have achieved significant success in medical imaging, such as tumor detection and cancer screening [8]. Also, with the application of computer vision techniques as features classification, localization, and segmentation, radiologists are increasingly assisted by computer-aided diagnosis. Notwithstanding the ability of deep learning algorithms, most of these implementations require massive amount of data which are often unavailable. Also, in the more successful deep learning supervised framework where the model is trained to map the relationship of features to a label, it is challenging to get as many annotated or labeled training samples [9], especially with novel diseases such as COVID-19. For this reason, in this research, we experiment the possibility of an automatic detection of COVID-19 using multiple deep learning techniques on available volumes of lung CT scans.

First, we design a custom VGG16 convolutional neural network model for the identification of COVID-19-infected CT scans. Also, we exploit the possibility of transfer learning for the virus classification, taking advantage of pretrained models as feature extractor and then fine-tuning the new data with the model. This initiative helps to reduce training time, computational power, and the shortage of data dilemma [10]. Using the logic of the generative adversarial network [11], we compare the discriminator model against the generator model, such that the generator learns to create features similar enough to the real data distribution, thereby deceiving the discriminator. Over time, the generator learns the underlying patterns in the input data as similar as possible to the original CT scans. By updating both models dynamically, we frame the generative model in a supervised learning structure which is capable of learning internal representations of data, albeit in a semisupervised manner [12]. Since GANs are efficient at learning density distributions of inputs, they provide a creative way to learn the

features of complex image structures such as CT scans. By extending the proficiency of GAN to a semisupervised framework, it resolves the problem of limited data particularly experienced with COVID-19. This is accomplished by training a subset of both labeled and unlabeled COVID-19 and normal images which could accurately generalize to unseen data. To achieve this goal, unlike the traditional GAN, we train a generator, supervised discriminator, and unsupervised discriminator model simultaneously.

## 2. Related Works

*2.1. CT Imagery and COVID-19 Latent Representation.* COVID-19 is a family of the coronaviruses whose genetic material is encoded in the ribonucleic acid (RNA). It is therefore a positive-sense single-stranded RNA virus which could be detected in a couple of ways, including via the detection of the virus' RNA or antibodies of the patient's immune system [13]. As a result, the diagnosis of the disease could be accomplished by reverse transcription-polymerase chain reaction (RT-PCR) [14]. However, the process of sample collection is quite complicated, and the RT-PCR detection suffers from low sensitivity at the initial stage. On the other hand, pneumonia inflammation of the lungs' air sacs makes it observable through a chest CT scan. As such, common COVID-19 infection manifestation such as the ground-glass opacity (GGO) becomes observable at the early stages. The condition of the pulmonary consolidation becomes detectable, especially at the later stages, allowing CT scans to produce a 3D view of the lungs [2]. This makes CT imaging a dominant diagnostic modality for COVID-19 diagnosis. The longitudinal changes and the relationships observed between multiple types of CT slices could then be carefully evaluated to provide essential information for COVID-19 diagnosis.

Although chest X-radiation (X-rays) and CT scans are the most commonly used modalities for pneumonia detection, CT scans are preferred as they provide cross-sectional images useful for 3D image reconstruction compared to X-rays which produce flattened 2D images. Furthermore, COVID-19 denotative traits are more effectively visible in a 3D view; therefore, CT scans have been widely accepted as the screening tool for the disease diagnosis [15]. Regardless of the positives, detection of COVID-19 infection from CT volumes remains a considerable challenge as there is a large variation in size, positioning, and texture of the CT data. The nature of the data, such as small interclass variance, blurry edges, and low contrast boundaries, means an intrinsic understanding of the fundamental latent representation must be ascertained to avoid false-negative detections [16].

*2.2. COVID-19 AI-Based CAD Systems.* Because of its high efficiency, deep learning models have been utilized for radiological imaging analysis and defect detection in patients. Over time, this system has been incorporated as an assistive tool for medical exploratory and decision-making analysis. Even with COVID-19, several systems have been proposed for the virus detection based on chest radiography images. A

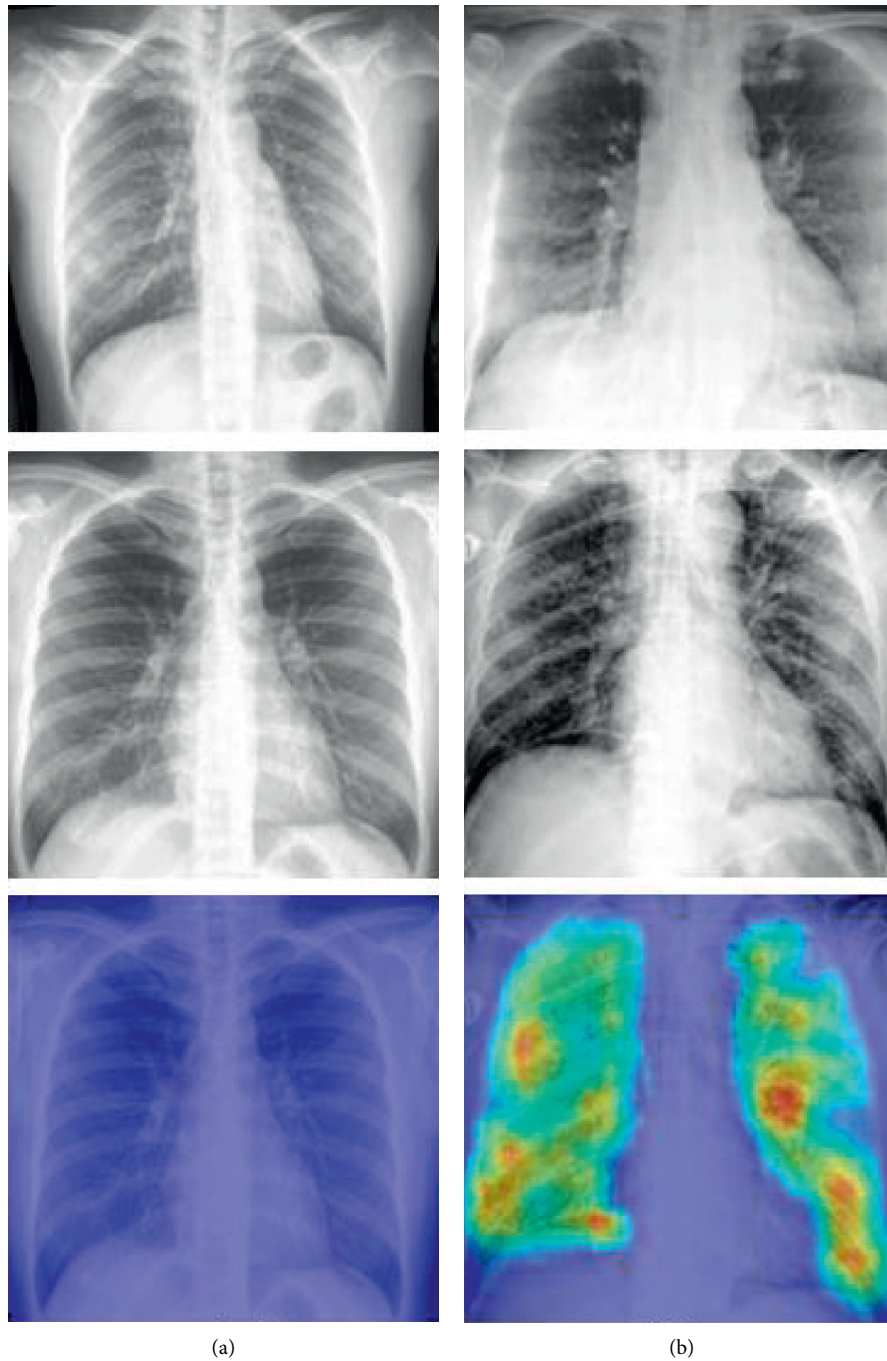


FIGURE 1: Sample preprocessed CT volume of the human lungs: (a) normal and (b) COVID-19 pneumonia patient [2].

COVID-Net model consisting of deep convolutional neural network was designed to learn characteristic abnormalities of COVID-19 diseases from large samples of patients' radiography images [17]. Emphasizing the low detection of RT-PCR during COVID-19 early stages, an early screening model was designed to map COVID-19 pneumonia compared to other viral pneumonia. This was established using location-attention classifier model from a 3-dimensional image set [18]. Also, a 4-stage infection extraction scheme was proposed by Rajinikanth et al. [19]. The scheme has an

artefact elimination filter, Otsu thresholding image enhancement, infected region segmentation, and a region-of-interest (ROI) binary extraction.

Using the Inception convolutional neural network architecture, CT images containing manually labeled pneumonia characteristic signs as regions of interest (ROIs) were trained for COVID-19 classification [20]. The model's experimental analysis indicates a comparable result with expert radiologists' analysis. To solve the problem of low-intensity contrast and high variation in CT slices, an infection

segmentation deep network consisting of a parallel partial decoder was designed [21]. The infection segmentation network generates a global map using an aggregate of the model’s high-level features. For data generation, a COVID-19 CT scan simulator was trained on an automatic segmentation model. The model achieves this by producing 2D images obtained from the decomposition of 3D images [22]. Multiple features from different views were extracted from CT slices of infected patients and then encoded using unified latent representation, which helps to learn every feature of each class via a backward neural network model [23].

**2.3. Weakly Supervised Lesion Classification.** As a step towards subduing the ongoing pandemic, rapid and accurate diagnosis of COVID-19 cases is crucial and of the essence in medical procedures [24]. Remarkably, artificial intelligence has been employed as a means of automatically analyzing medical data, including identification, classification, localization, and segmentation, mostly in a supervised context. Unfortunately, the supervised technique requires a massive amount of data and annotation, which are unavailable presently. To mitigate the problem of insufficient data, transfer learning has been adopted to maximize the availability of other antecedent datasets [25]. Using prior trained weights, COVID-19 models can benefit from increased accuracy. Also, semisupervised models have been explored to benefit the small number of labeled positive COVID-19 samples compared to the larger number of normal samples [26]. To achieve this, the generative adversarial network (GAN) architecture has been most effective. Accordingly, the unlabeled datasets are used to train the image generator, while the discriminator distinguishes the sample distribution [27].

The semisupervised GAN (SGAN) simultaneously trains a generator and a supervised and unsupervised discriminator, resulting in a supervised classifier that is capable of generalizing well to data samples which are yet unseen. As a way of dealing with limited data problem, a weakly supervised framework pretrained on a UNet model was applied for the COVID-19 infectious classification and lesion localization [28]. The method uses a combination of activation regions for connecting infectious components with high probability. A novel self-supervised model extracts features from COVID-19-positive and -negative samples, while the feature distances are learned and trained by a neural network [29]. Similarly, two 3D-ResNets and prior-attention mechanism trained as a double and discriminative classifier efficiently predict pneumonia identity probability of a CT scan [30].

### 3. Methods

**3.1. COVID-19-Net.** To achieve accurate classification of COVID-19 pneumonia lesion in CT scans, we experiment using the 2D convolutional neural network, taking advantage of its invariant property [31]. With CNN’s hierarchical and connectivity pattern of feature extraction [32], each layer of the model is able to accomplish extensive assembling

of complex patterns based on their receptive field [33]. Inspired by the VGG CNN framework, we implemented our COVID-19-Net using the VGG16 architecture [34]. Following the VGG framework, the model inputs an image size of  $224 * 224$  with a channel of 3. Each of the convolution layers has a kernel size of  $3 * 3$ , a stride of 1, and the same padding such that the input dimension is the same as the output. Every layer is then followed by a max-pooling layer of  $2 * 2$ -pixel window and a stride of 2. For proper classification efficiency, the rectified linear unit (ReLU) [35] is also introduced to induce nonlinearity.

$$z^{(l)} = W^{(l)} * x^{(l)} + b^{(l)}, \quad (1)$$

$$a^{(l)} = g(z^{(l)}), \quad (2)$$

where  $W$  and  $b$  are learnable parameters,  $x$  represents input image,  $l$  is the network layer, and  $g$  is the activation function. The configuration of the model follows a repeated dual block of convolutional layers followed by a pooling layer and then three stacks of three convolution blocks, each followed by a pooling layer. The model is concluded with three dense layers, where the last one is the output layer. The model output layer has two neurons representing the two classes of the dataset. Since it is a binary classification, the binary cross-entropy function is used for the model probabilistic loss computation.

**3.2. Transfer-Net.** Using the transfer learning technique, we construct several convolutional pretrained models as a feature extractor for COVID-19 classification. We apply five different pretrained models on the ImageNet dataset, which are the DenseNet121 [36], InceptionNetV3 [37], MobileNet [38], ResNet50 [39], and VGG16 [34] models. This allows us to save training time, as well as achieve better performance with fewer data availability. The comparison of these different models is made to investigate the effect of model size and depth on data generalization while avoiding overfitting and underfitting. As in Figure 2, the DenseNet121 model consists of 121 trainable layers, excluding the batch normalization layer. To overcome the problem of vanishing gradient in large convolutional networks, DenseNet121, in particular, combines three novel architectural schemes, which are highway networks, residual networks, and fractal networks, into the model design. This introduction helps the model to simplify pattern connectivity through maximum information flow and representational feature reuse.

The model takes advantage of CNN’s transfer learning using a pretrained model. In our transfer learning models, the last layers of the pretrained networks are removed and combined with more layers. This way, the initial network serves as a feature extractor and then further fine-tunes the new data. The pretrained models’ classifier layers are removed and replaced with three fully connected layers having 512, 256, and 2 neurons, respectively. This way, we achieve feature extraction by discovering the best representation for our dataset through the representational learning of the

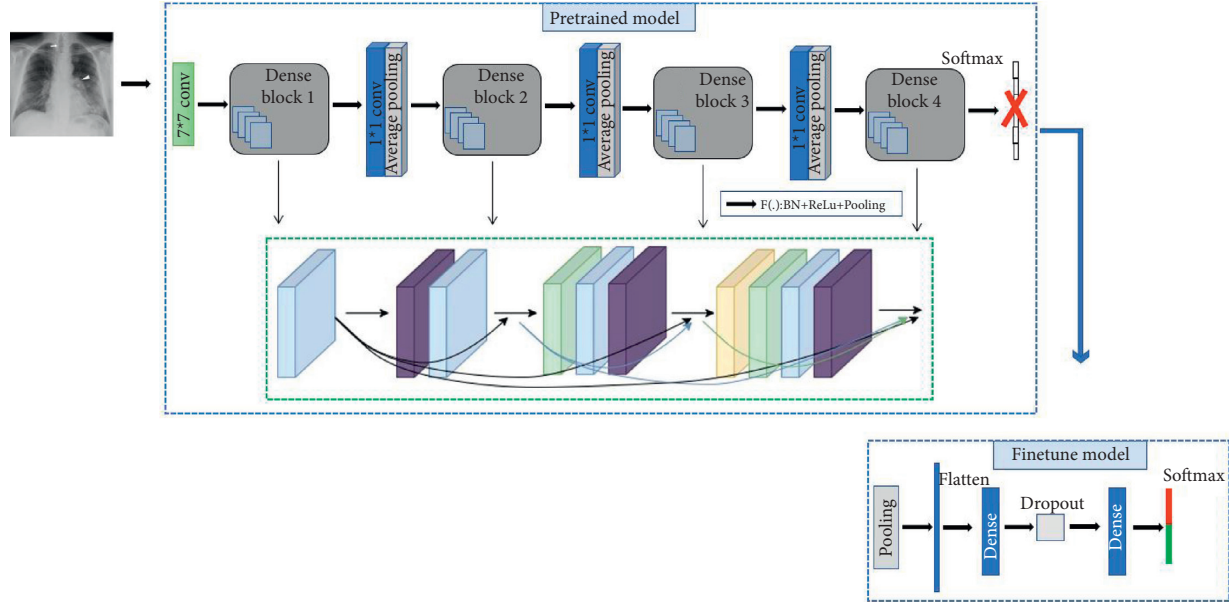


FIGURE 2: Architecture of the proposed fine-tune DenseNet121 pretrained model.

pretrained weights of the models. The newly added layers then help to fine-tune the representational learning of the modified model to our dataset, identifying the combination of features which are essential for the new training samples and, at the same time, reducing computational complexity and time.

Having source and target domains  $D_S$  and  $D_T$ , respectively, with domain  $D$  having source label  $Y_s$  and source feature space  $X_s$ , where  $x_1, \dots, x_n \in X$  and marginal probability distribution is represented as  $P(X)$ , then the conditional probability distribution of the source domain is given as  $P(Y_s|X_s)$ . Therefore, the objective of transfer learning is to learn the target conditional probability distribution  $P(Y_T|X_T)$  from the information gained from the source domain  $D_S$ .

**3.3. Generative Adversarial Network (GAN).** The generative adversarial model is framed in a supervised learning setting with a deep convolutional generator and discriminator [40]. Displayed in Figure 3, the generator learns to generate probability maps comparable to the distribution of the real data samples from random sample space. In the same manner, the discriminator learns to contradictistinguish if its input is from the generated samples or the real data distribution. Over time, the generated sample distribution becomes relative to the real data that the discriminator cannot easily differentiate the two. Mostly, the generator samples noise from a uniform distribution as input and then upsamples it through a learned transposed convolutional layer of 2 dimensions. The target of the generator is to upscale its input to the same dimension as the discriminator's input,  $224 * 224 * 3$  in our case, since the input images of the discriminator are in RGB format. Each of the transposed convolutional layers is appended with both a batch normalization and ReLU activation layer; thus, the generator loss  $L^{(G)}$  function is given as

$$L^{(G)} = \min[\log(D(x)) + \log(1 - D(G(z)))], \quad (3)$$

where  $D$  is the discriminator model,  $G$  is the generator model,  $z$  is the generator input noise distribution, and  $x$  is the real data distribution. Additionally, the discriminator that first trains on the real data distributions learns the mapping of the underlying data features to their corresponding classes, enabling it to identify the generated probability maps from the generator especially at the initial stages. So, the discriminator's output is a scalar probability of the input being a fake or real image. Generally, the discriminator loss  $L^{(D)}$  and combined adversarial loss  $L$  are given as

$$L^{(D)} = \max[\log(D(x)) + \log(1 - D(G(z)))], \quad (4)$$

$$L = \min_G \max_D [\log(D(x)) + \log(1 - D(G(z)))], \quad (5)$$

where  $D$  is the discriminator model,  $G$  is the generator model,  $z$  is the generator input noise distribution, and  $x$  is the real data distribution.

**3.4. Semisupervised COVID-Net.** The semisupervised model is majorly advantageous in cases where there are less labeled data but abundant unlabeled data. Both the labeled and unlabeled data are taken advantage of by training them with supervised and unsupervised discriminators, respectively. Extending the GAN model, we propose a semisupervised GAN (SSGAN) to take advantage of GAN's ability to learn data features. This includes the training of the model on both a labeled dataset and a larger unlabeled dataset. In most cases, the SSGAN model simultaneously trains a generator model, an unsupervised discriminator model, and a supervised discriminator to predict  $k + 1$  classes, which allows it to generalize properly to unseen data. Displayed in Figure 4, the supervised discriminator trains to predict the



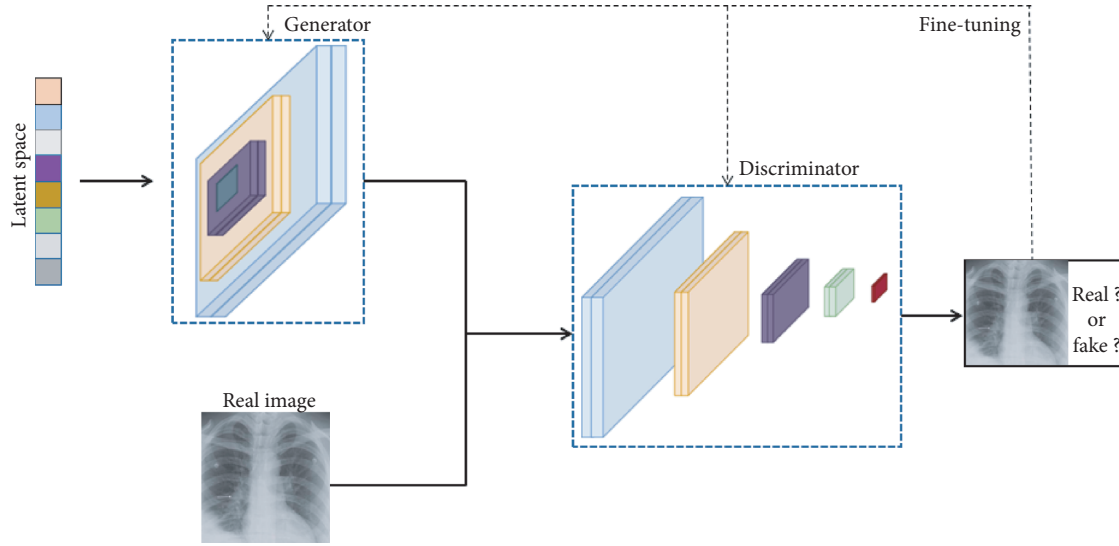


FIGURE 3: Illustration of a GAN classifier with the generator and discriminator model.

classes of the data plus an additional placeholder class using the softmax activation function, followed by optimization by applying the categorical cross-entropy loss function. Correspondingly, the unsupervised discriminator model trains to predict if an input image is fake or real with a sigmoid activation function and optimized with a binary cross-entropy loss function.

For our model, the generator has five transpose convolution layers which upsample the input noise to a  $224 * 224 * 3$  dimension for the discriminator. The discriminator is built on the ResNet18 architecture [39]. For the supervised discriminator, the last layer is modified to a binary classifier. The unsupervised discriminator is defined as the output layer prior to the supervised discriminator's classifier, and since the layer represents neuron activations, it is normalized using a customized lambda function defined by Salimans [41] to a value between 0 and 1 on  $k$  output classes. This allows an efficient implementation of reusing the output nodes of the supervised discriminator as well as in the unsupervised discriminator. The normalized sum of the unsupervised exponential outputs is computed as

$$D(x) = \frac{Z(x)}{Z(x) + 1}, \quad (6)$$

$$Z(x) = \sum_{k=1}^K \exp[l_k(x)], \quad (7)$$

where  $z$  is the generator's vector of input noise,  $x$  is the real data distribution, and  $l$  is the logit vector of  $k$  classes.

## 4. Experimental Analysis

**4.1. Datasets.** A detailed clinical and paraclinical feature investigation of COVID-19 was reported by Huang [42] who reported abnormalities in patients' CT images, having bilateral engagement. As a result, CT scans have often been used for the classification of infected patients. For this work,

we gathered 1400 images containing 700 infected COVID-19 images and 700 normal CT scans. The data were collected from multiple sources including the COVID-19 Open Research Dataset Challenge (CORD-19) by the Allen Institute for AI [43], the COVID-19\_Dataset from the University of Montreal [44], and the COVID-Chestxray-Dataset developed from various websites and publications [45].

The images are preprocessed by first transforming them into a tensor, and then, they are normalized to a pixel range of 0 to 1. The image dimension is fixed at  $224 * 224 * 3$ , representing the height, width, and color channel of the images. For training purpose, the data are categorized binarily, with one representing COVID-19 infected and zero representing normal images.

**4.2. Evaluation Metrics.** To analyze the efficiency of our models, we evaluate and compare our predictions using the accuracy, sensitivity, and specificity metrics [46]. These metrics are built on the correctness of the model's predictions compared to the true labels. The true-positive (TP) prediction represents a COVID-19 lung image correctly predicted as infected, while the false-negative (FN) prediction wrongly classifies a COVID-19 infected image as not infected or normal. Correspondingly, a true-negative (TN) prediction refers to a healthy person's CT scan rightly classified as non-COVID, while false-positive (FP) prediction represents the misprediction of a healthy scan as COVID-19 infected. With this, the ability of the model to differentiate classes correctly (accuracy), the proportion of true positives in infected class (sensitivity), and proportion of true negatives among the healthy class (specificity) is represented as [47]

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}, \quad (8)$$

$$\text{sensitivity} = \frac{TP}{TP + FN}, \quad (9)$$

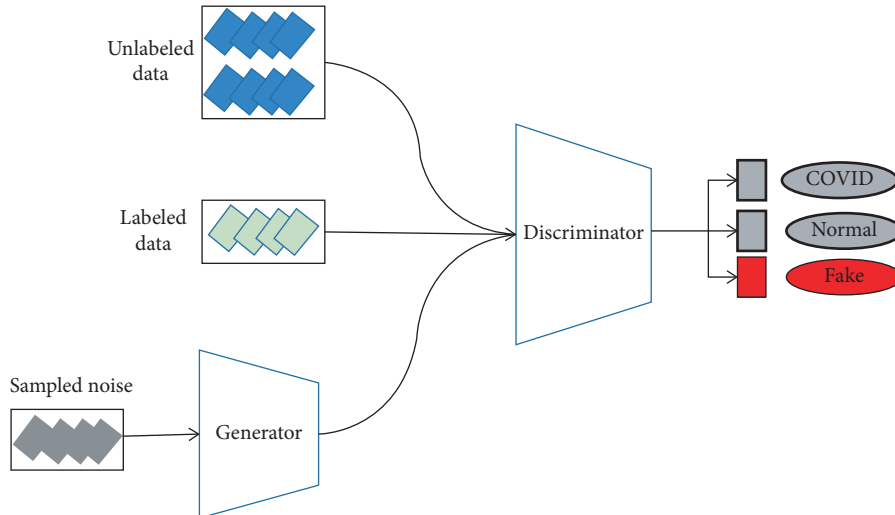


FIGURE 4: Architecture of the proposed semisupervised classifier with classes  $K + 1$  labels.

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (10)$$

**4.3. Training Details.** The COVID-19-Net and Transfer-Net were both trained using the Keras 2.3.0 API on a Windows NVIDIA 1080Ti GPU, while the semisupervised generative adversarial model was trained on a PyTorch 1.7.0 version API on an Ubuntu operating system with NVIDIA 1080Ti GPU.

The COVID-19-Net follows the VGG16 CNN architecture with 13 CNN layers and three dense layers. The CNN layers all have filter sizes of  $3 \times 3$  with the same padding, resulting in the exact dimension of the input image after convolution. The convolution layers are compartmentalized into five blocks, followed by a max-pooling layer. The pooling layers have a filter size of  $2 \times 2$  and a stride of  $2 \times 2$ , which results in a halved dimension of the input shape. The final dense layer has two neurons representing the binary classes of the data. For overcoming the vanishing gradient problem, each layer is followed by a rectified linear unit (ReLU) activation function, which results in a nonnegative return. The loss of the model is computed using the binary cross-entropy equation and minimized using the Adam backpropagation algorithm with 0.0001 learning rate. The model was trained for ten epochs with a batch size of 32. Figure 5 displays the accuracy and loss curve of the model as it converges.

The Transfer-Net models consist of five pretrained models, including InceptionNetV3, DenseNet121, MobileNet, VGG16, and ResNet50. For each of the models, the last classifier layers are removed and replaced with three dense ANN layers. First, the original outputs are flattened, followed by a dropout layer, then a dense layer of 256 neurons, and a final classifier layer of 2 outputs. The pretrained models were used as a feature extractor during training such that the weights of the appended layers were updated to fine-tune the model to the new data. The loss of the models is also computed using the binary cross-entropy

equation and minimized using the RMSProp backpropagation algorithm with  $2e - 5$  learning rate. The models are all trained for three epochs with a batch size of 32.

The semisupervised model follows the adversarial learning framework of the generative adversarial network (GAN). The semisupervised model consists of a generator and supervised and unsupervised discriminators. During training, a small portion of the dataset is set aside to be trained in a supervised manner with the supervised discriminator. These images are labeled appropriately. In our work, 100 images of each of the two classes are subsamples, totaling 200. 100 other images of each class are also set aside as a test set, totaling another 200. The remaining 1000 images are set as an unlabeled set. The generator acts as in a traditional GAN by creating a fake image sample generated from random noise vector. The discriminator, on the other hand, is classified into two: a supervised one that trains on the labeled dataset and an unsupervised one that trains on the unlabeled data. The discriminator is trained on three different data types, including the generator's generated fake images.

The generator receives a vector noise of length  $100 \times 1$  as input. The input is then upscaled using five transpose convolution layers to the  $224 \times 224 \times 3$ -dimension shape of the discriminator input. The first transpose convolution layer has a filter size of 7, a stride of 1, and zero paddings, resulting in a dimension of  $7 \times 7$ . The second and third layers both have filter, stride, and paddings of 4, 2, and 1, resulting in output shapes of  $14 \times 14$  and  $28 \times 28$ , respectively. The fourth layer results in an output shape of  $112 \times 112$  given a filter size of 4, stride size of 4, and zero paddings. The last layer then results in an output shape of  $224 \times 224 \times 3$  with the filter set as 4, the stride as 2, and padding as 1, where 3 equals the number of filters in the final layer.

The discriminator uses the ResNet18 architecture with the last layer's dense output modified to a neuron of two. The supervised discriminator is trained to predict the labeled classes of the data, that is, COVID-19 infected or healthy, whereas the unsupervised discriminator trains to determine if an input is either from a fake or real distribution. Both the

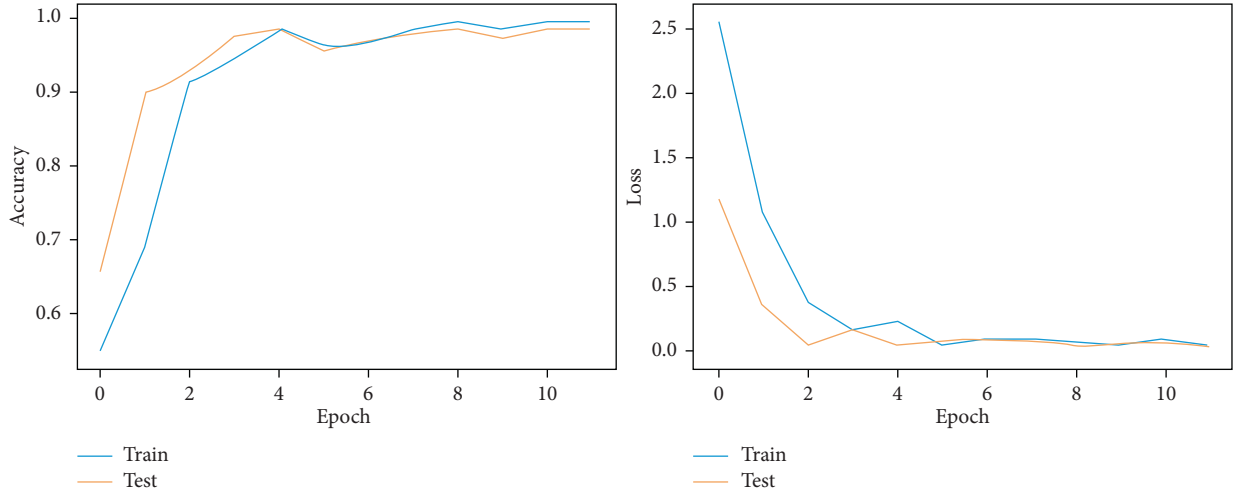


FIGURE 5: COVID-19-Net training accuracy and loss graph.

generator and discriminator losses are computed with the binary cross-entropy function and optimized using the Adam optimizer with learning rates of 0.002. This way, the generator is optimized via the unsupervised discriminator training as the model can extract and learn useful feature from the enormous unlabeled dataset, allowing the supervised model to apply the extracted features knowledge to the class label prediction. As such, the adversarial training maximizes the divergence of the correct label distribution and the predicted label.

#### 4.4. Statistical Analysis

**4.4.1. COVID-19-Net Classification Analysis.** The VGG16 designed COVID-19-Net is used to classify CT volumes into COVID-19 positive or healthy after being trained for 12 epochs. During training, of the total 700 images of each class available, 550 are set as a training set and the remaining 150 images are set as a test set. As highlighted in Table 1, our VGG16 with block max-pooling recorded an accuracy of 98.45%. The training loss decreased from 62.1614 in the first epoch to 0.0260 in the final epoch. As a result, the accuracy also increased from 55.09% in the first epoch to 98.45% in the last epoch. Evidently, the model is able to learn meaningful features from the data for the adequate classification and prediction of the input image class. In comparison to other classifier models, our model is able to predict the binary classes with more accuracy. Corresponding to the C3D algorithm, which predicted the accuracy of 96.8%, our COVID-19-Net scores a 1.64% more accuracy. Similarly, COVID-19-Net achieves a better result compared to the DeCoVNet and AD3D-MIL algorithms, which achieved 96.8% and 97.7% accuracy, respectively; thus, it outperforms them with 1.64% and 0.54%, respectively. The COVID-19-Net also achieves a 98.12% accuracy on the test set, showing that the model generalizes well to the dataset. It was, however, observed that when dropout was included in the model, the model’s classification result was adversely affected.

TABLE 1: Binary class classification results: COVID-19 and non-COVID.

Model	Train acc.	Test acc.
C3D [48]	—	96.8
DeConVNet [49]	—	96.8
AD3D-MIL [50]	—	97.9
COVID-19-Net	99.3	98.45

**4.4.2. Transfer-Net Classification Analysis.** Displayed in Figure 6, our Transfer-Net consists of the comparison of 5 pretrained CNN architectures to observe the advantages of transfer learning on COVID-19 classification. Since the models are trained on a different form of data (ImageNet), we use the pretrained models as feature extractors and then add new layers to fine-tune the pretrained weights to our data. The architectures include InceptionNetV3, DenseNet121, MobileNet, and the ResNet50 model. All of the five models are all trained for three epochs with a batch size of 32 and the same training set distribution as the COVID-19-Net.

In the training set displayed in Table 2, ResNet50 achieved the highest accuracy of 99.64%, followed by MobileNet, while InceptionNet achieved the least accuracy of 96.27%. For the test set, ResNet50 also maintained the highest score with 99%, while the DenseNet algorithm achieved a percentage of 92% as the least. Although the high values obtained by these models prove the advantage of transfer learning especially in the area of image classification and computer vision in general, the pattern of the prediction score of the models shows perhaps the likelihood that the deeper models are not as suitable for this task as the available dataset is simply not enough.

In comparison, the VGG16 architecture designed from scratch achieved better result than the pretrained model but it took longer time and required more computational resources. The transfer learning VGG16 architecture was only trained on 3 epochs, while the VGG16 architecture designed from scratch was trained on 10 epochs.

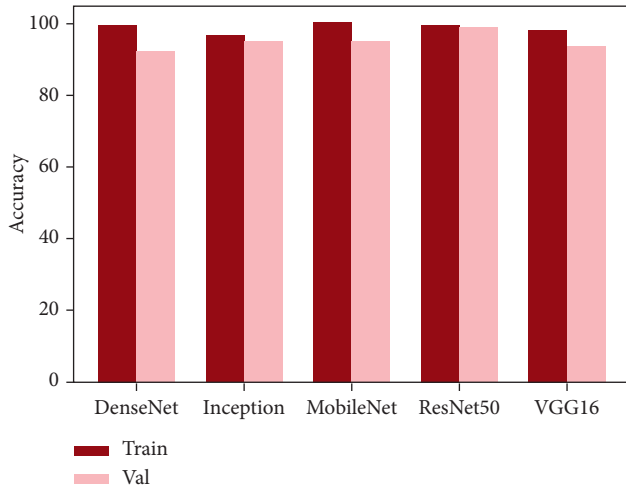


FIGURE 6: Results of the pretrained classifiers.

TABLE 2: Classification results on transfer learning models.

Model	Train acc.	Test acc.
DenseNet121	99.27	92.0
InceptionNetV3	96.27	95.33
VGG16	97.55	93.33
MobileNet	99.55	95.67
ResNet50	99.64	99.0

TABLE 3: Performance analysis of semisupervised learning.

Model	Metrics	
	Sensitivity	Specificity
EBD-Reduced [51]	0.83	0.72
Weakly-Sup-0.1 [28]	0.88	0.94
Weakly-Sup-0.2 [28]	0.93	0.83
EBD-98 [51]	0.96	0.59
Our model	0.96	0.92

**4.4.3. Semisupervised Generative Classifier Analysis.** After training, to evaluate the performance of our SSGAN, we compute the sensitivity and specificity of our model as detailed in Table 3, that is, the model’s ability to correctly determine COVID-19 infected (true positive) CT scans and the proportion of the healthy scans (true negative). Our model was able to achieve a prediction accuracy of 94%, with a total of 48 correct COVID-19 positives of 50 images, resulting in a sensitivity of 96%. Also, of the 50 healthy images, 46 are predicted correctly, yielding a specificity of 92%. In comparison to the EBD reduced data method [51], our model achieved 13% more sensitivity and 24% more specificity. Also, compared to the EBD-98 cases [51], our model achieved 2% more sensitivity and 12% more specificity. This is indeed a good result as the SSGAN model was trained on only 100 labeled datasets, taking advantage of the semisupervised learning framework to learn useful features from the more unlabeled data.

## 5. Conclusion

This study presents a computer-aided analysis of COVID-19 CT scan images. The study analyzes the applicability of deep learning and convolutional neural network for the extraction of features in images and for the classification of images to predetermined classes in a supervised manner. Similarly, we leverage the concept of transfer learning in machine learning to reduce both time and computational complexity yet accomplish an excellent result. We investigate the practicability of this by implementing five pretrained convolutional architectures which yield superior classification accuracies. Additionally, owing to the limited availability of COVID-19 data, we explore the possibility of semi-supervised learning by utilizing large unlabeled data to enrich a model’s ability to identify features and patterns in data. Using the highly effective generative adversarial learning technique, we develop a semisupervised classifier with two forms of discriminators for both labeled and unlabeled data, achieving state-of-the-art classification with few labeled training data. Our semisupervised model proves to be a very efficient diagnostic tool for COVID-19 classification with a prediction accuracy of 94%, sensitivity of 96%, and specificity of 92%.

## Data Availability

For this work, the authors gathered 1400 images: 700 infected COVID-19 images and 700 normal CT scans. The data were collected from multiple sources, including the COVID-19 Open Research Dataset Challenge (CORD-19) by the Allen Institute for AI, the COVID-19 Dataset from the University of Montreal, and the COVID-Chestxray-Dataset.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the NSFC-Guangdong Joint Fund under Grant U1401257; in part by the National Natural Science Foundation of China under Grants 61300090, 61133016, and 61272527; in part by the Science and Technology Plan Projects in Sichuan Province under Grant 2014JY0172; and by the Opening Project of Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology under Grant 2013A061401003.

## References

- [1] H. Ayaz, M. Ahmad, A. Sohaib et al., “Myoglobin-based classification of minced meat using hyperspectral imaging,” *Applied Sciences*, vol. 10, no. 19, p. 6862, 2020.
- [2] Y. Oh, S. Park, and J. C. Ye, “Deep learning COVID-19 features on CXR using limited training data sets,” *IEEE Transactions on Medical Imaging*, vol. 39, pp. 2688–2700, 2020.
- [3] F. A. Khan, “Computer-aided diagnosis for burnt skin images using deep convolutional neural network,” *Multimedia Tools and Applications*, vol. 24, 2020.

- [4] World Health Organization (WHO): [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it).
- [5] Coronavirus Disease, "(COVID-19) situation dashboard," 2020, <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>.
- [6] S. Wang, "A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis," *The European Respiratory Journal*, vol. 56, 2020.
- [7] P. Feng, "Time course of lung changes on chest CT during recovery from 2019 novel coronavirus (COVID-19) pneumonia," *Radiology*, vol. 295, no. 3, 2020.
- [8] A. Oluwasanmi, Z. Qin, and T. Lan, "Fusion of Gaussian mixture model and spatial fuzzy C-means for brain MR image segmentation," in *Proceedings of the International Conference on Computer Science and Application Engineering (CSAE 2017)*, Shanghai, China, October 2018.
- [9] A. Oluwasanmi, M. U. Aftab, E. Alabdulkreem, B. Kumeda, E. Y. Baagyere, and Z. Qin, "CaptionNet: automatic end-to-end Siamese difference captioning model with attention," *IEEE Access*, vol. 7, pp. 106773–106783, 2019.
- [10] A. Oluwasanmi, E. Frimpong, M. U. Aftab, E. Y. Baagyere, Z. Qin, and K. Ullah, "Fully convolutional CaptionNet: Siamese difference captioning attention model," *IEEE Access*, vol. 7, pp. 175929–175939, 2019.
- [11] A. Oluwasanmi, M. U. Aftab, A. Shokanbi, J. Jackson, B. Kumeda, and Z. Qin, "Attentively conditioned generative adversarial network for semantic segmentation," *IEEE Access*, vol. 8, pp. 31733–31741, 2020.
- [12] J. Goodfellow, "Generative adversarial nets," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, December 2014.
- [13] Z. Xu, L. Shi, Y. Wang et al., "Pathological findings of COVID-19 associated with acute respiratory distress syndrome," *The Lancet Respiratory Medicine*, vol. 8, no. 4, pp. 420–422, 2020.
- [14] T. Ai, "Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases," *Radiology*, vol. 296, no. 2, pp. 32–40, 2020.
- [15] E. J. Luz, P. L. Silva, R. Silva, and G. J. Moreira, "Towards an efficient deep learning model for COVID-19 patterns detection in X-ray images," 2020, <https://arxiv.org/abs/2004.05717>.
- [16] M. C. Azemin, R. Hassan, M. Tamrin, and M. A. Ali, "COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-ray images as training data: preliminary findings," *International Journal of Biomedical Imaging*, vol. 2020, Article ID 8828855, 2020.
- [17] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Scientific Reports*, vol. 10, 2020.
- [18] X. Xu, "Deep learning system to screen coronavirus disease 2019 pneumonia," 2020, <https://arxiv.org/abs/2002.09334>.
- [19] V. Rajinikanth, N. Dey, A. Raj, A. Hassanien, K. Santosh, and N. M. Raja, "Harmony-search and Otsu based system for coronavirus disease (COVID-19) detection using lung CT scan images," 2020, <https://arxiv.org/abs/2004.03431>.
- [20] S. Wang, "A deep learning algorithm using CT images to screen for corona virus disease (COVID-19)," *medRxiv*, 2020.
- [21] D.-P. Fan, T. Zhou, G.-P. Ji et al., "Inf-Net: automatic COVID-19 lung infection segmentation from CT images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2626–2637, 2020.
- [22] L. Zhou, Z. Li, J. Zhou et al., "A rapid, accurate and machine-agnostic segmentation and quantification method for CT-based COVID-19 diagnosis," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2638–2652, 2020.
- [23] H. Kang, L. Xia, F. Yan et al., "Diagnosis of coronavirus disease 2019 (COVID-19) with structured latent multi-view representation learning," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2606–2614, 2020.
- [24] X. He, "Sample-efficient deep learning for COVID-19 diagnosis based on CT scans," *medRxiv*, 2020.
- [25] Y. Pathak, "Deep transfer learning based classification model for COVID-19 disease," *Ingenierie et Recherche Biomedicale*, 2020.
- [26] N. Singh, S. B. Singh, E. H. Houssein, and Ahmad, "COVID-19: risk prediction through nature inspired algorithm," *World Journal of Engineering*, 2020.
- [27] M. Ahmad, S. Shabbir, D. Oliva, M. Mazzara, and S. Distefano, "Spatial-prior generalized fuzziness extreme learning machine autoencoder-based active learning for hyperspectral image classification," *Optik*, vol. 206, Article ID 163712, 2020.
- [28] S. Shabbir and M. Ahmad, "Hyperspectral image classification – traditional to deep models: a survey for future prospects," 2021, <https://arxiv.org/abs/2101.06116>.
- [29] Y. Li, D. Wei, J. Chen et al., "Efficient and effective training of COVID-19 classification networks with self-supervised dual-track learning to rank," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2787–2797, 2020.
- [30] J. Wang, Y. Bao, Y. Wen et al., "Prior-attention residual learning for more discriminative COVID-19 screening in CT images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2572–2583, 2020.
- [31] B. T. Anh, "A computer-vision based application for student behavior monitoring in classroom," *Applied Sciences*, vol. 9, 2019.
- [32] M. T. Qadri and M. Asif, "Automatic number plate recognition system for vehicle identification using optical character recognition," *International Conference on Education Technology and Computer*, vol. 338, 2009.
- [33] J. Zakria, J. Cai, M. Deng, U. Aftab, M. S. Khokhar, and R. Kumar, "Efficient and deep vehicle re-identification using multi-level feature extraction," *Applied Sciences*, vol. 9, 2019.
- [34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, <https://arxiv.org/abs/1409.1556>.
- [35] T. Kessler, G. Dorian, and J. Mack, "Application of a rectified linear unit (ReLU) based artificial neural network to cetane number predictions," in *Proceedings of the ASME 2017 Internal Combustion Engine Division Fall Technical Conference 2017*, Seattle, WA, USA, October 2017.
- [36] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE, Boston, MA, USA, June 2015.
- [37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, IEEE, Las Vegas, NV, USA, June 2016.
- [38] A. Howard, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, <https://arxiv.org/abs/1704.04861>.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.
- [40] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016, <https://arxiv.org/abs/1511.06434>.
  - [41] T. Salimans, “Improved techniques for training GANs,” 2016, <https://arxiv.org/abs/1606.03498>.
  - [42] C. Huang, Y. Wang, X. Li et al., “Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China,” *The Lancet*, vol. 395, no. 10223, pp. 497–506, 2020.
  - [43] L. L. Wang, “CORD-19: the COVID-19 open research dataset,” 2020, <https://arxiv.org/abs/2004.10706>.
  - [44] J. P. Cohen, “COVID-19 image data collection: prospective predictions are the future,” 2020, <https://arxiv.org/abs/2006.11988>.
  - [45] J. P. Cohen, P. Morrison, and L. Dao, “COVID-19 image data collection,” 2020, <https://arxiv.org/abs/2003.11597>.
  - [46] R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar, and R. Thomas, “Understanding and using sensitivity, specificity and predictive values,” *Indian Journal of Ophthalmology*, vol. 56, pp. 45–50, 2020.
  - [47] A. Swift, R. Heale, and A. Twycross, “What are sensitivity and specificity?” *Evidence Based Nursing*, vol. 23, no. 1, pp. 2–4, 2019.
  - [48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proceedings of the Eighth International Conference On Computer Vision*, pp. 4489–4497, Vancouver, Canada, July 2015.
  - [49] C. Zheng, “Deep learning-based detection for COVID-19 from chest CT using weak label,” 2020, <https://www.medrxiv.org/content/early/2020/03/26/2020.03.12.20027185>.
  - [50] Z. Han, B. Wei, Y. Hong et al., “Accurate screening of COVID-19 using attention-based deep 3D multiple instance learning,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2584–2594, 2020.
  - [51] A. D. Berenguer, “Explainable-by-design semi-supervised representation learning for COVID-19 diagnosis from CT imaging,” 2020, <https://arxiv.org/abs/2011.11719>.

## Research Article

# Multicriteria-Based Crowd Selection Using Ant Colony Optimization

Guan Wang,<sup>1,2</sup> Farhad Ali,<sup>3</sup> Jonghoon Yang,<sup>1</sup> Shah Nazir ,<sup>3</sup> Ting Yang ,<sup>2</sup> Abdullah Khan,<sup>3</sup> and Muhammad Imtiaz<sup>3</sup>

<sup>1</sup>The Graduate School, Sangmyung University, Seoul 03016, Republic of Korea

<sup>2</sup>School of Art and Communication, Nanhang Jincheng College, Nanjing 210056, China

<sup>3</sup>Department of Computer Science, University of Swabi, Swabi, Pakistan

Correspondence should be addressed to Shah Nazir; [snsahnzr@gmail.com](mailto:snsahnzr@gmail.com) and Ting Yang; [yangting1207@nuaa.edu.cn](mailto:yangting1207@nuaa.edu.cn)

Received 13 December 2020; Revised 28 December 2020; Accepted 13 January 2021; Published 23 January 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Guan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet-enabled technologies have provided a way for people to communicate and collaborate with each other. The collaboration and communication made crowdsourcing an efficient and effective activity. Crowdsourcing is a modern paradigm that employs cheap labors (crowd) for accomplishing different types of tasks. The task is usually posted online as an open call, and members of the crowd self-select a task to be carried out. Crowdsourcing involves initiators or crowdsourcers (an entity usually a person or an organization who initiate the crowdsourcing process and seek out the ability of crowd for a task), the crowd (online participant who is having a particular background, qualification, and experience for accomplishing task in crowdsourcing activity), crowdsourcing task (the activity in which the crowd contribute), the process (how the activity is carried out), and the crowdsourcing platform (software or market place) where requesters offer various tasks and crowd workers complete these tasks. As the crowdsourcing is carried out in the online environment, it gives rise to certain challenges. The major problem is the selection of crowd that is becoming a challenging issue with the growth in crowdsourcing popularity. Crowd selection has been significantly investigated in crowdsourcing processes. Nonetheless, it has observed that the selection is based only on a single feature of the crowd worker which was not sufficient for appropriate crowd selection. For addressing the problem of crowd selection, a novel “ant colony optimization-based crowd selection method” (ACO-CS) is presented in this paper that selects a crowd worker based on multicriteria features. By utilizing the proposed model, the efficiency and effectiveness of crowdsourcing activity will be increased.

## 1. Introduction

Nowadays, individuals voluntarily offer their own time, talent, and money to engage in activities that include helping the poor and making the planet a better place [1]. The Internet has made it easier for people to be connected and to be a part of a collaborative function, and this collaboration of people over the Internet is conceptualized into a term known as “crowdsourcing” [2]. Crowdsourcing is modern paradigms that employ cheap labors (crowd) for accomplishing different types of tasks. The task is usually posted online as an open call and members of the crowd self-select a task [3]. Crowdsourcing is an online participative activity in which

organizations make use of the heterogeneous group of people having knowledge and skills to complete the task with the announcement of an open call [4, 5]. Social networking improvements have made it possible for organizations to pool collective knowledge from people around the world, i.e., “crowds’ wisdom,” for finding best solutions to various problems [6]. The wide use of social networking services acts as a massive pool of workers. These workers vary in demography and in their population. The information present in their profiles is used for inferring abilities and preferences. Crowdsourcer registers specific crowds by utilizing built-in functionality such as in Facebook using private messages and in Twitter using “@”. There is, therefore, an evolving

trend of “managed crowdsourcing” where employees are actively selected [7, 8]. The process of crowdsourcing involves initiators or crowdsourcers (an entity usually a person or an organization who initiate the crowdsourcing process and seek out the ability of crowd for a task), the crowd (online participant who is having a particular background, qualification, and experience for accomplishing the task in crowdsourcing activity), crowdsourcing task (the activity in which the crowd contributes), the process (how the activity is carried out), and the crowdsourcing platform (software or market place where requesters offer various tasks and crowd workers complete these tasks)[9–12]. Crowdsourcing is widely used in various domains such as tagging images [13], schema matching [14], and entity resolution [15].

The crowdsourcing task is delivered to the group of people (crowd) who complete the tasks. The allocation of the task is an important characteristic in the crowdsourcing context, and it requires suitable techniques. If the allocation of tasks is carried out correctly, it provides best outcomes [11]. In crowdsourcing the task may be outsourced to a dispersed crowd (workers) who might be inexperienced on these tasks [16]. Crowd selection is becoming a challenging issue with the growth in crowdsourcing popularity. There may be an untrustworthy person, who sometimes makes errors in solving various types of tasks. Crowdsourcing will be effective if an appropriate crowd is selected [17–19]. Crowd workers differ in several dimensions, so it is mandatory that we delegate tasks based on different features [20]. A participant may be identified by basic information available in workers’ profiles, such as gender, nationality, education level, his majors, personality test score [21], his skills, and his willingness for performing tasks [22].

Diversity, largeness, and profiles difference of crowd workers across many dimensions (e.g., skills, motives, and socioeconomic backgrounds) are the foundation for the success of a given crowdsourced tasks. In addition to the involvement of dishonest workers the differences among workers may also contribute to varying quality of responses received [23–25].

Identification of high-quality workers is a significant, complex, and realistic problem as in the crowdsourcing activity various tasks are accomplished by global crowd whose size and nature is unknown. Crowd attitudes, behaviors, and skills must be identified prior to assigning tasks. High-quality work can be accomplished by workers possessing unique features such as the level of education, major, and age [21]. Crowd selection is a complex problem in which the skills and knowledge of huge crowd workers are matched with the requirements of a job [26]. Crowd selection or formation is an optimization problem that involves all types of approaches to build a crowd group to whom various tasks will be offered [27]. Various features of crowd workers were identified from existing studies with the aim to use it for crowd selection in crowdsourcing. To the best of our knowledge, the proposed “ACO-CS” is a novel approach as not a single prior study exists that addresses the problem of crowd selection in crowdsourcing using the ant colony approach. By utilizing this approach, the effectiveness and efficiency of crowdsourcing activity would greatly be

increased as it selects appropriate crowd workers according to multicriteria features for accomplishing various crowdsourced tasks.

## 2. Literature Review

Crowd selection is an important step in the activity of crowdsourcing process as the selection may make the activity effective one or may affect the activity. Various techniques were utilized for Crowd Selection in literatures which are discussed in subsequent sections.

*2.1. Crowd Selection Based on Trustworthiness.* Trust is the key element in choosing employees for a task [19, 27–29]. In different practices such as in the discovery of truth and in selection of workers, trust plays a significant role [30]. Employer can employ a trustworthy crowd worker [28]. SSC (strong social component) and C-AWSA (context-aware worker selection approach) are used to classify trustworthy workers, which is an accurate and useful algorithm for choosing trustworthy employees. It utilizes trust quality for optimization purposes. The forward search algorithm is used to measure trust of a worker [19]. Crowd trust efficiently chooses workers who are trustworthy according to an approval rate. The model effectively distinguishes dishonest workers and trustworthy workers [31].

*2.2. Crowd Selection on the Basis of Expertise Filtering.* Expertise [11, 26–28, 32–37] level of the crowd is estimated by expertise filtering. According to the expertise level, a right crowd for a task is selected [33, 34]. If they have expertise in the appropriate field, workers will be assigned tasks [27]. As an individual has task expertise, he will attempt to complete it with full attention [26]. Crowd of diverse expertise level is selected for tasks [35].

*2.3. Crowd Selection on the Basis of Individual Profiles.* Workers interested in participating on various tasks are required to build profiles that consist of various worker information, such as age, gender, skill, interest, and accomplished task history that will be stored on the platform in the worker profile database [38–40]. There are three forms of profiles. A declarative profile is created by workers themselves, the derived profile is determined from the system’s user interaction, and hybrid profile that contains declarative as well as derived information [41]. Profiling evaluates the ability of individuals to work [32]. Profiles contain three types of information: first, the voluntary information they provide about themselves; second, the information and criteria that the platforms collect about their job performance; third, the assessments of their customers. Unlike crowd discourses, these platforms allow employees to modify their individual profiles to varying degrees of independence. This is not a concession to the online workers of autonomy and self-determination, but it is because of the competitive structure of the global labor market. Platforms assist consumers in evaluating workers by providing



individually tailored profiles (in the absence of traditional recruitment documents and job interviews) [42]. Relevant workers for tasks are filtered using their profiles information. [34, 43]. Based on the level of task quality delivered, these profiles are updated [38].

*2.4. Crowd Selection on the Basis of Task-Related Qualification.* Several qualification criteria for workers may be defined by requesters. Such criteria often include test forms that are necessary to assess their qualifications [34]. Skill tests are also carried out to confirm worker qualification [44]. The workers are selected on the basis of relevant qualification they possess. For evaluating the ability of workers, qualification tests are conducted and a worker must pass the required examination to work on a project. It determines the capacity of workers in the activity. The qualification tests ensure that each employee has job related knowledge. Workers are selected on the basis of their score in qualification tests [45, 46].

*2.5. Crowd Selection on the Basis of Experience.* Workers are selected according to their experience in a task [47]. By utilizing the experience strategy, experienced workers are selected [48]. The selection of the experienced worker can make significant differences in the results of a task, i.e., it can produce high-quality results [35].

*2.6. Crowd Selection on the Basis of Skills.* Skills is a major personal attribute considered for appropriate participant selection [49]. Organization selects skilled work force for various tasks [42]. As the quality depends highly on workers' skills [43], an Initial screening of the crowd workers is carried out [50]. These screening are also referred as skill assessment which evaluates the crowd according to possessing skills and they are helpful in matching skilled labor to a task [26, 51]. Activity-based positions on the platform are allocated to workers with the essential training and skills. Crowd workers are automatically assigned to tasks if they possess the required skills [27].

*2.7. Need for the Proposed Study.* Existing studies focus only on single or few features of crowd for addressing the problem of crowd selection that is not sufficient for selection of appropriate crowd to carry out crowdsourcing activity. Our proposed model "ACO-CS" will increase the efficiency and effectiveness of crowdsourcing activity as it selects crowd based on multicriteria features; therefore, appropriate multifeatured crowd will be selected to accomplish various tasks in crowdsourcing activity.

### 3. Methodology

The selection of quality crowd workers is a challenging issue. The workers are identified by their unique features [21]. These features are identified in the literature analysis and the selection of crowd will highly depend on these identified features. The feature set consists of large redundant data

which will affect the appropriate selection of crowd workers. To remove redundancy and complexity, these are filtered out.

*3.1. Feature Selection.* A feature is an individual assessable asset of the process being investigated [52]. Feature selection (FS) is used in machine learning processes particularly in solving complex feature problems. Feeding a wide range of features into a model of recognition not only raises the strain of computation but also creates the issue widely known as the dimensionality curse. With feature selection, a large and complex dataset is reduced, as appropriate features are sorted out. Feature selection is broad and extends through many areas, including categorization of text, data mining, and identification of patterns and processing of signals [53]. Using the feature selection approach, a feature subset from the original set is selected and the accuracy of the original set is preserved. The efficacy and scalability will be increased by eliminating unnecessary and redundant features [54]. In dealing with larger feature datasets, the selection of features is obligatory. FS is a requirement in real-world problems due to the proliferation of noisy, meaningless, or deceptive characteristics.

Despite the availability of data with hundreds of variables leading to high-dimensional data, many feature selection strategies were used by researchers for selecting best features. The various feature selection techniques provide us with a way to reduce computing time, boost prediction efficiency, and better understanding of machine learning process or pattern recognition. Crowd consists of diverse and multidisciplinary people and this crowd possesses unique features. Crowd features are collected from existing research studies. These are positive and negative features. The identification of these features was necessary for distinguishing appropriate and inappropriate crowd workers. These features are then filtered out to remove the ambiguity and complexity. A set of crowd containing multifeatures (Table 1) was obtained as a result of combining the features that were captured during literature analysis. According to multicriteria features, our proposed method will select or reject various types of crowd. The various features of the crowd are represented in Table 2 (negative features), Table 3 (positive Features), and Table 1 overall (negative and positive) features.

*3.2. Background.* Ant colony optimization is a swarm intelligence technique which was introduced by M. Dorigo and his colleagues in 1990 [97]. They were inspired by the foraging behavior of certain species of ant. For leaving marks on encouraging direction that should be followed by other ant's colony (members), these ants deposit various pheromones (chemical) on the ground. To solve optimization problems [98], ant colony optimization approach is utilized. Two variables are used by ants in solving problems such as heuristic knowledge and the value of pheromones. Quality outcomes can be generated as a result of the mutual communication among artificial ants. This is obtained by pheromone trail values through indirect contact (sensed the

TABLE 1: Representation of the crowd worker set.

No.	Crowd	Features
1	C1	Professionals, collaborative, coordinative, fast, and adaptability
2	C2	Nonprofessional, noncollaborative, lazy, noncoordinative, untrained
3	C3	Nonexpertise, dependent, un-reliable, uneager, and malicious
4	C4	Expertise, autonomous, reliable, eager beavers, and problem solving
5	C5	Experience, loyal, cooperative, flexible, and right
6	C6	Unexperienced, disloyal, heterogeneous, nonflexible, and cheater
7	C7	Unskilled, noncreative, nonenergetic, in appropriate, and incompetent
8	C8	Skill, creative, energetic, appropriate, and competent
9	C9	Trustworthy, qualified, knowledge, smart, and identifiable
10	C10	Untrustworthy, novice, selfish, anonymous, and erroneous

TABLE 2: Representation of negative features crowd features.

S. no.	Negative features	Citations
1	Nonprofessionals	[36, 55, 56]
2	Nonexperts	[34, 36]
3	Untrustworthy	[19]
4	Unskilled workers	[29]
5	Anonymous	[27, 42, 46, 55, 57–59]
6	Untrained participants	[27, 60]
7	Malicious workers	[55, 60, 61]
8	Selfish workers	[62]
9	Novice	[43, 63–65]
11	Unexperienced	[57]
12	Fraudulent	[66]
13	Incompetence	[2]
14	Heterogeneous	[4, 38, 56, 67]

TABLE 3: Representation of positive crowd features.

S. no.	Positive features	Citation
1	Professionals	[32, 68]
2	Expertise	[22, 37, 69, 70]
3	Trustworthy	[19, 41, 71–73]
4	Skill	[1, 5, 6, 17, 18, 20, 22, 29, 51, 56, 63, 74–83]
5	Identifiable	[84]
6	Autonomous	[84]
7	Flexible	[69, 84]
8	Creative	[34, 79, 81]
9	Knowledge	[4–6, 11, 17, 32, 34, 36, 47, 57, 59, 60, 68, 74, 77, 78, 83, 85–88]
10	Appropriate	[27]
11	Eager beavers	[60]
12	Competent	[72, 89, 90]
13	Adaptable	[89]
14	Coordinative	[89]
15	Collaborative	[2, 83, 90]
16	Qualified/educated	[11, 41, 66]
17	Problem solving	[2, 34, 42, 67, 68, 77, 80, 81, 91]
18	Energetic	[1, 6]
19	Experience	[6, 17, 35, 43, 59, 83, 92–94]
20	Reliable/efficient (workers)	[5, 50, 58, 61, 95]
21	Smart	[50, 96]
22	Right	[18, 33]
23	Fast	[5]
24	Cooperative	[63]
25	Loyal	[88]

pheromone) of various ants. Ants do not change themselves but adjust the way other ants represent and view the problem adaptively [99]. ACO was mostly concerned with addressing the problems of ordering. One of ACO's recent trends is to address various existing problems in the industrial sector [98].

**3.3. Proposed Method.** In crowdsourcing operations, the proposed ACO model can be extended to the problem of crowd selection. There are different steps involved in implementing the ACO algorithm in the selection of crowds. Figure 1 shows the overall operation of the proposed system “ant colony optimization-based crowd selection method (ACO-CS).” The proposed method undergoes the process of crowd selection. The selection of crowd starts with the generation of ants that will traverse on various paths (edges) and will select crowds on the basis of the pheromone value present on different edges. If the ant traversal satisfy a stopping criteria, the ants stops (traversal terminates) and the best subset of crowds is generated that will latterly be used for assigning different tasks. If the traversal does not correspond to stopping criteria, then the pheromone value is updated and once again the process is initiated.

**3.3.1. Ant Colony Optimization-Based Crowd Selection Method.** A set of 10 crowd workers are present (Table 1). The crowd selection technique is to minimize crowd subset which will be less than the original set of crowd; a higher accuracy in the depiction of the original crowd set will be retained. The partial selection may be in any order between the solutions. Simultaneously, the potential crowd to be chosen is not generally affected by the prior crowd attached to the node. It is not, however, necessary that the solutions to the selection of crowds should be of equal scale. The following steps are considered in mapping the crowd selection problem to the ACO algorithm:

- (i) The graphical representation
- (ii) Pheromone and heuristic desirability
- (iii) Updating the pheromone value
- (iv) Outcomes formation

**(1) The Graphical Representation.** The problem of crowd selection can also be defined in terms of ant colony optimization problem. ACO generally represents the problem in a graphical form as represented in Figure 2. The nodes signify various types' crowds, and the edges reflect the corresponding crowd choice. The nodes are linked with each other to permit the selection of any crowd. An optimal subset of the crowd is selected when ant traverse over the graph, i.e., visit various nodes. The ant traversal must satisfy stopping criteria (select optimal multifeatures appropriate crowds). In Figure 2, initially, the ants A, B, C, D, E, F, G, H, I, and J from their nest are allowed to start traversing to different nodes such as C1 or C2 and subsequently to C3, C4, C5, C6, C7, C8, C9, and C10. These ants in the traversal process leave pheromones (represented in Figure 3) which is

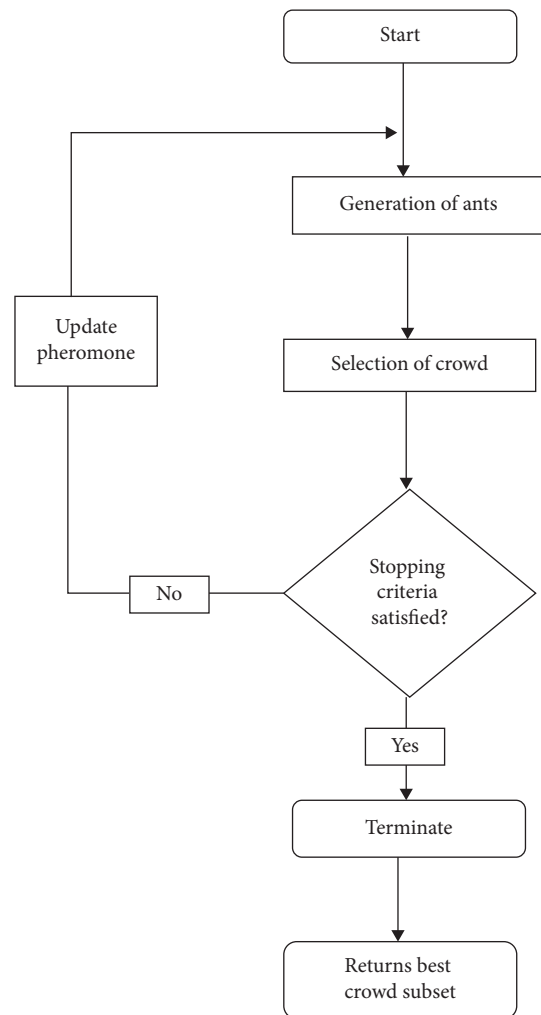


FIGURE 1: The proposed ant colony optimization-based crowd selection method (ACO-CS).

a chemical substance on different edges. The ants move according to the probability of the pheromones level on various edges, i.e., if the levels of pheromone are high, the ant will select only high pheromones values edges (bold line) and select only those specific nodes (Figure 4). The ant A from the nest will select node C1 and then using transition rule, the ant selects a crowd C4. Next, it chooses C5, C8, and C9. On reaching C9, the ant traversal satisfies stopping criteria and its stops its traversal and provides a partial solution of the original crowd set “C” that consists of crowd workers C1, C4, C5, C8, and C9. A high accuracy is achieved as an outcome of the crowd workers subset. The crowd subset is then used as a nominee for different tasks.

**(2) Pheromone and Heuristic Desirability.** Initially teams of crowd are evaluated to identify best crowd workers. A simple multistart local search method decides the initial selection of crowd alternatives. In general,  $(\eta_i)$  heuristic function is used in combination with the pheromone value in the ACO algorithm to make a right transition. Quality crowd workers are selected by calculating the pheromone and heuristic

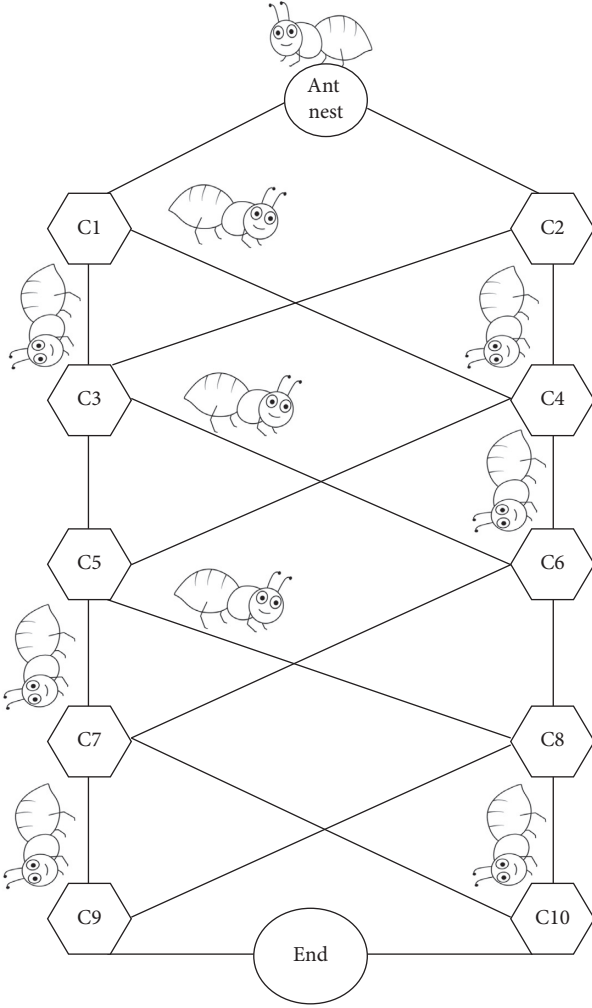


FIGURE 2: Ants traversals.

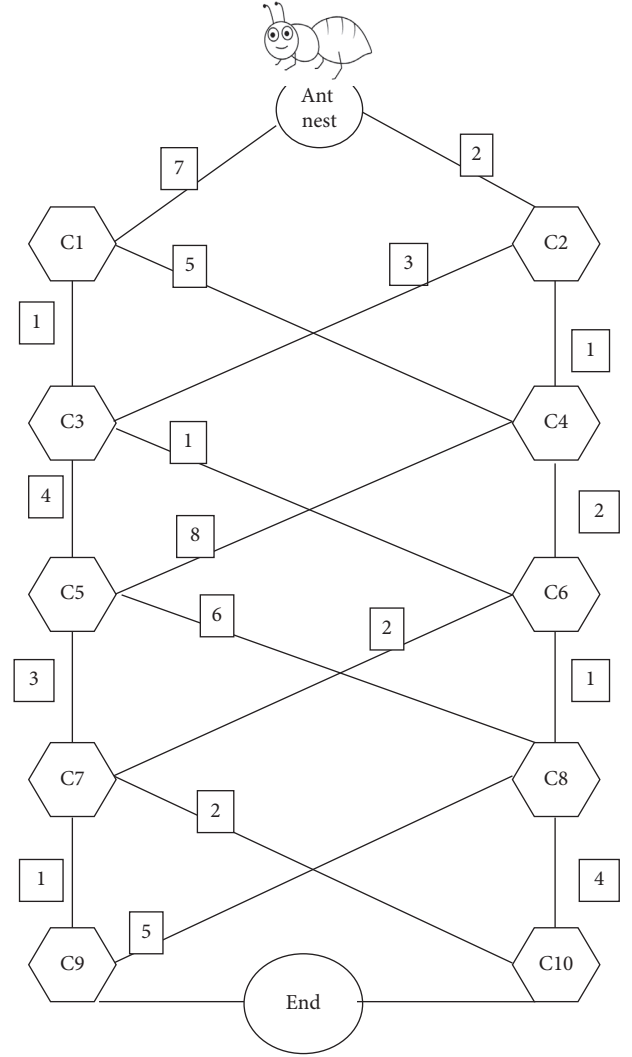


FIGURE 3: Pheromones excreted by ants on edges.

value. If a crowd worker is to be selected, then this value is assigned with a greater value “1.” On the other hand, if a crowd is not selected, then the pheromone value is kept smaller “0.” An ant in C1 determines whether C3 is to be selected or not, and the decision is taken in accordance with highest pheromones probability on edges and the probability can be calculated using the following formula:

$$P(\text{edge}) = \frac{P(\text{pheromones}(X_i))\eta_i}{\sum(P(\text{pheromone}(X_i))\eta_i)}. \quad (1)$$

The probability of each ant selecting a node is determined using equation (1) where  $P$  represents the probability,  $X_i$  represents edges, and  $\eta_i$  represents heuristic desirability. If an edge is to be selected, then  $\eta_i$  value is kept higher otherwise lower. The traversal and selection of a node (i.e., crowd) depend on the pheromone value. The ant will traverse on edge having a greater pheromone value.

(3) *Updating the Pheromone Value.* If the stopping condition is not satisfied by ants traversal, then the pheromone is modified, a new collection of ants is generated, and once

again the process iterates. The pheromone is modified according to the following formula on each edge:

$$\tau_1(\tau + 1) = (1 - \rho) \cdot \tau_1(\tau) + \rho \cdot \Delta\tau_1(\tau), \quad (2)$$

where  $\rho$  value may be considered 0 and 1, and it is the coefficient pheromone trail decay. If the stopping criterion is not satisfied (i.e., best crowd subset is not produced), then the ants modify pheromone. More pheromones are laid on the best solution nodes by the best ants, and as a result, optimal solutions are revealed.

(4) *Outcomes Formation.* The entire process of ACO-CS initiates the formation of randomly positioned numbers of ants. Then, these ants are positioned on a graph and the numbers of ants are equally set to the number of crowds (i.e., both are 10). The process of path building from a specific crowd starts with every ant. They cross the nodes from those starting positions in a probabilistic fashion until the stopping condition is fulfilled (optimal multifeatures appropriate crowds are selected). For an ideal subset, the resulting crowd

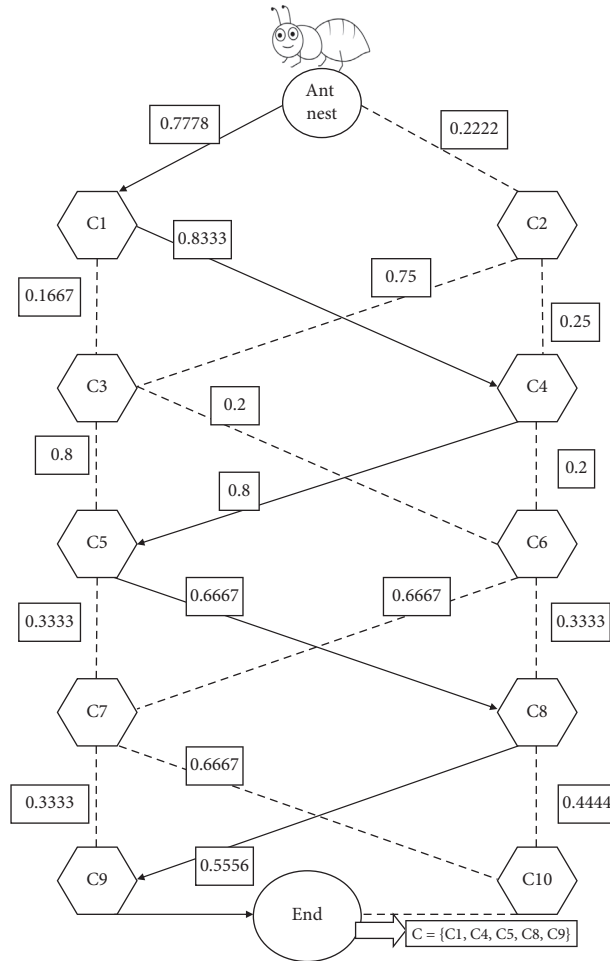


FIGURE 4: Pheromones probability wise path traversal.

subset is obtained and investigated. When best crowd subset has been found, then the experiment ends and it is noted (Figure 1). The pheromone update takes place when the conditions are not met, new ant’s colony is created, and the process repeats again.

**4. Results and Discussion**

The process of crowdsourcing involves Internet crowds [5, 90]. Crowds are recruited from Internet-enabled societies. Crowd workers possess some attributes, such as qualification, age, gender, language, worker place, skills, past service, and experience. Based on these characteristics, workers are selected [6, 11, 39, 100]. An employer should carefully pick workers to produce quality results [56]. The success of the organization depends on the allocation of tasks to members of the crowd, which requires adequate control systems, such as screening workers [11, 41]. When choosing the crowd, care must be taken as it is noted that the quality increases with a varied choice of crowds [50]. In our process of crowd selection, a crowd set (10 number of crowd “C1, C2, C3, C4, C5, C6, C7, C8, C9, and C10”) possessing multiple features (positive and negative features) are positioned in a graphical structure. The crowds are represented

as different nodes connected with each other with the help of edges; an equal number of ants (10 numbers) are generated for the purpose to traverse on various edges and to select various crowd nodes, i.e., partial solution of the crowd set. The partial solutions (crowd subset C) will be evaluated based on the pheromone probability on paths, if it meets the stopping criteria (i.e., it selects the best crowd subset C), then the ants will stop its traversal and will produce the best crowd subset having multiple features. If the ants do not satisfy the stopping criteria, the pheromones are updated, and for the second iteration, the process is initiated. The selection and rejection of nodes (crowd) depend on the probability of pheromone on each edge; if the value is higher, the nodes will be selected and best crowd subset will be produced in ant’s traversal and if the value of pheromone is less, the edge and in turn, the crowd node is rejected. Table 4 represents the ants and their selected path. The probability of edges is calculated to find the best traversing path and in turn selects only appropriate crowd that will be assigned with different tasks in various activities. In our crowd selection method, the crowd subset C1, C4, C5, C8, and C9 by ant A, traversal is selected as the evaluation of the probability of pheromone values on edges linking these nodes (crowds) were higher than the edges linking other nodes (Crowd).

TABLE 4: Representation of selected path by ants with pheromone probability.

Crowd	Selected crowd	Ants	Pheromone (value) on selected edges	Paths selection probability node/ $\Sigma$ (value) connected edges	Path selection probability of an ant (from starting node to last node) $\sum P$ (pheromone on edge)	Probability pheromone $P$ (pheromone on selected edges)/ $\sum P$ (pheromones on all edges)
	C1, C4, C5, C8, C9	A	7, 5, 8, 6, 5	0.7778, 0.8333, 0.8, 0.6667, 0.5556	3.6334	0.1464
	C2, C3, C5, C8, C9	B	2, 3, 4, 6, 5	0.2222, 0.75, 0.8, 0.6667, 0.5556	2.9945	0.1207
	C1, C3, C6, C7, C10	C	7, 1, 1, 2, 2	0.7778, 0.1667, 0.2, 0.6667, 0.6667	2.4779	0.0998
	C2, C3, C6, C8, C9	D	2, 3, 1, 1, 5	0.2222, 0.75, 0.2, 0.3333, 0.5556	2.0611	0.0831
C1, C2 C3, C4 C5, C6 C7, C8 C9, C10	C2, C3, C6, C7, C9	E	2, 3, 1, 2, 1	0.2222, 0.75, 0.2, 0.6667, 0.3333	2.1722	0.0875
	C1, C4, C6, C8, C10	F	7, 5, 2, 1, 4	0.7778, 0.8333, 0.2, 0.3333, 0.4444	2.5888	0.1043
	C2, C4, C6, C8, C10	G	2, 1, 2, 1, 4	0.2222, 0.25, 0.2, 0.3333, 0.4444	1.4499	0.0584
	C1, C3, C5, C7, C9	H	7, 1, 4, 3, 1	0.7778, 0.1667, 0.8, 0.3333, 0.3333	2.4111	0.0972
	C1, C4, C5, C7, C9	I	7, 5, 8, 3, 1	0.7778, 0.8333, 0.8, 0.3333, 0.3333	3.0777	0.1240
	C2, C3, C6, C8, C10	J	2, 3, 1, 1, 4	0.2222, 0.75, 0.2, 0.3333, 0.4444	1.9499	0.0786
					$\Sigma = 24.8165$	

## 5. Conclusion

In the proposed study, we proposed ant colony optimization-based crowd selection for the problem of crowd selection in crowdsourcing. The key contribution of our research is to select the crowd based on the multicriteria features as the selection of crowd in previous approaches was based on a single or few features that do not guarantee the appropriate selection of crowd and thus affects the crowdsourcing activity. With our presented approach (ACO-CS) best crowds are selected according to multicriteria features that will, in turn, make crowdsourcing activity efficient and effective. The model "ACO-CS" in this paper is presented theoretically with less number of crowds. In future, we will implement it practically as this model is effective (i.e., selects crowd on the basis of multicriteria features) than existing techniques used for crowd selection; therefore, it will play an important role in the crowd selection phase in crowdsourcing activity.

## Data Availability

No data were used to support the findings of this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] N. Mazlan, S. S. Syed Ahmad, and M. Kamalrudin, "Volunteer selection based on crowdsourcing approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 743–753, 2018.
- [2] J. Lee and D. Seo, "Crowdsourcing not all sourced by the crowd: an observation on the behavior of Wikipedia participants," *Technovation*, vol. 55–56, pp. 14–21, 2016.
- [3] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, pp. 1–4, 2006.
- [4] A. L. Zanatta, I. Steinmacher, L. S. Machado, C. R. B. De Souza, and R. Prikladnicki, "Barriers faced by newcomers to software-crowdsourcing projects," *IEEE Software*, vol. 34, no. 2, pp. 37–43, 2017.
- [5] A. Ellero, P. Ferretti, and G. Furlanetto, "Realtime crowdsourcing with payment of idle workers in the retainer model," *Procedia Economics and Finance*, vol. 32, pp. 20–26, 2015.
- [6] I. Dissanayake, J. Zhang, and B. Gu, "Virtual team performance in crowdsourcing contest: a social network perspective," in *Proceedings of the 2015 48th Hawaii International Conference on System Sciences*, pp. 4894–4897, Kauai, HI, USA, January 2015.
- [7] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *Proceedings of the 16th International Conference on Extending Database Technology*, pp. 637–648, Genoa, Italy, March 2013.
- [8] C. C. Cao, Y. Tong, L. Chen, and H. Jagadish, "Wisemarket: a new paradigm for managing wisdom of online social users," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 455–463, Chicago, IL, USA, August 2013.
- [9] E. Estellés-Arolas and F. González-Ladrón-De-Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information Science*, vol. 38, no. 2, pp. 189–200, 2012.
- [10] R. Ali, M. Hosseini, K. T. Phalp, and J. Taylor, "The four pillars of crowdsourcing: a reference model," in *Proceedings of the 2014 IEEE Eighth International Conference on Research*

- Challenges in Information Science (RCIS)*, Marrakech, Morocco, May 2014.
- [11] L. Machado, R. Prikładnicki, F. Meneguzzi, C. R. De Souza, and E. Carmel, "Task allocation for crowdsourcing using AI planning," in *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering*, pp. 36–40, Austin, TX, USA, May 2016.
  - [12] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "CrowdDB: answering queries with crowdsourcing," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, pp. 61–72, Athens, Greece, June 2011.
  - [13] X. S. Yang, R. Cheng, L. Mo, B. Kao, and D. W. Cheung, "On incentive-based tagging," in *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 685–696, Brisbane, Australia, April 2013.
  - [14] N. Q. V. Hung, N. T. Tam, Z. Miklós, and K. Aberer, "On leveraging crowdsourcing techniques for schema matching networks," in *Proceedings of the International Conference on Database Systems for Advanced Applications*, pp. 139–154, Wuhan, China, April 2013.
  - [15] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: crowdsourcing entity resolution," 2012, <https://arxiv.org/abs/1208.1927>.
  - [16] H. Li, B. Yu, and D. Zhou, "Error rate analysis of labeling by crowdsourcing," in *Proceedings of the ICML Workshop: Machine Learning Meets Crowdsourcing*, Atlanta, GA, USA, June 2013.
  - [17] M. Garcia Martinez and B. Walton, "The wisdom of crowds: the potential of online communities as a tool for data analysis," *Technovation*, vol. 34, no. 4, pp. 203–214, 2014.
  - [18] A. Dubey, G. Viridi, M. S. Kuriakose, and V. Arora, "Towards adopting alternative workforce for software engineering," in *Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pp. 16–23, Irvine, CA, USA, August 2016.
  - [19] Y. Zhao, G. Liu, K. Zheng, A. Liu, Z. Li, and X. Zhou, "A context-aware approach for trustworthy worker selection in social crowd," *World Wide Web*, vol. 20, pp. 1211–1235, 2017.
  - [20] A. Smirnov, A. Ponomarev, and N. Shilov, "Hybrid crowd-based decision support in business processes: the approach and reference model," *Procedia Technology*, vol. 16, pp. 376–384, 2014.
  - [21] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 165–176, Seoul, Korea, April 2014.
  - [22] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. M. Aiello, G. D. F. Morales, and D. Quercia, "Incentivizing social media users for mobile crowdsourcing," *International Journal of Human-Computer Studies*, vol. 102, pp. 4–13, 2017.
  - [23] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor, "Are your participants gaming the system? screening mechanical turk workers," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2399–2402, Atlanta, GA, USA, April 2010.
  - [24] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators? crowdsourcing abuse detection in user-generated content," in *Proceedings of the 12th ACM Conference on Electronic Commerce*, pp. 167–176, San Jose, CA, USA, June 2011.
  - [25] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *The Journal of Machine Learning Research*, vol. 13, pp. 491–518, 2012.
  - [26] J. T. Bush and R. M. Balven, "Catering to the crowd: an HRM perspective on crowd worker engagement," *Human Resource Management Review*, vol. 31, no. 1, Article ID 100670, 2018.
  - [27] A. Sari, A. Tosun, and G. I. Alptekin, "A systematic literature review on crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 153, pp. 200–219, 2019.
  - [28] H. Amintoosi, S. S. Kanhere, and M. Allahbakhsh, "Trust-based privacy-aware participant selection in social participatory sensing," *Journal of Information Security and Applications*, vol. 20, pp. 11–25, 2015.
  - [29] O. Folorunso and O. A. Mustapha, "A fuzzy expert system to trust-based access control in crowdsourcing environments," *Applied Computing and Informatics*, vol. 11, no. 2, pp. 116–129, 2015.
  - [30] W. Feng, Z. Yan, L. T. Yang, and Q. Zheng, "Anonymous authentication on trust in blockchain-based mobile crowdsourcing," *IEEE Internet of Things Journal*, 2020.
  - [31] B. Ye, Y. Wang, and L. Liu, "Crowd trust: a context-aware trust model for worker selection in crowdsourcing environments," in *Proceedings of the 2015 IEEE International Conference on Web Services*, pp. 121–128, New York, NY, USA, July 2015.
  - [32] K. L. Jeffcoat, T. J. Eveleigh, and B. Tanju, "A conceptual framework for increasing innovation through improved selection of specialized professionals," *Engineering Management Journal*, vol. 31, no. 1, pp. 22–34, 2019.
  - [33] Z. Zhao, D. Yan, W. Ng, and S. Gao, "A transfer learning based framework of crowd-selection on twitter," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, USA, August 2013.
  - [34] N. Luz, N. Silva, and P. Novais, "A survey of task-oriented crowdsourcing," *Artificial Intelligence Review*, vol. 44, pp. 187–213, 2015.
  - [35] Q. Cui, J. Wang, G. Yang, M. Xie, Q. Wang, and M. Li, "Who should be selected to perform a task in crowdsourced testing?" in *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pp. 75–84, Turin, Italy, July 2017.
  - [36] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.
  - [37] A. Moayedikia, W. Yeoh, K.-L. Ong, and Y. L. Boo, "Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach," *Decision Support Systems*, vol. 122, Article ID 113065, 2019.
  - [38] B. Satzger, H. Psailer, D. Schall, and S. Dustdar, "Auction-based crowdsourcing supporting skill management," *Information Systems*, vol. 38, no. 4, pp. 547–560, 2013.
  - [39] X. Peng, J. Gu, T. H. Tan et al., "CrowdService: serving the individuals through mobile crowdsourcing and service composition," in *Proceedings of the 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 214–219, Singapore, Singapore, September 2016.
  - [40] K. Abhinav, G. K. Bhatia, A. Dubey, S. Jain, and N. Bhardwaj, "TasRec: a framework for task recommendation in crowdsourcing," in *Proceedings of the 15th International Conference on Global Software Engineering*, pp. 86–95, Seoul, South Korea, May 2020.
  - [41] T. Awwad, N. Bennani, K. Ziegler, V. Sonigo, L. Brunie, and H. Kosch, "Efficient worker selection through history-based learning in crowdsourcing," in *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pp. 923–928, Turin, Italy, July 2017.

- [42] H. J. Pongratz, "Of crowds and talents: discursive constructions of global online labour," *New Technology, Work and Employment*, vol. 33, no. 1, pp. 58–73, 2018.
- [43] D. Sánchez-Charles, J. Nin, M. Solé, and V. Muntés-Mulero, "Worker ranking determination in crowdsourcing platforms using aggregation functions," in *Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1801–1808, Beijing, China, July 2014.
- [44] P. Adamska, M. Jużwin, and A. Wierzbicki, "Picking peaches or squeezing lemons: selecting crowdsourcing workers for reducing cost of redundancy," in *Proceedings of the International Conference on Computational Science*, pp. 510–523, Amsterdam, The Netherlands, June 2020.
- [45] E. R. Q. Weidema, C. López, S. Nayebaziz, F. Spanghero, and A. V. D. Hoek, "Toward microtask crowdsourcing software design work," in *Proceedings of the 2016 IEEE/ACM 3rd International Workshop on CrowdSourcing in Software Engineering (CSI-SE)*, pp. 41–44, Austin, TX, USA, May 2016.
- [46] F. R. Assis Neto and C. A. S. Santos, "Understanding crowdsourcing projects: a systematic review of tendencies, workflow, and quality management," *Information Processing & Management*, vol. 54, no. 4, pp. 490–506, 2018.
- [47] M. Alsayyari and S. Alyahya, "Supporting coordination in crowdsourced software testing services," in *Proceedings of the 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 69–75, Bamberg, Germany, March 2018.
- [48] S. Alyahya, "Crowdsourced software testing: a systematic literature review," *Information and Software Technology*, vol. 127, Article ID 106363, 2020.
- [49] X. Zhang, B. Gong, Y. Cao, Y. Ding, and J. Su, "Investigating participants' attributes for participant estimation in knowledge-intensive crowdsourcing: a fuzzy DEMATEL based approach," *Electronic Commerce Research*, vol. 145, pp. 1–32, 2020.
- [50] A. Dwarakanath, U. Chintala, N. C. Shrikanth et al., "CrowdBuild: a methodology for enterprise software development using crowdsourcing," in *Proceedings of the Second International Workshop on CrowdSourcing in Software Engineering*, Florence, Italy, May 2015.
- [51] M. Christoforaki and P. G. Ipeirotis, "A system for scalable and reliable technical-skill testing in online labor markets," *Computer Networks*, vol. 90, pp. 110–120, 2015.
- [52] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
- [53] S. Kashef and H. Nezamabadi-pour, "A new feature selection algorithm based on binary ant colony optimization," in *Proceedings of the 5th Conference on Information and Knowledge Technology*, pp. 50–54, Shiraz, Iran, May 2013.
- [54] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 491–502, 2005.
- [55] Z. Peng, X. Gui, J. An, R. Gui, and Y. Ji, "TDSRC: a task-distributing system of crowdsourcing based on social relation cognition," *Mobile Information Systems*, vol. 2019, Article ID 7413460, 2019.
- [56] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, "Efficient crowdsourcing of unknown experts using bounded multi-armed bandits," *Artificial Intelligence*, vol. 214, pp. 89–111, 2014.
- [57] M. Hosseini, A. Shahri, K. Phalp, J. Taylor, and R. Ali, "Crowdsourcing: a taxonomy and systematic mapping study," *Computer Science Review*, vol. 17, pp. 43–69, 2015.
- [58] A. Moayedikia, W. Yeoh, K.-L. Ong, and Y.-L. Boo, "Framework and literature analysis for crowdsourcing's answer aggregation," *Journal of Computer Information Systems*, vol. 60, no. 1, pp. 49–60, 2020.
- [59] K.-J. Stol, B. Caglayan, and B. Fitzgerald, "Competition-based crowdsourcing software development: a multi-method study from a customer perspective," *IEEE Transactions on Software Engineering*, vol. 45, no. 3, pp. 237–260, 2019.
- [60] T. D. LaToza and A. V. D. Hoek, "A vision of crowd development," in *Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, pp. 563–566, Florence, Italy, May 2015.
- [61] C. Qiu, A. Squicciarini, D. R. Khare, B. Carminati, and J. Caverlee, "CrowdEval: a cost-efficient strategy to evaluate crowdsourced worker's reliability," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, Stockholm, Sweden, July 2018.
- [62] P. Shi, M. Zhao, W. Wang et al., "Best of both worlds: mitigating imbalance of crowd worker strategic choices without a budget," *Knowledge-Based Systems*, vol. 163, pp. 1020–1031, 2019.
- [63] B. Jiang, J. Hamari, and A. Maedche, "Cooperation or competition - when do people contribute more? A field experiment on gamification of crowdsourcing," *International Journal of Human-Computer Studies*, vol. 127, pp. 7–24, 2019.
- [64] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio, "Crowdsourcing: a review and suggestions for future research," *International Journal of Management Reviews*, vol. 20, no. 2, pp. 343–363, 2018.
- [65] F. Moodley, J. V. Belle, and N. Hasteer, "Crowdsourced software development: exploring the motivational and inhibiting factors of the South African crowd," in *Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pp. 656–661, Noida, India, January 2017.
- [66] L. Zhai, H. Wang, and X. Li, "Optimal task partition with delay requirement in mobile crowdsourcing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5216495, 2019.
- [67] E. Schenk, C. Guittard, and J. Pénin, "Open or proprietary? choosing the right crowdsourcing platform for innovation," *Technological Forecasting and Social Change*, vol. 144, pp. 303–310, 2019.
- [68] G. D. Saxton, O. Oh, and R. Kishore, "Rules of crowd-sourcing: models, issues, and systems of control," *Information Systems Management*, vol. 30, no. 1, pp. 2–20, 2013.
- [69] Y. Amsterdamer, T. Milo, A. Somech, and B. Youngmann, "December: a declarative tool for crowd member selection," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1485–1488, 2016.
- [70] Y. Amsterdamer, T. Milo, A. Somech, and B. Youngmann, "Declarative user selection with soft constraints," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, November 2019.
- [71] R. L. Saremi, Y. Ye, G. Ruhe, and D. Messinger, "Leveraging crowdsourcing for team elasticity: an empirical evaluation at TopCoder," in *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, pp. 103–112, Buenos Aires, AR, USA, May 2017.



- [72] F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab, "Modelling cognitive bias in crowdsourcing systems," *Cognitive Systems Research*, vol. 58, pp. 1–18, 2019.
- [73] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.
- [74] L. G. Pee, E. Koh, and M. Goh, "Trait motivations of crowdsourcing and task choice: a distal-proximal perspective," *International Journal of Information Management*, vol. 40, pp. 28–41, 2018.
- [75] C. Qiu, A. C. Squicciarini, B. Carminati, J. Caverlee, and D. R. Khare, "CrowdSelect: increasing accuracy of crowdsourcing tasks through behavior prediction and user selection," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, Indianapolis, IN, USA, October 2016.
- [76] S. Sharma, N. Hasteer, and J. P. Van-Belle, "An exploratory study on perception of Indian crowd towards crowdsourcing software development," in *Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 901–905, Noida, India, May 2017.
- [77] M. Modaresnezhad, L. Iyer, P. Palvia, and V. Taras, "Information Technology (IT) enabled crowdsourcing: a conceptual framework," *Information Processing & Management*, vol. 57, no. 2, Article ID 102135, 2020.
- [78] M. Garcia Martinez, "Inspiring crowdsourcing communities to create novel solutions: competition design and the mediating role of trust," *Technological Forecasting and Social Change*, vol. 117, pp. 296–304, 2017.
- [79] T. Mack and C. Landau, "Winners, losers, and deniers: self-selection in crowd innovation contests and the roles of motivation, creativity, and skills," *Journal of Engineering and Technology Management*, vol. 37, pp. 52–64, 2015.
- [80] I. Dissanayake, N. Mehta, P. Palvia, V. Taras, and K. Amoako-Gyampah, "Competition matters! self-efficacy, effort, and performance in crowdsourcing teams," *Information & Management*, vol. 56, no. 8, Article ID 103158, 2019.
- [81] S. Standing and C. Standing, "The ethical use of crowdsourcing," *Business Ethics: A European Review*, vol. 27, no. 1, pp. 72–80, 2018.
- [82] H. Zheng, D. Li, and W. Hou, "Task design, motivation, and participation in crowdsourcing contests," *International Journal of Electronic Commerce*, vol. 15, no. 4, pp. 57–88, 2011.
- [83] G. Montelisciani, D. Gabelloni, G. Tazzini, and G. Fantoni, "Skills and wills: the keys to identify the right team in collaborative innovation platforms," *Technology Analysis & Strategic Management*, vol. 26, no. 6, pp. 687–702, 2014.
- [84] R. Saremi, "A hybrid simulation model for crowdsourced software development," in *Proceedings of the 2018 IEEE/ACM 5th International Workshop on Crowd Sourcing in Software Engineering (CSI-SE)*, pp. 28–29, Gothenburg, Sweden, June 2018.
- [85] B. Satzger, R. Zabolotnyi, S. Dustdar et al., "Chapter 8-toward collaborative software engineering leveraging the crowd," in *Economics-Driven Software Architecture*, I. Mistrik, R. Bahsoon, R. Kazman, and Y. Zhang, Eds., pp. 159–182, Morgan Kaufmann, Boston, MA, USA, 2014.
- [86] T. D. LaToza, W. B. Towne, A. V. D. Hoek, and J. D. Herbsleb, "Crowd development," in *Proceedings of the 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pp. 85–88, San Francisco, CA, USA, May 2013.
- [87] E. Segev, "Crowdsourcing contests," *European Journal of Operational Research*, vol. 281, no. 2, pp. 241–255, 2020.
- [88] J. Troll, I. Blohm, and J. M. Leimeister, "Why incorporating a platform-intermediary can increase crowdsourcees' engagement," *Business & Information Systems Engineering*, vol. 61, pp. 433–450, 01 2019.
- [89] O. Tokarchuk, R. Cuel, and M. Zamarian, "Analyzing crowd labor and designing incentives for humans in the loop," *IEEE Internet Computing*, vol. 16, no. 5, pp. 45–51, 2012.
- [90] A. L. Zanatta, L. Machado, and I. Steinmacher, "Competence, collaboration, and time management: barriers and recommendations for crowdworkers," in *Proceedings of the 2018 IEEE/ACM 5th International Workshop on Crowd Sourcing in Software Engineering (CSI-SE)*, pp. 9–16, Gothenburg, Sweden, May 2018.
- [91] X. Wang, H. J. Khasraghi, and H. Schneider, "Towards an understanding of participants' sustained participation in crowdsourcing contests," *Information Systems Management*, vol. 37, no. 7, pp. 1–14, 2019.
- [92] M. Alhamed and T. Storer, "Estimating software task effort in crowds," in *Proceedings of the 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 281–285, Cleveland, OH, USA, September, 2019.
- [93] X. Zhang, Z. Chen, C. Fang, and Z. Liu, "Guiding the crowds for android testing," in *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pp. 752–753, Austin, TX, USA, May 2016.
- [94] S. Franken, S. Kolvenbach, W. Prinz, I. Alvertis, and S. Koussouris, "CloudTeams: bridging the gap between developers and customers during software development processes," *Procedia Computer Science*, vol. 68, pp. 188–195, 2015.
- [95] S. Hosio, J. Goncalves, N. van Berkel, S. Klakegg, S. I. Konomi, and V. Kostakos, "Facilitating collocated crowdsourcing on situated displays," *Human-Computer Interaction*, vol. 33, no. 5–6, pp. 335–371, 2018.
- [96] E. Mourelatos and M. Tzagarakis, "An investigation of factors affecting the visits of online crowdsourcing and labor platforms," *NETNOMICS: Economic Research and Electronic Networking*, vol. 19, pp. 95–130, 2018.
- [97] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2–3, pp. 243–278, 2005.
- [98] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [99] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Vol. 453, Springer Science & Business Media, Berlin, Germany, 1998.
- [100] N. Leicht, I. Blohm, and J. M. Leimeister, "Leveraging the power of the crowd for software testing," *IEEE Software*, vol. 34, no. 2, pp. 62–69, 2017.

## Research Article

# Probabilistic Model-Based Malaria Disease Recognition System

Rahila Parveen <sup>1,2</sup>, Wei Song <sup>1</sup>, Baozhi Qiu <sup>2</sup>, Mairaj Nabi Bhatti <sup>3</sup>, Tallal Hassan <sup>4</sup>,  
and Ziyi Liu <sup>5</sup>

<sup>1</sup>Henan Academy of Big Data, Zhengzhou University, Henan, Zhengzhou, China

<sup>2</sup>School of Information Engineering, Zhengzhou University, Henan, Zhengzhou, China

<sup>3</sup>Department of Information Technology, Shaheed Benazir Bhutto University, Shaheed Benazirabad, Nawabshah, Pakistan

<sup>4</sup>School of Business, Zhengzhou University, Henan, Zhengzhou, China

<sup>5</sup>College of Engineering & Science, University of Detroit Mercy, Detroit, USA

Correspondence should be addressed to Wei Song; [iewsong@zzu.edu.cn](mailto:iewsong@zzu.edu.cn)

Received 14 October 2020; Revised 8 December 2020; Accepted 17 December 2020; Published 6 January 2021

Academic Editor: Dr Shahzad Sarfraz

Copyright © 2021 Rahila Parveen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we present a probabilistic-based method to predict malaria disease at an early stage. Malaria is a very dangerous disease that creates a lot of health problems. Therefore, there is a need for a system that helps us to recognize this disease at early stages through the visual symptoms and from the environmental data. In this paper, we proposed a Bayesian network (BN) model to predict the occurrences of malaria disease. The proposed BN model is built on different attributes of the patient's symptoms and environmental data which are divided into training and testing parts. Our proposed BN model when evaluated on the collected dataset found promising results with an accuracy of 81%. On the other hand, F1 score is also a good evaluation of these probabilistic models because there is a huge variation in class data. The complexity of these models is very high due to the increase of parent nodes in the given influence diagram, and the conditional probability table (CPT) also becomes more complex.

## 1. Introduction

The name of malaria is given to this disease in back 1740. Malaria is a genuine worldwide sickness and the main source of bleakness and deaths in tropical and subtropical nations. It influences somewhere in the range of 350 and 500 million persons and caused over a million casualties every year. However, malaria is both preventable and treatable. It is very much important for humans to treat themselves quickly when someone is affected by any disease at early stage. It is brought about by parasitic protozoa (a sort of unicellular microorganism) of the family *Plasmodium*. As per the World Health Organization (WHO) Report, [1] the most effected region from this disease in Sub-Saharan Africa and India is at least 85% of the whole world. The WHO shows that malaria cases are increased in the Bolivarian Republic of Venezuela as compared to Sub-Saharan Africa from 2010 to 2018.

This life-threatening disease is spread with the bite of female mosquitoes. These female mosquitos are responsible for spreading the *Plasmodium* parasite from one person to

another, and these for the most infect the humans in between twilight and dawning time bites. Due to these *Plasmodium* infection [2] patients, red blood cells are damaged, which leads to this disease. There are five major categories of parasites by which humans are getting infected with malaria, namely, *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium malariae*, *Plasmodium ovale*, and *Plasmodium knowlesi* [3]. All of the deadliest parasites are *Plasmodium falciparum* in this disease of malaria. If malaria is detected at an early stage, then it can be remediable and escapable. The main technique for the detection of malaria parasites is the microscopic diagnosing method which is used until today as a gold technique. Slender or thick Giemsa recolored blood spreads are analyzed with an amplification factor of 100x goal and 10x visual focal point under a microscopic method [4]. It is derived from a blood smear screen report that any person might be influenced by more than one malaria parasite at the same time [5]. Besides, there are five unique types of malaria parasites and every species experiences an alternate life cycle. Each phase of the existence cycle

experiences an adjustment in its shape, size, morphology, shading, and so forth. These stages are named as a ring, trophozoite, schizont, and gametocyte [2].

Additionally, there exists an immediate connection between high episodes of malaria malady and the change in climate conditions. The high volume of cases in intestinal sickness scourge territories and deficiency of talented experts frequently prompts delay in revealing the outcome which might be basic for malaria treatment [6]. The deferral in finding forestall brief treatment of sickness further prompts constant stage, at considerable expense to the persons and enormous expense to society. This disease mostly exists in rural or remote areas of underdeveloped countries where technical staff did not skill so that in the current era of the Internet of Things (IoT) technologies for the collection of environmental data, advanced artificial intelligence (AI) can be utilized to detect malaria by analyzing the visual symptoms of infected persons. The visual symptoms of this disease are high temperature, feeling cold, headache, vomiting, and pain in body muscles or feeling too much tired. The doctor's reports regarding the mentioned symptoms can be monitored 24/7 in the affected area of respective countries for malaria disease. Due to this, less human interaction will be involved and automatically all these data are gathered at a central point. As these symptoms are detected in any person, the panel of doctors will recommend them for the clinical test of malaria at its early stage. Due to this, life of patients can be saved by monitoring his health condition not only for malaria but also blood pressure, sugar level, and heartbeat. In this process of gathering patient's data, privacy is also considered as the main part as per the recommended communication method for IoT by Jalbani et al. [7].

To validate the visual symptoms detected by the proposed system in this paper, the clinical test recommended by the doctors for malaria disease will be analyzed. As per the current study, it is suggested that probabilistic models are fit for learning information for the prediction of malaria disease. The Bayesian system (BS) is one such approach and has been broadly applied in healthcare systems because of their capacity to viably deal with variable data. A BN is a graphical portrayal of probability circulation where nodes speak to uncertain factors and connections speak to coordinate probabilistic impact between the uncertain values [8]. The connection between a node and its parents is evaluated by a CPT, determining the likelihood of the irregular variable precast on all assortments of the estimations of the parents. The structure of the system encodes data about probabilistic freedom with the end goal that the CPTs alongside the sovereignty relations give a full particular of the joint likelihood conveyance over the irregular factors spoke to by nodes. By decaying a joint likelihood dispersion into an assortment of littler nearby disseminations (the CPTs), a BN gives an exceptionally minimized interpretation of the total joint appropriation. In BN indicative models, factors that can impact the determination, including clinical signs, side effects, and lab results, are remembered for the model to frame a causal relationship organize [9]. The BN is one such class of probabilistic-based recognition approaches. It

exceeds expectations from clinical report characterization and impersonates the artificial neural network (ANN) system engineering such as human knowledge [10]. The IoT devices extracted data, and clinical tests will be further analyzed by the proposed method of AI, the BN. By the extracted symptom data and clinical reports, the information will be trained into the proposed framework for the accuracy of malaria disease recognition. On the bases of trained data, this system will decide whether the patient is infected with malaria or not. Because of this, the long process of microscopic tests will be reduced for remote areas and human life can be saved with this smart system. With this proposed model, not only will malaria disease be predicted but also other diseases can be monitored. It seems that machine learning (ML) approaches are used to build the different models using different approaches; however, every approach has its own advantages and disadvantages. Rule-based methods are based on informed search strategies which allow the system to construct direct classification connections while using probabilistic methods which are based on casual dependency relationships. Many classifiers such as ANN, support vector machines (SVM), decision tree, and BN have been used in medical diagnosis to get acceptable results; however, complexities of these algorithms are different which depend on data sets used for prediction. A BN seems to be more interpretable compared with other methods such as rule-based methods [11].

As discussed earlier that there are different ML techniques available to use in modeling the disease to recognize them properly, however, BN is very much helpful for predicting the malaria disease with different features such as environmental features. The BN has capabilities to integrate with different features to provide the prediction of the particular disease. The BN also shows the cause and effect relationship efficiently whereas decision tree (DT) is only work on yes and no properties and it can limit the results in certain range.

The main disadvantages of DT is overfitting and underfitting when using the small amount of data. Another main advantage of using BN is that it only connects nodes that are probabilistically linked by some sort of causal dependency, which helps to reduce the computational cost of the system. A second main reason is that the BN nets are very much flexible; therefore, they are so adjustable [12].

The paper is divided into the following sections. In Section 2, the related work is given for this area of research. In Section 3, the proposed methodology is described for the Bayesian model to predict the malaria disease. In Section 4, the results and discussions are explained for the proposed model. In Section 5, the final words of conclusions are given regarding this research paper.

## 2. Related Work

Vijayalakshmi [13] has proposed a new visual geometry group-support vector machine (VGG-SVM) organizes utilizing move learning approach for perceiving tainted falciparum intestinal sickness. Here, a preprepared VGG is considered as a specialist learning model and it is focused to

characterize 1000 classes. SVM is an area explicit classifier used to group tainted and noncontaminated from malaria microscopic images.

Parveen et al. [14] have utilized neural networks (NNs) by mental undertaking and simulation of the brain. A multilayer feedforward system with a back proliferation learning algorithm is utilized. The effectiveness of the proposed framework is contrasted with other comparative frameworks.

Dong et al. [15] have used a completely automated process with no manual element extraction, and we picked a profound convolutional neural system (CNN) as the classifier. CNN can separate various leveled portrayals of the input information. They have used LeNet-5 to get proficiency with the intrinsic highlights of intestinal malaria contaminated and nontainted cells.

Bari et al. [16] have used identification, and analysis of lung cancer growth can be processed as images on three fundamental stages which are prehandling, division lastly followed by postpreparing. Any sort of diagnosing technique is expected to gather the blood tests to recognize malaria. There are two distinctive blood films are utilized for distinguishing proof of malaria in particular, thick and thin blood films by Elter et al. [17]. Thick blood films acknowledge the blood tests for the location of intestinal malaria parasite thickness. Slender blood films acknowledge the blood tests to recognize or portrayal of malaria parasites.

Maqsood et al. [18] have proposed identifying dark scale images utilizing hidden Markov tree (HMT) which is introduced. Utilizing the above-displaying structure, a joined spatial and fleeting separating procedure can expel Gaussian just as dot commotion from shading images and video successions significantly. By molesting the conditions among wavelet coefficients, better execution has been accomplished.

Cooper et al. [19] have developed a BN with 20 million hubs for *Bacillus anthracenes* outburst identification. The whole populace is displayed by a self-contained system with every individual in the system associated with the remainder of the system through a hub called sickness status. The undertaking is to compute the back prospect of the alarm hub given the construed illness statuses surprisingly in the public. The work does not think about spatial or sophisticated viewpoints.

Jiang and Wallstrom [20] have explored the utilization of BS for cryptosporidium occurrence place and projection. This examination centers essentially around the weaknesses of traditional techniques for time-arrangement investigation and improving them by utilizing BS. The model is tried on a recreated occurrence informational collection.

In Martha et al.'s study [21], an adaptable smart framework has been developed by using fuzzy logic in diagnosing malaria sickness. The framework recognized malaria sickness with high identification exactness. In any case, the framework had gone with issues: the framework intelligent module could not make a bidirectional conclusion and handle issues of impossibility.

In Mehanian et al.'s study [22], a smart framework that investigated malaria using CNN was developed. The

framework decided to *Plasmodium falciparum* malaria sickness to have high acknowledgment exactness. In any case, the framework had the going with insufficiencies to be explicit: the arrangement gotten from the NN is difficult to get a handle on and the learning method of the framework is tedious and capital-escalated, and the framework fails to perceive cerebral malaria and other mosquitoborne disorders due to the covering indications the sickness impacts to neurological and febrile infections.

Junior et al. [23] have assessed the utilization of ANN and BN with regards to the determination of asymptomatic malaria disease. They have figured out how to assemble ANNs and BNs dependent on immunological and epidemiological information gathered from people from an exceptionally endemic area for malaria in the Brazilian Amazon. The results were contrasted with those acquired utilizing light microscopy and a subatomic test (settled PCR). The Naïve-Bayesian has utilized as a probabilistic learning technique, and these classifiers are among the best-known algorithms for figuring out how to arrange text records such as e-mail spam separating [24]. Some exploration shows that it is likewise helpful for heart illness forecast.

Osubor and Chiemeké [25] have proposed a solution of the adaptive neurofuzzy inference system (ANFIS) that examined malaria disease was made. The framework decided malaria sickness with 98% disclosure precision. Notwithstanding the high acknowledgment exactness, the framework had the going with quandaries, for instance, inconvenience in recognizing cerebral malaria sickness and other mosquitoborne infirmities as a result of the covering reactions and the contamination grants to neurological and febrile maladies.

Djam et al. [26] have an expert framework called the fuzzy expert system for the management of malaria (FESMM) which relied upon fluffy principles to investigate malaria sickness called was made. The framework decided malaria to have high disclosure accuracy. Despite the framework's high distinguishing proof accuracy, it had the going with detriments: the framework could not recognize cerebral malaria and other mosquitoborne sicknesses.

Jiang and Wallstrom [20] have analyzed the utilization of BNs for cryptosporidium spreading location and forecast. This examination concentrates predominantly on the weaknesses of old-style strategies for time-arrangement investigation and improving them by utilizing BNs. The model is tried on a reproduced spreading informational index.

Sebastiani et al. [27] have developed an automatic BN for anticipating flu-like disease and the quantity of pneumonia and flu passing's dependent on past pediatric and grown-up instances of respiratory disorder. No natural factors are utilized.

Warfield [28] has proposed interpretive structural modeling (ISM) which was utilized to build up a visual progressive structure of complex frameworks. The method was utilized in overseeing decision making for complex issues. The contribution of the ISM strategy was unstructured, and indistinct data about the framework factors and

their interdependencies were utilized. The yield of ISM investigation was a very much characterized, ordered, and instructive model, which is valuable for some different purposes.

Colin et al. [29] have executed ISM to consider the interdependencies in gracefully chain hazards. The interrelationship among flexibly hazard factors, which they created, depended on the reliance and driving intensity of separate components. Their investigation is very much important for future researchers in this area. They contemplated 21 danger factors, and beginning interrelationships were created utilizing bunch conversations among the creators and individual specialists.

Singh and Kant [30] have organized nine boundaries in knowledge management (KM) in the business procedure. The KM hindrances were those which antagonistically influence the execution of KM in a business association. They executed the ISM strategy for their investigations. The shared connections among various boundaries were grouped in various levels and incorporated a visual relationship also. Be that as it may, the examination was deficient in clarifying how frail or solid relations were among different interconnected elements.

Hussain et al. [31] have proposed a novel methodology for identification of hand developments during impediment, and they reported consequences of isolating hand (s) from the head (face) locale during explicit signal advancement, utilizing shading and surface invariant recognition plot. Moreover, broadening their work for the improvement of a keen mentoring framework that requires distinguishing motions and foreseeing the understudy's emotional (mental) state progressively, they proposed and assessed a coordinated methodology of identification and understanding of mental states from hand signals utilizing information detailed before by Abbasi et al. [32]. The outcomes are promising in further turn of events.

The BN has played a vital role in healthcare systems. BN has got great attention in all over the world and maintains to obtain momentous importance for the facility to mingle the available evidence and perfect reasoning under uncertain situation by McLachlan et al. [33].

### 3. Proposed Methodology

The data have been collected from various sources regarding malaria-affected peoples. For the collection of information regarding the symptoms of malaria disease, different types of IoT devices have been utilized. For the environmental information, the research target was tropical and nontropical area weather conditions. Due to these, more effected areas from malaria can be identified on the basis of weather conditions changed, for example, in rain, weather humidity will be increased. In this condition, the number of malaria cases may increase due to the growth of mosquitoes in these conditions. This all gathered information will be processed in BN for getting perfect results of probability regarding malaria disease. The proposed methodology is described in Figure 1.

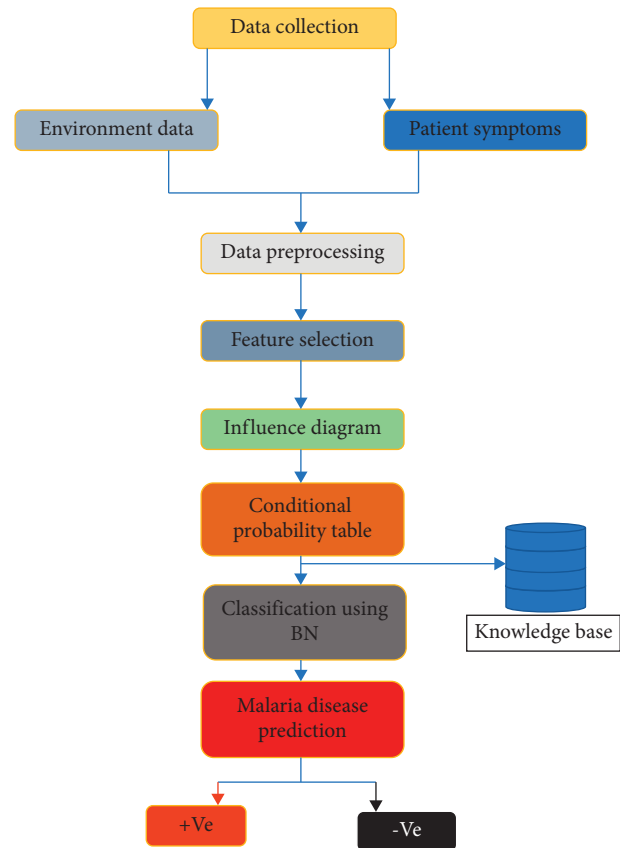


FIGURE 1: A proposed framework for malaria disease prediction.

**3.1. Visual Symptoms of Malaria Patients.** The symptoms of malaria disease may be visible from 8–25 days in infected peoples from this parasite. However, at the same time, these symptoms can be visible for some days later as mentioned above if those patients have used antibiotic medicine already in proactive measures from malaria disease. These symptoms can vary for each type of malaria disease parasite infection. Likewise, for the *Plasmodium vivax* and *Plasmodium ovale*, the patients feel skin trembling with high temperature and wilting after every two-day cycle. On the other hand, for *Plasmodium malariae* and *Plasmodium falciparum*, parasite-infected patients feel these symptoms after every three-day cycle in which high temperature may remain also for the 36–48 hours. And other main categories of malaria symptoms are headache along with high temperature and feeling too much fatigued. The doctors working on malaria recommend that at least two types of symptoms will be visible simultaneously in the patients infected with parasites. Alongside these four main types of symptoms as mentioned earlier, there are four other low-level types of symptoms also observed in malaria patients, such as spewing, amplification in irritation, cough, and pain in body muscles. In all of the parasite infections, plasmodium falciparum is more dangerous and life-threatening for infected patients. In this type of infection, the patients' symptoms can be visible after 9–30 days. The headache is the more prominent type of symptoms in this infection. This will help to increase the chance of more accurate prediction of malaria disease.

**3.2. Environmental Data Collection.** The environmental effects are also major sources for spreading the malaria infection in tropical and nontropical areas around the globe. The major carrier of these malaria parasites is female mosquitoes, and these types of environments can be too friendly for their growth. The environment conditions are also considered in this research paper as a data component for the prediction of malaria. The different sources have been utilized for the collection of environmental data the online resources such as weather.com and local metrological department data are the main contributor to this research, such as maximum temperature, minimum average temperature, humidity, rainfall months, and how many malaria cases are reported during that period. The humidity will increase during the rainfall seasons, and that is the period in which mosquitoes will grow in more numbers. After that these cases will be categorized as per the type of parasite infections. This will help near about 85% accurate prediction of malaria disease.

**3.3. Influence Diagram.** Howard [34] has developed an influence diagram which is known as relevance diagrams. These diagrams are based on acyclic directed graphs for the solution of decision issues. The motive behind this influence diagrams was to get a higher rate profit by selecting alternative decisions from given values. The influence diagrams are the same as BNs for the complete overview of domain architecture such as the structure of the decision problems. The influence diagrams are further divided into four nodes which are decision, chance, deterministic, and values and two kinds of arcs which are influences and informational. Typically, an arc in an influence outline indicates an influence, such as the way that the node at the tail of the circular segment impacts the worth or the probability dissemination over the potential estimations of the node at the top of the arc. A few arcs in influence graphs have causal importance. Specifically, a guided way from a choice node to a possibility node implies that the choice (for example, control of the diagram) will affect that opportunity node in the feeling of changing the probability appropriation over its results.

In Figure 2, the sample influence diagram is shown and can be obtained from the factorized form and where directed edges represent direct dependencies and the absence of edges shows the conditional independence. These types of the model are also known as belief network (BN) or graphical models and causal network as well. BN or BN is a directed acyclic graph represented with pairs  $G(V, E)$ .  $V$  stands for vertices or nodes which represents random variables (events) in our case symptoms and environmental variables, and  $E$  stands for edges or links between nodes which shows a causal dependency relationship. A direct link from variable  $X$ - $Z$  indicates that  $X$  can cause  $Z$ , or, in BN terminology,  $X$  is a parent of  $Z$ , and  $Z$  is a child of  $X$ .  $P$  is a probability distribution over vertices  $V$ . Discrete random variables are assigned to the node variables representing a finite set of mutually exclusive states and interpreted with a CPT that represents the conditional probability of the

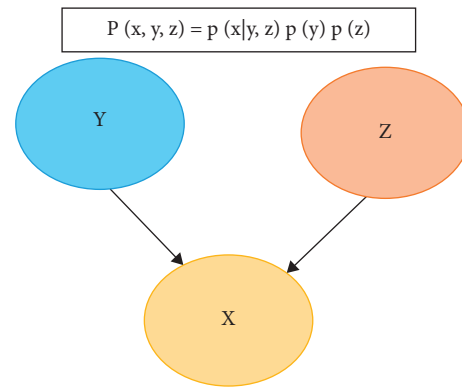


FIGURE 2: Sample influence diagram.

variable given the values of its parents in the given graph. In our proposed BN model, nodes are symptoms of malaria disease and environmental variables.

**3.4. Conditional Probability Tables.** It is a common observation that the effects of direct or indirect factors on the number of events are very difficult to model. However, BN is considered a very easy and powerful framework for graphical modeling of cause-effect relationships between different variables using influence diagrams. The sample CPT is shown in Figure 3.

The BN is divided into two parts: one is the qualitative part and the other is the quantitative part. The qualitative part is based on structural modeling and learning, which is used to make the structure of a directed acyclic graph (DAG) which is made of a set of vertices (nodes) and directed arrows (edges) which show the parent-child relationship between nodes. When two nodes are connected directly by an edge, the node with a link directed to a successive node is called a “parent node” of the succeeding node. The subsequent node is called a “child node.” Child nodes are also known as conditionally dependent nodes on their parent nodes. For instance, nodes  $Z$  and  $Y$  are the parent nodes of node  $X$  (Figure 2). The quantitative part of a BN is also known as parameter learning, which is used to find out the conditional probability distribution of each node, according to the established BN structure of the proposed scenario which is based on Bayes’ theorem. The same scenario has been used in our proposed BN prediction model where symptoms are given to the BN model and the CPT has been calculated using Bayes theorem.

## 4. Results and Discussion

For the prediction of malaria disease based on environmental information and the patient’s symptoms by using the probabilistic model, for a better prediction with less time, the BN model is used in this paper. If the patient is not diagnosed at the early stage of this parasite virus, then this will be life-threatening for them. The malaria disease exists in the underdeveloped countries where a shortage of experts as a laborer’s technician to diagnose it. As technology evolved such as IoT networks and AI in the field of healthcare systems, by

Variable Y		Variable Z	
Yes	0.4	Yes	0.2
No	0.6	No	0.8

Variable X			
Y	Z	Yes	No
Yes	Yes	0.55	0.45
Yes	No	0.4	0.6
No	Yes	0.55	0.45
No	No	0.9	0.1

FIGURE 3: Sample CPT.

utilizing these two technologies, the malaria disease can be predicted at an early stage around 80–85%. The current research has been done in the diagnosis methods which are processed by microscopic tests. There are chances that the medical staff can be infected from this virus. The proposed model in the research paper will predict malaria disease remotely with the help of IoT devices and sensors. The environmental information has been collected from various sources regarding weather conditions of those tropical and nontropical areas where the malaria parasites exist. This disease spreads due to the bites of female mosquitoes with the infection of parasite virus so that there is also the season of rising these mosquitoes in different weather conditions. The humans may be infected more during more rain falling and humid conditions. Or they may be at the start of winter or the end of winter. However, the study suggests that this infection rate increases during the few last months of summer because in this period the mosquitoes are rising too much due to heavy rain falls and humidity. In this paper, at least 16 types of environmental and patient's symptoms have been considered for the probabilistic.

For data description, two types of data are collected from the patient's symptoms and environmental data. The patient's symptoms are collected from the Peoples Medical Hospital (PMH) Shaheed Benazirabad, Nawabshah, Pakistan, regarding infected patients. And the environmental data are collected from online resources. The collected data are shown in Table 1.

**4.1. Bayesian Network Model.** The BN architecture is a graphical representation, which demonstrates it as a qualitative base for the communication between variable sets within a model. The architecture of the directed graph could be cloned into the fundamental construction for the designed domain, or it may not be necessary. At the point when the structure is causal, it gives a helpful, measured knowledge into the associations among the factors and considers the prediction of impacts of outside control. In light of the restrictive conditions, a BN factorizes the joint dissemination of factors. The BN registers the circulation probabilities in a given arrangement of factors by utilizing earlier data of different factors by Jensen [8]. The arrangement of nodes and coordinated circular segments are the attributes of a BN, where nodes speak to the framework factors and the bend speaks to

TABLE 1: Data description table.

Description	Value
Total record	250
Training	150
Testing	100
<i>Patient's symptoms</i>	
Fever	
Chills	
Sweats	
Shaking	
Jaundice	
Headache	
Fatigue	13
Low energy	
Nausea	
Vomiting	
Myalgia	
Stomach upset	
Diarrhea	
<i>Environmental data</i>	
Rainfall	
Humidity	03
Temperature	

the reason impact relationship of conditions among the factors. Every node has its likelihood of an event. On account of a root node, such likelihood is from the earlier one and is resolved for the others by surmising. The nodes which are not coordinated towards some other nodes are the parent nodes. A youngster node is a node where a node gets any edge/coordinated curves. The probabilities of parent nodes and a restrictive likelihood (CPT) were the bases for BN calculations. The model has been created using GeNIe/SMILE [35]. The CPT contained the data for contingent probabilities. Figure 4 shows the sample BN model.

BS is noncyclic coordinated diagrams that represent to factorizations of joint probability appropriations. Each joint probability dissemination over  $n$  arbitrary factors can be factorized in  $n$  times and composed as a result of probability disseminations of every one of the factors contingent on different factors. The basic properties of Bayesian networks, alongside the CPT related to their nodes, take into account probabilistic thinking inside the model. Probabilistic thinking inside a BN is prompted by watching proof. A node that has been watched is called a proof node. Watched nodes become launched, which implies, in the least difficult case, that their result is known with sureness. The effect of the proof can be spread through the system, changing the probability conveyance of different hubs that are probabilistically identified with the proof.

**4.2. BN Model Trained Data.** This cycle sums at the establishments to a dull use of Bayes's hypothesis to update the probability appropriations of all nodes in the system. Various methods of applying Bayes' hypothesis and distinctive requests for updating lead to various calculations. The current calculations for thinking in BS can be partitioned into three gatherings: message passing, chart decrease, and

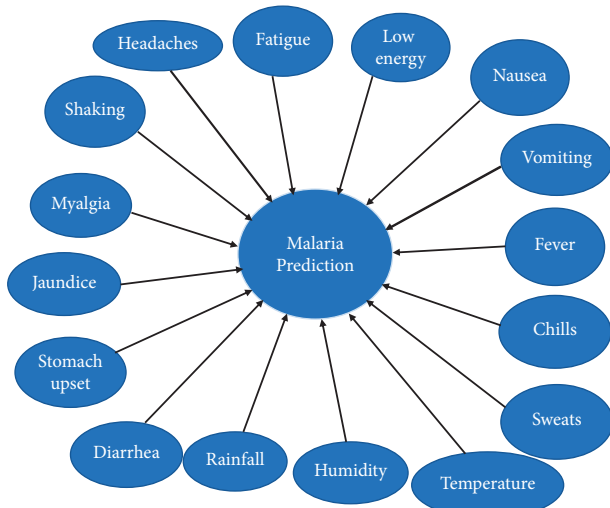


FIGURE 4: Proposed BN model.

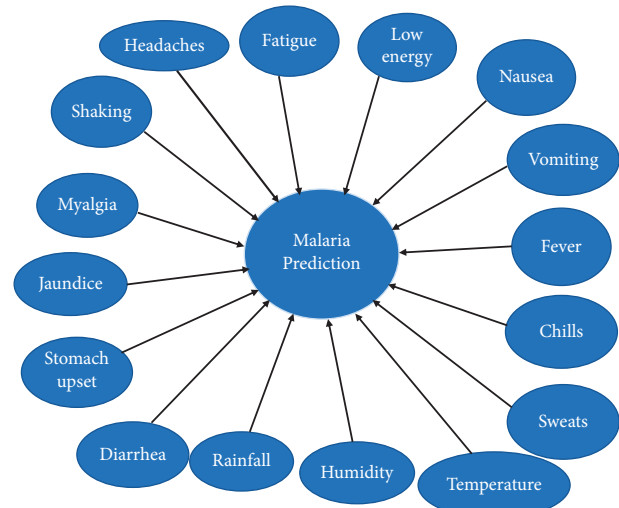


FIGURE 5: Proposed trained BN model.

stochastic reenactment. The express portrayal of freedoms takes into account the expanded computational manageability of probabilistic thinking. Probabilistic surmising in separately associated BN is proficient. Sadly, careful calculations for duplicate associated systems are obligated to exponential multifaceted nature in the number of nodes in the system. The environmental information and patient's symptoms are trained in the BN model as shown in Figure 5.

**4.3. Testing of Proposed BN Model.** To test the samples, the proposed BN model has been used to predict the occurrence of malaria in a particular patient. Figure 6 shows the result of the proposed model applied to the data of a patient. Different symptoms and environmental data have been given to the model, and the model reported the accuracy 81%. 100 samples are tested on the proposed BN model.

Figure 7 shows the negative test result of a patient.

**4.3.1. Conditional Probability Table Generation.** In the probabilistic domain, the graphs or diagrams will show a much more detailed view of information regarding architecture but not about the numerical informational. Those values are encrypted into conditional probabilistic delivery matrices which are equal to the factorization form and these are known as CPT connected with the nodes. Significantly, there will consistently be nodes in the system without any forerunners. These nodes are described by their earlier negligible probability dissemination. Any probability in the joint probability dispersion can be resolved from these expressly spoken of earlier and restrictive probabilities. After data collection, there is a need of preprocessing those data for training and testing of the BN model with respect to CPT for the prediction of malaria disease. And after that, this information will be preprocessed into the BN model along with an influence diagram and CPT for the prediction of malaria disease. With this, malaria disease can be predicted at an early stage, and due to this, the life of the patient can be saved.

**4.4. Proposed BN Model Evaluation.** A confusion matrix is a matrix that can be used to measure the performance of the classifier in supervised ML methods. It is used to evaluate the performance of the classification model on the given test data set whereas truth values are given. The confusion matrix has been used to visualize the accuracy of a classifier by comparing the actual data and predicted data classes. In this research work, 100 patient's records are used to test the performance of the proposed BN model. Our model out of 100 records correctly classified 81 patients as malaria positive and 19 correctly classified as negative. Therefore, the overall accuracy of the classifier is 81% shown in Table 2. After accuracy measurements, the F1 score is also very important to understand the performance of the classifier which shows the weighted average of Precision and Recall. In some problems, the F1 score is considered more useful than accuracy when there is an uneven class distribution of the features.

The statistical significant test is used to measure the probability of different observed relationships of the data and provide the important information regarding the proposed research findings. However, statistical significance may report confusing results due to less numbers and does not have any relationship with practical findings of the particular research. Nonstatistical significance does not mean that there is no significant difference between the different groups or shows no effect of the findings by Amrhein et al. [36].

**4.5. Computational Complexity of Probabilistic BN Model.** It has been remembered that the complexity of probabilistic models comes in the category of NP-hard in worst-case scenarios because the complexity of these probabilistic models is based on the exponential growth of the CPT in the number of parents and structural modeling problem of connectivity of DAGs. If the number of nodes increased, CPT becomes more complex; for example, if there are 10 nodes, then 2048 parameters are there, so care must be taken in mind when using probabilistic inference.



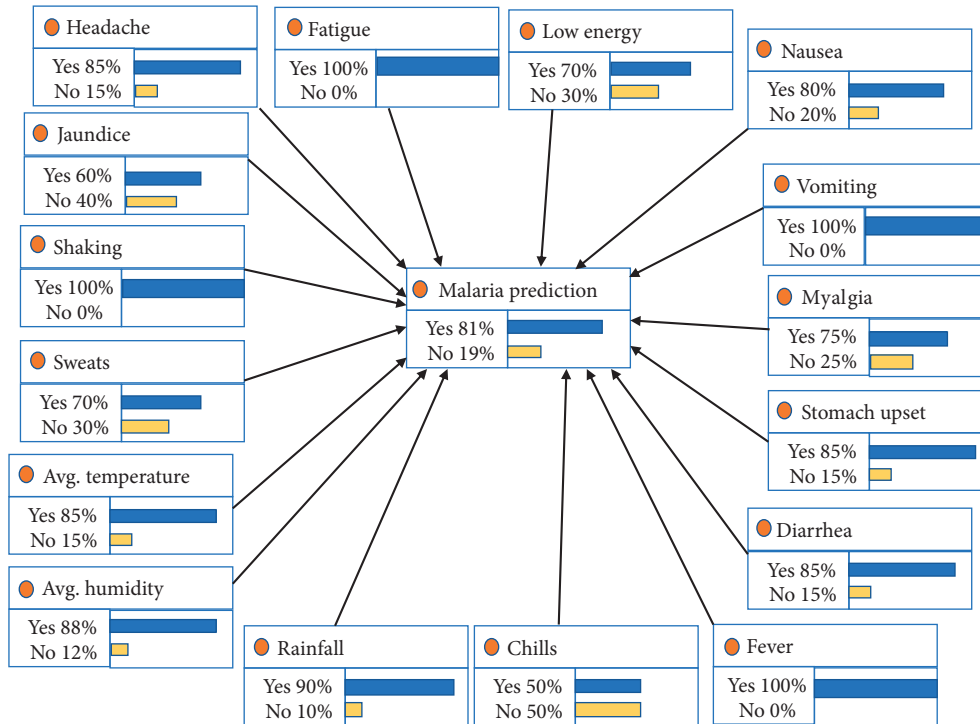


FIGURE 6: Proposed BN model predicted positive malaria disease.

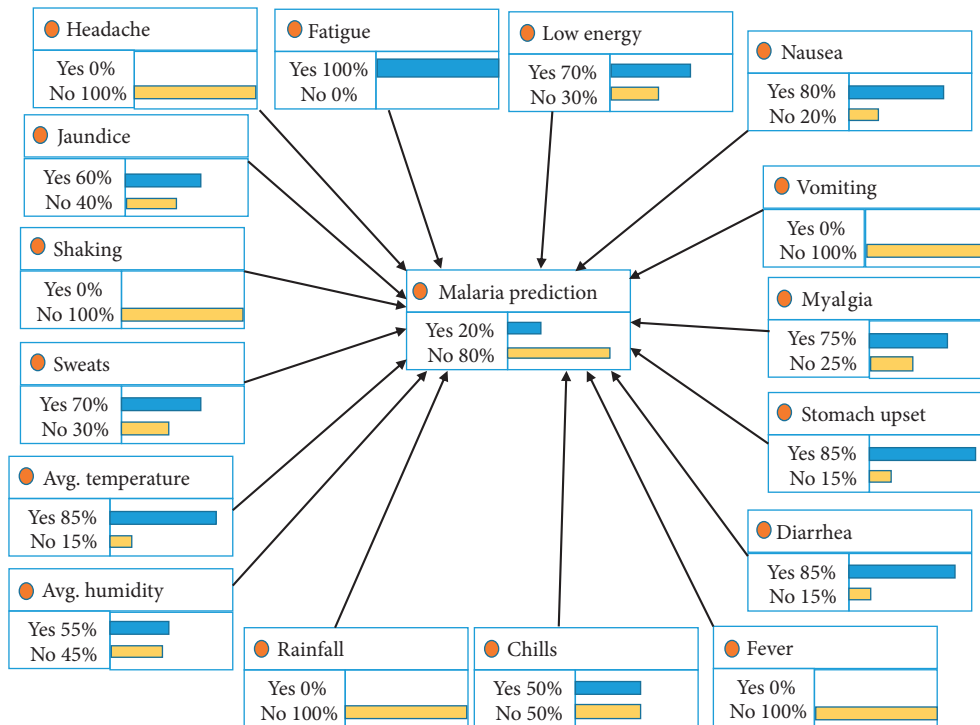


FIGURE 7: Proposed BN model predicted negative malaria disease.

TABLE 2: Confusion matrix.

Actual	Predicted		No
	Yes	No	
	Yes	46	14
	No	5	35
Measure	Value	Derivations	
Sensitivity	0.9020	TPR = TP/(TP + FN)	
Specificity	0.7143	SPC = TN/(FP + TN)	
Precision	0.7667	PPV = TP/(TP + FP)	
Negative predictive value	0.8750	NPV = TN/(TN + FN)	
False positive rate	0.2857	FPR = FP/(FP + TN)	
False discovery rate	0.2333	FDR = FP/(FP + TP)	
False negative rate	0.0980	FNR = FN/(FN + TP)	
Accuracy	0.8100	ACC = (TP + TN)/(P + N)	
F1 score	0.8288	F1 = 2TP/(2TP + FP + FN)	
Mathews correlation coefficient	0.6288	$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$	

## 5. Conclusion

Malaria is a life-threatening disease to the patients if it is not detected timely. For quick detection of this disease and without infecting medical staff from this virus, in this research paper, the AI method of BN is used for the prediction of malaria disease. The two kinds of parameters, the environmental information and patient's symptoms, are considered to predict it. In this process, the environmental information is collected from various types of online resources and local metrological departments of the respective areas regarding weather condition predictions. For this, the weather conditions are collected from tropical and nontropical countries of the world because, as per old or current research, these countries are still more affected by malaria. However, in these countries, the mosquitoes with the virus of parasites are at rising in the season especially in summer and at the start or end of winter. The weather conditions are included in the proposed system which are rain falling, humidity rate, and weather is hot or cold, taken as input into the proposed framework. In this paper, we have used the BN model to predict malaria disease using different symptoms and environmental data. We used BN-based inference from the GeNIe/SMILE tool to train and test our proposed framework. We built our own BN model and trained on our data set.

The proposed model could correctly predict malaria as positive or negative. There are 13 different symptoms of malaria, and 3 environmental factors are used in our proposed system. In this paper, we present a framework based on the BN model for the prediction of malaria disease which provides 81% accuracy of results on a given data set.

The main contribution of this paper is to use clinical diagnosis of symptoms of any disease is based on visual features that showed by patients. This method seems to be useful where there is no any urgent facility of testing the blood samples are available at early stages of the disease, and this may be helpful to control this disease to prevent the human life. In this research paper, additional feature of environmental conditions is also considered which may help to diagnose the patients properly.

The BN inference model is considered an NP-hard problem due to the increasing number of parent nodes in the influence diagram. The CPT also becomes complex which creates a problem for the machine. Therefore, care must be taken into account when selecting the nodes for designing any probabilistic model.

*5.1. Future Work.* In future work, there may be a possibility to increase the data set to get a more accurate result. Even there is a huge possibility of using a different variation of the BN model for the prediction of malaria and many other diseases also. For the collection of data regarding patient's symptoms and environmental information, the IoT devices and sensors will play a vital role. Due to the combination of AI and IoT technologies, the prediction of malaria or other diseases can be detected at an early stage.

## Data Availability

Two types of data are collected from the patient's symptoms and environmental data. The patient's symptoms are collected from the hospitals regarding infected patients. And the environmental data are collected from online resources.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work was supported by the International Cooperation Project, Department of Science & Technology, Henan Province, China, under grant no. 172102410065; Basic Research Project of the Education Department of Henan Province under grant no. 17A520057; Frontier Interdisciplinary Project of Zhengzhou University under grant no. XKZDQY202010; and the ZZU-Kingduns Joint Laboratory of Cyber Security.

## References

- [1] World Health Organization, *The World Malaria Report 2019 at a Glance*, World Health Organization, Geneva, Switzerland, 2019, <https://www.who.int/news-room/feature-stories/detail/world-malaria-report-2019>.
- [2] Z. May, S. S. A. M. Aziz, and R. Salamat, "Automated quantification and classification of malaria parasites in thin blood smears," in *Proceedings of the IEEE International Conference on Signal and Image Processing Applications*, pp. 369–373, Melaka, Malaysia, October 2013.
- [3] P. Khanna and S. Kumar, "Malaria Parasite Classification Employing Chan–Vese Algorithm and SVM for Healthcare," in *Proceedings of the 1st International Conference on Computing, Communications, and Cyber-Security*, pp. 697–711, Springer, New York; NY; United States, November 2020.
- [4] S. N. Chavan and A. M. Sutkar, "Malaria disease identification and analysis using image processing," *International Journal Latest Trends Engineering Technology*, vol. 3, pp. 218–223, 2014.
- [5] World Malaria Report, 2020, <https://www.who.int/malaria/publications/world-malaria-report-2017/en/>.
- [6] S. Rajaraman, S. K. Antani, M. Poostchi et al., "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *Peer Journal*, vol. 6, Article ID e4568, 2018.
- [7] K. B. Jalbani, A. H. Jalbani, and S. S. Soomro, "IoT Security: To Secure IoT Devices with Two-Factor Authentication by Using a Secure Protocol," *Industrial Internet of Things and Cyber-Physical Systems: Transforming the Conventional to Digital*, pp. 98–118, 2020.
- [8] F. V. Jensen, *An Introduction to Bayesian Networks*, vol. 210, pp. 1–178, UCL Press, London, UK, 1996.
- [9] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, CRC Press, Boca Raton, Florida, USA, 2010.
- [10] V. V. Makkapati and R. M. Rao, "Ontology-based malaria parasite stage and species identification from peripheral blood smear images," in *Proceedings of the IEEE Annual International Conference on Engineering in Medicine and Biology Society*, pp. 6138–6141, Boston, Massachusetts, August 2011.
- [11] N. Shafaf and H. Malek, "Applications of machine learning approaches in emergency medicine," *Archives of Academic Emergency Medicine*, vol. 7, no. 1, 2019.
- [12] V. Bhavana and T. Adilakshmi, "Comparison of decision tree classifier and Bayes classifier using WEKA," *International Journal of Computer Applications*, vol. 176, no. 3, 2017.
- [13] A. Vijayalakshmi, "Deep learning approach to detect malaria from microscopic images," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15297–15317, 2020.
- [14] R. Parveen, A. H. Jalbani, M. Shaikh et al., "Prediction of malaria using artificial neural network," *IJCSNS*, vol. 17, no. 12, p. 79, 2017.
- [15] Y. Dong, Z. Jiang, H. Shen, and W. D. Pan, "Classification accuracies of malaria-infected cells using deep convolutional neural networks based on decompressed images," in *Proceedings of the IEEE Southeast Conference*, pp. 1–6, San Jose California, March 2017.
- [16] M. Bari, A. Ahmed, M. Sabir, and S. Naveed, "Lung Cancer Detection Using Digital Image Processing Techniques: A Review," *Mehran University Research Journal of Engineering & Technology*, Jamshoro, vol. 38, no. 2, p. 351, Pakistan, 2019.
- [17] M. Elter, E. Haßmeyer, and T. Zerfaß, "Detection of malaria parasites in thick blood films," in *Proceedings of the IEEE Annual International Conference on Engineering in Medicine and Biology Society*, pp. 5140–5144, Boston, Massachusetts, August 2011.
- [18] A. Maqsood, I. Touqir, A. M. Siddiqui, and M. Haider, *Wavelet-Based Video Denoising Using Probabilistic Models*, *Mehran University Research Journal of Engineering & Technology*, Jamshoro, vol. 38, no. 1, p. 17, Pakistan, 2019.
- [19] G. F. Cooper, D. Dash, J. Levander, W. K. Wong, W. Hogan, and M. Wagner, "Bayesian Biosurveillance of Disease Outbreaks," 2012, [https://www.dbmi.pitt.edu/sites/default/files/Cooper\\_13.pdf](https://www.dbmi.pitt.edu/sites/default/files/Cooper_13.pdf).
- [20] X. Jiang and G. L. Wallstrom, "A bayesian network for outbreak detection and prediction," *Proceedings of National Conference on Artificial Intelligence*, AAAI Press, MIT Press, vol. 21, no. 2, p. 1155, Menlo Park, CA, USA, 2006.
- [21] S. Martha, N. Emilian, A. Frederick et al., "Deep Learning Mobile Application towards Malaria Diagnosis," in *Proceedings of the 9th International Conference on Learning Representations*, pp. 1–6, Vienna, Austria, May 2020.
- [22] C. Mehanian, M. Jaiswal, C. Delahunt et al., "Computer-automated malaria diagnosis and quantitation using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 116–125, Venice, Italy, October 2017.
- [23] A. M. B. Junior, A. A. Duarte, M. B. Netto, and B. B. Andrade, "Artificial neural networks and bayesian networks as supporting tools for diagnosis of asymptomatic malaria," in *Proceedings of the IEEE 12th International Conference on E-Health Networking, Applications and Services*, pp. 106–111, Lyon, France, July 2010.
- [24] Naive-Bayes Classification Algorithm, <http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf>, 2020.
- [25] V. I. Osuor and S. C. Chiemeké, "An adaptive neuro-fuzzy inference system for the diagnosis of malaria," *Nigerian Society for Experimental Biology Journal*, vol. 14, no. 4, 2017.
- [26] X. Y. Djam, G. M. Wajiga, Y. H. Kimbi, and N. V. Blamah, "A fuzzy expert system for the management of malaria," *International Journal of Pure and Applied Sciences and Technology*, vol. 5, no. 2, pp. 84–108, 2011.
- [27] P. Sebastiani, K. D. Mandl, P. Szolovits, I. S. Kohane, and M. F. Ramoni, "A bayesian dynamic model for influenza surveillance," *Statistics in Medicine*, vol. 25, no. 11, pp. 1803–1816, 2006.
- [28] J. N. Warfield, "Developing interconnection matrices in structural modeling," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-4, no. 1, pp. 81–87, 1974.
- [29] J. Colin, D. Estampe, H. C. Pfohl, P. Gallus, and D. Thomas, "Interpretive structural modeling of supply chain risks," *International Journal of Physical Distribution & Logistics Management*, vol. 41, no. 9, pp. 839–859, 2011.
- [30] M. D. Singh and R. Kant, "Knowledge management barriers: an interpretive structural modeling approach," *International Journal of Management Science & Engineering Management*, vol. 3, no. 2, pp. 141–150, 2008.
- [31] A. Hussain, A. R. Abbasi, and N. Afzulpurkar, "Detecting & Interpreting Self-Manipulating Hand Movements for Student's Affect Prediction," *Human-Centric Computing and Information Sciences*, vol. 2, no. 1, p. 14, 2012.
- [32] A. R. Abbasi, M. N. Dailey, N. V. Afzulpurkar, and T. Uno, "Student mental state inference from unintentional body gestures using dynamic bayesian networks," *Journal on Multimodal User Interfaces*, vol. 3, no. 1, pp. 21–31, 2010.
- [33] S. McLachlan, K. Dube, G. A. Hitman, N. Fenton, and E. Kyrimi, "Bayesian Networks in Healthcare: Distribution by

- Medical Condition,” *Artificial Intelligence in Medicine*, vol. 7, 2020.
- [34] R. A. Howard, “Readings on the Principles and Applications of Decision Analysis,” *Professional Collection Strategic Decisions Group*, vol. 2, 1984.
- [35] *GeNIe Modeler or SMILE Engine*, <https://www.bayesfusion.com/>, 2020.
- [36] V. Amrhein, S. Greenland, and B. McShane, “Scientists rise up against statistical significance,” *Nature*, vol. 567, no. 7748, pp. 305–307, 2019.

## Review Article

# Machine Learning Techniques for Quantification of Knee Segmentation from MRI

Sujeet More <sup>1</sup>, Jimmy Singla <sup>1</sup>, Ahed Abugabah,<sup>2</sup> and Ahmad Ali AlZubi<sup>3</sup>

<sup>1</sup>School of Computer Science and Engineering, Lovely Professional University, Kapurthala, India

<sup>2</sup>College of Technological Innovation, Zayed University, Dubai, UAE

<sup>3</sup>Computer Science Department, King Saud University, Riyadh, Saudi Arabia

Correspondence should be addressed to Jimmy Singla; jimmy.21733@lpu.co.in

Received 26 October 2020; Revised 16 November 2020; Accepted 27 November 2020; Published 7 December 2020

Academic Editor: Dr. Shahzad Sarfraz

Copyright © 2020 Sujeet More et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Magnetic resonance imaging (MRI) is precise and efficient for interpreting the soft and hard tissues. Moreover, for the detailed diagnosis of varied diseases such as knee rheumatoid arthritis (RA), segmentation of the knee magnetic resonance image is a challenging and complex task that has been explored broadly. However, the accuracy and reproducibility of segmentation approaches may require prior extraction of tissues from MR images. The advances in computational methods for segmentation are reliant on several parameters such as the complexity of the tissue, quality, and acquisition process involved. This review paper focuses and briefly describes the challenges faced by segmentation techniques from magnetic resonance images followed by an overview of diverse categories of segmentation approaches. The review paper also focuses on automatic approaches and semiautomatic approaches which are extensively used with performance metrics and sufficient achievement for clinical trial assistance. Furthermore, the results of different approaches related to MR sequences used to image the knee tissues and future aspects of the segmentation are discussed.

## 1. Introduction

Arthritis is one of the serious, prevalent joint diseases that cause disability and health issues in a large population. This arthritis is categorized with progressive degradation of joint tissues with a variety of abnormalities [1] and is a serious issue in recent years. Nearly sixty to seventy percent of people older than 60 years suffer from arthritis [2, 3]. Osteoarthritis is the damage of joint cartilage and leads to damage of functionality in the knee and hips, and the early signs can be observed with tears in cartilage. But rheumatoid arthritis is an autoimmune disease that largely affects the soft tissues around joints, bones, and cartilage. Therefore, the thickness and volume of the knee joint are the important parameters to evaluate rheumatoid arthritis.

Different imaging modalities can be deployed to estimate the quantitative measures of knee arthritis. But the ability of magnetic resonance imaging to provide the quality imaging

of bones with soft tissues, cartilage, and tendons as illustrated in Figure 1, for the diagnosis and treatment of diverse diseases, is extensively utilized. Magnetic resonance images provide information about damage and inflammation with more sensitivity than other modalities. The magnetic resonance images may contain hundreds of slices depending on the sample rate. The magnetic resonance images are a valuable instrument for the treatment and study of different ailments. The significant cost of labor, hours for analyzing a single scan by the radiologist, and insensitive for detection of progression of arthritis make the treatment more difficult, expensive, and inefficient.

Regardless of which disease is under study, the processing architecture includes a step known as segmentation for extracting the quantitative measures. The segmentation is a process of selecting the area of interest (AOI) concerning certain characteristics. Let  $X$ ,  $Y$ , and  $Z$  be a finite grid of  $p$ ,  $q$  pixels with  $Y$  set of intensities and  $Z$  objects, respectively.

$$\begin{aligned}
 X &= (x = (p, q): p = 0, 1, \dots, p-1, q = 0, 1, \dots, q-1), \\
 Y &= (0, 1, \dots, y-1), \\
 Z &= (0, 1, \dots, z-1).
 \end{aligned}
 \tag{1}$$

Let  $g: X \rightarrow Y$  and  $m: X \rightarrow Z$  denote the image and an area, respectively. The image segmentation divides an image  $g$  into  $Z$ -connected subimages.

This process of segmentation is important in providing the information about knee structure and the progress of the disease to the radiologist for diagnosis. However, this is a perilous and complex task for numerous reasons such as irregular shape, size, and connecting tissues. Therefore, many studies have focused on the progress of different methods to segment the knee magnetic resonance images [4, 5]. Although many studies on automatic and semiautomatic approaches are conducted, this segmentation from the magnetic resonance image is a future research problem [6] for the development of fully automated accurate and precise techniques. Thus, this scientific review provides a comprehensive knowledge of different computational methods utilized for the segmentation of magnetic resonance images. The review also focuses on exploring the challenges during segmentation.

Consequently, articles were screened for selection based on title and abstract. Only original papers that are published in journals and conferences are selected. Books, book chapters, reports, and thesis are excluded from selection criteria. Articles that are written in English and focused on rheumatoid arthritis disease are selected for research. Articles written in other than English and not related to other diseases are excluded. Articles using the machine learning approach are included. Articles that entirely use image processing methods are excluded from research.

This scientific article is organized as follows: Section 2 discusses the challenges faced in the segmentation of magnetic resonance images. Section 3 describes the various approaches followed by automated and semiautomated segmentation methods with their metrics in Section 4. Section 5 includes discussion on existing work reviewed in this article. Section 6 accomplishes the conclusion of the review article.

## 2. Challenges in Segmentation of Magnetic Resonance Image

In the recent past, several efforts have been made for the segmentation of magnetic resonance images with the progression of arthritis. Generally, every algorithm is evaluated with different parameters on the image. These algorithms work well with a few sets of data but do not provide efficient performance measures. This is because of different features associated with knee structure that result in difficulties with the segmentation of magnetic resonance images.

*2.1. Structure of Tissue.* The structure of tissues is measured in terms of thickness. In normal humans, the knee cartilage

density is about 2 mm to 4 mm with a curled surface with no blood vessels, and this density decreases at the rate of  $\sim 2.5\%$  to  $50\%$  [7, 8] every year in rheumatoid arthritis. These structures become thin and irregular in shape and are a challenging task in all slices. However, the edges between tibia and femur cartilage exhibit complications as represented in Figure 2(b), for providing reliable measures.

*2.2. Magnetic Field Strength.* The signal quantity in the mode of fissile quality is determined in magnetic resonance images. As the frequency of the magnetic field, strength 1.5 T, increases linearly, the signal-to-noise ratio (SNR) is amplified in magnetic resonance imaging [9]. Furthermore, there is proportionality between SNR and scan time. An increase in scan time increases the SNR value and often leads to artifacts in the image. For a high field, i.e., 3 T or 7 T strength, the visual quality and segmentation process of cartilage are improved. However, in 3 T magnetic strength, the T1-weighted images appear to be different than normal, and in 7 T strength, the T2-weighted images interrupt the scanning process and deploy more artifacts in the image leading to difficulty in finding a region of interest and other features [10].

*2.3. Image Artifacts and Properties.* MR imaging images are always liable to dissimilar types of artifacts. These relics are represented in the form of the pulse signal, affected volume, and chemical variations that precede an incorrect diagnosis. These artifacts are instigated by the magnetic field, resulting in uneven tissue representation. The chemical variations result in shady and bright spots on the edges of the tissues. These artifacts result in misinterpretation and may wrongly include the area of interest consisting of artifacts. The decreased quality of the edges due to volume effect and variations in signal intensities makes it difficult to develop computational approaches.

The local variation in tissue properties that exhibits several problems within an image is identified as low visibility areas. Figure 3 illustrates the variations and decreasing the edge quality of the knee magnetic resonance image.

As discussed, several issues and challenges need to be considered during the segmentation process for computational efficiency. The above challenges discussed are not only for automatic approaches but also for semiautomatic and manual approaches. The various segmentation techniques are reviewed and examined in the subsequent section.

## 3. Knee Tissue Segmentation Approaches

The knee bone is the major and largest bone within the human body and is the one that is most affected by rheumatoid arthritis disease. The cartilage and bone segments and features are important for the study of this rheumatoid arthritis disease. Recently, bone shape was proposed for predicting the rheumatoid arthritis progression [12]. This bone segmentation is a very critical assessment as connected cartilage degeneration is possible after the reconstruction of ACL [13]. In the class of approaches, the knee cartilage is

Inclusion criteria	Exclusion criteria
(i) Original articles and conference papers	(i) Books, book chapters, reports, and thesis
(ii) Articles published in English	(ii) Articles written in other than English
(iii) Disease related to only rheumatoid arthritis	(iii) Disease other than rheumatoid arthritis
(iv) Diagnosis using machine learning approach	(iv) Articles using completely image processing approach
(v) Articles only focused on diagnosis	(v) Articles related to treatment

FIGURE 1: Inclusion and exclusion criteria for selecting papers.

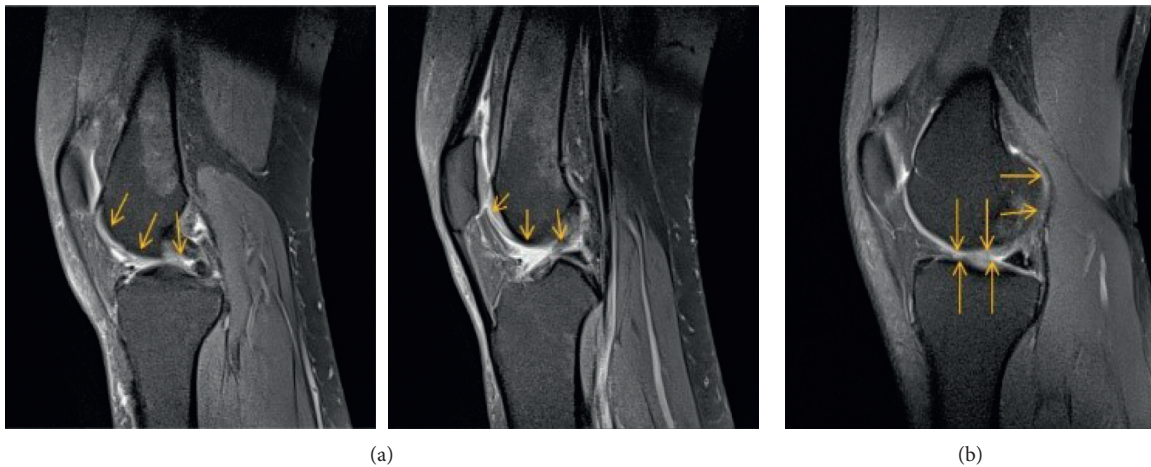


FIGURE 2: (a) Irregular cartilage tissue. (b) Diffused edges between the tibia and femur bone.

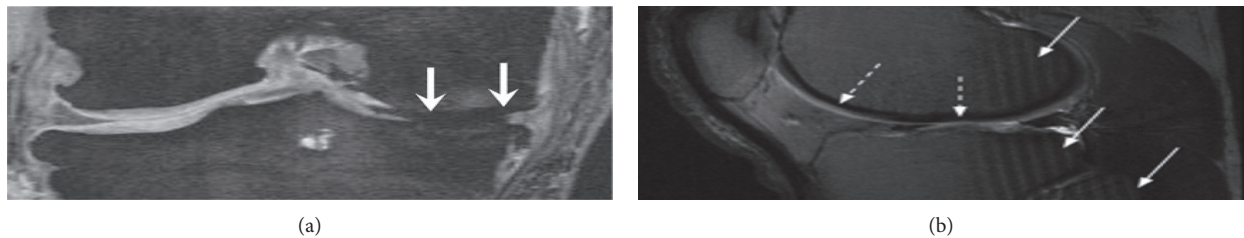


FIGURE 3: Artifacts in knee magnetic resonance image: (a) susceptibility and (b) artifacts produced by chemical (dashed arrow) and variation artifact (arrow) (reproduced from reference [11]).

physically or semiautomatically segmented, and area of interest (AOI) is estimated. There are numerous approaches for segmentation of bone and cartilage that are reviewed in this section, as illustrated in Figure 4.

### 3.1. Manual and Traditional Computational Approaches.

It is often seen that the traditional and manual methods used for segmentation result in reliable outcomes. In these traditional and manual methods, the tissues are manually segmented slice by slice from magnetic resonance images. Though the precision, sensitivity, and specificity of the

manual methods are considered as the gold standard and high compared to automatic and semiautomatic techniques, it requires time-consuming efforts by the experts and provides inter- or intraobserver inconsistency by different professionals [14]. These manual techniques are not practically utilized for clinical trials, for example, Figure 5 representing femur bone which is manually segmented.

The manual strategy for the identification of segmented regions from the medical image is implemented with high precision. A statistical model [15] is developed to solve the issues related to knee image segmentation. This method used 60 images for identification of knee thickness with an

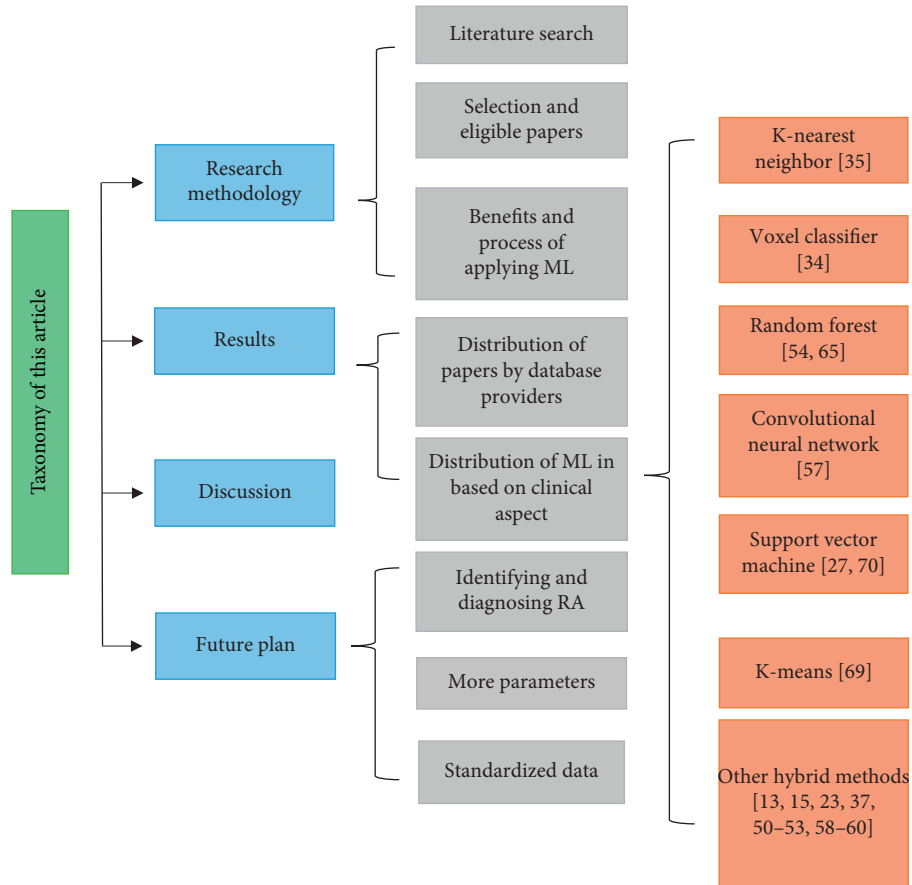


FIGURE 4: Taxonomy used in this article.



FIGURE 5: Manual segmentation of femur bone.

extreme and tiniest density of the cartilage. The method was used on the public dataset with 150 knee images [16], and mostly the data consisted of T1-weighted images with an

index of  $\sim 73\%$  and  $\sim 74\%$  of femur and tibia segments, respectively.

**3.1.1. Region-Based Technique.** A region or area is poised of pixels of the neighbours, and the boundary is estimated by the difference between two regions. In this article, we discuss the most prevalent region-based techniques region growing and thresholding and edge-based method [17]. The initialization and process of segmentation start with some seed points. This process divides the regions based on the intensity of the pixel. The process requires only one seed point and homogenous properties of neighbouring pixels. Figure 6 represents the region growing technique, where two seed points are selected to segment the tissue.

**3.1.2. Threshold or Intensity-Based Technique.** The simplest and speediest method for segmentation is thresholding. The method assumes the regions with a diverse gray level. The diverse parts of the image are identified with histogram intensity. The intensity is divided into twofold: the first portion known as the foreground is having the pixel intensity higher than or equal to a threshold, second portion known as a background having the pixel intensity less than the threshold value as shown in equation:



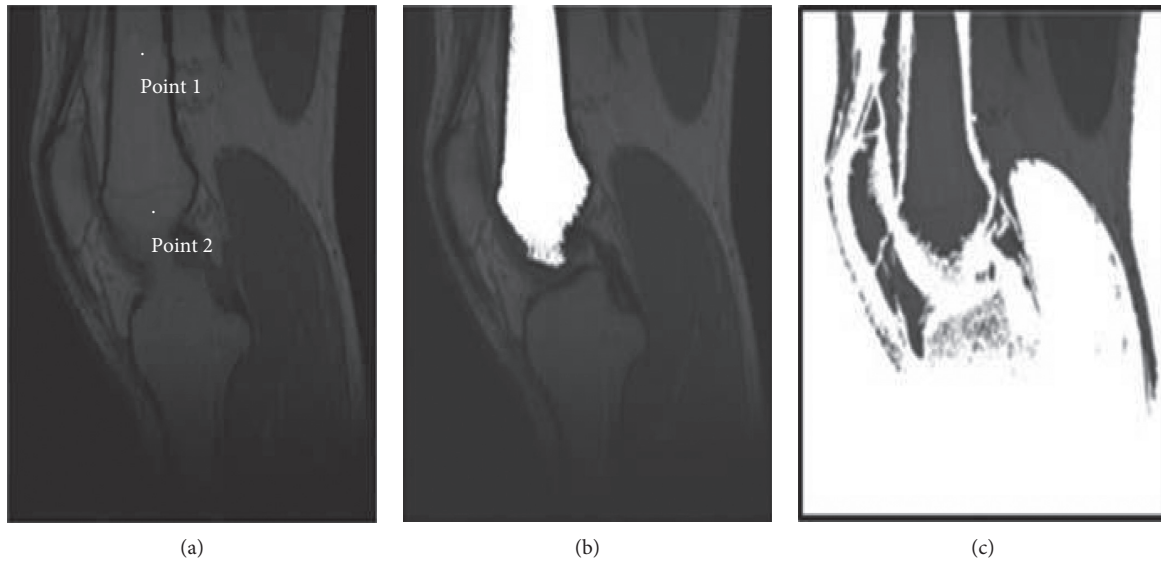


FIGURE 6: (a) Initial image with two seed points. (b) Outcome of region growing for one seed point. (c) Outcome of region growing for two seed points.

$$f(x, y) = \begin{cases} \text{foreground } g(x, y) \geq P \\ \text{background } g(x, y) < P \end{cases}, \quad (2)$$

where  $f(x, y)$  is the pixel strength at  $(x, y)$  location and  $P$  is the threshold value. If the threshold value is inappropriate, then it leads to inaccurate segmentation outcomes [18]. This thresholding parameter occurs as a problem in magnetic resonance images and makes an image partition complex.

The global thresholding does not provide better results for some types of images, like the images that do not have low contrast and low pixel intensity across the image. For these types of images, the thresholding provides a better outcome in some parts of the image and fails in other parts of the image [19]. In the local thresholding, the image is partitioned into vertical and horizontal lines. The local thresholding needs more time compared to global thresholding.

**3.1.3. Edge-Based Technique.** The edges or boundaries are relevant information in the image. The incidence of the image is known by the number of changes in pixel strength that is utilized for identifying the different regions of those changes [20]. The automatic classification of boundaries is achieved by some high filters that avoid low-intensity data (uniform data) and preserves high-intensity data (edge data). These edges are extracted for utilization in sophisticated approaches [21]. The tissues appear to be weak or break that are not distinguishable by some methods with edge-based region and distance information.

**3.1.4. Atlas or Graph-Based Technique.** The gray pixels of the image are utilized to create a graph with gray indices [22]. The atlas and graphs are designed manually by experts who assign labels to the tissue structures. The segmentation process is applied for transferring the image information

from labelled data to subjects. Hence, the efficiency of the image segmentation relies on image registering and resemblance between subjects and atlas. The multiatlas method [23, 24] that is a sequence of multiple graphs is used to collect the information in terms of color, structure, and texture of the tissues with gray values. Furthermore, the image is partitioned into several multiple small regions and segmented with foreground and background areas.

A three label approach that utilizes the atlas approach [25] for bone and cartilage segmentation is applied and evaluated on knee images. This study extensively described the performance of different classifiers, and the mean Jaccard value is 75.3% and 75.6% for femur and tibia bones. The graph method proposed in [26] offers a subject extraction with dimensionality where the weights of edges are related to boundary properties. The method evaluated the images obtained from the OAI database with a better performance parameter.

**3.2. Fuzzy-Based Approaches.** The fuzzy  $c$ -means approach provides a better classification of medical images. The disease is being classified into four levels based on severity which is introduced in [27]. The knee arthritis is categorized as normal, minimal, doubtful, and moderate as per the standard practice of the KL score. The morphometric measures of the knee assigned as osteoarthritis are investigated with image processing methods. The outcomes intensify and reinforce the traditional approaches with the harshness of the application. The tissues such as gray-level and white-level intensified are segmented with bias-filed corrected fuzzy  $C$ -means in [28]. A combination of the level-set and fuzzy method clusters the gray and white matter from the magnetic resonance images.

The graphical processing units are providing fast computation along with the fuzzy method that is implemented in [29]. A large dataset can be processed by accelerating the

fuzzy *c*-means with GPU to tackle the computational time around 2.24 times faster. The specific region of interest (ROI) is extracted, and the active contour of the cartilage with noninterpolated areas is estimated in [30]. The identical pixels are assigned with some classes by using the membership function to detect the pathological changes in the tissue structure of the knee magnetic resonance image.

**3.3. Machine Learning Approaches.** In the recent past, a great effort has been employed on machine learning practices for solving the segmentation problem [31]. The unsupervised learning approach does not need training or labelling the data. The labelling of the image is accomplished by exploiting shapes in the voxel strength features. The supervised learning approach requires training for the algorithm and learns from the voxel examples. The support vector machines with multiple sets of images [32] are used for the evaluation of knee cartilage with better sensitivity. 36 different dimensional features are selected and experimented on 4 different classifiers with an AUC of 76%. The hidden biomedical information is analyzed clearly for the progression of arthritis.

The subcortical segmentation method is used to construct the graph [33]. The reliable and automated segmentation of structures from magnetic resonance images is more important for shape analysis and volumetric study. The random forest classifier is used for assigning a cost for each node in the graph and evaluates the dice index and putamen measures. A method based on the K-NN strategy [34] showed successful segmentation for cartilage. The cartilage is segmented from the background that leads to false-positive values. Furthermore, the study was modified and presented with SVM [35]. The main difficulty of SVM is the distribution and independence of data instances. To incorporate the contextual data such as intensity and structural information, the ML approach builds feature vector [36] with simultaneous use of multiple images. The method utilizes the T1-weighted images for cartilage segmentation problems that are multicontrast and deposited with fat and water gradient. A method based on pattern recognition [32] for gradient data of the knee cartilage segment reports averages dice value to 0.76.

**3.4. Deep Learning Approaches.** A priori knowledge of shape analysis using a convolution neural network [37] is proposed that incorporates the SSM. The method uses 40 validation and 50 subjects for the estimation of knee images. The voxel accuracy of 89% is estimated with manual adjustment, and the correlation between the developed method and the ground truth is investigated. The automated segmentation is based on the CNN approach [38]. This method applies volumetric structure on images and manually segments the magnetic resonance images. The automatic segmentation of images achieved better outcomes with a dice similarity score of 0.95 and a precision of 0.95 on femur tissue.

The contextual constrained neural network combined with level set evolution [39] is used for the segmentation of knee magnetic resonance images. The movement of the

patella in the knee T2-weighted image should be normal, so tracking this with segmenting the femur and patella is required. The performance of the T1-weighted knee magnetic resonance images in terms of ASD and RMSD is estimated with the convolution network proposed in [40]. The deep learning approach has some promising issues, such as high dependency on the quality and amount of training data that tends to overfit of data [41].

The 3D deformable approach introduced in [42] uses CNN with 3D simplex deformable modeling. The method performs pixelwise, multiclass classification which has been tested on the public knee image dataset. This method provides state-of-the-art performance with superior accuracy and segmentation error of VOE. An extended version of the method is proposed in [43] for segmenting the cartilage lesion detection. The T2-weighted fast spin-echo magnetic resonance images of 175 subjects are used with CNN. The ROC and *k* statistics were used for analyzing the performance and intraobserver detection of cartilage lesions. The volumetric assessment of knee cartilage is introduced in [44]. The dice score and VOE are estimated for different architectures using the CNN approach.

## 4. Performance of Segmentation Approaches

Most of the studies that involve segmentation of the cartilage have been evaluated by estimating the performance parameters against the ground truth. This ground truth is being prepared using manual extraction methods by an expert. To estimate the efficiency of progressive methods for estimating the cartilage boundaries, different metrics such as sensitivity, specificity, reliability, dice similarity coefficient (DSC), accuracy, and efficiency are considered from [6, 45, 46]. Based on the ground truth and developed methods, the sensitivity, specificity, DSC, and accuracy are measured using equations (3)–(6), respectively:

$$\text{sensitivity} = \frac{TP}{TP + FN}, \quad (3)$$

$$\text{specificity} = \frac{TN}{TN + FP}, \quad (4)$$

$$\text{DSC} = \frac{2TP}{((TP + FP) + (FN + TP))}, \quad (5)$$

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (6)$$

where TP is true positive, i.e., area correctly labelled as cartilage area, TN is true negative, i.e., area correctly labelled as noncartilage area, FP is false positive, i.e., area incorrectly labelled as cartilage area, and FN is false negative, i.e., area incorrectly labelled as noncartilage area. These parameters are estimated with base values over the segmented areas.

Sensitivity is the parameter for a ratio of true positives and measures approximate to 100% for accurate classification of cartilage areas of ground truth and developed methods. In this parameter, the total quantity of small false negatives represents high sensitivity. The specificity

parameter provides the ratio of true negatives that are appropriately identified through comparative analysis. A combined effect of both sensitivity and specificity reflects the best efficiency of the developed approach. Accurateness is a measure for representing the ground truth agreeing with the developed methods. It shows how many correctly classified cartilage areas are included and noncartilage areas are excluded. Other performance parameters are also considered by many researchers.

*4.1. Automatic Segmentation Approaches.* Automatic segmentation from magnetic resonance image is being achieved for widespread studies in rheumatoid arthritis, which makes the process that requires to be accomplished fast and consistent. Research provides more sophisticated methods to segment the images and moves towards an entirely automatic process. Once the tissues are segmented, the quantitative or measurable analysis is carried out by estimating the volume, density, and structure of the tissues. Most of the automated techniques that are grounded on voxel arrangement require a large number of training dataset. The most used machine learning approach is K-nearest neighbour applied with a multivariant classifier.

Several studies have discussed the utilization of voxel classification to segment the tissues. Although these voxel methods can segment the magnetic resonance images, the data required is the primary concern. The main limitation of this kind of method is the generation of a new dataset. The automatic methods used by different researchers with different metrics are demonstrated in Table 1. All methods discussed in the table are utilizing intensity or pixel for the computation. Generally, image segmentation can be achieved by using numerous parameters such as edges, regions, clusters, and image intensity. The threshold-based segmentation approach is limited by tissue properties such as (a) varying intensities of cartilage region, (b) low signal intensity, and (c) low visibility of tissues. The development of these techniques is still under research, and a lot yet has to be discovered. Current developments in the area of segmentation have been aimed at multiple pulse sequence of magnetic resonance images.

To automatically segment the tissues from the magnetic resonance image, the successive approach is performed with automated segmentation by extracting probabilities using the k-NN classifier. The statistical shape and models are more precise segmentation methods. However, the utilization of these methods has some limitations due to the initialization of sensitivity [62].

*4.2. Semiautomatic Segmentation Approaches.* This approach requires minimal user intervention during the segmentation process. These approaches are considered for reducing the efforts required for manual segmentation. These approaches with the need for human interaction complete some tasks of pattern recognition and image processing. The semiautomated approach uses some of the methods such as snake contours, gradient vectors, watershed, graph search, and region growing for segmentation. As

depicted in Table 2, the methods used for semiautomatic segmentation have resulted in great success in extracting the soft and hard tissue features. A repetitive averaging filter is used for replicating Gaussian function in the deblurring process. To dehaze the images by segmentation using the super pixels technique [72], intensities are summed and the transmission map is estimated. This method preserves the features such as texture and edges of real-time images.

The semiautomated approach requires less computational time and provides an accurate outcome in the studies when compared with manual techniques, but includes the human interaction, viewer inconsistency that needs to be performed. Figure 7 represents the femur and tibia tissue segmentation using the atlas-based approach for sagittal and axial images. The method discussed in [14] provides a multicontrast image that has a rich set of the feature given for classifier using support vector machine technique. Furthermore, besides the local features, the global features are also included in the anatomical direction of femur and tibia bones. SVM is extended, and new SVM-DRF includes the independent estimation of pixels to incorporate the spatial dependency between neighbouring pixels.

## 5. Discussion

The assessment of tissues is the crucial stage in assessing the progression of rheumatoid arthritis. The segmentation process is not only needed for treatment but also used for quantitative parameter estimation. This has become a need due to complex variations related to different tissues in the knee magnetic resonance image. The segmentation techniques discussed above are broadly classified as traditional or manual, machine learning approach, deep learning approach, automatic approach, and semiautomatic approach. Manual segmentation is a tedious task and has result variability amongst professionals within the same image dataset. Semiautomatic segmentation is a less complex task compared to manual segmentation but is the same in terms of variations of manual segmentation. The intraobserver reproducibility for the density is measured in several areas of cartilage [73]. The discussed techniques are included such as region growing, active contour, and graph-cut. All these techniques require less human intervention and provide better sensitivity, specificity, and dice similarity scores. To overcome the drawback of inter- or intraobserver, many researchers came up with automated segmentation approaches that are discussed and reviewed in Section 4.

Table 3, provides a comparative analysis of other articles and this article. This paper shows an indetail insight selecting all the features related to rheumatoid arthritis. In Table 3, this review work selects synovial fluid and meniscus that is the main source for rheumatoid arthritis. The use of magnetic resonance images and machine learning approaches is the key parameter for selecting the review articles. This review gives an overview of different techniques utilized for the segmentation of tissues with resolving major challenges in the magnetic resonance image. The techniques available require more time for computation for segmentation of the

TABLE 1: Automated segmentation approaches.

Study	MR sample	Number of subjects	Methods used	Efficiency measures
[47]	3D DESS	33 subjects	Region growing and seed selection	DSC 82.8%, 83.1%, and 72.6% for femur, tibia, and patella, respectively DSC 0.80
[15]	T1-weighted 3D DESS	20 subjects	Side selection and coupled cartilage segmentation	Sensitivity 90.0 Specificity 99.8
[48]	3D DESS	20 subjects	Spatial fuzzy C-mean	DSC of femur, tibia, and patella are 87.1, 81.1, and 84.8, respectively
[37]	DESS	50 subjects	Statistical shape model	Accuracy $74 \pm 5$ DSC 86.1%
[49]	2D scan	100 subjects	Random walker algorithm	DSC 0.8758 DSC 90.28
[50]	DESS	8 subjects	DRLSE algorithm	Sensitivity 91.14 Specificity 99.12 Sensitivity 84.2 Specificity 99.9 Dice volume 0.81
[35]	3D DESS	10 subjects	K-nearest neighbor	DSC 0.80 $\pm$ 0.04
[34]	3D image	139 subjects	Voxel classification	
[51]	DESS	13 subjects	Hierarchical classifier and random forest	Surface positioning errors femur $0.03 \pm 0.19$ tibia $0.10 \pm 0.17$
[52]	T1-weighted 3D SPGR and T2-weighted GRE	9 subjects	Three-label segmentation	Median dice coefficient femur 0.935 and tibia 0.938
[53]	3D SPGR	155 subjects	Longitudinal three label segmentation	Femur DSC 75% Sensitivity 78% Specificity 99.9%
[54]	Sagittal DESS	88 subjects	Convolution neural network	DSC 83.8%
[55]	DESS	40 subjects	Active appearance model (statistical model)	RMSE for femur 1.49, tibia 1.21
[56]	PDW TSE	14 subjects	Distance-weighted directional gradient	Intensity level, 2~4
[57]	T1-weighted	30 subjects	Multiatlas segmentation constrained graph	Femur average surface distance 0.36 mm
[58]	2D ASM	140 subjects	Rigid multiatlas registration	Dice volume tibia 0.8 and femur 0.87
[59]	T1-weighted SPGR	20 subjects	Statistical method	DSC of femur 0.871, tibia 0.852, and patella 0.645. Sensitivity of femur 0.947, tibia 0.949, and patella 0.909. Specificity of femur 0.988, tibia 0.993, and patella 0.996.
[60]	T1-weighted FS	14 subjects	Validation of statistical method	Meniscus DSC 0.75, sensitivity 0.72, and specificity 1.00
[61]	T1-weighted	100 subjects	Coarse to fine method with min-cut	DSC of femur 0.947, and tibia 0.968

TABLE 2: Semiautomated segmentation approaches.

Study	MR sample	Number of subjects	Methods used	Efficiency measures
[63]	PDW SPAIR	12 subjects	Random forest	Femur DSC 94.9% Tibia DSC 92.5%
[27]	T1-weighted axial	103 subjects	Support vector machine	Accuracy 72%
[58]	CCBR	159 subjects	PLDS method	Dice volume 0.82
[64]	3D SGPR	4 subjects	Region growing	Error -6.53%
[65]	T1-weighted image	5 subjects	Active shape model	Mean error -0.57
[66]	Flash 3D	15 subjects	B-spline with manual adjustment	Interobserver 3.3 to 13.6
[45]	3DMR	20 subjects	Active contour	Thickness 0.996 and 0.998 for femur and tibia DSC 89.5%
[4]	SPGR	7 subjects	Watershed transformation	Sensitivity 90% Specificity 99.9%

TABLE 2: Continued.

Study	MR sample	Number of subjects	Methods used	Efficiency measures
[23]	3D DESS	320 slices	Graph-cut algorithm	DSC 94.3% DSC 0.77 and 0.80
[67]	Flash GRE	50 subjects	K-means with manual adjustment	Sensitivity 83.1 and 85.3 Specificity 99.9 and 099.9 for femur and tibia bones, respectively
[68]	DESS	17 subjects	Support vector machine	DSC patella 0.82, tibia 0.83, and femur 0.86
[23]	3T MR images	10 subjects	Graph-cut method	DSC 0.943
[13]	3D DESS	12 subjects	Active contour model	Root mean square 0.8% to 1.3%
[69]	3D DESS	10 subjects	Mesh morphing approach	Mean S.D of femur 0.87, tibia 0.40 and, patella 0.53
[70]	T1-weighted	15 subjects	Watershed method	Cartilage volume tibia 3.3 mm

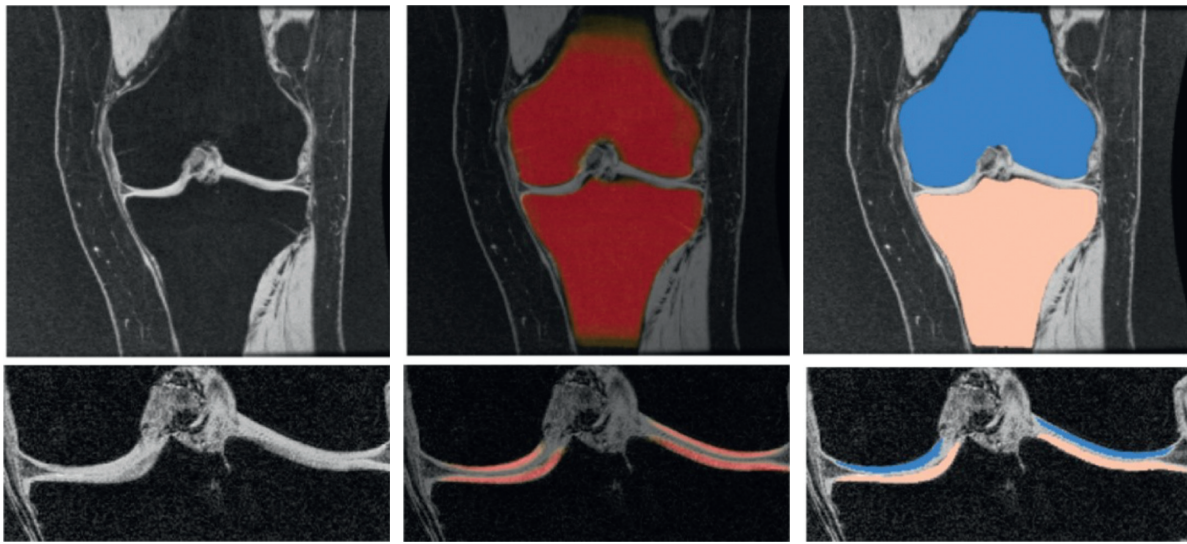


FIGURE 7: First row: segmentation of knee bone (left: original image, middle: automated segmentation, and right: segmented bone). Second row (left: original image, middle: automated segmentation, and right: cartilage segmented) (reproduced from reference [25]).

TABLE 3: Comparative analysis with other published review articles.

Article	Datatype considered	Medical disease considered		Classification and summarization of existing work			
	Magnetic resonance imaging	Machine learning	Cartilage	Meniscus	Ligaments and lesions	Patella, tibia, and femur bone	Synovial fluid
[74]	√	√	X	X	√	X	X
[75]	√	√	√	X	X	X	X
[76]	√	√	X	√	X	X	X
[77]	X	√	X	X	X	X	√
[78]	√	√	X	X	X	√	X
This article	√	√	√	√	√	√	√

entire knee magnetic resonance image. Some of the automated and semiautomated approaches discussed above are highlighted with the main strengths and drawbacks. However, the growing rate of data and increasing rate of computational power make the machine learning and deep learning approaches most promising for future applications concerning IoT applications [79–83].

## 6. Conclusion

This systematic review approximates the magnetic resonance imaging key parameters reminiscent of rheumatoid arthritis. Our nature of the study selects sensitivity, specificity, and dice similarity index for an approximation of imaging results. However, from the results of our study, more frequent

parameters for rheumatoid arthritis disease such as synovial fluid volume, meniscus volume, the dissimilar structure of tibia, femur, and tears of ligaments. Different segmentation techniques are considered for review with pros and cons of themselves that may increase the productivity of a novel hybrid approach.

Our future work includes selecting more parameters that are directly related to rheumatoid arthritis. The synovial fluid and cystic lesions are the less considered features than the cartilage and other bone tissues. These considered features may be used in the future for targeted treatment, in those most required for rheumatoid arthritis diagnosis.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by Zayed University, Office of Research under Grant no. R18088.

## References

- [1] S. Castañeda, J. A. Roman-Blas, R. Largo, and G. Herrero-Beaumont, "Subchondral bone as a key target for osteoarthritis treatment," *Biochemical Pharmacology*, vol. 83, no. 3, pp. 315–323, 2012.
- [2] D. Lubar, P. H. White, L. F. Callahan et al., "A national public health agenda for osteoarthritis 2010," *Seminars in Arthritis and Rheumatism*, vol. 39, no. 5, pp. 323–326, 2010.
- [3] A. O. Akinpelu, T. O. Alonge, B. A. Adekanla, and A. C. Odole, "Prevalence and pattern of symptomatic knee osteoarthritis in Nigeria: a community-based study," *The Internet Journal of Allied Health Sciences and Practice*, vol. 7, no. 3, pp. 1–7, 2009.
- [4] V. Grau, A. U. J. Mewes, M. Alcañiz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, 2004.
- [5] B. Zhang, "Computer-aided knee joint magnetic resonance image segmentation—a survey," 2018, <https://arxiv.org/abs/1802.04894>.
- [6] D. Kumar, A. Gandhamal, S. Talbar, and A. F. M. Hani, "Knee articular cartilage segmentation from MR images: a review," *ACM Computing Surveys*, vol. 51, no. 5, p. 1, 2019.
- [7] B. S. Bentley and R. V. Hill, "Assessing macroscopic and microscopic indicators of osteoarthritis in the distal interphalangeal joints: a cadaveric study," *Clinical Anatomy*, vol. 20, no. 7, pp. 799–807, 2007.
- [8] T. G. Williams, A. P. Holmes, M. Bowes et al., "Measurement and visualisation of focal cartilage thickness change by MRI in a study of knee osteoarthritis using a novel image analysis tool," *The British Journal of Radiology*, vol. 83, no. 995, pp. 940–948, 2010.
- [9] P. R. Kornaat, S. B. Reeder, S. Koo et al., "MR imaging of articular cartilage at 1.5 T and 3.0 T: comparison of SPGR and SSFP sequences," *Osteoarthritis and Cartilage*, vol. 13, no. 4, pp. 338–344, 2005.
- [10] R. Stahl, R. Krug, D. A. C. Kelley et al., "Assessment of cartilage-dedicated sequences at ultra-high-field MRI: comparison of imaging performance and diagnostic confidence between 3.0 and 7.0 T with respect to osteoarthritis-induced changes at the knee joint," *Skeletal Radiology*, vol. 38, no. 8, pp. 771–783, 2009.
- [11] P. Lang, F. Noorbakhsh, and H. Yoshioka, "MR imaging of articular cartilage: current state and recent developments," *Radiologic Clinics of North America*, vol. 43, no. 4, pp. 629–639, 2005.
- [12] A. Arovitola and L. Gallo, "Knee bone segmentation from MRI: a classification and literature review," *Biocybernetics and Biomedical Engineering*, vol. 36, no. 2, pp. 437–449, 2016.
- [13] M. H. Brem, P. K. Lang, G. Neumann et al., "Magnetic resonance image segmentation using semi-automated software for quantification of knee articular cartilage-initial evaluation of a technique for paired scans," *Skeletal Radiology*, vol. 38, no. 5, pp. 505–511, 2009.
- [14] K. Zhang, W. Lu, and P. Marziliano, "Automatic knee cartilage segmentation from multi-contrast MR images using support vector machine classification with spatial dependencies," *Magnetic Resonance Imaging*, vol. 31, no. 10, pp. 1731–1743, 2013.
- [15] H. Seim, D. Kainmueller, H. Lamecker, M. Bindernagel, J. Malinowski, and S. Zachow, "Model-based auto-segmentation of knee bones and cartilage in MRI data," in *Proceedings of the 13th International Conference on Medical Image Computing and Computer Assisted Intervention*, Beijing, China, 2010.
- [16] T. Heimann and B. Morrison, "Segmentation of knee images: a grand challenge," in *Proceedings of the 2010 MICCAI Workshop on Medical Image Analysis for the Clinic*, Beijing, China, 2010.
- [17] A. Norouzi, M. S. M. Rahim, A. Altameem et al., "Medical image segmentation methods, algorithms, and applications," *IETE Technical Review*, vol. 31, no. 3, pp. 199–213, 2014.
- [18] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.
- [19] A. Rad, M. S. Mohd Rahim, A. Rehman, A. Altameem, and T. Saba, "Evaluation of current dental radiographs segmentation approaches in computer-aided applications," *IETE Technical Review*, vol. 30, no. 3, pp. 210–222, 2013.
- [20] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, "LOGISMOS—layered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint," *IEEE Transactions on Medical Imaging*, vol. 29, no. 12, pp. 2023–37, 2010.
- [21] M. S. Swanson, J. W. Prescott, T. M. Best et al., "Semi-automated segmentation to assess the lateral meniscus in normal and osteoarthritic knees," *Osteoarthritis and Cartilage*, vol. 18, no. 3, pp. 344–353, 2010.
- [22] J.-G. Lee, S. Gumus, C. H. Moon, C. K. Kwok, and K. T. Bae, "Fully automated segmentation of cartilage from the MR images of knee using a multi-atlas and local structural analysis method," *Medical Physics*, vol. 41, no. 9, Article ID 092303, 2014.
- [23] H. Shim, S. Chang, C. Tao, J.-H. Wang, C. K. Kwok, and K. T. Bae, "Knee cartilage: efficient and reproducible segmentation on high-spatial-resolution MR images with the semiautomated graph-cut algorithm method," *Radiology*, vol. 251, no. 2, pp. 548–556, 2009.
- [24] L. Shan, C. Charles, and M. Niethammer, "Automatic multi-atlas-based cartilage segmentation from knee MR images," in *Proceedings of the 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, Barcelona, Spain, 2012.

- [25] L. Shan, C. Zach, C. Charles, and M. Niethammer, "Automatic atlas-based three-label cartilage segmentation from MR knee images," *Medical Image Analysis*, vol. 18, no. 7, pp. 1233–1246, 2014.
- [26] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [27] A. Alkan, "Analysis of knee osteoarthritis by using fuzzy c-means clustering and SVM classification," *Scientific Research and Essays*, vol. 6, no. 20, pp. 4213–4219, 2011.
- [28] P. Agarwal, S. Kumar, R. Singh, P. Agarwal, and M. Bhattacharya, "A combination of bias-field corrected fuzzy C-means and level set approach for brain MRI image segmentation," in *Proceedings of the 2015 2nd International Conference on Soft Computing and Machine Intelligence (ISCMCI)*, Hong Kong, China, 2016.
- [29] M. Al-Ayyoub, A. M. Abu-Dalo, Y. Jararweh, M. Jarrah, and M. A. Sa'd, "A GPU-based implementations of the fuzzy C-means algorithms for medical image segmentation," *The Journal of Supercomputing*, vol. 71, no. 8, pp. 3149–3162, 2015.
- [30] J. Kubicek and M. Penhaker, "Fuzzy algorithm for segmentation of images in extraction of objects from MRI," in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Delhi, India, 2014.
- [31] S. Koo, B. Hargreaves, T. Andriacchi, and G. Gold, "Automatic segmentation of articular cartilage from MRI: a multi-contrast and multi-dimensional approach," in *Proceedings of the 16th Scientific Meeting of the International Society for Magnetic Resonance in Medicine*, Toronto, Canada, 2008.
- [32] Y. Du, R. Almajalid, J. Shan, and M. Zhang, "A novel method to predict knee osteoarthritis progression on MRI using machine learning methods," *IEEE Transactions on Nano-Bioscience*, vol. 17, no. 3, pp. 228–236, 2018.
- [33] Z. Guo, S. Kashyap, M. Sonka, and I. Oguz, "Machine learning in a graph framework for subcortical segmentation," *Medical Imaging 2017: Image Processing*, vol. 10133, 2017.
- [34] J. Folkesson, E. B. Dam, O. F. Olsen, P. C. Pettersen, and C. Christiansen, "Segmenting articular cartilage automatically using a voxel classification approach," *IEEE Transactions on Medical Imaging*, vol. 26, no. 1, pp. 106–115, 2007.
- [35] J. Folkesson, O. F. Olsen, P. Pettersen, E. Dam, and C. Christiansen, "Combining binary classifiers for automatic cartilage segmentation in knee MRI," *Lecture Notes in Computer Science*, Springer, vol. 3765, pp. 230–239, Berlin, Germany, 2005.
- [36] K. Zhang, J. Deng, and W. Lu, "Segmenting human knee cartilage automatically from multi-contrast MR images using support vector machines and discriminative random fields," in *Proceedings of the 2011 18th IEEE International Conference on Image Processing*, Brussels, Belgium, 2011.
- [37] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: data from the osteoarthritis initiative," *Medical Image Analysis*, vol. 52, pp. 109–118, 2019.
- [38] C. M. Deniz, S. Xiang, R. S. Hallyburton et al., "Segmentation of the proximal femur from MR images using deep convolutional neural networks," *Scientific Reports*, vol. 8, no. 1, pp. 1–14, 2018.
- [39] H.-C. Lan, T.-R. Chang, W.-C. Liao, Y.-N. Chung, and P.-C. Chung, "Knee MR image segmentation combining contextual constrained neural network and level set evolution," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Nashville, TN, USA, 2009.
- [40] H. Lee, H. Hong, and J. Kim, "BCD-NET: a novel method for cartilage segmentation of knee MRI via deep segmentation networks with bone-cartilage-complex modeling," in *Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, USA, April 2018.
- [41] J.-G. Lee, S. Jun, Y.-W. Cho et al., "Deep learning in medical imaging: general overview," *Korean Journal of Radiology*, vol. 18, no. 4, pp. 570–584, 2017.
- [42] F. Liu, Z. Zhou, H. Jang, A. Samsonov, G. Zhao, and R. Kijowski, "Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging," *Magnetic Resonance in Medicine*, vol. 79, no. 4, pp. 2379–2391, 2018.
- [43] F. Liu, Z. Zhou, A. Samsonov et al., "Deep learning approach for evaluating knee MR images: achieving high diagnostic performance for cartilage lesion detection," *Radiology*, vol. 289, no. 1, pp. 160–169, 2018.
- [44] A. Raj, S. Vishwanathan, B. Ajani, K. Krishnan, and H. Agarwal, "Automatic knee cartilage segmentation using fully volumetric convolutional neural networks for evaluation of osteoarthritis," in *Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, USA, 2018.
- [45] C. Kauffmann, P. Gravel, B. Godbout et al., "Computer-aided method for quantification of cartilage thickness and volume changes using MRI: validation study using a synthetic model," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 8, pp. 978–988, 2003.
- [46] J. Tang, S. Millington, S. T. Acton, J. Crandall, and S. Hurwitz, "Surface extraction and thickness measurement of the articular cartilage from MR images using directional gradient vector flow snakes," *IEEE Transactions on Bio-Medical Engineering*, vol. 53, no. 5, pp. 896–907, 2006.
- [47] C. N. Öztürk and S. Albayrak, "Automatic segmentation of cartilage in high-field magnetic resonance images of the knee joint with an improved voxel-classification-driven region-growing algorithm using vicinity-correlated subsampling," *Computers in Biology and Medicine*, vol. 72, pp. 90–107, 2016.
- [48] C. Ahn, T. D. Bui, Y.-w. Lee, J. Shin, and H. Park, "Fully automated, level set-based segmentation for knee MRIs using an adaptive force function and template: data from the osteoarthritis initiative," *BioMedical Engineering Online*, vol. 15, no. 1, pp. 1–14, 2016.
- [49] P. R. Desai and I. Hacihaliloglu, "Enhancement and automated segmentation of ultrasound knee cartilage for early diagnosis of knee osteoarthritis," in *Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, USA, April 2018.
- [50] A. Gandhamal, S. Talbar, S. Gajre, R. Razak, A. F. M. Hani, and D. Kumar, "Fully automated subchondral bone segmentation from knee MR images: data from the osteoarthritis initiative," *Computers in Biology and Medicine*, vol. 88, pp. 110–125, 2017.
- [51] S. Kashyap, H. Zhang, K. Rao, and M. Sonka, "Learning-based cost functions for 3-D and 4-D multi-surface multi-object segmentation of knee MRI: data from the osteoarthritis

- initiative," *IEEE Transactions on Medical Imaging*, vol. 37, no. 5, pp. 1103–1113, 2018.
- [52] L. Shan, C. Zach, and M. Niethammer, "Automatic three-label bone segmentation from knee MR images," in *Proceedings of the 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Rotterdam, Netherlands, 2010.
- [53] L. Shan, C. Charles, and M. Niethammer, "Longitudinal three-label segmentation of knee cartilage," in *Proceedings of the 2013 IEEE 10th International Symposium on Biomedical Imaging*, San Francisco, CA, USA, 2013.
- [54] A. Tack, A. Mukhopadhyay, and S. Zachow, "Knee menisci segmentation using convolutional neural networks: data from the osteoarthritis initiative," *Osteoarthritis and Cartilage*, vol. 26, no. 5, pp. 680–688, 2018.
- [55] G. Vincent, C. Wolstenholme, I. Scott, and M. Bowes, "Fully automatic segmentation of the knee joint using active appearance models," *Medical Image Analysis for the Clinic: A Grand Challenge*, CreateSpace, Scotts Valley, CA, USA, 2010.
- [56] F. Huang, X. Chen, D. Ye, and S. Hertel, "A distance weighted directional gradient method for fully automatic bone segmentation of knee MRI," in *Proceedings of the 16th Scientific Meeting of the International Society of Magnetic Resonance in Medicine*, Toronto, Canada, 2008.
- [57] C. Chu, J. Bai, X. Wu, and G. Zheng, "MASCg: multi-atlas segmentation constrained graph method for accurate segmentation of hip CT images," *Medical Image Analysis*, vol. 26, no. 1, pp. 173–184, 2015.
- [58] E. B. Dam, "Simple methods for scanner drift normalization validated for automatic segmentation of knee magnetic resonance imaging—with data from the osteoarthritis initiative," 2017, <https://arxiv.org/abs/1712.08425>.
- [59] J. Fripp, S. Ourselin, S. K. Warfield, and S. Crozier, "Automatic segmentation of the bones from MR images of the knee," in *Proceedings of the 2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 336–339, Arlington, VA, USA, 2007.
- [60] J. Fripp, P. Bourgeat, C. Engstrom, S. Ourselin, S. Crozier, and O. Salvado, "Automated segmentation of the menisci from MR images," in *Proceedings of the 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Boston, MA, USA, 2009.
- [61] S. Hyun, S. Lee, I. Dong, and S. Uk, "Automatic bone segmentation in knee MR images using a coarse-to-fine strategy," vol. 8314, pp. 1–6, 2012.
- [62] E. B. Dam, J. Folkesson, P. C. Pettersen, and C. Christiansen, "Semi-automatic knee cartilage segmentation," in *Proceedings of the Medical Imaging 2006: Image Processing*, San Diego, CA, USA, 2006.
- [63] M. Marčan and I. Voiculescu, "Unsupervised segmentation of MRI knees using image partition forests," in *Proceedings of the Medical Imaging 2016: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 9788, San Diego, CA, USA, 2016.
- [64] A. Iplani, D. G. Disler, T. J. Holmes, T. R. Mccauley, and J. P. Cousins, "Articular cartilage volume in the knee: semi-automated determination from three-dimensional reformations of MR images," *Radiology*, vol. 198, no. 3, pp. 855–859, 1996.
- [65] S. Solloway, C. E. Hutchinson, J. C. Waterton, and C. J. Taylor, "The use of active shape models for making thickness measurements of articular cartilage from MR images," *Magnetic Resonance in Medicine*, vol. 37, no. 6, pp. 943–952, 1997.
- [66] D. T. Sandwell, "Thermal isostasy: response of a moving lithosphere to a distributed heat source," *Journal of Geophysical Research: Solid Earth*, vol. 87, no. B2, pp. 1001–1014, 1982.
- [67] P. Dodin, J. Pelletier, J. Martel-Pelletier, and F. Abram, "Automatic human knee cartilage segmentation from 3-D magnetic resonance images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2699–2711, 2010.
- [68] J. Fripp, S. Crozier, S. K. Warfield, S. Ourselin, and S. Ourselin, "Automatic segmentation and quantitative analysis of the articular cartilages from magnetic resonance images of the knee," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 55–64, 2010.
- [69] M. A. Baldwin, J. E. Langenderfer, P. J. Rullkoetter, and P. J. Laz, "Development of subject-specific and statistical shape models of the knee using an efficient segmentation and mesh-morphing approach," *Computer Methods and Programs in Biomedicine*, vol. 97, no. 3, pp. 232–240, 2010.
- [70] F. K. Patel and M. Singh, "Segmentation of cartilage from knee MRI images using the watershed algorithm," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 2, pp. 1727–1730, 2018.
- [71] H. Hassan, A. K. Bashir, R. Abbasi, W. Ahmad, and B. Luo, "Single image defocus estimation by modified Gaussian function," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 6, pp. 1–14, 2019.
- [72] H. Hassan, A. K. Bashir, M. Ahmad et al., "Real-time image dehazing by superpixels segmentation and guidance filter," *Journal of Real-Time Image Processing*, pp. 1–21, 2020.
- [73] A. Saygılı and S. Albayrak, "A new computer-based approach for fully automated segmentation of knee meniscus from magnetic resonance images," *Biocybernetics and Biomedical Engineering*, vol. 37, no. 3, pp. 432–442, 2017.
- [74] N. Bien, P. Rajpurkar, R. L. Ball et al., "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of MRNet," *PLoS Medicine*, vol. 15, no. 11, Article ID e1002699, 2018.
- [75] B. G. Ashinsky, M. Bouhrara, C. E. Coletta et al., "Predicting early symptomatic osteoarthritis in the human knee using machine learning classification of magnetic resonance images from the osteoarthritis initiative," *Journal of Orthopaedic Research*, vol. 35, no. 10, pp. 2243–2250, 2017.
- [76] A. Saygılı and S. Albayrak, "An efficient and fast computer-aided method for fully automated diagnosis of meniscal tears from magnetic resonance images," *Artificial Intelligence in Medicine*, vol. 97, no. 13, pp. 118–130, 2019.
- [77] R. Hemalatha, V. Vijaybaskar, and T. Thamizhvani, "Automatic localization of anatomical regions in medical ultrasound images of rheumatoid arthritis using deep learning," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 233, no. 6, pp. 657–667, 2019.
- [78] Z. Zhou, G. Zhao, R. Kijowski, and F. Liu, "Deep convolutional neural network for segmentation of knee joint anatomy," *Magnetic Resonance in Medicine*, vol. 80, no. 6, pp. 2759–2770, 2018.
- [79] N. Tsafack, S. Sankar, B. Abd-El-Atty et al., "A new chaotic map with dynamic analysis and encryption application in internet of health things," *IEEE Access*, vol. 8, Article ID 137731, 2020.
- [80] M. Al-Maitah, A. A. AlZubi, and A. Alarifi, "An optimal storage utilization technique for IoT devices using sequential machine learning," *Computer Networks*, vol. 152, pp. 98–105, 2019.



- [81] A. A. AlZubi, A. Alarifi, and W. Alnumay, "Machine learning-based channel analysis for user concentric optical switching networks," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 1178–1194, 2020.
- [82] S. Deep, X. Zheng, A. Jolfaei, D. Yu, P. Ostovari, and A. Kashif Bashir, "A survey of security and privacy issues in the internet of things from the layered context," *Transactions on Emerging Telecommunications Technologies*, Article ID e3935, 2020.
- [83] Z. Zheng, T. Wang, J. Wen, S. Mumtaz, A. K. Bashir, and S. H. Chauhdary, "Differentially private high-dimensional data publication in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2640–2650, 2020.