Wireless Communications and Mobile Computing

Cyber-Physical Mobile Computing, Communications, and Sensing for Industrial Internet of Things and Industry 4.0 2021

Lead Guest Editor: Mohammad Khosravi
Guest Editors: Alireza Jolfaei, Varun Menon, and Shunmei Meng

Cyber-Physical Mobile Computing, Communications, and Sensing for Industrial Internet of Things and Industry 4.0 2021

Wireless Communications and Mobile Computing

Cyber-Physical Mobile Computing, Communications, and Sensing for Industrial Internet of Things and Industry 4.0 2021

Lead Guest Editor: Mohammad Khosravi
Guest Editors: Alireza Jolfaei, Varun Menon, and Shunmei Meng

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9060340, 13 pages
https://doi.org/10.1155/2022/9060340

Research Article

An Accurate Heart Disease Prognosis Using Machine Intelligence and IoMT

Jamshid Pirgazi[1] **Ali Ghanbari Sorkhi**[1] **and Majid Iranpour Mobarkeh**[2]

[1] *Department of Computer Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran*
[2] *Department of Computer Engineering and IT, Payam Noor University, Tehran, Iran*

Correspondence should be addressed to Jamshid Pirgazi; j.pirgazi@mazust.ac.ir

Received 25 February 2022; Revised 10 May 2022; Accepted 18 June 2022; Published 1 July 2022

Academic Editor: Mohammad R Khosravi

In recent years, Internet of Medical Things (IoMT) and machine learning (ML) have played a major role in the healthcare industry and prediction of in time diagnosis of diseases. Heart disease has long been considered one of the most common and lethal causes of death. Accordingly, in this paper, a multiple-step method using IoMT and ML has been proposed for diagnosis of heart disease based on image and numerical resources. In the first step, transfer learning based on convolutional neural network (CNN) is used for feature extraction. In the second step, three methods of distributed stochastic neighbor embedding (t-SNE), F-score, and correlation-based feature selection (CFS) are utilized to select the best features. In the end, a combination of outputs of three classifiers including Gaussian Bayes (GB), support vector machine (SVM), and random forest (RF) according to the majority voting is employed for diagnosis of the conditions of heart disease patients. The results were evaluated on the two UCI datasets. The results indicate the improvement of performance compared to other methods.

1. Introduction

According to the World Health Organization (WHO) statistics, cardiovascular disease is one of the leading causes of death worldwide, accounting for 17.9 million deaths each year [1]. The main causes of heart disease are various unhealthy activities such as high cholesterol, obesity, an increase in triglyceride levels, and high blood pressure, among others. Sleep problems, irregular heartbeat, swollen legs, and, in some cases, weight gain of 1 to 2 kilograms per day all increase the risk of heart disease [2, 3]. All these symptoms are common within various diseases leading to death in the near future; therefore, the correct diagnosis is difficult.

Smart healthcare presents healthcare platforms which make use of tools such as IoT, wearable appliances, and wireless Internet connection for signing in health evidences and resource connection, organizations, and individuals. IoT, artificial intelligence (AI), big data, cloud networks, 5G, and advanced biotechnology are some of the smart healthcare networks used in disease screening and diagnosis and medical research [4].

As previously mentioned, IoT and IoMT play a great part in the healthcare in prediction of time and chronic illness diagnosis. The volume of information required by the healthcare, security factors, power of processing, and accuracy of information is very important in terms of diagnostic prediction for many illnesses. To tackle these challenges, AI algorithms in previous researches are used to increase the precision of patients' data [5].

IoMT refers to disease diagnosis without human intervention through the development of intelligent sensors, smart devices, and advanced lightweight communication protocols. IoMT-based healthcare, swallowable sensor tracking, mobile health, smart hospitals, and improved treatment of chronic diseases have been shown in [6].

IoMT is a new network-based technique for connecting medical devices and their applications to healthcare information technology systems. In [7], in addition to providing treatment to orthopedic patients, the IoMT approach examines the possibilities of facing with COVID-19 pandemic.

In the recent years, ML is widely utilized in healthcare industry to analyze big data for initial prediction of diseases leading to the improvement of the quality of healthcare [8, 9]. ML can be used to solve complex health issues and give accurate results. Healthcare industry is one of the largest industries in which ML has shown to be functional. Creating accurate and multidimensional datasets are very important and play a critical role in the functionality of ML algorithms. IoMT enables medical facilities and healthcare products to share real-time data to create a great volume of data for ML [10].

Lately, large amount of research data and patients' cases have become accessible. There are many open sources for gaining access to patients' records, and research can be done to be able to use computer technologies for patient identification and accurate disease diagnosis in order to prevent the lethality of these illnesses. Today, ML and AI are well recognized to play major roles in healthcare industry, and various models of ML and deep learning (DL) can be employed to classify and diagnose diseases or to predict results. Complete analysis of genome data can easily be done using different models of ML [11–13].

Several studies have utilized different models of ML for classification and diagnosis of heart diseases. CART automatic classifier based on classification and regression of congestive heart failure [14], using deep neural network for best feature selection and ECG performance improvement [15], proposing a clinical decision support system for diagnosis of heart failures and its prevention during initial stages of the disease [16], and also rule-based natural language processing (NLP) [17] are among these researches.

In today's digital age, healthcare generates a large amount of patient data. For physicians, manual control of these data is difficult, whereas IoT can manage the produced data very efficiently. IoT records large amounts of data and is capable of diagnosing diseases using machine algorithms with the purpose of applying different methods of ML on the produced data. A ML approach is proposed for initial heart disease prediction in relation to IoT [10].

Cardiac image processing approaches which are obtained from DL manage and supervise large medical data gathered by the IoT. Deep IoMT is a common DL and IoT platform that is in charge of extracting precise cardiac image data of usual instruments and devices. Energy depletion, finite battery life, and high PLR (packet loss ratio) are critical issues that must be addressed in universal medical care. Wearable devices must be stable (i.e., have a longer battery life), energy efficient, and valid in order to improve an affordable and inclusive healthcare environment. In this regard, a new efficient approach based on the consciously enhanced efficient-aware approach (EEA) of self-adaptive power control to decrease energy utilization while increasing validity and battery life is proposed in [18]. For remote cardiac imaging of elderly patients, a new common DL-IoMT framework (DL-based layered architecture for IoMT) has also been proposed.

Medical image classification is critical in the prediction and early detection of critical illnesses. Medical imaging is the most essential record of patient's health which helps to control and cure illnesses, which is one of the important applications of IoMT. In [19], an improved classification of optimal DL for the lung cancer classification, brain imaging, and Alzheimer's disease is introduced. The researches show that medical image classification is based on optimal feature selection using the DL by combining preprocessing, feature selection, and classification. The primary goal of model extraction is to select an effective feature for medical image classification. The opposition-based crow search (OCS) approach is recommended to enhance the efficiency of the DL classifier. In addition, multitextured, gray-level features are chosen for analysis. Finally, it is claimed that the optimal features made better the result of classification.

This study presents a method based on data collected by IoT. In this regard, a general method is presented for numerical and image data. At first, the proposed method examines the type of data resource. If input data were from image resources, in the first step, features are extracted from this type of resource using transfer learning. CNN-based deep network is used for this purpose. Fully connected layer has been utilized for feature extraction, whereas if the input data were from numerical sources, the first step is ignored. The proposed method's next steps include feature selection and classification phases, which are independent of the input resource. In the feature selection step, three methods of distributed stochastic neighbor embedding (t-SNE), F-score, and correlation-based feature selection (CFS) have been used. An individual classifier has been trained for each method of feature selection. In this paper, three classifiers of SVM, GB, and RF have been employed. In the end, voting is used for final label selection. The results demonstrate that the proposed method performs well.

The rest of this paper is organized as follows. Section 2 discusses previous research in this area. Section 3 examines the proposed method and its details. Section 4 compares the performance of the proposed method to some of the successful models in this field, and Section 5 concludes the paper.

2. Literature

With the recent advances in medical data processing and machine learning, many researchers have been consistently active in this field. One of the most challenging medical data is data related to heart diseases which have drawn many researchers' attention. In [20, 21], multiple machine learning methods were examined for the prediction of heart diseases in which recursive neural network (RNN) and decision tree (DT) were reported to have gained the best results.

In [22], deep neural network (DNN) with the name of Heart Evaluation for Algorithmic Risk-reduction and Optimization five (HEARO-5) was proposed. This method which is consisted of regularization has shown positive results on UCI dataset. In [23], for classifying imbalanced clinical data, a neural network with a convolution layer was used. This study takes advantage of a two-step approach feature weight based on least absolute shrinkage and selection operator (LASSO) and then identification of critical features based on majority voting for achieving more accuracy in classified imbalanced data.

In [24], to increase the performance of the classifier, feature selection approaches based on fast correlation-based feature selection (FCBF) were used to choose efficient features. In this method, classification is done using K-nearest neighbor (KNN), SVM, Naive Bayes (NB), RF, and multilayer perceptron (MLP) optimized using particle swarm optimization (PSO) with ant colony optimization (ACO) [25]. NB, SVM, and RF methods were employed for extraction and classification of the most relevant features in [26, 27].

A k-means method with particle swamp was proposed in [28] for detecting hazard factors in coronary heart disease treatment (CAD). The extracted data are classified using MLP, multinomial logistic regression (MLR), and algorithms of phase rule, as well as C4.5. It was claimed that the results demonstrated the appropriate accuracy of the proposed method on the datasets presented by medical college in India. In [29], heart disease prediction has been done using methods of data mining, ML, and DL, and neural network method was claimed to be more functional than other methods. In [30], genetic algorithms and neural networks were employed for diagnosis of heart disease.

3. Proposed Method

The general procedure of the proposed method is shown in Figure 1. As it can be seen, this method is made up of three major steps. In the first step, two different approaches with respect to the input resource are used. If data are numerical, only feature vector gets used for the next step; however, if data are image, the feature vector must be extracted. For the purpose of extracting features from images, transfer learning based on CNN has been used. In this stage, fully connected layer is utilized after convolution layers for feature extraction. The second step of the proposed method is made up of feature selection. This step is independent of the input resource. Three methods of t-SNE, F-score, and CFS have been put to use for feature selection. In the third step of the proposed method, for each feature vector of the previous step, three different classifiers of SVM, GB, and RF are used. In the end, majority voting has been used for selection of the favorable output. Labels of the three classifiers used in the last step are the input of the current step. Eventually, the final input label is selected. In the following, different sections of the proposed method will be described.

3.1. Feature Extraction Based on Image Resource. The extraction of features is a critical issue in classification [31]. As illustrated in Figure 1, one of the main steps of the proposed method is feature extraction. In the step of feature extraction, if the resource is image, it must turn into a feature vector. Methods based on DL are among the most successful methods for feature extraction; however, unfortunately, the numbers of images related to heart diseases are very low; therefore, in this step, transfer learning has been utilized for feature extraction (Figure 2). A pretrained CNN network is used in this step as well. This network is merely used for feature extraction that the output of fully connected layer is selected as the feature vector.

Transfer learning is an issue of great significance which focuses on knowledge retention of problem-solving and its usage to solve a different but related problem. Since datasets are not sufficiently available, CNN network is not initially trained; thus, pretrained network weights aid to solve more issues concerning feature extraction or configuration. Very deep networks are costly to be trained. More complex models require more time for training using hundreds of systems with expensive CPUs.

Transfer learning maps a model that has already been trained in specific areas to a new model in new domains; thus, the time required for training by using this method is reduced [32]. Furthermore, in complex models, transfer learning decrease the need for a large number of training samples. Because the number of images available in the field of heart disease is limited, this method is used to compute the initial weights from the well-known ImageNet dataset. The ResNet, AlexNet, VGG-16, and VGG-19 architectures trained on ImageNet are evaluated based on a set of validations. VGG-16 architecture has shown the best performance due to experimental results. As shown in Figure 2, this paper uses CNN-based transfer learning to extract features.

3.2. Feature Selection. As it is shown in Figure 1, in this section, the feature vector extracted from the previous step is used as the input for feature selection. In this step, three methods of feature selection including t-SNE, F-score, and CFS are used which are further elaborated in the following.

3.3. Correlation-Based Feature Selection (CFS). As a filter method, CFS classifies and evaluates feature subsets based on subsets that are highly correlated with the class but unrelated to one another [33]. Irrelevant features should be ignored if they have a low correlation with the class. Aside from that, the duplicated features can be identified because they are closely related to the remaining ones. The feature can be accepted if it predicts the label that no other features predict. The evaluation function of CFS' feature subset is as follows:

$$M_s = \frac{\overline{k r_{cf}}}{\sqrt{k + k(k-1) r_{ff}}}. \tag{1}$$

In this equation, M_s shows the heuristic "merit" of a feature subset S including k features, and also, $\overrightarrow{r_{cf}}$ and $\overline{r_{ff}}$ represent the mean feature-class correlation $(f \in S)$ and the average feature intercorrelation, respectively. The calculation from this equation has the usage to predict not only the feature subsets but also the redundant ones [34].

3.4. F-Score. F-score by evaluating the difference between two real numbers sets presents a simple feature selection filter method [35] which for feature i is calculated as follows:

$$F - \text{score}(i) = \frac{\sum_{k=1}^{m} \left(\bar{x}_i^k - \bar{x}_i\right)^2}{\sum_{k=1}^{m} \left(1/n^k - 1\right) \sum_{j=1}^{n^k} \left(x_{j,i}^k - \bar{x}_i^k\right)^2}. \tag{2}$$

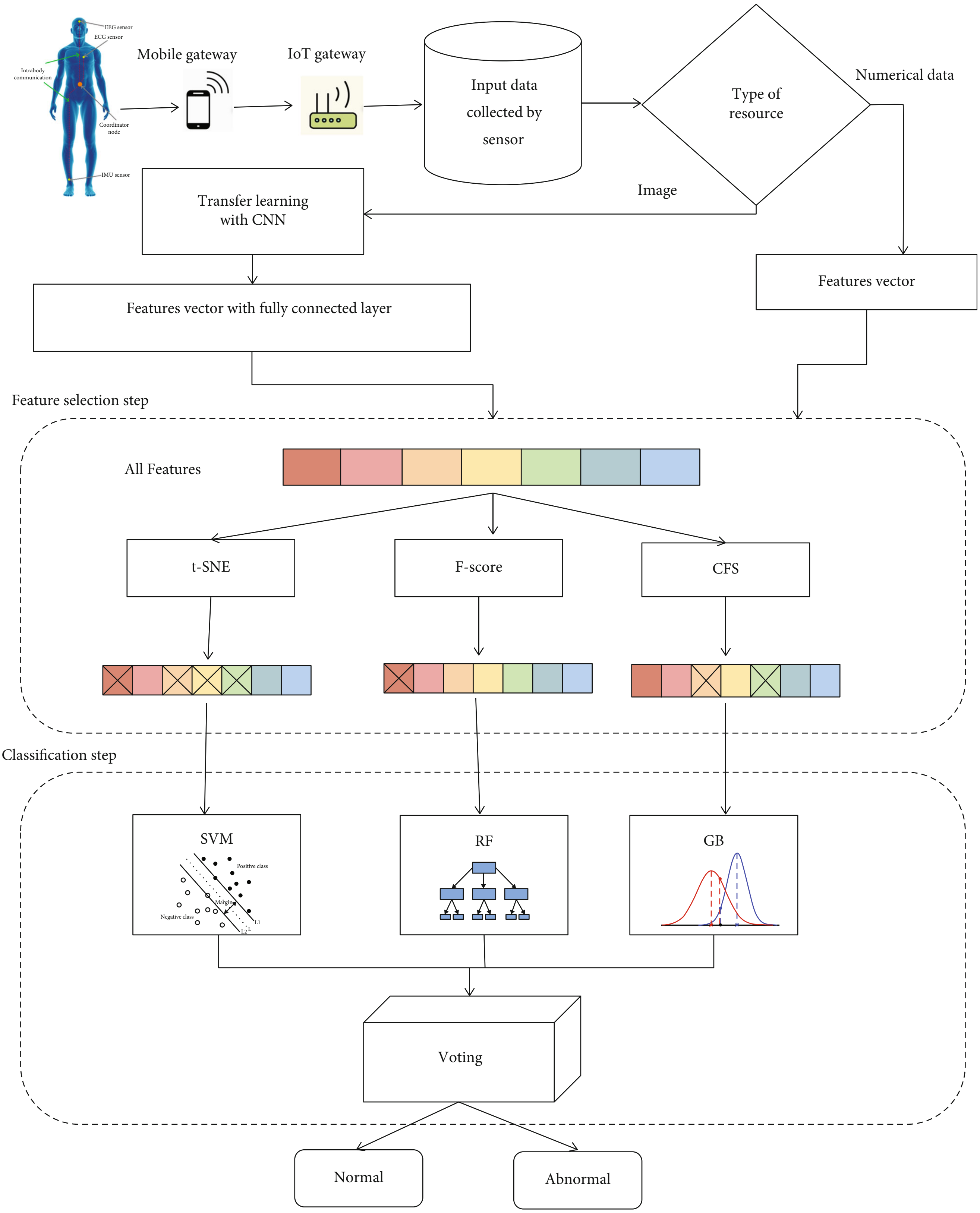


Figure 1: An overview of the proposed method.

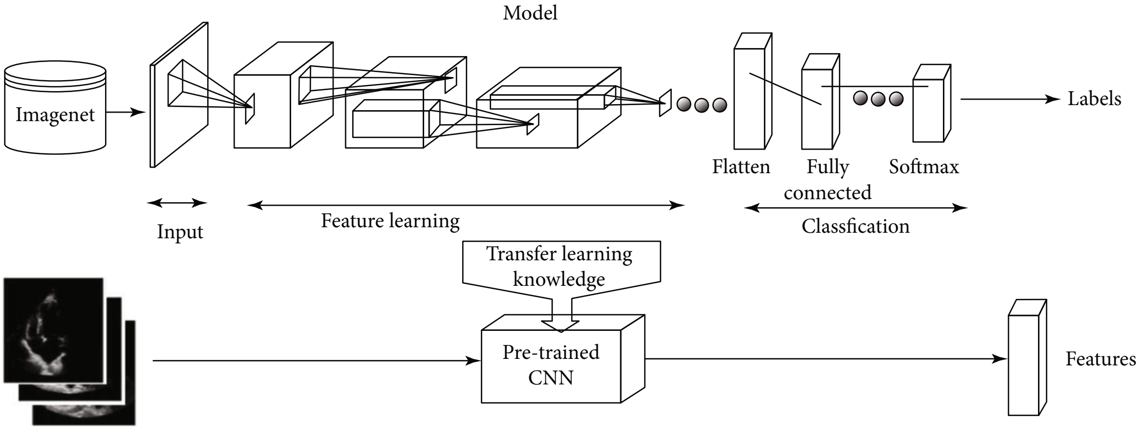


Figure 2: Using transfer learning to extract features form image resources.

Table 1: Hyperparameters of the basic classifiers.

Methods	Parameters	Amounts
SVM	C_SVM	1
	Kernel_SVM	Radial basis function (RBF)
	Degree_SVM	3
	Gamma_SVM	Scale
	Coef0_SVM	0
GB	Priors_GB	None
	Var-smoothing_GB	1e-08
RF	Min_samples_split_RF	2
	Min_samples_leaf_RF	1

In the above equation, m refers to the number of classes, n^k shows the samples number of class k, $\bar{x}_i$ presents the mean of feature i among data, also $\bar{x}_i^k$ demonstrates the mean of feature i in class k, and $x_{j,i}^k$ shows the amount of feature i in the sample j of the class k. If F-score related to a feature is high, it shows that the respected feature includes proper information which belongs to classification.

3.5. Distributed Stochastic Neighbor Embedding (t-SNE). This method is an unsupervised nonlinear method which is used for discovery and reduction of data dimensions. In other words, it will provide the user with an understanding of the manner of data organization in a high-dimensional space. This method has been introduced in 2008 by Laurens van der Maatens and Geoffery Hinton [36]. The main difference between this method and principal component analysis (PCA) is that PCA is a method of reducing the linear dimensions which attempts to maximize the variance and preserve the large distance between the pares, while t-SNE preserves PCA in preserving the small distance between pares by using local similarities. t-SNE algorithm computes a similarity measure between the pare of samples in large-dimensional data and low-dimensional space. Then, it attempts to optimize these two similarity measures using a cost function. This process is undertaken through three main steps. They are as follows:

(1) In the first step, the interpoint similarity in high-dimensional space is measured. To better understand this, suppose a set of scattered data points in a two-dimensional space. For each data point of x_i, the Gaussian distribution is spread around that point by the user. Then, the density of all x_i points will be computed based on that Gaussian distribution. Then, renormalization is applied to all data points. This will result in a set of P_{ij} probabilities for all data points. These probabilities are proportional to their similarities. This actually means that if x_1 and x_2 data points possess a similar value under the Gaussian circle, their proportions and similarities will be equal consequently; hence, the local similarities will hold true in the structure of high-dimensional space

(2) The second step is quite similar to the first; but conversely, Student's T-distribution with one level of freedom is used instead of Gaussian distribution which is also known as the Cauchy distribution. This will result in a second set of Q_{ij} probabilities in a low-dimensional space

(3) The last step is associated with the reflection of high-dimensional space probabilities P_{ij} through low-dimensional space probabilities Q_{ij} in the best possible manner. The basic requirement here is the similarity of the two mappings. The difference between two-dimensional space probability distributions is computed through the Kullback-Leibler (KL) divergence criteria. This study does not elaborate upon KL. The only point to be considered is that it is an asymmetrical approach in which the effective comparison of P_{ij} and Q_{ij} values does not suffice. Eventually, the optimal value of the KL cost function is found using gradient descent

3.6. Classification. An ensemble classifier is used on the reduced feature vector. In these types of classifications,

Table 2: Description of Cleveland dataset [39].

No.	Name of attribute	Description
1	Age	Age in years
2	Sex	Male is equal 1 and female is equal 0
3	CP	Type of chest pain
4	Trestbps	A criterion which shows resting blood pressure
5	Chol	A criterion which shows serum cholesterol
6	FBS	A variable which is boolean, when fasting blood sugar > 120 mg/dl is true otherwise it is false
7	Restecg	A criterion which shows resting electrocardiographic results
8	Thalach	A criterion which shows maximum heart rate
9	Exang	A binary variable which shows exercise-induced angina
10	Oldpeak	A criterion which shows ST depression
11	Slope	A criterion which shows the peak exercise ST segment
12	CA	A criterion which shows major vessel number
13	Thal	A criterion which shows heart rate
14	NUM	A criterion which shows heart disease status

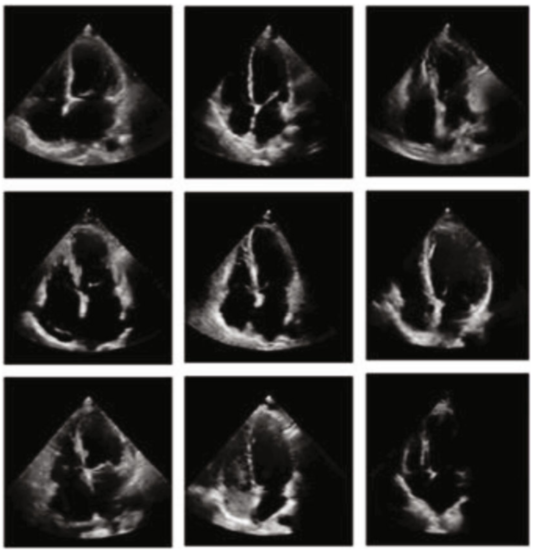


Figure 3: Sample image from dataset.

combination of a number of basic classifiers creates an accurate and robust classification. One of the most common ways to combine classifiers is majority voting. As shown in Figure 1, since the diversity of the consisting classifiers gives rise to the power of an ensemble classifier, the SVM, BG, and RF are suggested as basic classifiers. Therefore, it is expected that the sample data to be covered in the maximum range and the generalizability of the classification to be increased. It is better not to use the classification with the similar results in group classification. In order to reduce the classification error, it is important to choose the appropriate classifier and combination strategy.

Support vectors in the SVM model are the most important component of the model, which is obtained through convex optimization. In this model, the classification margin creates the maximum distance within classes. The main assumption in Bayesian classifier is statistical independence between features and in most cases maximizes the performance of the acquisition. In this classifier, model parameters are estimated with a small set of training data. Random forest is a simple machine learning technique that usually produces outstanding results even when its hyperparameters are not adjusted. This technique is one of the most extensively used machine learning algorithms for both regression and classification because of its simplicity and usability [37, 38]. This method works based on building a large number of decision trees. In the proposed method, the classifications are combined by voting according to label repetitions. The main reason for choosing three different classifiers, SVM, BG, and RF, as the basic classifier which is the main component in constructing ensemble classifiers is "diversity." All of these classifiers are trained differently leading to the increase of the level of classification diversity and ensemble generalization.

4. Experimental Results

This section summarizes the results of experiments conducted to evaluate the suggested method's performance. It should be noted that all the presented methods and analysis of their results are done on same datasets and similar hardware. All the implementation is done on a computer with Core (TM) i7 M620 CPU, 4GB memory card, and T4 graphic card with Python as programming language as well as Keras framework. It also should be mentioned that Scikit-learn-0.22.0 toolbox has been used for classification and all the parameters in this toolbox also have been utilized by default. For instance, SVC employs the "one vs. one" approach for ensemble classification. Table 1 shows the main classifier parameters.

4.1. Database. The Cleveland dataset from UCI is used to evaluate the proposed method. This dataset is available at http://archive.ics.uci.edu/ml/datasets.php. Cleveland dataset

Table 3: Descriptions of echocardiogram dataset [4].

Name of attribute	Description
Survival	This variable indicates the number of months the patient survives
Still-alive	A variable which is binary, still-alive is shown by 1 and dead by 0
Age at heart attack	Age of heart attack occurrence (in years)
Pericardial effusion	A variable which is binary. Fluid around the heart is shown by 1 and no fluid by 0
Fractional shortening	A criterion which measures contractility around the heart
Epss	Another criterion which measures contractility (E-point septal separation)
Lvdd	A criterion which measures the size of the heart (left ventricular end-diastolic dimension)
Wall motion score	A criterion which measures the movement of the left ventricle segments
Wall motion index	This criterion depends on number of segments seen that can be used instead of the wall motion score
Mult	An ignorable var which is derivative
Name	Patient's name
Group	Meaningless
Alive at 1	A variable which is boolean, patient was dead after one year is shown by 0 and patient was alive at one year by 1

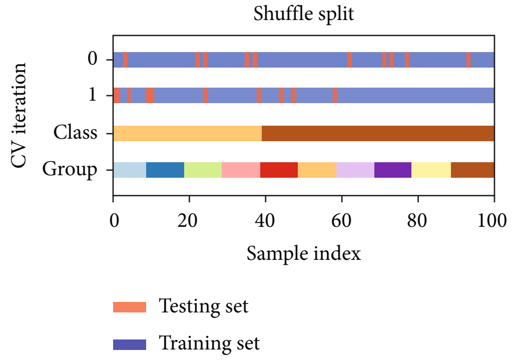


Figure 4: Stages of sample selection in testing and training sets.

owns 76 attributes and 303 samples. Nonetheless, only 14 attributes of Cleveland dataset were put to use for training and testing. These features are further elaborated in Table 2. These types of data have been used as numerical resources in the present paper.

In the following, echocardiogram images have been employed as image resources. Figure 3 shows some examples of these images. The suitable attributes are described in Table 3. UCI database was used for echocardiography image retrieval using 66 normal images from 30 participants and 66 abnormal images from 30 subjects [4]. When the variables of "survival" and "still-alive" are combined together, it shows whether the patient has stayed alive at least one year after the heart attack or not.

In the experiments performed to evaluate the proposed method, 10-fold cross-validation was used. The steps for building a training and test set are described in Figure 4. Accordingly, in each repetition, 10% of the data were used as a test set and the rest as a training set. In addition, 10% of the training image sets have been used to create the validation set.

4.2. Evaluation Criteria. Several quantitative criteria including specificity (Spe), accuracy (ACC), recall (sensitivity) (RE), precision (PR), and F1 are used to show the performance of the proposed method [40].

Generally, accuracy (ACC) refers to a model's ability to accurately predict the output label. Equation (3) depicts the accuracy criterion. It also should be mentioned that variance and mean in 10 numbers of repetitions are considered to calculate accuracy for 10-fold cross validation. This criterion examines the training level and functionality of the model, although it has no further information regarding the model accurate functionality.

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{total examples}}. \quad (3)$$

In equation (4), precision criterion is shown that is appropriate for amounts with high false positive.

$$\text{PR} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (4)$$

In equation (5), recall (sensitivity) criterion is shown that is appropriate for amounts with high false negative.

$$\text{RE} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (5)$$

In equation (6), specificity criterion is shown.

$$\text{Spe} = \frac{\text{TN}}{\text{TP} + \text{FN}}. \quad (6)$$

F1 criterion is shown in equation (7). This criterion also contains accuracy and recall (sensitivity) criteria. F1 approaches 0 and 1, respectively, in its worst and best cases.

$$F_1 = \frac{2 * \text{RE} * \text{PR}}{\text{PR} + \text{RE}}. \quad (7)$$

In the aforementioned equations, TP presents the number of images which is correctly allocated to C_i class by

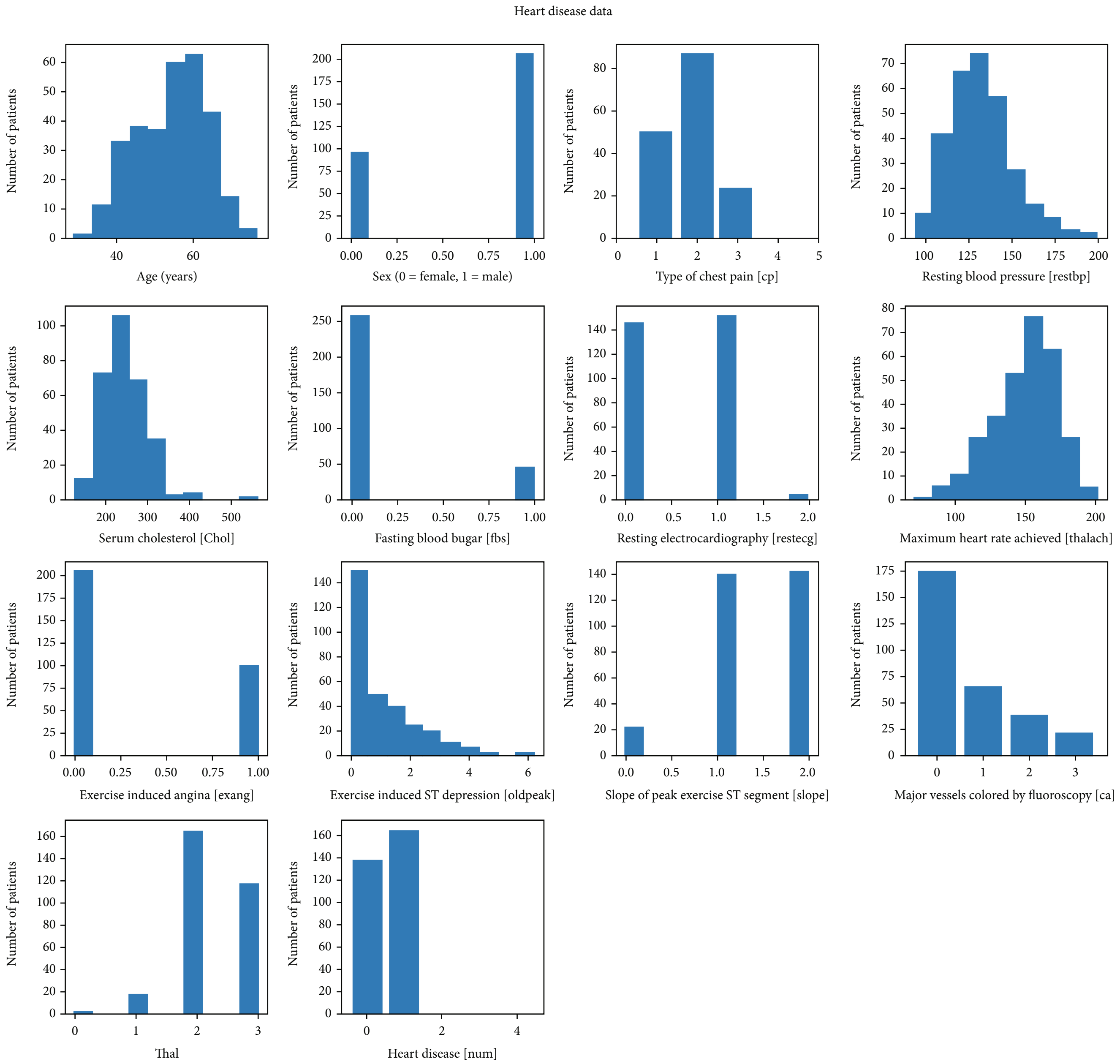


Figure 5: Histogram of the number of patients based on different attributes of Cleveland dataset.

classifier and FN presents the number of images from class C_i which are wrongly allocated to other classes using classifier. FP presents the number of images belonging to class C_i which are allocated to other classes. TN criterion is the number of images which do not belong to class C_i nor allocated to this class using classifier.

4.3. Results. In this section, we investigate the proposed method's performance on two datasets with varying input resources. In the first dataset, data are numerical and extracted from Cleveland dataset. As it was previously mentioned, these types of data directly go into the step of feature selection as inputs. In this section, in order to show the influence of each attribute, the attributes of this dataset are examined. Figure 5 illustrates the histogram of the number of patients per attribute. As it is evident, the amount of most attributes is imbalanced among patients.

Figure 6 shows the frequency of attributes according to the individuals' condition (healthy or sick). With respect to the aforementioned figure, it is certain that amounts of some of the attributes have more significant relationships with the condition of samples and show more separability toward individuals' conditions. This relationship and separability, however, is less noticeable in some of the attributes.

The system's performance can be influenced by choosing the right features. Three feature selection approaches are employed in this case: t-SNE, F-score, and CFS.

As stated in the proposed method, for the three classifiers SVM, RF, and GB, the extracted features based on t-SNE, F-score, and CFS methods have been used, respectively.

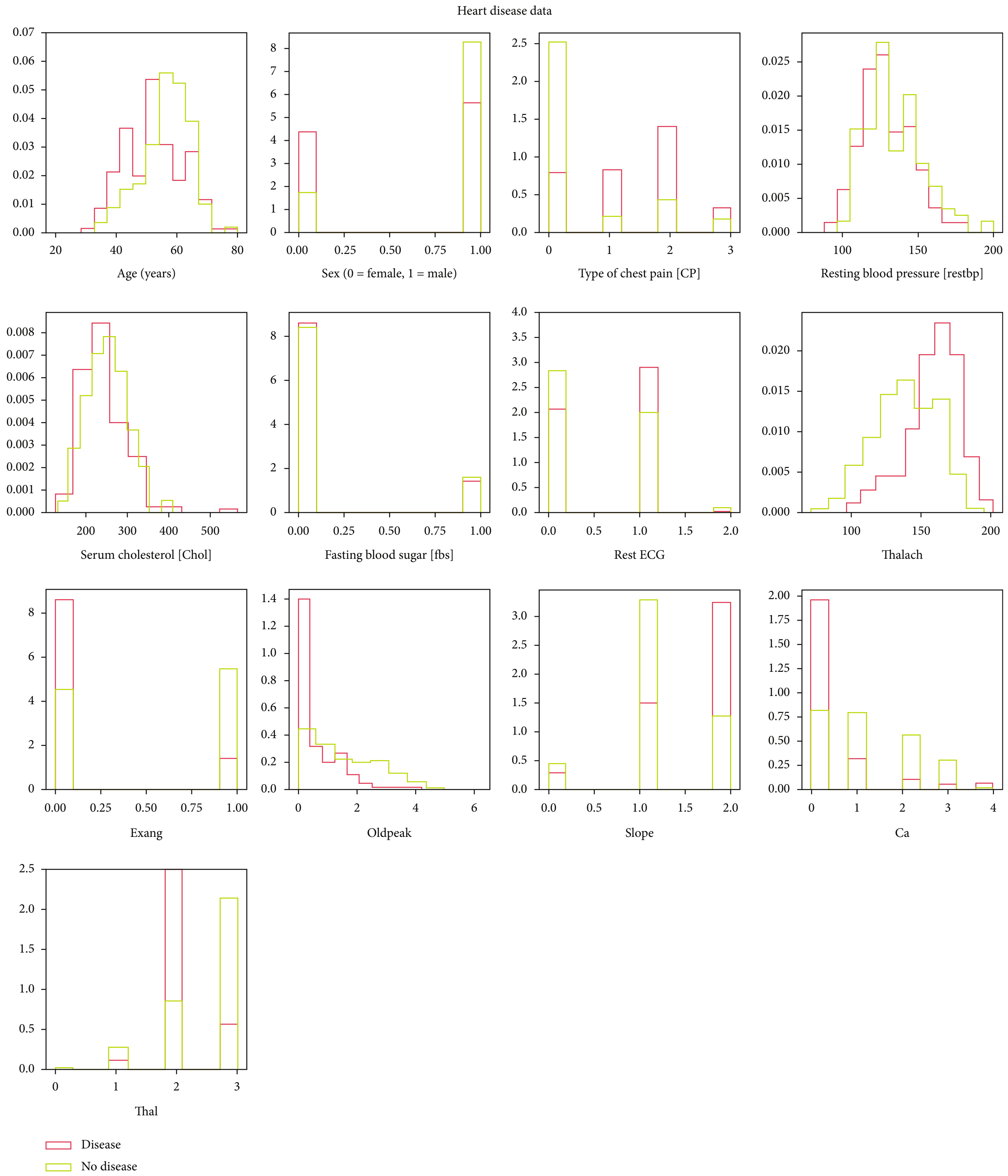


Figure 6: Frequency of different attributes in two conditions of the healthy and the sick in Cleveland dataset.

Each classifier's features are chosen using a validation set. Table 4 displays the outcomes of each approach in the validation set. It should be noted that the mean accuracy for 10 iterations is reported in this table. According to the results obtained in both types of input sources (image or numerical), the t-SNE feature has the best performance in the SVM classification, the F-score feature in the RF classification, and the CFS feature in the GB classification, respectively.

Table 4: Results of different classifiers based on different feature selection methods in validation set.

Type of data	Method		Accuracy	
		t-SNE	F-score	CFS
Numerical resources	SVM	90.12 (±0.032)	88.34 (±0.570)	81.22 (±0.0078)
	RF	85.12 (±0.0322)	86.43 (±0.120)	83.11 (±0.056)
	GB	90.21 (±0.0167)	78.45 (±0.077)	88.25 (±0.110)
Image resources	92.60(±0.570)	93.12 (±0.061)	95.65 (±0.018)	SVM
	94.16(±0.420)	96.32 (±0.045)	89.32 (±0.130)	RF
	95.78(±0.220)	86.25 (±0.190)	90.74 (±0.470)	GB

Table 5: Results of the proposed method in comparison with other methods based on numerical resources in Cleveland dataset.

Method	Accuracy	Precision	Recall	Specificity	F-score
Logistic regression [8]	83.3	—	86.3	82.3	—
K-neighbors [8]	84.8	—	85.0	77.7	—
SVM [8]	83.2	—	78.2	78.7	—
Random forest [8]	80.3	—	78.2	78.7	—
Decision tree [8]	82.3	—	78.5	78.9	—
DL [8]	94.2	—	82.3	83.1	—
K-nearest neighbor [5]	75.73	—	—	—	—
Decision trees [5]	72.45	—	—	—	—
Random forest [5]	75.73	—	—	—	—
Multilayer perceptron [5]	67.54	—	—	—	—
Naïve Bayes [5]	76.26	—	—	—	—
Linear support vector machine [5]	77.73	—	—	—	—
Faster R-CNN with SE-ResNeXt-101 [4]	98.00	96.16	98.47	96.02	97.58
Proposed method	98.7	96.61	99.18	96.65	98.48

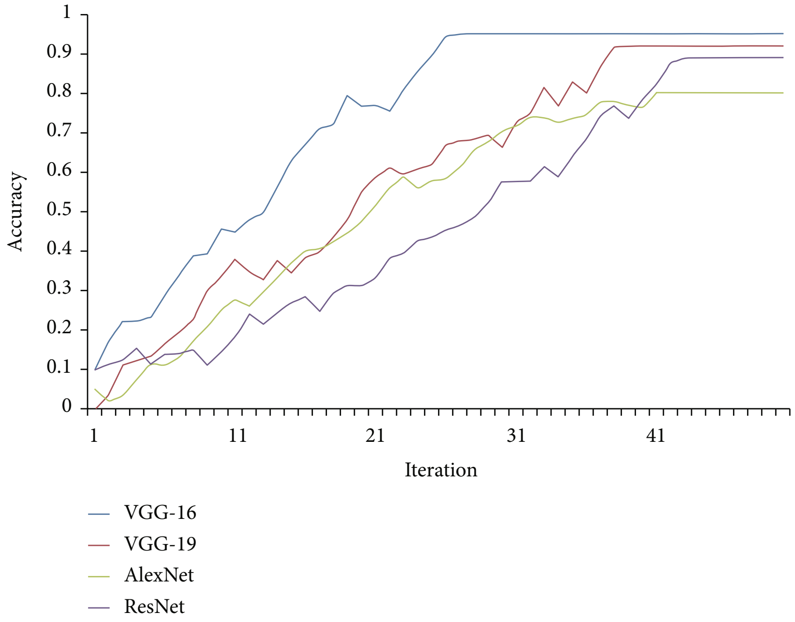


Figure 7: The initial results based on different architectures in image classification.

Table 6: Results of the proposed method in comparison with other methods based on image resources.

Method	Accuracy	Precision	Recall	Specificity	F-score
VGG-19 [4]	95.23	93.96	94.80	93.19	95.58
ResNeXt-101 [4]	96.15	94.00	95.42	92.98	95.99
Inception-ResNet-v2 [4]	96.48	94.07	96.14	94.11	96.04
SE-ResNet-101 [4]	97.94	95.18	97.31	95.03	98.25
Faster R-CNN with SE-ResNeXt-101 [4]	99.15	98.06	98.95	96.32	99.02
Proposed method	99.84	98.64	99.61	97.19	99.12

Table 7: Results of the proposed method with the different voting method.

Type of data	Accuracy	Method
Numerical resources	97.32	Proposed method-weighted majority voting
	98.7	Proposed method-majority voting
Image resources	98.00	Proposed method-weighted majority voting
	99.84	Proposed method-majority voting

The proposed method's results are shown in Table 5. As is obvious, the proposed method outperformed all of the other methods.

In the following, the performance of the proposed method based on the image resource is examined. It was noted in the proposed method section that the choice of convolutional network design affects the method's performance; hence, four different architectures were investigated: AlexNet, ResNet, VGG-16, and VGG-19. Training occurs solely in the fully connected layers, which is identical to an MLP network used for classification, and the convolutional layers needed to extract the feature are not learned due to the usage of transfer learning. The output layer has the same number of layers as the number of classes and is made up of two layers. The accuracy performance of each type of architecture with 50 repetitions to train fully connected layers is shown in Figure 7. This comparison shows that the VGG-16 architecture performs better, and as a result, this architecture has been used to extract features. The results show that a fully connected neural network (e.g., MLP) reports accuracy of 96.4% for image classification, and this approach can improve performance.

Table 6 shows the results of the proposed method, and as it can be seen, the proposed method has proved to have a suitable performance on these types of data.

In this section, the voting method is evaluated with two different perspectives. In the proposed method, the same weight for each classifier is considered. Table 7 shows the results obtained from the proposed method based on weighted majority voting with different weights for each classifier. As can be seen in the below table, the proposed method has performed better.

5. Conclusion

Many researchers have been interested in using ML to diagnose heart diseases in recent years. In this paper, IoMT is used for receiving input data based on numerical and image resources. In this paper, to diagnose the condition of heart disease patients, a hybrid method based on feature extraction from images using transfer learning, feature selection using t-SNE, F-score, and CFS, and classification using the combined output of three classifiers including GB, SVM, and RF using majority voting is used. It was indicated that feature selection or a subset of suitable features is a fundamental part of these types of systems and highly influences the accuracy of their performance.

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 that they have no conflicts of interest.

References

[1] World Health Organization, *Cardiovascular Diseases*, WHO, Eneva, Switzerland, 2020.

[2] American Heart Association, *Classes of Heart Failure*, American Heart Association, Chicago, IL, USA, 2020, https://www.heart.org/en/health-topics/heart-failure/what-is-heartfailure/classes-of-heart-failure.

[3] American Heart Association, *Heart Failure*, American Heart-Association, Chicago, IL, USA, 2020, https://www.heart.org/en/health-topics/heart-failure.

[4] S. Manimurugan, S. Almutairi, M. M. Aborokbah et al., "Two-stage classification model for the prediction of heart disease using IoMT and artificial intelligence," *Sensors*, vol. 22, no. 2, p. 476, 2022.

[5] A. Kishor and W. Jeberson, "Diagnosis of heart disease using internet of things and machine learning algorithms," in

Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, pp. 691–702, Springer, Singapore, 2021.

[6] S. Vishnu, S. R. J. Ramson, and R. Jegan, "Internet of medical things (IoMT)-an overview," in *2020 5th international conference on devices, circuits and systems (ICDCS)*, Coimbatore, India, 2020.

[7] R. P. Singh, M. Javaid, A. Haleem, R. Vaishya, and S. Ali, "Internet of medical things (IoMT) for orthopaedic in COVID-19 pandemic: roles, challenges, and applications," *Journal of Clinical Orthopaedics and Trauma*, vol. 11, no. 4, pp. 713–717, 2020.

[8] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 8387680, 11 pages, 2021.

[9] A. G. Sorkhi, Z. Abbasi, M. I. Mobarakeh, and J. Pirgazi, "Drug–target interaction prediction using unifying of graph regularized nuclear norm with bilinear factorization," *BMC Bioinformatics*, vol. 22, no. 1, pp. 1–23, 2021.

[10] Z. Al-Makhadmeh and A. Tolba, "Utilizing IoT wearable medical device for heart disease prediction using higher order Boltzmann model: a classification approach," *Measurement*, vol. 147, p. 106815, 2019.

[11] S. Shalev-Shwartz and S. Ben-David, "Understanding machine learning," in *From 4eory to Algorithms*, Cambridge University Press, Cambridge, UK, 2020.

[12] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," in *Data Mining, Inference, and Prediction*, Springer, Cham, Switzerland, 2020.

[13] S. Marsland, *Machine Learning*, An Algorithmic Perspective CRC Press, Boca Raton, FL, USA, 2020.

[14] P. Melillo, N. De Luca, M. Bracale, and L. Pecchia, "Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 727–733, 2013.

[15] M. M. A. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, and R. R. Yager, "Deep learning approach for active classification of electrocardiogram signals," *Information Sciences*, vol. 345, pp. 340–354, 2016.

[16] G. Guidi, M. C. Pettenati, P. Melillo, and E. Iadanza, "A machine learning system to improve heart failure patient assistance," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1750–1756, 2014.

[17] R. Zhang, S. Ma, L. Shanahan, J. Munroe, S. Horn, and S. Speedie, "Automatic methods to extract New York heart association classification from clinical notes," in *Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1296–1299, Kansas City, MO, USA, November 2017.

[18] T. Zhang, A. H. Sodhro, Z. Luo et al., "A joint deep learning and internet of medical things driven framework for elderly patients," *IEEE Access*, vol. 8, pp. 75822–75832, 2020.

[19] R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, and K. Shankar, "Optimal feature selection-based medical image classification using deep learning model in internet of medical things," *IEEE Access*, vol. 8, pp. 58006–58017, 2020.

[20] K. Saxena and R. Sharma, "Efficient heart disease prediction system," *Procedia Computer Science*, vol. 85, pp. 962–969, 2016.

[21] A. Gavhane, G. Kokkula, I. Pandya, and K. Devadkar, "Prediction of heart disease using machine learning," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1275–1278, Coimbatore, India, 2018.

[22] N.-S. Tomov and S. Tomov, "On deep neural networks for detecting heart disease," 2018, https://arxiv.org/abs/1808.07168.

[23] A. Dutta, T. Batabyal, M. Basu, and S. T. Acton, "An efficient convolutional neural network for coronary heart disease prediction," *Expert Systems with Applications*, vol. 159, article 113408, 2020.

[24] Y. Khourdifi and M. Bahaj, "Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 242–252, 2019.

[25] J. Patel, A. A. Khaked, J. Patel, and J. Patel, "Heart disease prediction using machine learning," in *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security*, pp. 653–665, Springer, Singapore, 2021.

[26] A. N. Repaka, S. D. Ravikanti, and R. G. Franklin, "Design and implementing heart disease prediction using Naives Bayesian," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 292–297, Tirunelveli, India, 2019.

[27] S. Bashir, Z. S. Khan, F. H. Khan, A. Anjum, and K. Bashir, "Improving heart disease prediction using feature selection approaches," in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 619–623, Islamabad, Pakistan, 2019.

[28] L. Verma, S. Srivastava, and P. C. Negi, "A hybrid data mining model to predict coronary artery disease cases using noninvasive clinical data," *Journal of Medical Systems*, vol. 40, no. 7, p. 178, 2016.

[29] H. Sharma and M. A. Rizvi, "Prediction of heart disease using machine learning algorithms: a survey," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 8, pp. 99–104, 2017.

[30] K. Uyar and A. İlhan, "Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks," *Procedia Computer Science*, vol. 120, pp. 588–593, 2017.

[31] A. G. Sorkhi, J. Pirgazi, and V. Ghasemi, "A hybrid feature extraction scheme for efficient malonylation site prediction," *Scientific Reports*, vol. 12, no. 1, pp. 1–16, 2022.

[32] S. M. R. Hashemi, H. Hassanpour, E. Kozegar, and T. Tan, "Cystoscopic image classification by unsupervised feature learning and fusion of classifiers," *IEEE Access*, vol. 9, pp. 126610–126622, 2021.

[33] M. A. Hall, *Correlation-Based Feature Selection for Machine Learning*, Citeseer, 1999.

[34] N. Sánchez-Marono, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection–a comparative study," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 178–187, Springer, Berlin, Heidelberg, 2007.

[35] S. Ding, "Feature selection based F-score and ACO algorithm in support vector machine," in *2009 Second International Symposium on Knowledge Acquisition and Modeling*, vol. 1, pp. 19–23, Wuhan, China, 2009.

[36] P. Jiang, W. Ning, Y. Shi et al., "FSL-Kla: a few-shot learning-based multi-feature hybrid system for lactylation site prediction," *Computational and Structural Biotechnology Journal*, vol. 19, 2021.

[37] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[38] M. Rahimi and M. A. Riahi, "Reservoir facies classification based on random forest and geostatistics methods in an off-shore oilfield," *Journal of Applied Geophysics*, vol. 201, article 104640, 2022.

[39] M. A. Khan and F. Algarni, "A healthcare monitoring system for the diagnosis of heart disease in the IoMT cloud environment using MSSO-ANFIS," *IEEE Access*, vol. 8, pp. 122259–122269, 2020.

[40] J. Pirgazi, A. R. Khanteymoori, and M. Jalilkhani, "TIGRNCRN: trustful inference of gene regulatory network using clustering and refining the network," *Journal of Bioinformatics and Computational Biology*, vol. 17, no. 3, article 1950018, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2234962, 13 pages
https://doi.org/10.1155/2022/2234962

Research Article

Workplace Flexibility Analysis in Cyberphysical Human Systems Using Cloud-Enabled Deep Autoencoder Networks

Danial Farashaei,[1] Amin Honarbakhsh ,[1,2] Seyed Mojtaba Movahedifar,[1] and Eghbal Shakeri[3]

[1]*Department of Civil Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran*
[2]*New Materials Technology and Processing Research Center, Department of Civil Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran*
[3]*Department of Civil and Environmental Engineering, Construction Engineering Management, Amirkabir University of Technology, Tehran, Iran*

Correspondence should be addressed to Amin Honarbakhsh; aminhonarbakhsh1985@gmail.com

Received 25 February 2022; Revised 8 April 2022; Accepted 21 April 2022; Published 3 June 2022

Academic Editor: Mohammad R Khosravi

To examine the correlation between worker safety, workplace interpersonal problems, and individual flexibility within a cyberphysical human system (CPHS), we employed a stacked autoencoder (SAE) approach and a cloud-based computing environment. The study's statistical population includes construction companies in Mashhad, Iran. To collect data, descriptive surveys and applied research approaches are employed. Thus, data is collected using a cloud-based platform, data processing tools, and information analysis methods. It is our main objective to figure out how to reduce construction accidents and make people safer. Our study used a sample of 200 people to study the entire study population because it is difficult to study the entire study population. There were 151 valid questionnaires collected after the questionnaire distribution. We developed a 28-item questionnaire as part of the study in addition to the Questionnaire on Experience and Evaluation of Work (QEEW). Implementing an optimized SAE network can reduce dangerous situations, physical injuries, supervisor conflict, workplace stress, interpersonal conflict, and colleagues' involvement. As a consequence of the large amount of data needed for quick analysis and mechanism construction, cloud computing performed admirably. The study of interpersonal conflicts and individual flexibility among construction workers was necessary because only limited research had been conducted on these topics.

1. Introduction

To achieve the company's goals in the urban and mass construction industries and to establish a successful CPHS ecosystem, human resources (HR) are crucial [1]. These businesses would not be able to accomplish their objectives if people's demands were not taken into account [2]. Human resources play a crucial part in the growth of urban centers and mass construction as a result of this. By employing a sound HR management philosophy, businesses may strike the right balance between employee and management needs [3]. The key goal of this phase is to establish a pleasant working atmosphere and provide employees with adequate working circumstances. Workplace safety is a hot concern in HR management [4, 5]. According to the evidence, employers, contractors, and employees may have been negligent. It is critical not to overlook the current flaws and legal loopholes [6]. Occupational psychologists have focused their emphasis on the stress and pressures linked with the workplace in recent decades [7]. The majority of mental illnesses and tension problems are caused by occupational stress. As a result, it is critical to address this problem and better understand workplace stress.

Depending on the nature of their employment, human resource professionals may be exposed to a number of mental health difficulties. Due to stress, an individual's capacity to do work and its needs may have a disproportional relationship [8].

Due to workplace inequities, workplace conflicts or interpersonal disagreements are likely. People's thoughts and attitudes are influenced by their psychological, social, and personality qualities [9]. As a result of these professional problems, personal conflicts emerge [10, 11]. Because of interpersonal disputes, there may be a lot of tension, dread, and distrust in the workplace [12, 13]. These interpersonal conflicts have profound behavioral and functional effects. Workplace interpersonal conflict has been demonstrated to negatively influence productivity and employee morale [14], as well as organizational commitment [15].

Adaptability is a must-have skill in today's world of developmental pathology and mental health. Flexibility was once thought to result from a person's ability to be self-aware and adaptable. In the face of adversity, people are capable of adapting and reaching beneficial outcomes. External influences, on the other hand, play a role in resilience. Researchers now consider various factors when evaluating individual flexibility, such as problem-solving skills, personality, temperament, environmental events, and challenges. Flexibility is a dynamic procedure that includes adaptation in the face of adversity. As a result, a flexible person possesses several outstanding qualities. These abilities include the ability to take calculated risks and improve conditions. Positive attributes from the inside and out, as well as other factors, can help to improve a situation. In light of the preceding, this study was aimed at looking at conflicts between construction site supervisors and coworkers. When it comes to physical and mental health concerns, the study will look at characteristics such as gender, age (male or female), challenging work situations, work experience, and job features to see if there is a link between these factors and physical, mental, and health problems.

In order to improve the accuracy of software systems' predictions without requiring explicit requests, machine learning (ML) is used [16, 17]. Predictions about future output values are generated by ML algorithms based on historical data. In addition, cloud computing refers to Internet-based services that are delivered to users over the Internet [18, 19]. This type of resource consists of data storage and retrieval devices, networks, and software. In recent years, the integration of machine learning and analysis-oriented methodologies has simplified information processing in the field of networking. Cloud computing allows anyone and everyone access to numerous technologies without being an expert in each. Using the cloud will benefit consumers since they can focus on their main business rather than worrying about technology (IT).

Accidents and fatalities are common in the construction industry, making safety a crucial concern. Occupational conflicts and injuries are common in construction, which is a high-risk occupation. However, to meet the current development and progress challenges, quality human resources are crucial. Given the risks workers face in this industry, human resources need to be of higher quality. Due to these divergent viewpoints and attitudes, conflicts and disagreements often arise regarding how resources should be used. In addition, how to account for the performance of artisans, employees, and supervisors is crucial to examine the relation between flexibility and interpersonal conflicts. Because the study was aimed at investigating interpersonal conflicts and flexibility, the research was aimed at address the following issues: (1) academic level, (2) daily job hours, (3) intellectual background and mental, and (4) adaptability skills.

A deep neural network (DNN) [20] is a method of ML that can self-learn and produce outputs independent of their inputs. Rather than storing data in databases, data is now stored in networks. Cloud computing provides the perfect platform for deep learning. Automation of large-scale data analysis is made possible by using deep learning and cloud computing. Furthermore, it enables employees to store information related to interpersonal conflicts at work securely.

The preceding cases examine interpersonal conflict. To be able to minimize the impact of interpersonal conflict, a suitable and accurate process must be created using a fusion and deep hybrid model of an optimal deep neural network structure.

The research and analysis of various studies may include additional significant components. Can individual flexibility affect interpersonal conflicts? This study was aimed at answering that question.

We consider individual adaptability, interpersonal tensions, and autoencoding of the output through fusion-hybrid algorithms.

We examine the correlation between interpersonal difficulties at work and physical safety utilizing a fused autoencoder model (stacked autoencoder) and cloud environment. In our previous study, artificial neural networks were used to examine workplace conflict and individual flexibility by collecting cloud-based information and integrating neural networks [21]. In light of this, we employ autoencoder network fusion rather than neural network fusion. After studying the relationship between worker safety, workplace conflict, and individual adaptability, we also developed our high-performance computerized system.

The following aspects have been significantly improved as a result of our initiatives:

(1) An enlarged deep decision-making structure was used to investigate the relationship between physical safety outcomes and workplace interpersonal differences

(2) This study examines the association between workplace interpersonal conflicts and job tension and stress by employing an optimal deep decision-making model

(3) The enhanced-fusion deep SAE used by cloud computing has improved data integration and security

In the second section, we discuss related efforts. In Section 3, an optimization technique is applied to study the relationship between interpersonal disagreements and individual flexibility. Section 4 delves more into the findings and conclusions. Section 5 concludes with a synopsis of the major issues and conclusions.

2. Related Work

Interpersonal conflicts are considered a violation of an organization's norms, regulations, and formal procedures. In addition, an aim-based systematization approach allows it to be further subdivided into conflicts with coworkers and conflicts with the organization. The result may be understaffing, absenteeism, furniture and equipment damage, and bizarre conduct toward coworkers and strangers. Conflict-related emotional states and attitudes have caused aberrant behavior in individuals and settings for many years [22]. Conflict in relationships has a cascading effect on all parties involved. The avoidance of this situation is not only impossible, but instead, it fosters greater appreciation for the origins and significance of human connections. Relationships with no conflict are not as interested as those with conflict [23].

Long-term problems can result in violence if they remain unresolved. A harmful effect of emotions, anxiety, and defense on initial bonds links interpersonal conflict inextricably [24]. The term interpersonal conflict refers to conflicts with other people. A dispute of this kind weakens relationships by causing tension, anxiety, and other unpleasant emotions that compromise one's ability to perform daily tasks. Relationship difficulties are associated with a higher incidence of self-harm. Interpersonal conflicts can be exacerbated by a variety of factors, such as disagreements about ideas, priorities, values, and motivations, as well as cultural differences [25, 26]. The construction industry shares specific characteristics with organizational conflicts. Disputes occur when individuals try to accomplish objectives that oppose one another, resulting in a waste of money and time. It is necessary to understand the subject in order to determine if a conflict exists. Conflict can have detrimental effects on human resources [27, 28].

According to research, deviant behavior is more prevalent when people are at odds with their coworkers or superiors [29]. "Work stress," which can be triggered by a range of conditions, is one of the most prevalent types of conflict-related stress. The causes of workplace stress may include inadequate reward systems, poor payment systems, job insecurity, fear of job loss, physical characteristics, a low or high workload, or night shifts. Engineering and building involve such a broad range of operations that conflicts resulting from them are particularly common. Construction is one of the most dangerous industries in the world due to the high rate of accidents and the absence of workplace control. Although it contributes to a significant part of the global economy, it is unreliable because of its complexity, danger, and unsanitary working conditions [30]. Construction site accidents are second only to mine accidents in terms of frequency. According to current figures, the construction industry is responsible for around 30% of all occupational accidents in the country. Accidents in this industry result in a 15% fatality rate, which is a notable figure. On the other hand, the bulk of these occurrences is tied to job conflicts and the stress that accompany them.

Stress at work is commonly attributed to interpersonal conflicts at work (ICW) [31]. Conflicts between ICWs and their managers and coworkers are two of the most common ICW types on construction sites. Also examined were factors that differentiate interpersonal disputes from aberrant behaviors, as well as direct and indirect effects of interpersonal conflicts on deviant behavior. A key component to this equation is personal adaptability. There is no universally accepted definition of personal flexibility despite substantial research on the topic. Although it was first investigated about four decades ago with a peak of interest in the mid-1980s, academics have studied it for years. As well as participating in meaningful activities regularly, psychological well-being is characterized by a positive attitude toward oneself and others [32]. Also, it requires adapting one's thoughts and actions to continually changing conditions [33].

3. The Suggested Procedure

Figure 1 illustrates the general framework of the proposed method for recognizing conflicts created in the workshop environment. Following is a description of each section of the method.

3.1. Statistical Characteristics and Data Collection. Statistical methods are used to analyze the numerical data collected from samples. There are two primary characteristics of quantitative analysis: (1) Evaluating individual and organizational characteristics by collecting data and (2) experimental studies, correlational studies, and surveys can all be used to conduct comparative research. It is possible to classify quantitative research as descriptive (comparative and correlational), relational (comparative), or practical (quasiexperimental, real, and single case).

In the paper, which is based on the process analysis method developed by Chen et al. [34], safety precautions and stress levels of construction workers are discussed.

The results of the analysis are generalized after data collection and sample examination. These methodologies need to produce scientifically generalizable and probabilistic results after data collection and sample examination. Many strategies exist for selecting samples for this purpose [35]. We used a judgmental sampling approach in our study. Although certain facilities are easy to access, this sampling strategy probably limits generalizability, but it is the only way to gather data from specific individuals in a statistical population. It is necessary to use a full system with specific properties in order to collect the relevant data. We chose research subjects based on the following criteria:

(1). A construction project that is progressing at a rate greater than 60%. In addition to the fact that many people work in such an environment on a daily basis, many construction projects are underway to further realism

There must be a minimum of 10,000 square meters of infrastructure. The single criterion led to the exclusion of many traditional structures from the list.

(2). The project involves construction over four stories in height. The purpose of this criterion is to ensure that the selected structures are at least a certain height above the surrounding ground. Altitude, however, increases the likelihood of errors and hazards. Operators at heights must pay more

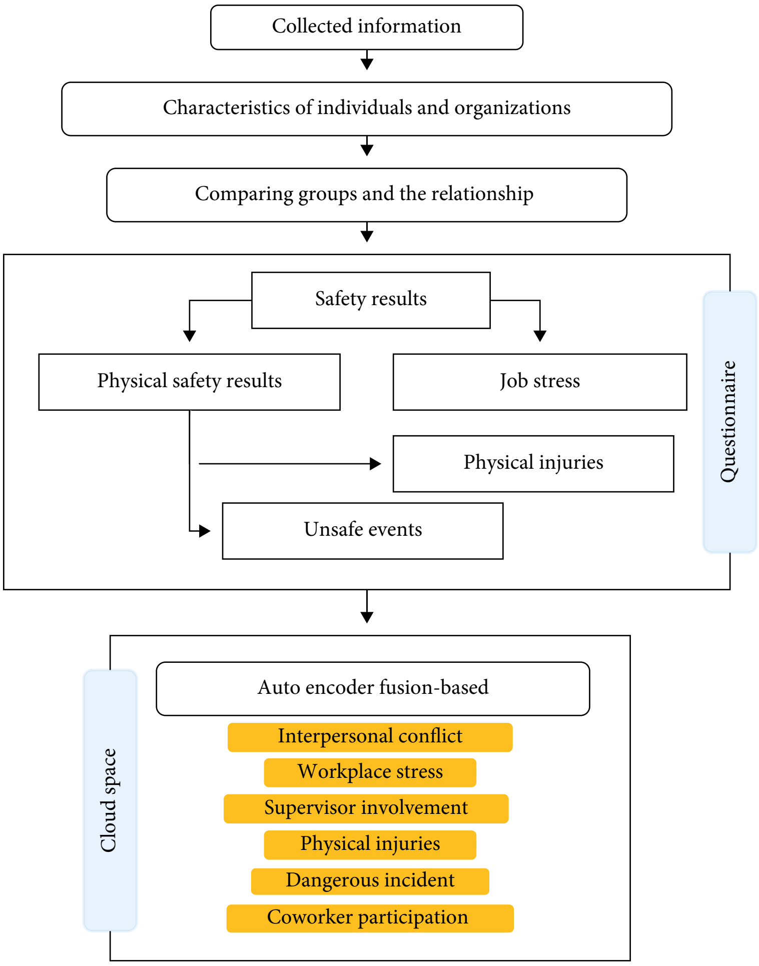


Figure 1: As shown in the figure, the proposed method consists of two common parts: data collection and processing.

attention to safety instructions due to the increased level of attention required

In conclusion, determining the sample size for research is an important step that should be taken at the beginning of a project. Insufficient samples make it impossible to generalize the results to the statistical population; however, even large samples are not necessarily indicative of reality, which is why monitoring is fundamental. In order to avoid wasting money, labor, and other resources during a study, the sample size should not be excessive. According to the aforementioned criteria, twenty construction sites in Mashhad, Iran, were selected for the study's sample of construction workers. Ten questionnaires were provided to each construction site, and 151 questionnaires were returned by participants.

A standard questionnaire with items on a five-point Likert scale is the primary data collection tool. Typically, respondents were asked to fill out a written questionnaire. As construction workers are often illiterate, the researcher or his assistants could also fill out the forms by interviewing them. We can be reached in a variety of ways, including by phone, email, or any other method. Questionnaires can be mailed and then returned. The questionnaire should be distributed and received online in a web-friendly format or via email if the necessary facilities are available. As a result, the number of respondents increases, while the speed and precision with which the survey is conducted, received, evaluated, and reported increases. During the fieldwork phase of the study, the authors explained the questions to participants and instructed them how to answer them.

3.2. Analysis Procedure. A questionnaire and statistical approaches were used in the current study. Data from the survey was imported into MATLAB before SPSS was used to analyze it. Lastly, a cloud server can be used to update the processing tools. When Van Weldon and Meyman developed their own questionnaire in 1992 and 1994, they drew inspiration from the QEEW [34, 35]. Studying workplace health was conducted for the purpose of gaining a better understanding of it. The 28 items in this survey are divided into seven subcategories (i.e., conflict with colleagues, unsafe events, physical injuries, conflict with supervisors, physical safety outcomes, job stress, and interpersonal conflict). A questionnaire's validity can be determined by Cronbach's alpha coefficient. Validity and reliability refer to the consistency of results over time and similar circumstances.

In order to determine the skewness of one's opinions, attitudes, and beliefs, Cronbach's alpha coefficient can be used. Scales can be used to quantify the consistency with which respondents answer survey questions. Diverse individuals, things, and actions are assigned numerical values along a continuum. Since it is based on the concept of comparability, the Likert scale is the most commonly utilized scale in social study. The components of the survey have been assigned a numerical value (e.g., a Likert scale of 1-5). An individual's proclivity is determined by the sum of their scores. There is a formula for computing Cronbach's alpha in the following section:

$$\alpha = \left(1 - \left(\sigma^{-2} \times \sum_{i-1}^{k} S_i^2\right)\right) \times k \times (k-1)^{-1}, \qquad (1)$$

where σ, S^2, and k represent the variance of the total scores, the variance of item i, and the number of scale items, respectively. In light of the definition of Cronbach's alpha, the following conclusion can be reached:

(i) According to Cronbach's alpha, the strength of the relationship between the questions can be determined

(ii) When the mean variance is large, it is difficult to use Cronbach's alpha

(iii) With an increase in the number of items, Cronbach's alpha changes either positively or negatively according to the type of relationship between the questions

(iv) As sample size increases, Cronbach's alpha increases since mean-variance decreases

3.3. Fusion of Autoencoder Models. Automatic encoders are used to create stacked autoencoders (SAE). An autoencoder's hidden layers are connected to the hidden layers of the next autoencoder in a neural network. Training requires that the previous autohidden encoder layer be used as an input to the following one. The SAE architecture that was used in this experiment is shown in Figure 2. The SAE can be used to create new abstractions by stacking existing ones on top of each other. Following the reconstruction of the hidden layer using data collected via questionnaires and statistical analysis, the final output combines the high-level characteristics of employees and coworkers. Properties determine an object's conductivity distribution. Logistic regression is used to determine the conductivity distribution.

High-level characteristics are presented regarding the DNN. $U = \{\mathrm{G}(1), \mathrm{G}(2), \cdots, \mathrm{G}(\mathrm{M})\}$ is one of the symbols used, while M (training set number) and $G(k) \in [0,1]^m$ are the other two (normalized characteristics). The letter m denotes an unknown number of attribute values in a collection of randomly generated attribute sequences.

3.4. Cloud Computing. The mean, standard deviation, variance, and median of a set of data were calculated using SPSS software. SPSS also includes data management and mining. MATLAB can also be used to create mathematical and statistical software as well as user interfaces. Debugging, creating m-files, and modifying workspace variables are also available. MATLAB 2021b was the only version that included Fusion SAE and design analysis capabilities to users. Our qualitative research was carried out using SPSS.

Since data may be sent from a mobile device to a cloud computing center, mobile cloud computing can be implemented into the structure. Base stations develop and operate network and device interfaces to be simple to use (which may be transformers, access points, or satellites). Servers connected to mobile network services get information about workers' whereabouts and identities in a mobile work environment. Mobile networks can monitor data such as employee authentication and access privileges within this important context. On-the-job experts can access databanks holding critical information about agents and subscribers. Furthermore, industry experts strongly recommend data movement from the Internet to the cloud. Customers can make identical cloud service requests using their mobile devices thanks to cloud controllers in the cloud. Web servers, apps, and databases can all be constructed using virtualization and service-oriented architecture (SOA).

4. Experimental Results

This study focuses on Iranian construction workers in Mashhad. The descriptive information we use in our study include gender, work hours, employment history, and age. Figure 3 shows the data on work experience, hours worked, and age factors, but there are no female construction workers.

Moreover, Figure 3 shows that 44.8 percent and 45.7 percent of those surveyed who worked between 6 and 8 hours a day experienced fatigue and decreased safety, respectively. With 17.9% of those aged 20 and below, 33.1% aged 21-30, and 28.5% aged 31-40, and there are 20.5 percent of those over 40 in the study, as shown in Figure 3. As shown in Figure 3, only 15.9% of workers have less than five years of work experience, 27.2% have six to ten years, 23.8% have eleven to fifteen years, and 33.1% have more than fifteen years, respectively.

4.1. Software Environment Setting. Analytical and statistical tests were conducted in the 2020 MATLAB programming environment, and the results of the simulation of a software model were included in the statistical testing. Intel (R), Core (TM), and Core i7 CPUs, 8 GB of RAM, and a 64-bit operating system power the modeling system.

In order to minimize overfitting, ten different architectures were tested, each utilizing the DNN with a few modifications in the first hidden variable layer (among the tested folds) in order to find the design with the lowest mean square error (MSE). Thus, if the mean error square factor is less than a predetermined value (0.05), the selected network is chosen as a starting point; if it is larger, the least MSE and corresponding structure is chosen.

It is crucial that the DNN model has a sufficient number of hidden layers in order to be able to learn. Our models are often fine-tuned by varying the learning rates and number of

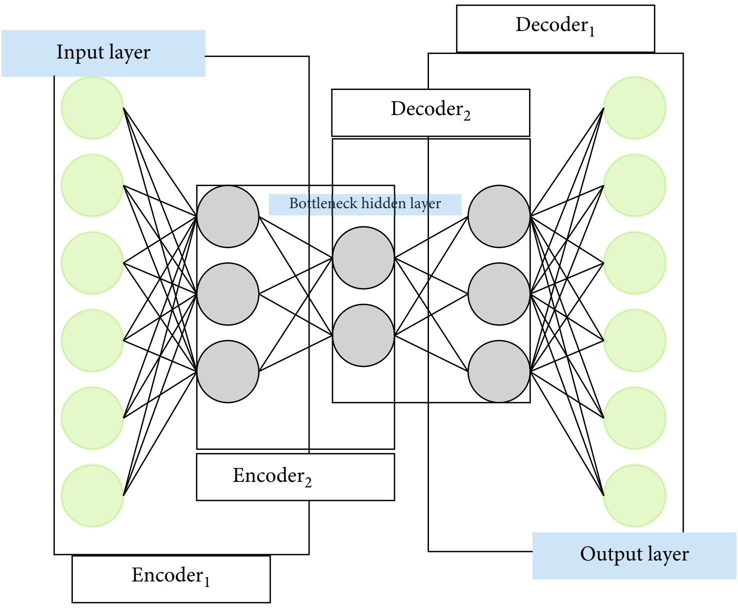


Figure 2: The configuration of the suggested model as SAE classifier.

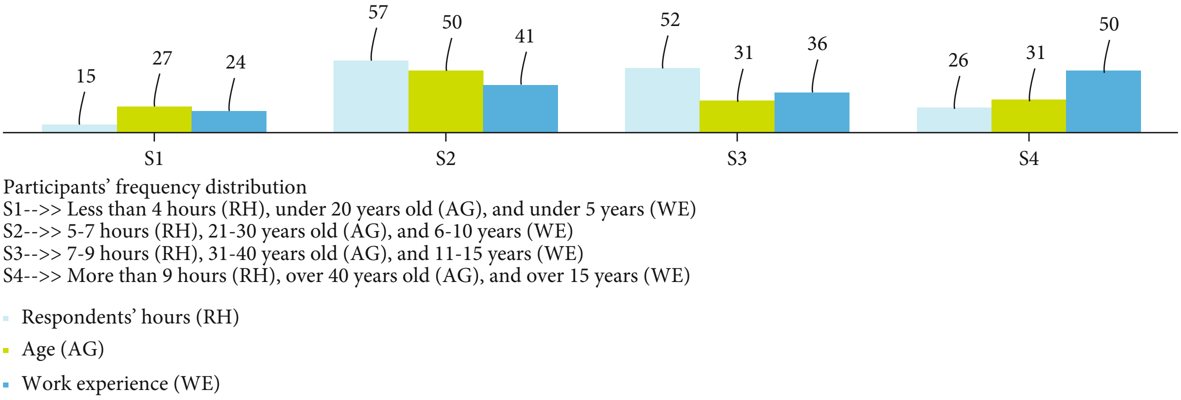


Figure 3: Participants' frequency distribution demographic scheme of respondents based on the working hours, age, and work experience.

iterations for each layer. Generally speaking, we construct networks of 3 to 5 layers. A minimum of 150 nodes were present in all three layers of the hidden and input layers, respectively. Pretraining learning rates decreased by approximately 0.1 to 0.01 based on the training stage. We fine-tuned the system by iterating between one and five hundred times. In the pretraining session, each ten-person group was subjected to 100 epochs of training in the same network configuration, among others. At this point, the batch size was reduced to 40. The MSE was used to evaluate the performance of the network.

4.2. Inferential Outcomes. To assess the relationship between variables, we used the correlation matrix (CM) test. Table 1 demonstrates the correlations among the various variables. In this table, Var1, Var2, Var3, Var4, Var5, Var6, and Var7 are conflict with colleagues, insecure events, physical injuries, conflict with observers, physical safety results, job stress, and interpersonal conflict, respectively.

Statistically, all the factors have a significant correlation. An association exists between injuries, risky situations, unhappiness with coworkers, interpersonal conflict, and conflicts with observers. Individuals with high degrees of interpersonal conflict, including conflict with supervisors ($r = 0.362$), unsafe working conditions and injuries ($r = 0.351$), and conflict with coworkers ($r = 0.354$), are more likely to report high degrees of stress in the workplace.

Anxiety in the workplace is closely associated with conflicts with supervisors ($r = 0.718$), injuries, accidents, and other dangerous situations ($r = 0.856$), as well as disagreements with coworkers ($r = 0.926$). Insecure situations ($r = 0.646$), physical injuries ($r = 0.642$), and disagreements with coworkers ($r = 0.689$) all have strong correlations. Workplace issues seem to have a strong correlation. The correlation coefficient ($r = 0.830$) shows that coworker insecurity and disagreement are related.

The descriptive statistics and relationships between variables are presented in Table 1. These results reveal that the

Table 1: This table shows the CM test between various variables.

No.	Var.	Avg.	STD	1	2	3	4	5	6	7
1	Var1	9.73	2.41	0.387*	0.918**	0.844**	0.712**	0.838**	0.835**	1
2	Var2	9.42	2.56	0.289*	0.858**	0.783**	0.644**	0.739**	1	—
3	Var3	9.96	2.40	0.367*	0.833**	0.794**	0.676**	1	—	—
4	Var4	9.76	3.03	0.356*	0.712**	0.674**	1	—	—	—
5	Var5	9.85	2.65	0.328*	0.848**	1	—	—	—	—
6	Var6	9.91	2.45	0.362*	1	—	—	—	—	—
7	Var7	9.88	2.23	1	—	—	—	—	—	—

Moreover, the star sign is $p < 0.005$, and the double star sign is $p < 0.001$ correlations.

variables management commitment to safety, observer perception of safety, peer perception of safety, safety knowledge, and individual adaptability all have a negative correlation with symptoms. On a bodily level, they are affected by dangerous events and stress-related ailments. Positive relationships were found between occupational stress and overtime, dangerous occurrences, and psychological stress indicators. Table 1 demonstrates a strong correlation between management commitment to safety, observer safety perception, peer safety perception, safety knowledge, and individual adaptability. Overtime was found to be positively associated with job stress. Positive correlations were found between physical symptoms, dangerous incidents, and work stress. To summarize, management commitment to safety and observer perception of safety showed the strongest negative correlations with physical symptoms and risky incidents; peer perception of safety had the strongest negative correlations with psychological stress symptoms. Work stress was found to have the most negative associations with physical symptoms, dangerous incidents, and psychological stress.

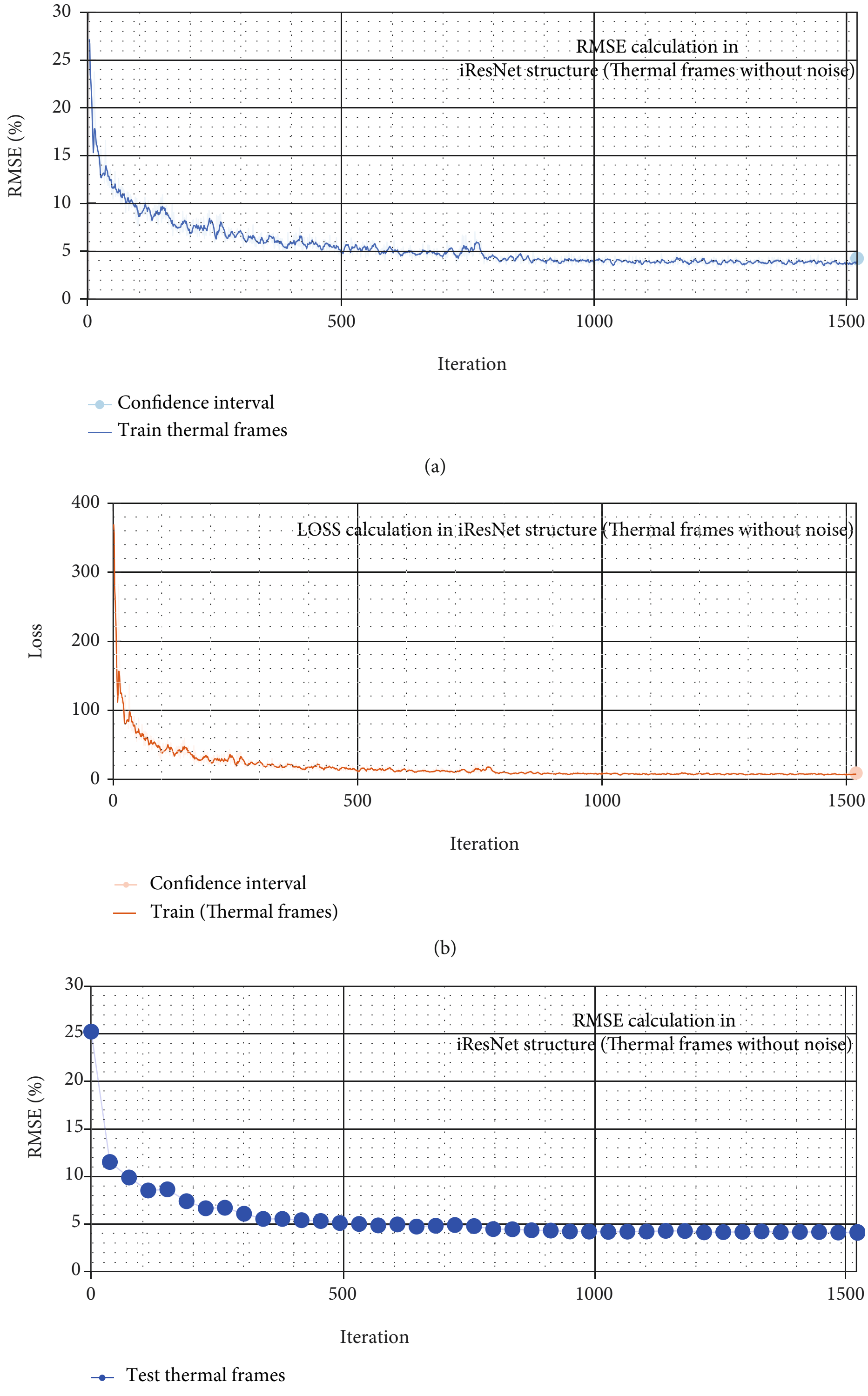
In the final evaluation of the generated models, we look at the impact of various input factors on error amounts. Sensitivity analysis is required for every SAE model to get rid of superfluous input. The information cost of the model drops when this data is deleted, and the accuracy of the model increases. Goal functions are supposed to target safety outcomes for employees. The variables or control parameters affect the output function based on the input information (e.g., physical injuries, interpersonal conflict, unsafe events, job stress, conflict with supervisors, and conflict with colleagues). Every phase has its own set of control variables that affect how quickly and thoroughly the input layers and hidden input function are examined. Several variables other than the dependent variable were examined to test the key hypotheses.

4.3. Model Analysis. In comparison to other approaches within the same family, the SAE network is a lot more accurate at classifying attributes. We provide the root mean square error (RMSE), convergence, and loss functions, as well as the smallest RMSE. Further, the layer-wise reduction factor might be increased to reduce complexity. Besides, a small change in RMSE means that the features in a decision-making and segregation network are not properly separated. Through the SAE network and a few repetitions, a very large data set from workplaces can be reduced to the most significant patterns. In real-time or near-real-time applications, feature values may be beneficial, but they may not be necessary in other cases. For each set of workplace scenarios, Figure 4 shows the convergence and loss functions.

4.4. Discussion. Knowledge of risk, risk management, safety regulations, and procedures influences construction workers' attitudes toward safety. According to the introduction, there is a link between these attitudes and workers' safety-related activities. According to Chen et al., [34] there is a lack of research on worker safety and interpersonal conflict, particularly in underdeveloped nations. The authors used the data from building workshops to examine how an individual adapts and conflicts with colleagues.

Questions for the questionnaire's safety component were derived from prior studies conducted in high-risk industries like construction. It was determined that seven variables needed to be considered before the final questionnaire could be developed. People's proclivity for interpersonal conflict is influenced by many factors, including the amount of stress they experience at work, the hazards they encounter while working, and their relationship with their supervisors. Physical injuries, unsafe environments, and clashes with coworkers can exacerbate interpersonal conflict. This study also incorporates earlier research findings. Road construction and injuries are two factors contributing to unsafe behavior. Employees' attitudes toward safety, as well as their adherence to and participation in it, are assessed in this study. Additionally, staff commitment and involvement were found, as well as an understanding of the importance of workplace safety [36].

According to a review of the articles, some research supports the comparison of safety findings across industries. According to Chen et al. [34], the building industry in Ontario has a safety environment of 35. The highest score for the overall safety climate was "neither agree nor disagree," which received 3.69 out of a possible 5. As a result, construction employees do not work in an environment that promotes safety. The disparity in overall safety environment rankings could also be explained as a result of the high value placed on safety in the US based on various methodologies or sample sizes. It is likely that these two variables are

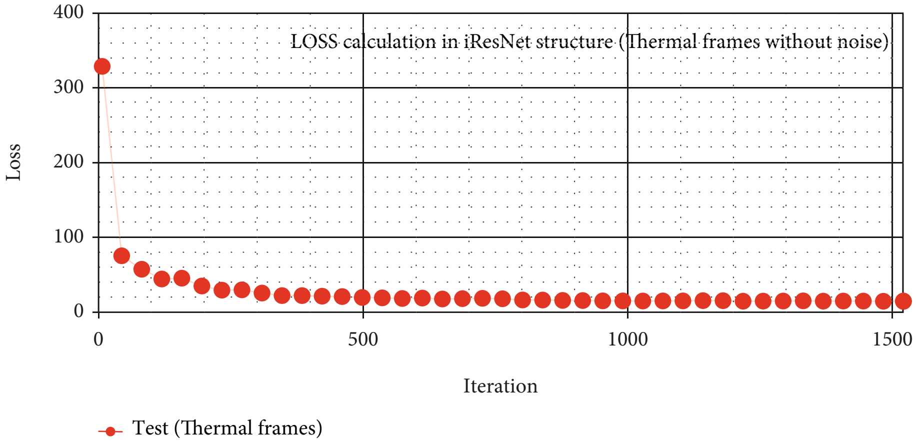


(a)

(b)

(c)

Figure 4: Continued.



(d)

Figure 4: This figure depicts the RMSE convergence and loss function of the workplace flexibility analysis, (a) and (c) are the RMSE calculation based on train and test samples, and (b) and (d) are the LOSS computation based on train and test samples.

responsible for the disparity between the total scores in the safety environment. The comparison will help ensure that the workshop's safety culture remains strong by managing the field of safety training based on the work procedures that are regularly employed. Construction and manufacturing industries use this practice regularly.

When developing safety training programs, more attention should be paid to changing employees' attitudes about risky acts and environmental hazards they have to pass through. Injuries related to this job are likely due to the nature of the working relationship. A lack of interest by management in addressing unsafe situations may have encouraged those who work in hazardous environments to keep doing so. A risky act or a danger already present in the workplace could cause it. Using workers' own personal experiences, researchers can infer their reactions to potentially hazardous situations. Even if they have not been involved in a personal accident, they are still at risk. But they had no idea what was happening.

According to a study by Chen et al. [34], workers' safety performance is directly linked to psychological stress. As part of a study of construction disasters, Haslam et al. [37] conducted focus groups. Approximately, 70 percent of the incidents were caused by worker or team issues; 56 percent were caused by equipment (including personal protective equipment); 27 percent were caused by the propriety and condition of materials; and 27 percent were caused by inadequacies in risk management. Steel construction workers were studied for their safety behavior and coping mechanisms. The researchers found a significant correlation between an individual's age and their safety behavior ($p = 0.016$; $r = 0.301$).

The correlation ($p = 0.260$; $r = 0.315$) did not mention the relation between education level and safety behavior. The researchers found a relationship between tenure and safe work practices ($p = 0.001$; $r = 0.422$). Therefore, the association ($p = 1.0$; $r = 0.015$) was insufficient to define coping strategy and safety behavior appropriately.

There is some evidence between psychological stress (such as workplace stress and job satisfaction) and safety performance, according to a study by Siu et al. [38]. Examples include accident rates reported by workers and workplace injuries. A questionnaire was given to construction workers in 27 Hong Kong development zones. Worker safety in hazardous situations can be enhanced by recognizing a variety of stressors that affect two distinct categories of workers, said Leung et al. [39]. Study participants were divided into two groups based on the type of stress they were experiencing: work-related stress or emotional stress. Construction workers' emotional stress was found to be the most influential risk factor for occupational injury events, while job overload and interrole conflict predicted emotional stress.

As demonstrated in this section, which focuses on correlations between single elements, there is no link between workers' well-being and interpersonal conflict. At the $p < 0.05$ significance level, work-related stress, conflicts with supervisors, physical injuries, dangerous accidents, and colleague participation all had a significant association.

4.5. Our Inferences. The study's findings verified a number of the study's assumptions, including the following:

(i) Interpersonal conflicts in the workplace have a beneficial effect on physical safety

(ii) ICWs are associated with occupational stress

(iii) ICWs and bodily injury are inextricably linked

(iv) Workplace injuries are related with conflict among coworkers

(v) ICW conflict is positively associated with instances of occupational insecurity

(vi) There is a correlation between increased workplace stress and supervisor engagement

(vii) There is a correlation between insecurity and conflict with observers (ICWs)

The following concerns have been identified as a result of this analysis:

(1) A weak safety culture is inextricably connected to a lack of success in adopting safety measures. In the absence of a safe environment, dangerous behavior may be encouraged. Unfortunately, construction workers have learned their craft by trial and error, which has resulted in the spread of detrimental practices using the workplace as a model

(2) This study discovered a high correlation between construction site accidents and a lack of knowledge of potentially hazardous conditions

(3) When it comes to ML and cloud computing, data security is critical. Cloud computing, which refers to a shared pool of programmable computer resources, can be utilized to establish a demand-based network. Additional alternatives include collocated application programming interface (API), API-enabled programs, and cloud computing services such as cloud desktop storage and cloud data gateways. This means that it is simple to install and manage, taking only a few minutes. Cloud computing, the study found, has a considerable impact on the creation of employee ties

Historically, safety criteria were not generally accepted, and cultural differences were shown to affect them. Since different entities enforce different management and safety requirements, they influence elements in one business that may be wrong in another. This study was placed on a construction site in an underprivileged country such as Iran.

Due to the particular nature of construction management approaches and circumstances, such as the presence of contractors, workers who are primarily involved in experimental building, and employees who lack technical skills, establishing a safety culture in construction workshops is challenging. Safety will become much more critical in this field of employment. Corporate culture and employee attitudes can contribute to the removal of safety barriers. To enhance workplace safety, it is critical to identify areas for improvement and then work to reinforce those areas. When safety procedures are followed properly, workplace incidents decrease, and safety initiatives are more successful. Both management and employees must adjust to these developments. The authors' recommendations for more research are aimed at addressing the study's limitations:

A toolbox meeting is being portrayed as a training session. Daily, workshop employees are exposed to potentially hazardous situations and are taught how to recognize them. Worker safety seminars should be held to educate employees on safety programs and build their trust in management's ability to perform them.

When workers are educated experimentally to work in experimental environments, these behavior patterns are abolished, and workers are encouraged to engage in appropriate actions when confronted with hazardous conditions; they are less likely to engage in insecure behaviors. Priority is given to employee well-being over all other issues: accountability and duty on the part of managers for promoting safer working environments and procedures. Any business should prioritize training its employees on potential workplace hazards and how to avoid them. Regular interaction between managers and employees, as well as cloud-based technologies that make it simple to get safety-related information at work, should be established as a way of accessible workplace safety communication. Each job has unique safety equipment that must be kept on hand at all times. Cloud computing offers the ability to improve responsiveness and precision in issue solving while also saving money and time.

DL should be utilized to address problems and challenges such as low accuracy in different ML systems rather than traditional ANN [40]. Even though artificial neural network outputs are intended for learning, DL has been employed to overcome a variety of categorization difficulties. According to industry experts, cloud-based data processing should also be more accurate and secure.

Concerns about cloud computing's complexity are legitimate. Before embarking on any cloud computing strategy or implementation, it is necessary to conduct a thorough study of all data sets, services, and workloads [41]. Automation and abstraction were employed to handle the cloud computing tools and information management. This section of the manual discusses procedures and field tools, as well as data entry and justification for resource deletion or update.

4.6. Challenges and Future Aspects. Data management and analysis were shown to be protected by machine learning and cloud computing in this study. A demand-based network can be built using cloud computing, a shared collection of programmable computer resources. More options for cloud computing include APIs for web services or API-enabled programs such as cloud desktop storage and cloud data gateways. Getting this up and running will only take a few minutes. The study found that cloud computing has a significant impact on the development of employee connections. Before this discovery, there was no consensus on safety issues because of cultural differences. Different management and safety standards can impact the influencing elements in one industry. An impoverished nation like Iran was the setting for this study. The challenge of establishing a safety culture in a construction workshop is due to construction-specific management approaches and conditions such as the presence of contractors, experimental workers, and employees who lack technical skills. In this line of work, safety will become increasingly imperative.

Safety barriers can be reduced by changing organizational culture and employee attitudes. It is most effective to identify the areas that can be improved in order to promote workplace safety. As a result, workplace accidents are reduced, and safety initiatives are more effectively implemented. Both management and the workforce must adopt these changes. In addition to pointing out the study's weaknesses, the authors have made recommendations for future research: a training session called a "toolbox meeting" is being conducted. The factory workers are constantly exposed to potentially hazardous conditions, and they are taught to recognize them.

In order to provide workers with a better understanding of safety procedures, training sessions should be scheduled. As a result, workers will feel more confident about management's ability to enforce them. Workers in a dangerous environment must experiment to learn effective skills and avoid ingrained ineffective habits. This is why it is imperative to train them in this way. Employees' health and happiness must always come first. It is the management's responsibility to ensure the safety of workers. Companies should focus on educating employees about workplace hazards and how to prevent them.

Regular employee and management interactions, as well as cloud-based services that make it easy to obtain safety-related information, should be implemented to facilitate accessible workplace safety communication. Regardless of the time of day or night, safety equipment should always be readily available for any job. In addition to improving responsiveness and problem-solving capabilities, cloud computing reduces costs and enhances timeliness. For problems like low accuracy when it comes to machine learning systems, deep learning should be used instead of traditional artificial neural networks. In addition to addressing many categorization problems, deep learning was developed to develop the outputs of artificial neural networks for learning. Experts believe that cloud computing should improve data processing accuracy. Cloud computing plans and implementations must begin with a thorough analysis of all data sets, services, and workloads [41]. The guidebook discusses cloud computing, data management, and workflows in this section. Workflows and field tools are also included, as are adding data and justifying changes to resources. We will investigate different conflicts between workers in the workshop and use new methods such as deep transfer learning [42] and long short-term memory (LSTM) structures in the future. In other words, these structures can be trained in advance and used for classification and prediction in various fields of workplace flexibility analysis.

5. Conclusion

Construction employees' attitudes toward safety are influenced by procedures, safety rules, and risk management. People who hold these views are more likely to engage in safe work practices, according to previous research. There are, however, surprisingly few studies examining the relationship between interpersonal variability, individual flexibility, and construction workers' safety outcomes, particularly in developing countries. Numerous studies have examined the effects of individual flexibility and interpersonal variance on construction site safety.

In this study, behavioral and safety questionnaires were used. The questionnaire's first few items were based on research on construction workers and other high-risk occupations. It was developed based on seven distinct criteria following an initial evaluation. Conflict can arise from various factors, such as disagreements between coworkers, work-related stress, injuries, and risky incidents. As a result, past investigations have been made public. People's behavior is influenced by a variety of factors, such as physical injury and inefficient road design. Furthermore, this study examines how employees feel about safety and their willingness to engage and commit to it. With cloud computing, it is possible to gain a better understanding of individual and workplace flexibility, as well as construction worker safety and interpersonal variability.

With deep learning, authors will be able to deal with issues like low accuracy in ML systems. The output of ANNs was easy to understand, which made machine learning convenient for them. Using cloud computing, the authors' data can be managed more accurately and securely. LSTM structures and deep transfer learning will be used in the future to develop a system for investigating conflicts between workshop workers.

Data Availability

The codes and data are all available from the corresponding authors.

Conflicts of Interest

There is no conflict of interest.

References

[1] J. Konopik, C. Jahn, T. Schuster, N. Hoßbach, and A. Pflaum, "Mastering the digital transformation through organizational capabilities: a conceptual framework," *Digital Business*, vol. 2, no. 2, p. 100019, 2022.

[2] N. Oliveira and F. Lumineau, "The dark side of interorganizational relationships: an integrative review and research agenda," *Journal of Management*, vol. 45, no. 1, pp. 231–261, 2019.

[3] T. Yigitcanlar and S. Teriman, "Rethinking sustainable urban development: towards an integrated planning and development process," *International journal of Environmental Science and Technology*, vol. 12, no. 1, pp. 341–352, 2015.

[4] O. Babalola, E. O. Ibem, and I. C. Ezema, "Implementation of lean practices in the construction industry: a systematic review," *Building and Environment*, vol. 148, pp. 34–43, 2019.

[5] C. Alexander-White, D. Bury, M. Cronin et al., "A 10-step framework for use of read-across (RAX) in next generation risk assessment (NGRA) for cosmetics safety assessment," *Regulatory Toxicology and Pharmacology*, vol. 129, p. 105094, 2022.

[6] J. P. Byrne, E. Conway, A. M. McDermott et al., "How the organisation of medical work shapes the everyday work experiences underpinning doctor migration trends: the case of Irish-trained emigrant doctors in Australia," *Health Policy*, vol. 125, no. 4, pp. 467–473, 2021.

[7] J. Hessels, C. A. Rietveld, and P. van der Zwan, "Self-employment and work-related stress: the mediating role of job control and job demand," *Journal of Business Venturing*, vol. 32, no. 2, pp. 178–196, 2017.

[8] O. M. Karatepe, H. Rezapouraghdam, and R. Hassannia, "Does employee engagement mediate the influence of psychological contract breach on pro-environmental behaviors and intent to remain with the organization in the hotel industry?," *Journal of Hospitality Marketing & Management*, vol. 30, no. 3, pp. 326–353, 2021.

[9] M. Destin, M. Rheinschmidt-Same, and J. A. Richeson, "Status-based identity," *Perspectives on Psychological Science*, vol. 12, no. 2, pp. 270–289, 2017.

[10] L. S. Beh and L. H. Loo, "Job stress and coping mechanisms among nursing staff in public health services," *International Journal of Academic Research in Business and Social Sciences*, vol. 2, no. 7, p. 131, 2012.

[11] R. Z. A. R. Ibrahim, J. Saputra, A. A. Bakar et al., "Role of supply chain management on the job control and social support for relationship between work-family conflict and job satisfaction," *International Journal of Supply Chain Management*, vol. 8, no. 4, pp. 907–913, 2019.

[12] A. Anand and A. Dalmasso, "Supervisor effects on employee knowledge sharing behaviour in SMEs," *Journal of the Knowledge Economy*, vol. 11, no. 4, pp. 1430–1453, 2020.

[13] A. Maqsoom, A. Mughees, U. Safdar, B. Afsar, and Z. Badar ul Ali, "Intrinsic psychosocial stressors and construction worker productivity: impact of employee age and industry experience," *Economic research-Ekonomska istraživanja*, vol. 31, no. 1, pp. 1880–1902, 2018.

[14] Q. Liang, M. Y. Leung, and K. Ahmed, "How adoption of coping behaviors determines construction workers' safety: a quantitative and qualitative investigation," *Safety Science*, vol. 133, p. 105035, 2021.

[15] L. Campbell-Sills, S. L. Cohan, and M. B. Stein, "Relationship of resilience to personality, coping, and psychiatric symptoms in young adults," *Behaviour Research and Therapy*, vol. 44, no. 4, pp. 585–599, 2006.

[16] A. Rezaee, K. Rezaee, J. Haddadnia, and H. T. Gorji, "Supervised meta-heuristic extreme learning machine for multiple sclerosis detection based on multiple feature descriptors in MR images," *SN Applied Sciences*, vol. 2, no. 5, pp. 1–19, 2020.

[17] N. Tavasoli, K. Rezaee, M. Momenzadeh, and M. Sehhati, "An ensemble soft weighted gene selection-based approach and cancer classification using modified metaheuristic learning," *Journal of Computational Design and Engineering*, vol. 8, no. 4, pp. 1172–1189, 2021.

[18] S. Meng, W. Huang, X. Yin et al., "Security-aware dynamic scheduling for real-time optimization in cloud-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4219–4228, 2021.

[19] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2910–2918, 2021.

[20] K. Rezaee, S. J. Mousavirad, M. R. Khosravi, M. K. Moghimi, and M. Heidari, "An autonomous UAV-assisted distance-aware crowd sensing platform using deep ShuffleNet transfer learning," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[21] D. Farashaei, A. Honarbakhsh, S. M. Movahedifar, and E. Shakeri, "Individual flexibility and workplace conflict: cloud-based data collection and fusion of neural networks," *Wireless Networks*, vol. 2022, pp. 1–16, 2022.

[22] Y. Pan and L. Zhang, "Roles of artificial intelligence in construction engineering and management: a critical review and future trends," *Automation in Construction*, vol. 122, p. 103517, 2021.

[23] Y. Chen, G. Fu, F. Liang, J. Wei, J. He, and J. Bai, "Symptoms, hope, self-management behaviors, and quality of life among Chinese preoperative patient with symptomatic valvular heart diseases," *Journal of Transcultural Nursing*, vol. 31, no. 3, pp. 284–293, 2020.

[24] Y. Ten Hoeve, J. Brouwer, and S. Kunnen, "Turnover prevention: the direct and indirect association between organizational job stressors, negative emotions and professional commitment in novice nurses," *Journal of Advanced Nursing*, vol. 76, no. 3, pp. 836–845, 2020.

[25] M. Golparvar, "Unconventional functions of deviant behaviors in the relationship between job stress and emotional exhaustion: three study findings," *Current Psychology*, vol. 35, no. 3, pp. 269–284, 2016.

[26] M. W. Teoh, Y. Wang, and A. Kwek, "Coping with emotional labor in high stress hospitality work environments," *Journal of Hospitality Marketing & Management*, vol. 28, no. 8, pp. 883–904, 2019.

[27] C. H. Liu, J. S. Horng, S. F. Chou, Y. C. Huang, and A. Y. Chang, "How to create competitive advantage: the moderate role of organizational learning as a link between shared value, dynamic capability, differential strategy, and social capital," *Asia Pacific Journal of Tourism Research*, vol. 23, no. 8, pp. 747–764, 2018.

[28] T. Fitriastuti and A. Vanderstraeten, "Being out of the loop: workplace deviance as a mediator of the impact of impression management on workplace exclusion," *Sustainability*, vol. 14, no. 2, p. 1004, 2022.

[29] E. Di Carlo, "Antecedents of deviant behavior: psychological and non-psychological factors and ethical justifications," *Employee Responsibilities and Rights Journal*, vol. 34, pp. 169–191, 2022.

[30] P. Findlay, C. Lindsay, J. McQuarrie, M. Bennie, E. D. Corcoran, and R. Van Der Meer, "Employer choice and job quality," *Work and Occupations*, vol. 44, no. 1, pp. 113–136, 2017.

[31] Y. Chen, B. McCabe, and D. Hyatt, "Relationship between individual resilience, interpersonal conflicts at work, and safety outcomes of construction workers," *Journal of Construction Engineering and Management*, vol. 143, no. 8, p. 04017042, 2017.

[32] A. Masuda and E. C. Tully, "The role of mindfulness and psychological flexibility in somatization, depression, anxiety, and general psychological distress in a nonclinical college sample," *Journal of Evidence-Based Complementary & Alternative Medicine*, vol. 17, no. 1, pp. 66–71, 2012.

[33] E. Wensing and S. Crompvoets, "Workplace flexibility in the ADF: anathema or panacea?," *Australian Defence Force Journal*, vol. 196, no. 196, pp. 79–93, 2015.

[34] Y. Chen, B. McCabe, and D. Hyatt, "Impact of individual resilience and safety climate on safety performance and psychological stress of construction workers: a case study of the Ontario construction industry," *Journal of Safety Research*, vol. 61, pp. 167–176, 2017.

[35] R. Harmsen, M. Helms-Lorenz, R. Maulana, K. van Veen, and M. van Veldhoven, "Measuring general and specific stress causes and stress responses among beginning secondary school teachers in the Netherlands," *International Journal of*

Research & Method in Education, vol. 42, no. 1, pp. 91–108, 2019.

[36] N. V. Schwatka and J. C. Rosecrance, "Safety climate and safety behaviors in the construction industry: the importance of co-workers commitment to safety," *Work*, vol. 54, no. 2, pp. 401–413, 2016.

[37] C. Haslam, T. Cruwys, M. Milne, C. H. Kan, and S. A. Haslam, "Group ties protect cognitive health by promoting social identification and social support," *Journal of Aging and Health*, vol. 28, no. 2, pp. 244–266, 2016.

[38] O. L. Siu, D. R. Phillips, and T. W. Leung, "Safety climate and safety performance among construction workers in Hong Kong: the role of psychological strains as mediators," *Accident Analysis & Prevention*, vol. 36, no. 3, pp. 359–366, 2004.

[39] M. Y. Leung, Y. S. Chan, and K. W. Yuen, "Impacts of stressors and stress on the injury incidents of construction workers in Hong Kong," *Journal of Construction Engineering and Management*, vol. 136, no. 10, pp. 1093–1103, 2010.

[40] K. Rezaee, A. Badiei, and S. Meshgini, "A hybrid deep transfer learning based approach for COVID-19 classification in chest X-ray images," in *2020 27th national and 5th international Iranian conference on biomedical engineering (ICBME)*, pp. 234–241, 2020.

[41] O. S. Mwilu, I. Comyn-Wattiau, and N. Prat, "Design science research contribution to business intelligence in the cloud – a systematic literature review," *Future Generation Computer Systems*, vol. 63, pp. 108–122, 2016.

[42] M. S. Anari, K. Rezaee, and A. Ahmadi, "TraitLWNet: a novel predictor of personality trait by analyzing Persian handwriting based on lightweight deep convolutional neural network," *Multimedia Tools and Applications*, vol. 81, no. 8, pp. 10673–10693, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1213536, 11 pages
https://doi.org/10.1155/2022/1213536

Research Article

Data Analysis in Green Industrial Processes with Modified Chemical Efficiency and Environmental Impact: Smart Urea Production and CO_2 Removal

Fatemeh Khandaghi[1,2,3] and Shahoo Abdollahi[3]

[1]*School of Chemical Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[2]*School of Chemical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran*
[3]*Research & Development, DanubTech Co., Tehran, Iran*

Correspondence should be addressed to Fatemeh Khandaghi; f_khandaghi@vu.iust.ac.ir

Received 21 February 2022; Revised 13 May 2022; Accepted 20 May 2022; Published 3 June 2022

Academic Editor: Varun Menon

Agriculture can benefit from urea fertilizer because it contains a lot of nitrogen at a reasonable cost. Urea fertilizer can be stored easily and does not pose a fire hazard over time. Due to its acidifying properties, urea fertilizer is an ideal fertilizer for many plants. Input and feed of the urea unit are taken from the output of the ammonia unit (CO_2 and NH_3). Hence, in this study, two methods of CO_2 recovery from combustion gases and CO_2 recycling in ammonia units will be used to increase urea production to realize low-carbon and industrial systems (including green agriculture). CO_2 recovery also reduces environmental pollution, which is a very important factor in sustainable cities and societies. The results showed that CO reduction increases the overall efficiency compared to the data reported in the world for the same process, which is due to the reduction of CO input to methanize. Collecting information around the globe for constructing the same green system considering various conditions in each environment makes complicated situations in terms of how to design the process and the observed outcomes. However, we could find a new smart design to build the green system in our case study where it is completely acceptable compared to the same systems' outputs. The obtained results indicate that the temperature of the shift reactor can be brought closer to 365°C without reducing the selectivity of the catalysts, which in turn would increase the CO conversion rate, the CO_2 output, and the overall efficiency of the unit. Finally, it is shown that the rate of CO escape from shift reactors is decreased.

1. Introduction

Nowadays, green industry concept is becoming a major challenge in modern societies [1]. In this regard, the human population is expected to reach 9,500,000 by 2050. It means food shortages occur as per capita demand almost doubles. Also, the number of acres of agricultural land is declining as a result of economic growth, residential development, and climate change [2].

A large number of fertilizers especially nitrogen fertilizers like urea (NH_2CONH_2) are needed to have a modern society as well as a green industry [3]. As it is obvious, urea is high in nitrogen and nitrogen is a vital nutrient source for plant product development. In fact, chlorophyll, proteins, and protein-carrying compounds are all made from nitrogen [4]. Compared to other nutrients, nitrogen is a primary and continuous nutrient for plant growth [5, 6]. Nitrogen contributes initially to rapid growth of roots and leaves and chlorophyll production and also increases biomass accumulation and yields [7, 8]. Agricultural products need nitrogen during critical growth stages, which is usually provided by conventional urea fertilizers [9, 10].

In order to achieve sustainable agriculture, it is essential to use pesticides and chemical fertilizers in the minimum amount and optimally [9]. Globally, about 200 million tons of urea fertilizer are produced each year [11, 12]. Since urea can be combined with other solid fertilizers, this fertilizer has become the most widely used fertilizer in the world

[13–15]. In this way, we can further the development of the green industry through its optimal production of fertilizer [5].

Among all solid nitrogen fertilizers, urea contains the most nitrogen (46.7%) [16] and it is hydrolyzed to ammonia and carbon dioxide in the soil. Bacteria in the soil oxidize ammonia produced in this process into nitrate, which can then be absorbed by plants. Often, urea is used in multicomponent formulations of solid fertilizers. Since urea is highly soluble in water, it is also very suitable for use in fertilizer solutions [17].

Urea is produced industrially by the reaction of ammonia and carbon dioxide at high pressures (13 to 30 MPa) and high temperatures (170 to 200°C) [18]. Interestingly, urea was first chemically produced by Farben in 1920 [19]. Urea fertilizer has been produced since the early 1950s. The advantages of urea have led many researchers to investigate how to improve conversion rates and the quality of products [20].

Nowadays, several technologies including chemical solvent adsorption, biological stabilization, membrane separation, and hydrate-based separation are used to recover CO_2.

With the growing population of the world and a need to meet its food needs, agriculture has become increasingly challenging despite the limited amount of land available to grow food. The total ammonia emissions from agriculture can be attributed to mineral nitrogen fertilization (especially urea). To reduce ammonia emissions and increase fertilizer efficiency, it is crucial to minimize ammonia evaporation. As such, it is necessary to reduce the negative impact of human pressure on the environment, including air pollution, soil erosion, and water pollution [21, 22]. The proportion of ammonia to carbon dioxide produced in ammonia production units is higher than the ratio required in its downstream unit (urea unit) due to the lightness of the feed gas. Accordingly, in this study, the CO_2 shortage will be compensated by the CO_2 recovery from flue gas or CO_2 absorption tower was increased in order to increase the urea production capacity. Green systems can be categorized into different parts. We will have green energy, for example, when we find ways to reduce energy consumption. By requiring fewer materials to produce something, we have green materials as well. Air pollution reduction is also one of the most important uses of green systems. Hence, the proposed framework has the potential to achieve green industrial policy. In fact, the proposed framework benefits from economic, social, and environmental aspects.

Considering the importance of this topic, the purpose of this article is as follows:

(I) CO_2 recycling in ammonia unit and combustion gases

(II) Investigating how to reduce air pollution in petroleum units

(III) Increasing the production of urea fertilizer

Gathering the different data for the green systems is a challenging issue when we attend to the outputs and complicated processes in each scenario. Fortunately, we could find a smart solution to make the green system in our case study where it is considerable in comparison to other similar systems' outputs.

2. Materials and Methods

In order to facilitate readers, ammonia and urea production procedures are described in Sections 2.1 and 2.2, respectively. Moreover, Section 2.3 depicts the reaction of CO_2 with aqueous ammonia and Section 2.4 describes CO_2 hydration in an aqueous solution, while Section 2.5 demonstrates how to increase the amount of urea. It should be noted that this research was experimental, and data were collected from experiments. A data mining approach is then used to preprocess the collected data. Statistical methods are used to determine the applicability of the proposed framework. Our entire process was carried out on a laptop with the following configuration:

(i) Core i7

(ii) 8 GB ram DDR4

(iii) NVIDIA graphic with 2 GB ram

2.1. Ammonia Production Process. Ammonia was first discovered in the 8th century by Jabir Ibn Hayyan as salt [23]. Valentinus Basilius investigated that ammonia can be obtained by alkaline reactions on ammonia salts in the 15th century [24]. The development of new technology and a need for nitrogen and nitric acid fertilizers as the basis for explosives led scientists to reformulate these compounds from oil and gas [25]. Various industries and agriculture have developed in different countries as a result of ammonia production, including agriculture in Iran, which is now a major market [26, 27]. Nitrogen (ambient air) and hydrogen (hydrocarbons or water electrolysis) are combined to produce ammonia. Equation (1) depicts the general reaction for the production of ammonia [28]. Furthermore, Figure 1 illustrates the schematic of the ammonia production process.

$$N_2 + 3H_2 \longrightarrow 2NH_3 \Delta H700 = -52.5\,\text{kJ/mol} \tag{1}$$

2.2. Urea Production Process. According to Equation (2), urea is formed when the mixture of ammonia liquid and carbon dioxide gas is pressured at 143 kg/cm^2 and heated to about 170-180°C. It should be noted that this process is calorific and rapid, converting ammonia and carbon dioxide into liquid ammonium carbamate (NH_2COONH_4), while urea (NH_2CONH_2) and water are extracted in a slow interaction that is depicted in Equation (3). Consequently, it is essential to provide the conditions in this unit for performing the above two reactions.

$$2NH_3 + CO_2 \leftrightarrow NH_2COONH_4 \tag{2}$$

$$NH_2COONH_4 \leftrightarrow NH_2CONH_2 + H_2O \tag{3}$$

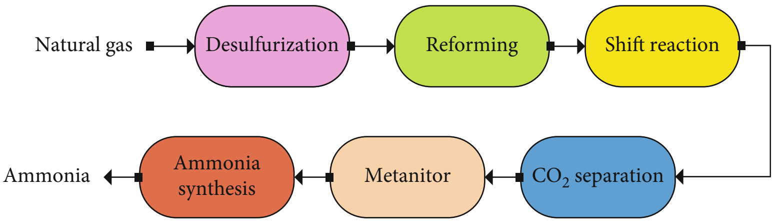


Figure 1: Schematic of the ammonia production process.

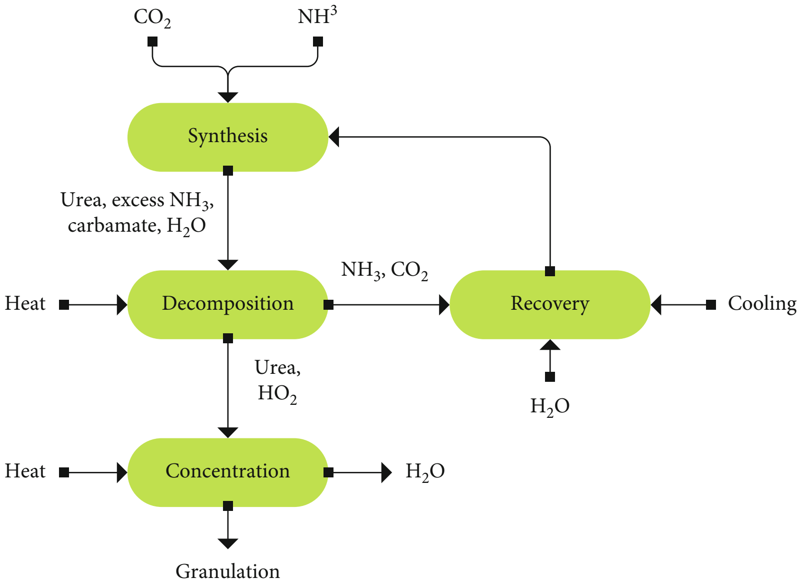


Figure 2: Urea production unit flowchart.

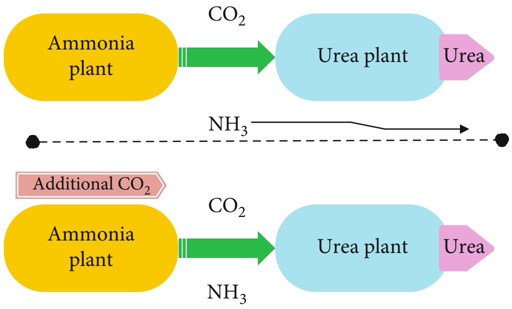


Figure 3: NH_3/CO_2 production ratio.

The urea unit receives ammonia and carbon dioxide from the ammonia unit. Pumps dispense liquid ammonia under a pressure of 160 kg/cm^2, and compressors compress carbon dioxide under a pressure of 146 kg/cm^2 [29]. Then, these two streams enter the unit that synthesizes urea. This unit must have the potential to operate at these temperatures and pressures. During the reaction of ammonia and carbon dioxide in a carbamate maker, high-pressure fuels form ammonium carbamate. The reaction is very fast and exothermic, and the generated heat is used to create low-pressure steam (4.5 kg/cm^2). Steam is used in several parts of the unit that the ammonium carbamate is then converted to urea and water. The urea production process is shown in Figure 2.

2.3. The Reaction of CO_2 with Aqueous Ammonia.

Carbon dioxide and ammonia react mainly in the liquid phase of the gas-liquid interface. Furthermore, Equation (4) illustrates this chemical reaction.

$$CO_{2(g)} + 2NH_{3(aq)} \longrightarrow NH_2COONH_{4(aq)} \tag{4}$$

As a matter of fact, Equations (5) and (6) consist of two steps as below.

$$CO_{2(g)} + NH_{3(aq)} \longrightarrow NH_2COONH_{(aq)} \tag{5}$$

$$NH_{3(aq)} + NH_2COOH_{(aq)} \longrightarrow NH^+_{4(aq)} + NH_2COO^- \tag{6}$$

NH_2COONH_4 is then decomposed in solution to produce free ammonia

$$H_2O + NH_2COO^- \leftrightarrow NH_3 + HCO_3^- \tag{7}$$

$$NH_3 + H_2O \leftrightarrow NH_4 + OH^- \tag{8}$$

Equation (5) is very swift and irreversible, whereas Equation (6) is momentary and sudden. Also, Equation (7) is relatively slow and does not affect the absorption intensity directly. Therefore, the reaction between aqueous ammonia solution and carbon dioxide is primarily controlled by

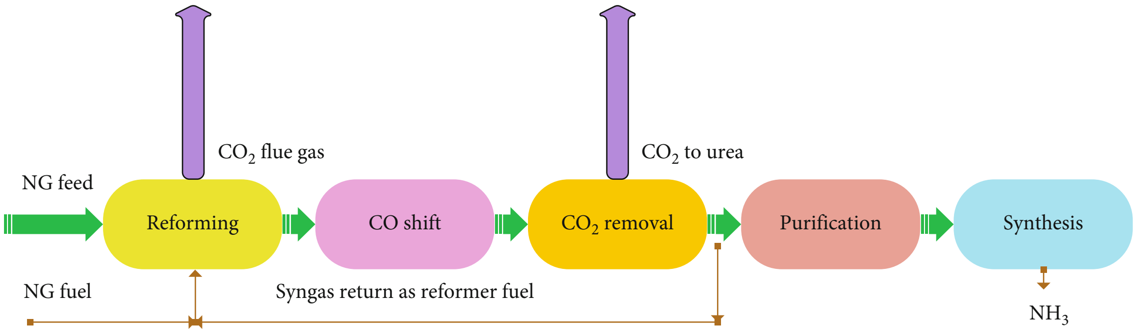


Figure 4: IncreasingCO_2 production by returning synthesized gas.

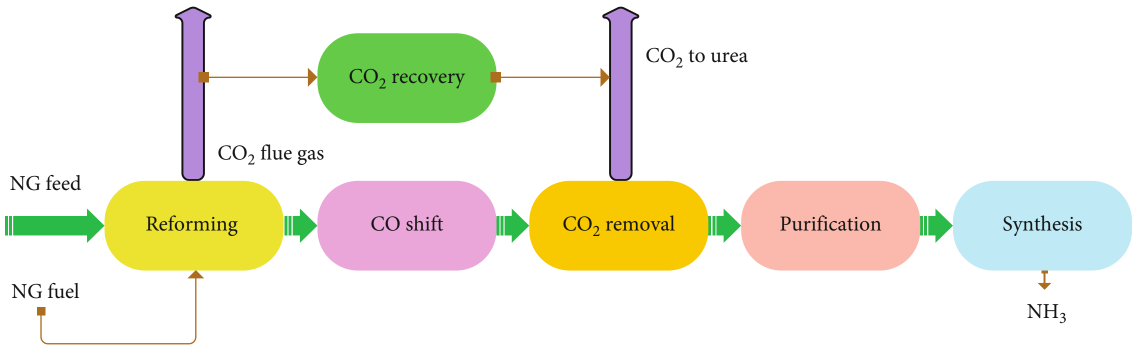


Figure 5: CO_2 recovery from combustion gases and flue.

Table 1: Comparison of two methods of CO_2 production.

	CO_2 recovery from combustion gases	CO_2 obtained from CO_2 recycling in ammonia unit
New equipment	✓	×
Investment	High	Low
Cost	High	High

Equation (5). CO_2 and NH_3 are, respectively, quadratic and first-degree reactions with a temperature of -257.85°C; the constant value of velocity k_2 is about 300 (L/mol·s).

2.4. CO_2 Hydration in Aqueous Solution. Ammonia solution has very weak alkalinity. As a result, immersion in the aqueous solution of CO_2 will occur in the liquid phase

$$CO_2 + H_2O \leftrightarrow HCO_3^- + H^+ \quad (9)$$

$$CO_2 + OH^- \leftrightarrow HCO_3^- \quad (10)$$

At 24.85°C the contribution of Equations (9) and (10) to the total reaction rate is very small, since $k = 0.026\,s^{-1}$. Therefore, they can usually be ignored. However, at 20°C, the value of k_{OH-} is 5747.9 m^3/kmol.

2.5. Increasing the Capacity of Urea Production Unit. The output of the ammonia unit (CO_2 and NH_3) constitutes the input and feed to the urea unit. There is often excess ammonia in urea units that cannot be converted into urea due to a lack of CO_2. In fact, the ratio of ammonia and CO_2 production is not independent of each other. The ideal ratio of CO_2 to NH_3 in the production of ammonia from pure methane (CH_4), air, and water is 1/14 (t/t) [30]. However, depending on the components of the natural gas and the losses during the process, this value can be lower or higher in the actual process. In contrast to the approximate amount, one unit of urea consumes more carbon dioxide and ammonia [31–33]. As mentioned, there is always extra ammonia; hence, the major challenge is insufficient CO_2 production. It means that additional CO_2 to utilize all of the ammonia available for urea production is required (Figure 3).

As follows, this paper suggests two methods for preparing the needed CO_2.

2.5.1. CO_2 Obtained from CO_2 Recycling in Ammonia Unit. Typically, the CO_2 feed for urea production is obtained by separating CO_2 from the synthetic gases in the CO_2 separation section [34]. Hence, by transferring more synthetic gases through this unit, the amount of CO_2 in this area can be increased. Therefore, the low output current from the CO_2 separation unit, which contains additional synthesis gas, is not required for ammonia production and is ignored and sent to the reformer, where it is used as a gaseous fuel. In this process, the gaseous fuel consumption per unit increases, resulting in higher throughput and therefore more work in areas such as desulfurization, reforming, and heat loss recovery. Meanwhile, the natural gas is returned to the reformer as fuel, reducing the consumption of natural gas. Natural gas consumption is still expected to increase as it is illustrated in Figure 4.

Table 2: Results of the applied changes.

Temperature	Description	Time (h)					
		9:40	9:50	9:55	11:00	11:20	13:00
AT-10048	Process gas to T-2001	0.413	0.4143	0.415	0.3906	0.3914	0.3955
FQI-2002	T-2002 over head	58046	58112	58279	58264	58380	58318
TI-1085	E-2003 inlet	126.89	126.89	126.94	127.14	127.34	127.17
TI-2011	CO_2 absorber over head	43.43	43.45	43.45	43.74	43.76	43.34
TI-2013	T-2003 over head	86.62	86.57	86.60	87.36	87.46	87.50
TI-2019	T-2002 over head	45.74	45.78	45.17	46.24	46.42	46.34
TI-2003	Co_2 stripper bottom	121.45	121.53	121.55	121.81	121.88	121.83
TI-1084	E-2001 PG outlet	165.16	165.28	165.79	166.78	165.11	165.72
TI-1082	BFW to E-1012	173.53	173.57	173.56	173.60	173.80	173.66
TI-2014	Lean soln pump disch	42.7	42.74	42.79	43.12	43.18	43.23
TI-2017	AMDEA semilean solution	83.08	83.04	83.01	83.96	84.11	84.26
TI-2012	T-2001 liquid outlet	87.62	87.64	87.70	88.53	88.63	88.70
TI-1087	D-2001 inlet	61	61.14	61.24	61.15	61.50	61.38
TI-1088	Reboiler circulation	112.11	112.14	112.06	112.41	112.60	112.60

2.5.2. CO_2 Recovery from Combustion Gases. As shown in Figure 5, CO_2 recovery unit is connected to the exhaust gases from the reformer. The combustion gases from reformers and boilers have a high carbon dioxide content making them another potential source of CO_2 for urea production units. The combustion gases of ammonia unit reformer furnaces contain 12% carbon dioxide. By separating CO_2 from this smoke and providing it to the downstream unit, which is urea, we can prevent its release into the atmosphere and air pollution.

During this process, the flue gas passes through a large tower, called an absorber tower. As a result of the adsorption process, exhaust gases are exposed to adsorbent fluid (amines dissolved in water), which creates a weak chemical bond between carbon dioxide and the amines. This carbon dioxide is then transferred to a tower known as a stripper, where the solvent is heated and CO_2 is separated from amines. Amines are then reused to reabsorb CO_2. The remaining ammonia in the CO_2-free output stream is removed in the wash tower. Additionally, fans and pumps are used to compensate for the pressure drop in the absorption tower. They are also used to pump the amino solution and to cool the water, as well as to compress or cool the CO_2 before it is transferred [35].

It should be noted that the CO_2 recovered in this way is of high quality and can be mixed with the existing CO_2 input to the CO_2 compressor to participate in the urea synthesis. Additionally, amines are used as solvents to counteract the side effects of combustion gases such as O_2, NOx, and SO_2.

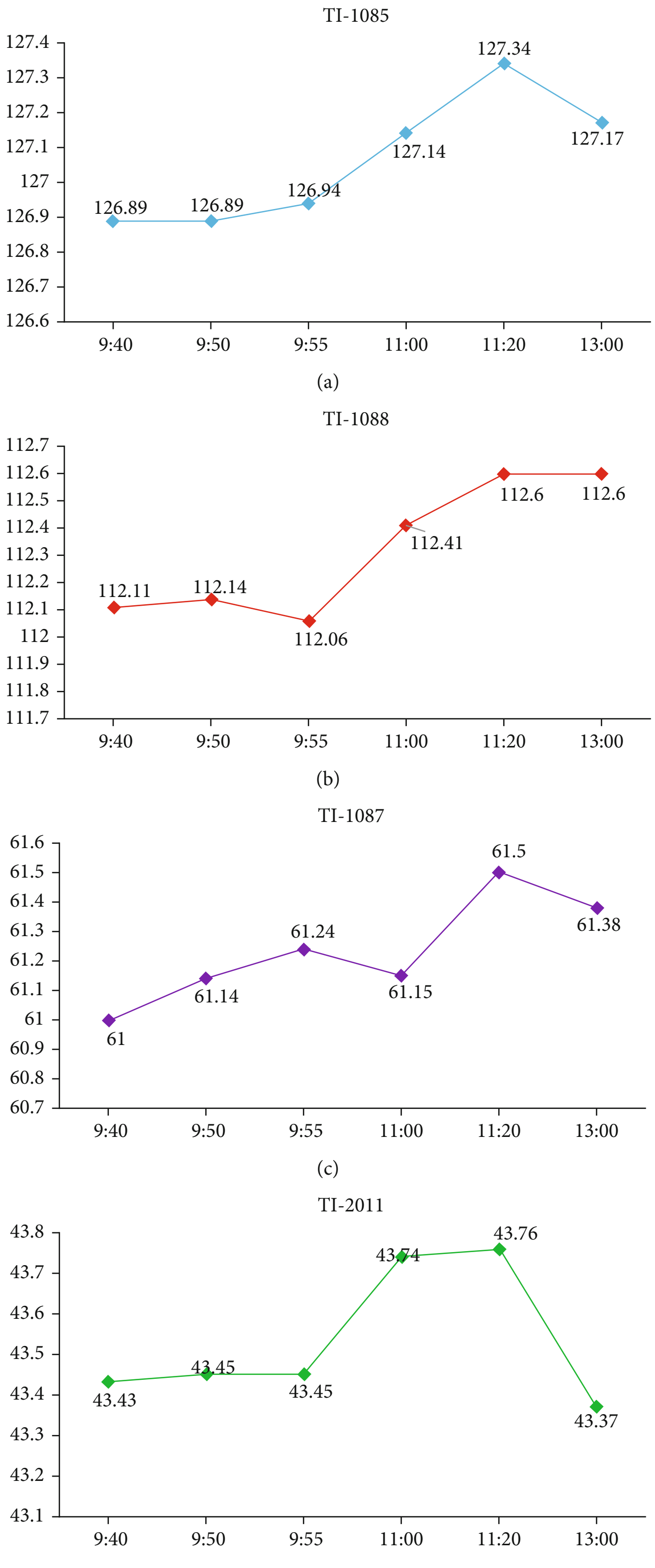
It is worth mentioning that carbon dioxide emissions are heavily influenced by operation costs. Although almost all CO_2 can be separated, separating the last few percent requires a large amount of energy and is expensive. Normally, amines can separate CO_2 from gas in about 59% of cases. Operation cost for this process includes the cost of steam to recover the solvent as well as electricity cost to supply pumps and fans. Moreover, there are additional costs for other stuff like leveling and replacing the waste solvent with fresh and additional solvent.

Table 1 compares these two methods in detail. According to Table 1, both methods have their advantages and disadvantages. The optimal method is determined by the amount of CO_2 required and the cost of energy. Recovering CO_2 from combustion gases is the best choice and solution for large amounts of CO_2 or a unit with a high gas cost, while, if a small amount of CO_2 is needed or the energy price is not too high, increasing the production of synthetic gas can be considered. It is worth mentioning that air pollution is another important factor to select the optimal solution. By capturing CO_2 from combustion gases, the amount of CO_2 emitted from the unit to the atmosphere will be greatly reduced. The development of various chemicals for CO_2 recovery has been studied in recent years to improve the processes and reduce recovery costs. The proposed framework can be used wherever there is CO_2 available as well as heat for starting the urea fertilizer production process. Therefore, ships seem like a good choice for testing the applicability of the proposed framework on moving objects.

3. Results and Discussion

The study was carried out at Pardis Petrochemical's second ammonia unit. The ratio of molar steam to gas flow in the reformer was set at 3.2. In addition, steam flow is 145 tons per hour and natural gas is 56 thousand normal cubic meters per hour according to the design. The steam input was therefore increased by one ton per hour for the experimental study, which reached 146 tons per hour after the necessary coordination. Temperatures in the carbon dioxide removal section were monitored. Moreover, Table 2 illustrates how the temperature indicator (TI) changes when the steam flow increases.

The TI-1085 displays the gas temperature as input to the E-2003 converter. Process gas heat is used to heat the aerated



(a)

(b)

(c)

(d)

Figure 6: Continued.

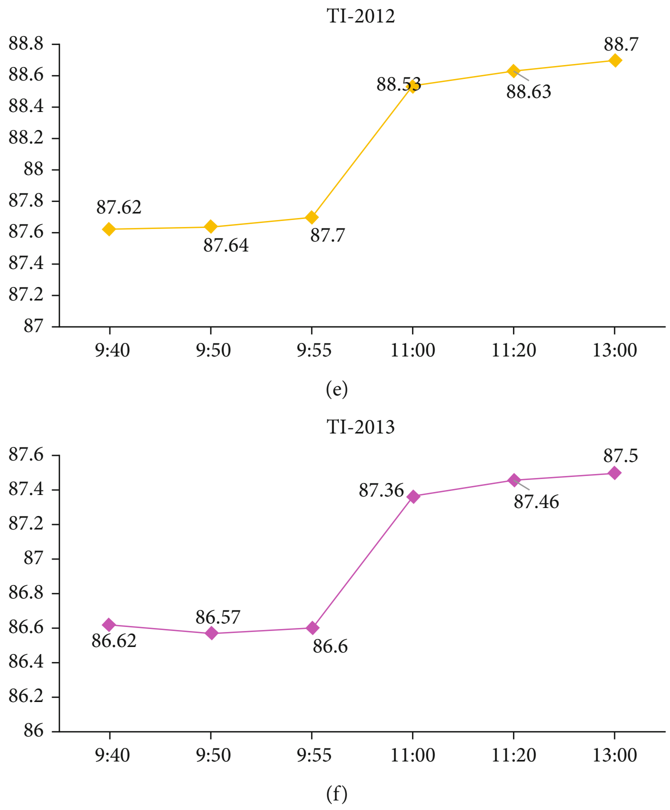


Figure 6: Temperature changes of (a) input process gas to the E-2003 converter over time, (b) outlet water from E-2003, (c) process gas after E-2003, (d) gas imaged from adsorption tower, (e) MDEA liquid at the outlet of the adsorption tower, and (f) rivet gas output from the flash drum.

water. According to Figure 6, when excess steam is present, it transfers more heat to the water, which makes it warmer.

Figure 6 provides an evaluation of the outlet water temperature of the E-2003 converter (TI-1088). TI_1087 represents the process temperature after the E-2003 converter. The process gas transfers its heat to water up to the condensate temperature. When this temperature is reached, water vapor condenses and separates from the gas flow in the D-2001 drum. Excess heat is then released from the converter. The TI-2011 also indicates the temperature of the purified gas from the adsorption tower (Figure 6). As the steam rises, the temperature increases. TI-2012 MDEA liquid temperature is the output of the absorption tower, while TI-2013 is the gas temperature of the flash drum output tower. This gas is an unabsorbed gas in the MDEA system. In fact, some of the process gas that comes out of the absorption tower with MDEA is flashed in this drum and removed from the system.

CO_2 temperature is the output of the discharge tower in TI-2019, while TI-2003 is the disposal tower's low temperature that is the two-phase temperature of the MDEA system and provides the necessary heat to the disposal tower. Moreover, the temperature of the semilean amine moving towards the adsorption tower to reabsorb CO_2 is TI–2017. The lean amine temperature for adsorption of CO_2 and the exhaust gas temperature of the E-2001 converter are TI–2014 and TI-1084, respectively. It heats the water from the air conditioner to make steam. Furthermore, TI-1082 is the outlet water temperature of the E-2001 converter (Figure 7).

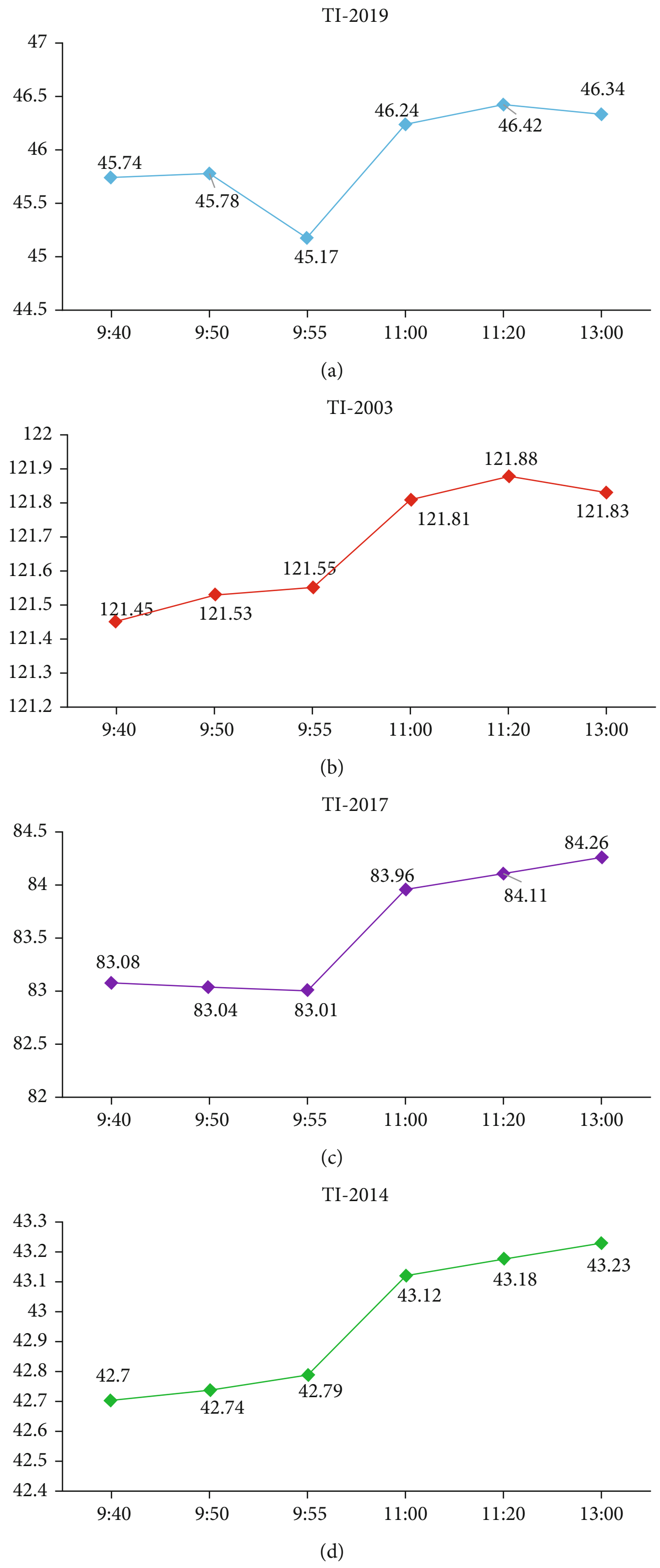
Results indicated that it is possible to increase the heat of a gas stream by adding steam to it, which increases its heating capacity. However, there are limitations to how much steam can be used. Figures 8 and 9 illustrate the effect of two other factors, including FQI-2002 and AT-1004B.

FQI-2002 represents the flow rate of CO_2 extracted in the CO_2 removal section. This amount has increased with the addition of steam, which is related to the rise in discharge tower temperatures. In order to conduct further investigation, the high temperature of the tower needs to be maintained constant, which is not possible given the unit conditions (Figure 9).

The second aspect that must be examined is the amount of CO entering the adsorption tower (process according to T-2001). Clearly, this value has decreased, meaning that the unit is operating in better conditions, which is why FQI-2002 has increased. With a lower amount of CO escaping in this section compared to the previous sections, the amount of CO_2 being converted has increased, which has provided very favorable conditions (Figure 9).

The concentration of ammonia and the pH of the solution were measured through titration. The data in Table 3 illustrate CO_2 uptake in aqueous ammonia at 20°C. Based on the results, $k_{OH-}[OH^-]$ is much smaller than $k_2[NH_3]$.

The ratio of $k_{OH-}[OH^-]$ to the reaction rate constant $([k_2[NH_3] + k_{OH-}[OH^-])$ is approximately 6.92% if 2% of the solution is aqueous ammonia. Furthermore, the contribution of $k_{OH-}[OH^-]$ decreases as the concentration of



(a)

(b)

(c)

(d)

Figure 7: Continued.

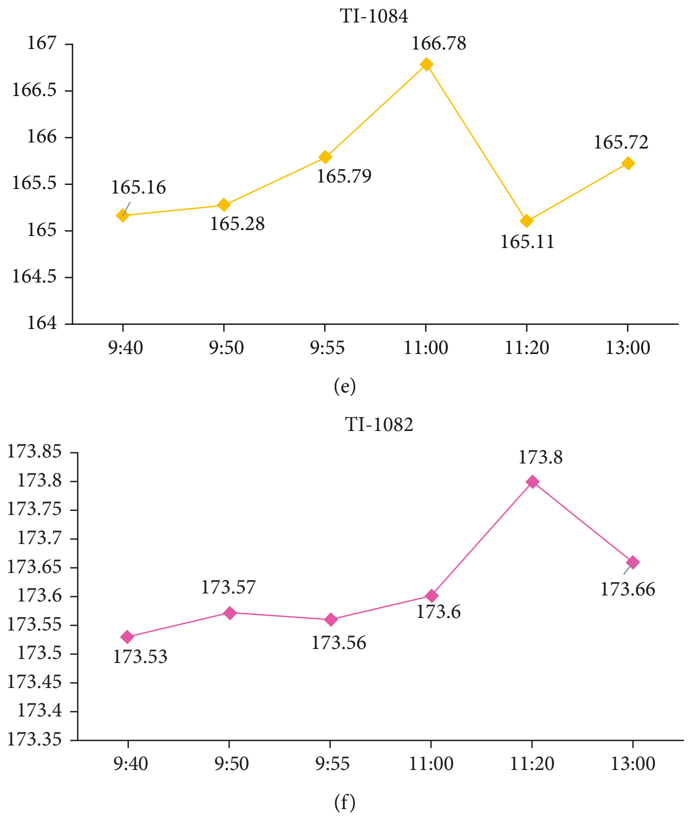


Figure 7: Temperature changes of (a) CO_2 at the exhaust tower, (b) discharge tower, (c) semilean liquid, (d) lean liquid, (e) exhaust gas from the exchanger (E-2001), and (f) outlet water from the E-2001 converter.

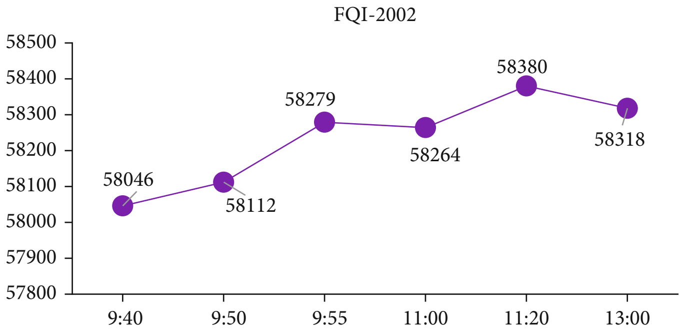


Figure 8: EQI rate of carbon dioxide extracted (Nm^3/h).

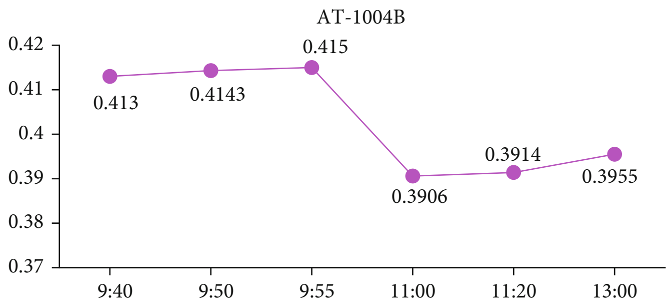


Figure 9: The amount of CO entering the adsorption tower.

Table 3: Synthetic data for CO_2 uptake in aqueous ammonia solution.

Temperature (°C)	C (NH_3) (%)	[NH_3] (kmol m^{-3})	[OH$^-$] (kmol m^{-3})	$k_2[NH_3]$ (A) (S^{-1})	$k_{OH-}[OH^-]$ (B) (S^{-1})	$B/(A+B)$ (%)
20°C	2	1.16	0.0045	348	25.87	6.92
20°C	4	2.3	0.0063	690	36.21	4.99
20°C	6	3.43	0.0077	1032	44.26	4.11
20°C	8	4.54	0.0088	1362	50.58	3.58
20°C	12	6.71	0.0107	2013	61.5	2.96
20°C	16	8.81	0.0123	2543	70.7	2.51

aqueous ammonia increases. In this case, the share of reaction explained by Equation (9) in the total reaction rate is less than 7% when the mass fraction of aqueous ammonia solution exceeds 2%. Since the rate of reaction between CO_2 and ammonia solution is mainly determined by Equation (10), it can be ignored that CO_2 reacts with negative ions. The reaction between NH_3 and CO_2 is extremely fast, and the absorption of CO_2 in ammonia occurs in a fast first-order reaction. In reality, cars are a major source of CO_2 emissions. We will have less air pollution if we reduce CO_2 emissions from cars. Therefore, future work will focus on how to store CO_2 and convert it into a less hazardous material in hybrid vehicles to reduce air pollution. As a result of the obtained results in this paper, removing CO_2 not only reduces environmental impact but also reduces extra ammonia, resulting in a more environmentally friendly industry [36–38].

4. Conclusion

In this paper, laboratory experiments were conducted to measure the carbon dioxide adsorption rate in an aqueous ammonia solution by using a filled tower. The temperature has a significant effect on CO_2 uptake into aqueous ammonia solutions, as demonstrated by the obtained results. An optimal temperature for this experiment was found to be around 40°C; temperatures above which had detrimental effects on absorption rates. In a filled tower, CO_2 adsorption rates are determined by the main operating parameters. Furthermore, it increases with liquid flow intensity, gas flow intensity, ammonia concentration, and input CO_2 concentration. It was also observed that ϕ increases with the partial pressure of the gas mass (PCO_2). On the other hand, the total mass transfer coefficient (KGav) is determined by calculating the slope of straight lines on the diagram of the adsorption rate of the gas mass versus its partial pressure at various concentrations of ammonia. Consequently, it seems that an aqueous ammonia solution is an ideal adsorbent for absorbing CO_2 from combustion gases. This solution can be used as an adsorbent with low operating costs that absorbs CO_2 from the exhaust gases of the ammonia unit in order to boost urea production.

In the CO_2 adsorption unit, the biggest loss was attributed to adsorption and disposal towers. It is possible to improve the thermodynamic conditions of the tower by altering the feed conditions and using reboilers and side condensers. An analysis of tower diagrams shows that a uniform distribution of the driving force leads to higher thermodynamic efficiency. Changes to the feed of the ammonia unit and an increased amount of vapor have resulted in a rise in temperature in the reboiler of the disposal tower.

As found in this paper, converting CO to CO_2 increases the overall efficiency of the unit. In fact, less CO is needed to methanize. It was also found that the temperature of the shift reactor could be approached up to 365°C. In this case, the rate of conversion of CO to CO_2 and the overall efficiency of the unit increase, while the rate of CO escape from shift reactors decreases.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. M. Stewart and T. L. Roberts, “Food security and the role of fertilizer in supporting it,” *Procedia Engineering*, vol. 46, pp. 76–82, 2012.

[2] L. Chen, Z. Xie, X. Zhuang, X. Chen, and X. Jing, “Controlled release of urea encapsulated by starch-g-poly(l-lactide),” *Carbohydrate Polymers*, vol. 72, no. 2, pp. 342–348, 2008.

[3] W. M. Stewart, D. W. Dibb, A. E. Johnston, and T. J. Smyth, “The contribution of commercial fertilizer nutrients to food production,” *Agronomy Journal*, vol. 97, no. 1, pp. 1–6, 2005.

[4] T. H. Trinh, K. Kushaari, A. S. Shuib, L. Ismail, and B. Azeem, “Modelling the release of nitrogen from controlled release fertiliser: constant and decay release,” *Biosystems Engineering*, vol. 130, pp. 34–42, 2015.

[5] E. Schultz, T. DeSutter, L. Sharma et al., “Response of sunflower to nitrogen and phosphorus in North Dakota,” *Agronomy Journal*, vol. 110, no. 2, pp. 685–695, 2018.

[6] M. Mahmood, M. Maaz Maqsood, T. Hussain Awan, M. Tahir Mahmood, M. Maqsood, and R. Sarwar, “Effect of different levels of nitrogen and intra-row plant spacing on yield and yield components of maize,” *Pakistan Journal of Agricultural Sciences*, vol. 38, no. 2, pp. 1-2, 2001, 2021, https://www.researchgate.net/publication/269630567.

[7] M. A. Shehzad and M. Maqsood, “Integrated nitrogen and boron fertilization improves the productivity and oil quality of sunflower grown in a calcareous soil,” *Turkish Journal of Field Crops*, vol. 20, no. 2, pp. 213–222, 2015.

[8] M. Awais, A. Wajid, A. Ahmad et al., “Nitrogen fertilization and narrow plant spacing stimulates sunflower productivity,” *Turkish Journal of Field Crops*, vol. 20, no. 1, pp. 99–108, 2015.

[9] G. Yang, H. Tang, Y. Nie, and X. Zhang, “Responses of cotton growth, yield, and biomass to nitrogen split application ratio,” *European Journal of Agronomy*, vol. 35, no. 3, pp. 164–170, 2011.

[10] M. Awais, A. Wajid, A. Ahmad, and A. Bakhsh, “Narrow plant spacing and nitrogen application enhances sunflower (helianthus annuus l.) productivity,” *Pakistan Journal of Agricultural Sciences*, vol. 50, no. 4, pp. 689–697, 2013, 2021, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1068.6039&rep=rep1&type=pdf..

[11] B. Beig, M. B. K. Niazi, Z. Jahan, A. Hussain, M. H. Zia, and M. T. Mehran, “Coating materials for slow release of nitrogen from urea fertilizer: a review,” *Journal of Plant Nutrition*, vol. 43, no. 10, pp. 1510–1533, 2020.

[12] IFASTAT2021, https://www.ifastat.org/.

[13] S. Nasreen and S. M. Imamul Huq, “Effect of sulphur fertilizer on yield and nutrient uptake of sunflower crop in an Albaquept soil,” *Pakistan J. Biol. Sci.*, vol. 5, no. 5, pp. 533–536, 2002.

[14] S. Najafian and M. Zahedifar, “Antioxidant activity and essential oil composition of Satureja hortensis L. as influenced by sulfur fertilizer,” *Journal of the Science of Food and Agriculture*, vol. 95, no. 12, pp. 2404–2408, 2015.

[15] G. H. Debaba, A. Hartono, U. Sudadi, and L. T. Indriyati, “Establishing soil phosphorus critical level for potato (solanum tuberosum l.) in andisol of Lembang, Indonesia,” *J. ISSAAS*, vol. 25, no. 1, pp. 11–20, 2019, 2021, http://issaasphil.org/wp-content/uploads/2019/06/2.-Debaba-et-al-2019-Potato-Soil-Indonesia-FINAL.pdf..

[16] B. Azeem, K. Kushaari, Z. B. Man, A. Basit, and T. H. Thanh, “Review on materials & methods to produce controlled release coated urea fertilizer,” *Journal of Controlled Release*, vol. 181, no. 1, pp. 11–21, 2014.

[17] T. L. de Souza, D. R. Guelfi, A. L. Silva, A. B. Andrade, W. F. T. Chagas, and E. L. Cancellier, “Ammonia and carbon dioxide emissions by stabilized conventional nitrogen fertilizers and controlled release in corn crop,” *Ciência e Agrotecnologia*, vol. 41, no. 5, pp. 494–510, 2017.

[18] M. A. Isla, H. A. Irazoqui, and C. M. Genoud, “Simulation of a urea synthesis reactor. 1. Thermodynamic framework,” *Industrial and Engineering Chemistry Research*, vol. 32, no. 11, pp. 2662–2670, 1993.

[19] M. Dente, S. Pierucci, A. Sogaro, G. Carloni, and E. Rigolli, “Simulation program for urea plants,” *Computers and Chemical Engineering*, vol. 12, no. 5, pp. 389–400, 1988.

[20] H. A. Irazoqui, M. A. Isla, and C. M. Genoud, “Simulation of a urea synthesis reactor. 2. Reactor model,” *Industrial and Engineering Chemistry Research*, vol. 32, no. 11, pp. 2671–2680, 1993.

[21] M. Skorupka, A. Nosalewicz, P. Krasilnikov, and M. A. Taboada, “Ammonia volatilization from fertilizer urea—a new challenge for agriculture and industry in view of growing global demand for food and energy crops,” *Agriculture*, vol. 11, no. 9, p. 822, 2021.

[22] M. Mokarram, M. J. Mokarram, A. R. Zarei, and B. Safarinejadian, “Using adaptive neuro-fuzzy network (ANFIS) to predict underground water quality in west of Fars province during 2003 to 2013 period,” *Iranian Journal Of Ecohydrology*, vol. 4, no. 2, pp. 547–559, 2017.

[23] E. Carsanba, S. Papanikolaou, P. Fickers, B. Agirman, and H. Erten, “Citric acid production by Yarrowia lipolytica,” in *In Non-conventional yeasts: from basic research to application*, pp. 91–117, Springer, Cham, 2019.

[24] K. Cecon, “Chemical translation: the case of Robert Boyle’s experiments on sensible qualities,” *Annals Of Science*, vol. 68, no. 2, pp. 179–198, 2011.

[25] R. Aranda, L. A. Stern, M. E. Dietz, M. C. McCormick, J. A. Barrow, and R. F. Mothershead, “Forensic utility of isotope ratio analysis of the explosive urea nitrate and its precursors,” *Forensic Science International*, vol. 206, no. 1-3, pp. 143–149, 2011.

[26] S. M. Safieddin Ardebili and A. Khademalrasoul, “An assessment of feasibility and potential of gaseous biofuel production from agricultural/animal wastes: a case study,” *Biomass Conversion and Biorefinery*, pp. 1–10, 2020.

[27] M. J. Mokarram, M. Gitizadeh, T. Niknam, and S. Niknam, “Robust and effective parallel process to coordinate multi-area economic dispatch (MAED) problems in the presence of uncertainty,” *IET Generation Transmission and Distribution*, vol. 13, no. 18, pp. 4197–4205, 2019.

[28] L. Principe, “Chemical translation’ and the role of impurities in alchemy: examples from basil Valentine’s Triumph-Wagen,” *Ambix*, vol. 34, no. 1, pp. 21–30, 1987.

[29] O. A. Salman, “Polymer coating on urea prills to reduce dissolution rate,” *Journal of Agricultural and Food Chemistry*, vol. 36, no. 3, pp. 616–621, 1988.

[30] M. Samer, W. Berg, H. J. Müller et al., “Radioactive 85Kr and CO_2 balance for ventilation rate measurements and gaseous emissions quantification through naturally ventilated barns,” *Transactions of the ASABE*, vol. 54, no. 3, pp. 1137–1148, 2011.

[31] F. Barzagli, F. Mani, and M. Peruzzini, “Carbon dioxide uptake as ammonia and amine carbamates and their efficient conversion into urea and 1,3-disubstituted ureas,” *Journal of CO2 Utilization*, vol. 13, pp. 81–89, 2016.

[32] M. Mokarram, M. J. Mokarram, M. Gitizadeh, T. Niknam, and J. Aghaei, “A novel optimal placing of solar farms utilizing multi-criteria decision- making (MCDA) and feature selection,” *Journal of Cleaner Production*, vol. 261, p. 121098, 2020.

[33] M. K. Moghimi and F. Mohanna, “Real-time underwater image enhancement: a systematic review,” *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1509–1525, 2021.

[34] E. Alper and O. Yuksel Orhan, “CO_2 utilization: developments in conversion processes,” *Petroleum*, vol. 3, no. 1, pp. 109–126, 2017.

[35] H. T. Oh, Y. Ju, K. Chung, and C. H. Lee, “Techno-economic analysis of advanced stripper configurations for post- combustion CO_2 capture amine processes,” *Energy*, vol. 206, p. 118164, 2020.

[36] M. J. Mokarram, M. Gitizadeh, T. Niknam, and K. E. Okedu, “A decentralized granular-based method to analyze multi-area energy management systems including DGs, batteries and electric vehicle parking lots,” *Journal of Energy Storage*, vol. 42, p. 103128, 2021.

[37] M. K. Moghimi and F. Mohanna, “A joint adaptive evolutionary model towards optical image contrast enhancement and geometrical reconstruction approach in underwater remote sensing,” *Applied Sciences*, vol. 1, no. 10, pp. 1–12, 2019.

[38] M. K. Moghimi and F. Mohanna, “Real-time underwater image resolution enhancement using super-resolution with deep convolutional neural networks,” *Journal of Real-Time Image Processing*, vol. 18, pp. 1653–1667, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9680519, 17 pages
https://doi.org/10.1155/2022/9680519

Research Article

Fabric Defect Segmentation System Based on a Lightweight GAN for Industrial Internet of Things

Bo Li,[1] **Yongkai Zou**,[1] **Rongbo Zhu**,[2] **Wei Yao**,[1] **Jun Wang**,[3] **and Shaohua Wan**[4]

[1]*College of Computer Science, South-Central Minzu University, Wuhan 430074, China*
[2]*College of Informatics, Huazhong Agricultural University, Wuhan 430070, China*
[3]*Yuxiang Technology (Hangzhou) Co. Ltd., Hangzhou 310024, China*
[4]*School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430074, China*

Correspondence should be addressed to Rongbo Zhu; rbzhu@mail.hzau.edu.cn

Received 23 February 2022; Accepted 20 April 2022; Published 23 May 2022

Academic Editor: Varun Menon

Machine vision systems based on deep learning play an important role in the industrial Internet of things (IIoT) and Industry 4.0 applications, especially for product quality monitoring. Fabric defect detection is an important task in the industrial production of textiles and is crucial for product quality assurance. In actual production, the detection of many small and weak target defects remains challenging. Furthermore, industrial production requires high production rates and small model sizes in practice. This study proposes a lightweight segmentation system that meets real-time industrial production requirements. Herein, first, the defect sample image was repaired based on the image repair mechanism of the generative adversarial network model. Then, the difference between the defect sample and the repaired sample was obtained and subsequent processing, such as denoising and enhancement, was done. Finally, the defect areas were segmented. Our model was specifically designed for the segmentation of weak and small defects. This was achieved through adversarial training, optimization of an objective function, and image processing. Experimental comparisons show that the intersection over union of the three different datasets is 77.84%, 77.85%, and 73.6% and that our model is superior to the conventional semantic segmentation model. Furthermore, our model has good image restoration quality with a low mean absolute error and high structural similarity index. Additionally, our model is lightweight, has good real-time performance, and is suitable for applications in the IIoT and industrial production lines, such as embedded systems.

1. Introduction

In recent years, the industrial Internet of things (IIoT) has accelerated its integration into traditional industries and, therefore, has evolved into various applications. With the deployment of machine vision systems on the edge side, an automated production inspection line can be established for product defect detection; the inspection results can be transmitted to the cloud to provide data support to satisfy different customer needs. Migrating complete or partial tasks to the edge can diminish the network bandwidth, computing, and storage requirements of a cloud center [1]. Fabric defect detection has attracted significant research attention in the textile industry. In industrial production, it is essential to segment fabric defects to ensure the high quality of fabric products [2]. With the development of machine learning and machine vision technology, machine vision-based methods to solve textile quality control problems have gradually become an industry trend because of their high accuracy, fast detection speed, and low labor cost [3, 4].

Many researchers have used various algorithms and models for the automatic detection and segmentation of fabric defects [5, 6]. Two methods are applied for defect detection, one of which involves the use of classical image analysis algorithms, such as texture models [7], Fourier analysis [8], and Gabor filters [9].

The second method is based on deep learning algorithms [10, 11] that can often achieve good results. However, in practical applications, some problems remain, for example, compared to normal samples, fewer defect samples can be obtained during production, and there are few observable types of defects. Additionally, the conventional labeling of defect samples is time-consuming and labor-intensive.

Traditional segmentation networks have insufficient segmentation capabilities for small and weak defect samples. In recent years, the generative adversarial network (GAN) model [12] has become increasingly favored and valued by researchers because of the strong modeling ability of the discriminator. It can continuously judge the difference between the segmentation results from the generator and the ground truth. The discriminator and the generator are optimized to obtain a segmentation feature map of multicontext features, which enables the generation of segmented images that are infinitely close to the ground truth. Therefore, in this study, we introduce a GAN as a fabric defect segmentation model.

Although the number of defect samples in the production is generally small and the types of defects that appear are also few, expanding the training sample set by heavy manual annotation work is not the best choice. By learning a small quantity of samples, the probability distribution of the texture and other features of the normal sample can be obtained. Some unknown random defects that are excluded from existing training samples often appear in industrial production. Regardless of the type and characteristics of the defect, differences exist between the normal sample area and the defect area. Therefore, the defect area can be determined based on these differences. Zhao et al. proposed a defect detection model based on positive samples, which first repairs the defect area and then determines the defect area by comparison [13]. The method of combining GANs and autoencoders is used to repair the defect image; then, local binary pattern (LBP) features are used to detect defects. The LBP method has a good effect on large-scale defects. However, the contours detected by the LBP features may be inaccurate for small and weak defects. In this study, we achieved the segmentation of the defect area based on the GAN image repair mechanism. In addition, many of the models have a large size and do not consider actual production needs.

Our research motivation is to develop a lightweight real-time system suitable for industrial production, confronting the more difficult detection problems of weak and small defects. Many existing models have good effects on conventional fabric defects. However, there are some small defects with low contrast, which affect the further improvement of product quality. Therefore, the detection and segmentation of these weak and small defects have become important research tasks. In addition, in actual production, the model must be as light as possible, occupy a small space, and meet real-time requirements. Deep learning systems are deployed at the edge and can play a very important role in the IIoT [14], and our system can be deployed on local production lines or at the edge of the IIoT.

In response to the abovementioned problems, herein, we studied the detection and segmentation of multiple types of defects in actual production samples, focusing on the segmentation of weak and small defects. A method for quickly constructing a large number of samples that conform to the true probability distribution is proposed. In addition, a model designed and optimized to occupy a small space and have a fast segmentation speed is presented, which can also be applied to industrial fabric production. A GAN model was used to realize the segmentation of fabric defects. Adversarial training not only makes the model more stable but also increases the accuracy of defect segmentation.

In summary, the main contributions of this article are summarized as follows.

(1) A fabric defect segmentation system suitable for industrial applications is proposed. The system is composed of a defect sample-synthesizing module without manual annotation, defect repair module, and defect segmentation module.

(2) Using a combined image processing method, we designed a defect segmentation model for weak and small defects, including an objective function for confrontation training, a normalization method, and a learning rate decay strategy, which contribute to the accurate segmentation of defects.

(3) The segmentation model proposed herein has the advantages of being lightweight and functions in real time, which is especially suitable for applications in IIoT and industrial production lines, such as embedded systems.

The remainder of this paper is organized as follows: Section 2 summarizes related work in recent years, Section 3 introduces our methodology, and Section 4 discusses the training process and model optimization method. Experimental results are presented and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Many classical algorithm models have been used in fabric defect segmentation and detection research, for example, wavelet analysis [15, 16], fuzzy C-means method [17], Gabor filter [18], established texture distribution model [19], Elo Rating algorithm [20], Bayesian classifier based on statistical features [21], and XGBoost classifier based on the genetic algorithm [22]. These methods have achieved good results in solving many application problems in related scenarios.

In recent years, with the development of deep learning theory, many valuable models have been proposed for semantic segmentation, such as the FCN [23], faster R-CNN [24], spatial pyramid pooling [25], U-Net [26], YOLO [27], RefineNet [28], SegNet [29], DeepLab [30], Fisher criterion [31], encoder-decoder [32], two-parallel-branch deep network [33], LSTM [34], EfficientDet [35], and multiscale network [36].

In the application field, fabric defect detection is also a type of object surface detection. Some researchers have conducted research in this field. For instance, surface defect detection is based on deep learning methods [37, 38].

In fabric production, defects are only a small part of the abnormal samples. In recent years, some researchers have proposed anomaly detection methods and models [39] for image anomaly data detection. With the development of the GAN model, new models and theories have emerged consistently, such as conditional GANs [40, 41], cycle-consistent GANs [42], and style-based GANs [43]. In recent years, owing to the powerful learning ability of the GAN model, some researchers have also used GAN models for anomaly detection [44].

Schlegl et al. proposed AnoGAN [45], which is trained on positive samples to learn a mapping from the latent space. Akcay et al. subsequently proposed GANomaly [46]; their approach only requires a generator and a discriminator as in a standard GAN architecture, which is an improvement compared to AnoGAN and EGBAD. Perera et al. proposed OCGAN [47], and Ngo et al. proposed Fence-GAN, which corrects the GAN loss and has a better anomaly classification accuracy [48]. Zhao et al. proposed a defect detection framework based only on positive sample training [13]. The defect area in the sample is first repaired, and then, the model compares the input defect sample with the restored sample to determine the exact defect area. Furthermore, Wang et al. proposed a method using the GAN model with locality-preferred recoding for visual anomaly detection [49]. Nema et al. proposed an unpaired GAN model for brain tumor segmentation [50]. To identify small changes in small structures, Murugesan et al. proposed a new context-based loss function and a new architecture, SegGLGAN [51]. Liu et al. proposed a multistage GAN [52] model for fabric defect detection, which can automatically generate multiple defect samples.

In addition, Huang et al. recently adopted a deep learning model to segment defects, requiring only a small number of training samples [53].

In recent years, some researchers have proposed lightweight systems for applications, such as in underwater object [54], salient object [55], and blind road detection and crosswalks [56].

Although previous studies have made their own contributions, limited research on the balance of requirements of lightweight, real-time, and high-accuracy fabric defect detection and segmentation has been conducted in enterprise production. This article designs a system that includes three modules for the abovementioned problems and a model for the segmentation of weak and small defects.

3. Our Proposed Methodology

3.1. Problem Statement and Framework Description

3.1.1. Problem Statement. A fabric image with defects can be divided into normal and several defective regions. The normal region meets certain characteristics such as grayscale, color, and texture, whereas the defective regions do not meet the characteristics of the normal region.

Let I denote a fabric image and S denote a predicate with the same properties, and the image includes n regions R_i $(i = 1, 2, \cdots, n)$. Among them, R_1 is a normal region and the other regions are defective regions that satisfy

$$\begin{gathered}\bigcup_{i=1}^{n} R_i = I, \\ R_i \cap R_j = \varnothing, \\ S(R_i) = \text{true}, \\ S\left(R_i \cup R_j\right) = \text{false}.\end{gathered} \tag{1}$$

Defect segmentation results can be described as

$$I_{\text{mask}} = \begin{cases} 0, & \text{if } P(x, y) \in R_1, \\ 1, & \text{else}, \end{cases} \tag{2}$$

where $i = 1, 2, \cdots, n, j = 1, 2, \cdots, n, \text{and } i \neq j.P(x, y)$ is the image pixel, I_{mask} is the result after defect segmentation, and the region with a pixel value of one in I_{mask} is the defect region.

3.1.2. Framework Description. This study is a two-step segmentation method based on a GAN model, as shown in Figure 1. The first step is to repair the defect image to obtain the corresponding repaired image. The second step is to compare the two images to obtain the difference result and obtain the mask result of the defect area using image processing methods such as denoising, linear transformation, and binarization processing. The experiment was divided into three modules: a module that synthesizes defect samples, defect repair module, and defect segmentation module.

3.2. Synthesizing Defect Samples. Our experiments used samples taken from the equipment during the fabric production process. Owing to the rapid improvement of production processes, few defect samples can be obtained. To meet the needs of sample training, we designed a method to quickly obtain a large number of experimental samples without manual labeling. As shown in Figure 2, first, the defect areas are separated to obtain the defect block using only a small number of existing defect samples combined with the corresponding labeling information.

We used the "sliding cutting" method with a sliding step and cutting resolution. By sliding and cropping each sample, new images with cutting resolution were obtained. By determining whether the label corresponding to the cropped image contains a label, determining whether the cropped image is a defect image can be easy. In this manner, we can obtain a large number of normal samples with a cutting resolution.

Then, these few defect blocks are randomly pasted into the existing normal background by programming while recording the labeling information at the same time. In this way, a set of samples, including the defect, repaired, and mask images, is quickly obtained. Here, "random" includes the random selection of defect blocks and random pasting positions. Figure 2 shows an example of a method for artificially synthesizing and constructing defect samples.

This method not only quickly provides a large number of defect samples that are close to the original defect sample distribution but also directly obtains the corresponding

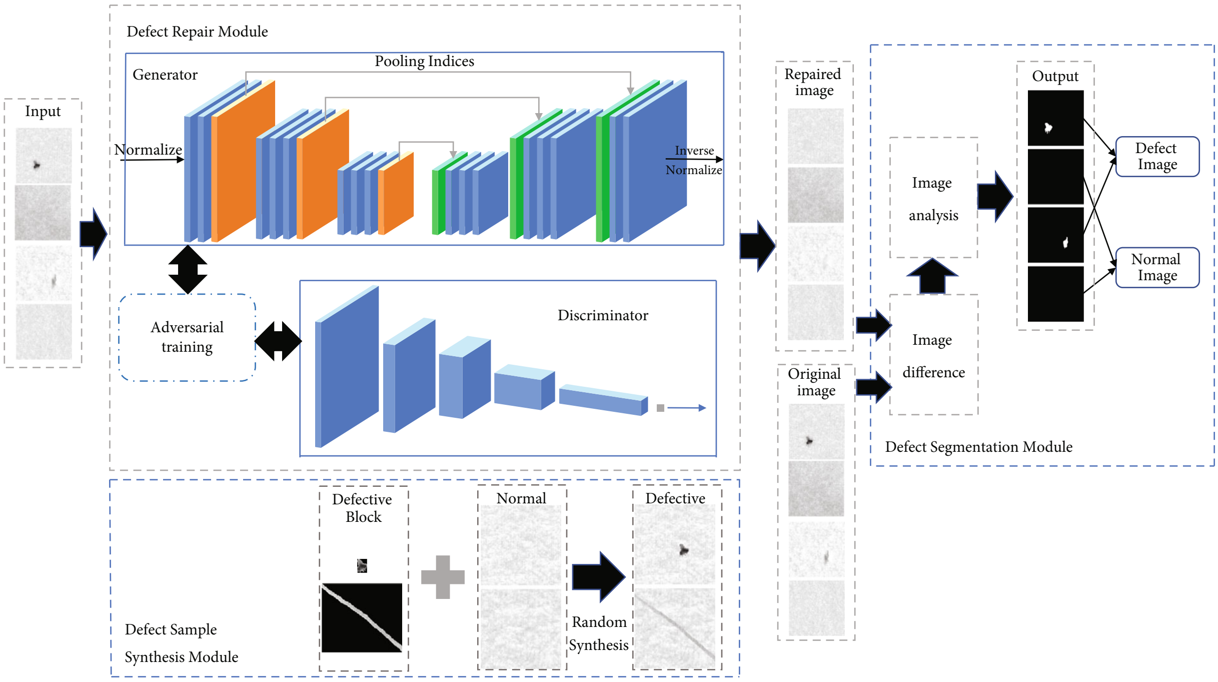


Figure 1: Architecture of our proposed system.

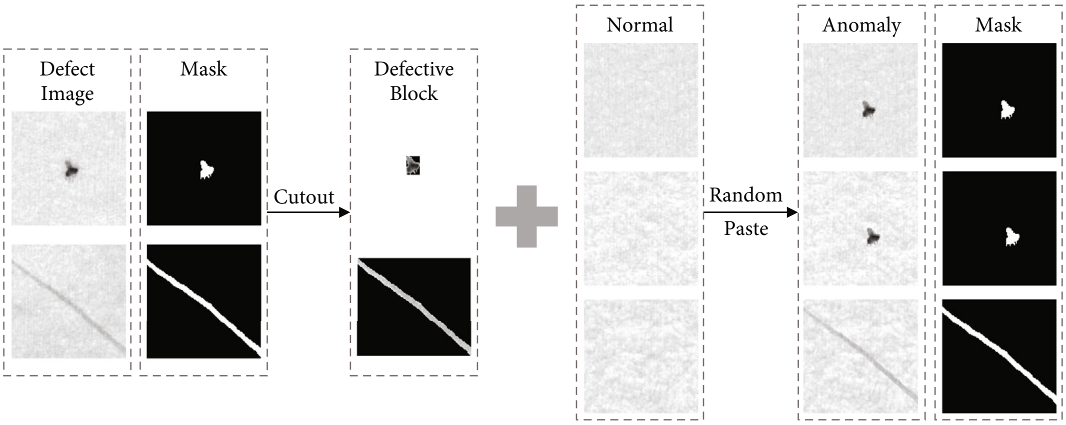


Figure 2: Method for synthesizing defect samples.

annotation information, which can replace the tedious work of manual labeling. Figure 3 illustrates some examples of synthesized defect samples.

3.3. Defect Repairing. Considering the real-time requirements of industrial production, we designed a simplified SegNet model. Compared with FCN [23] and U-Net [26], SegNet [29] uses the position information during maximum pooling. This does not require learning and, therefore, reduces the number of end-to-end training parameters. SegNet cleverly achieves upsampling by recording the position of the maximum value during pooling, and because there is no deconvolution process, it improves the training speed of the model.

In this study, we propose a concise SegNet with a reduced number of network layers. As shown in Figure 4, the model uses fewer coding and decoding layers but can retain more detail for repairing the image while also significantly reducing the storage space occupied by the model. In the encoding process, convolution and maximum pooling are alternately used to complete the downsampling of the image. This process is followed only three times (the original SegNet involves five downsampling times). In the decoding process, maximum depooling and convolution are alternately used and are performed only three times. Furthermore, the LeakyReLU activation function was used directly for the output. Pooling indices (location information during pooling) are used to transfer the decoder, record the location information during pooling, and directly place the value back to the original location for unpooling.

Figure 5 shows the structure of the discriminant model, which uses a six-layer convolutional encoder structure. After

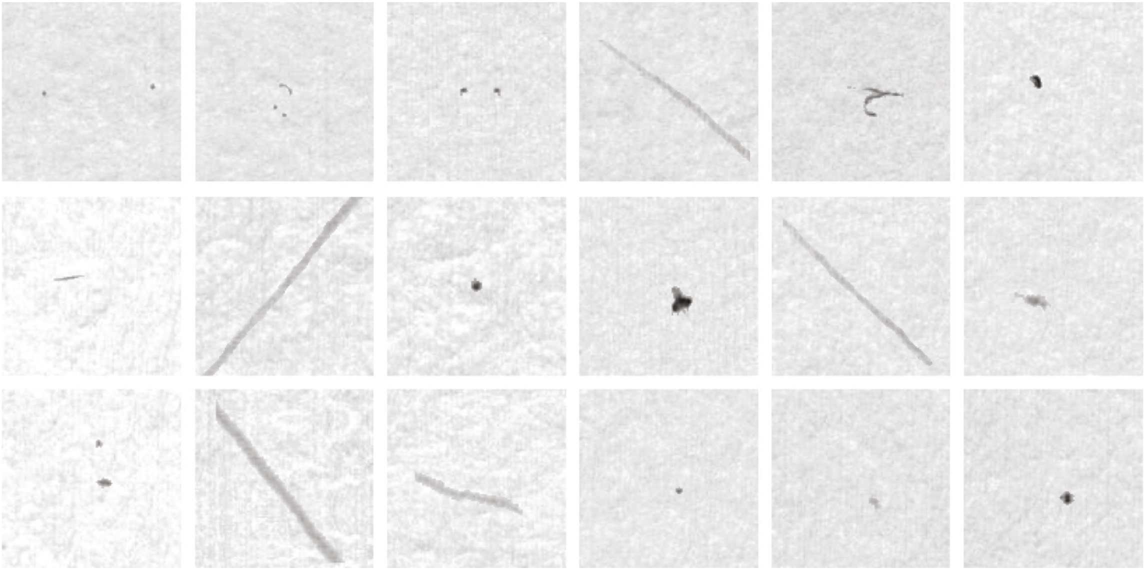


Figure 3: Examples of synthesized defect samples.

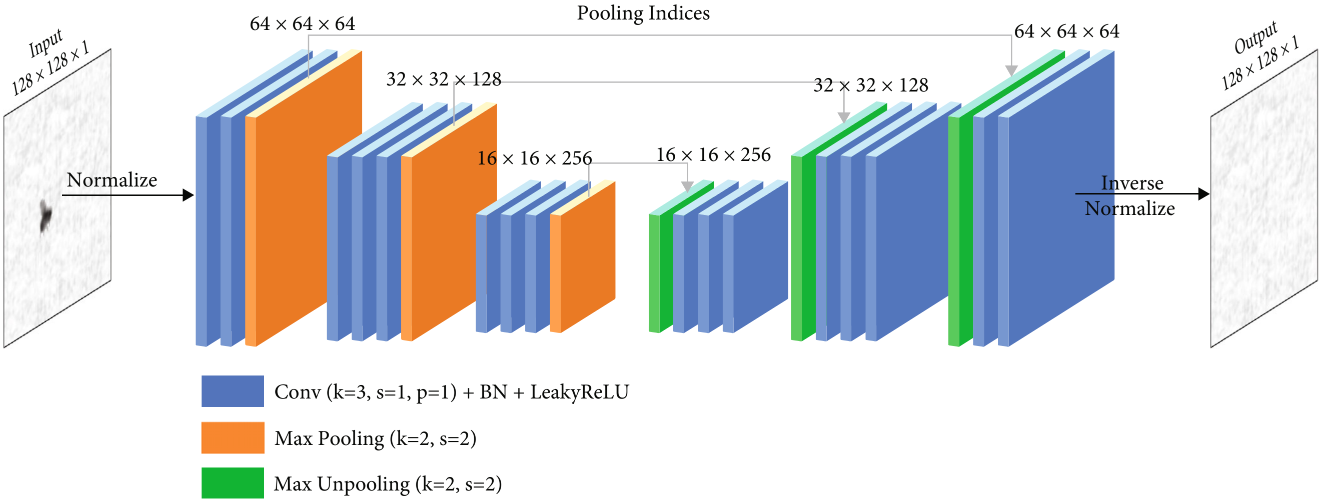


Figure 4: Generated network structure.

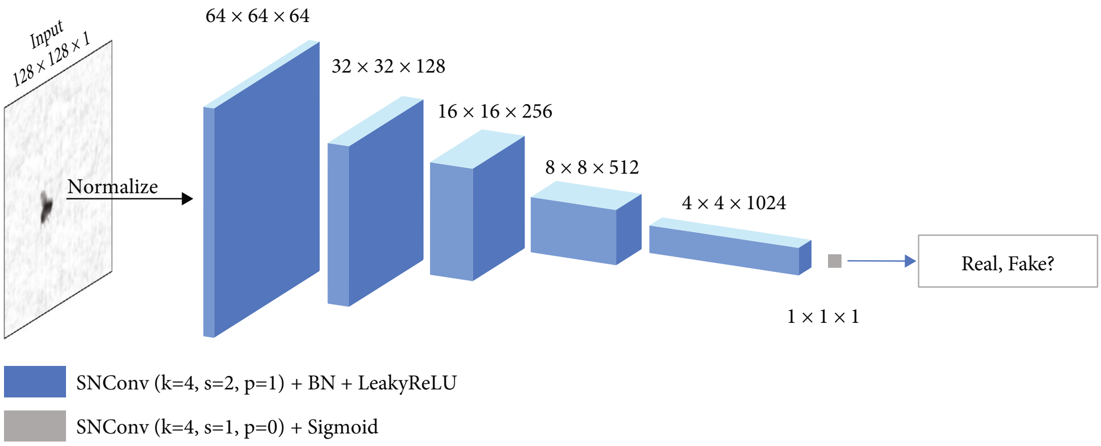


Figure 5: Discriminant network structure.

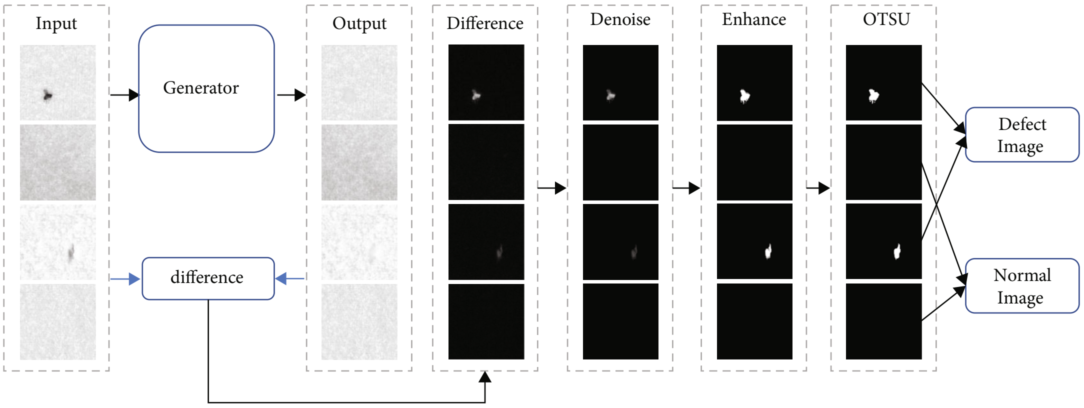


Figure 6: Process of segmenting defect images.

convolution, it is activated by the LeakyReLU function, and the last output layer uses the sigmoid function. Then, a score is obtained to determine whether the input image is truly normal based on the probability value.

3.4. Defect Segmentation. The end-to-end defect repair model is finally obtained through alternate training of the G and D networks. The test samples were inputted into the generated network to obtain a normal image.

There are two situations in this study: if there are any defects in the input sample, the model repairs the defects; otherwise, there is no significant difference between the output of the model and the input if the sample is normal. The original image and the repaired image need to be compared to obtain the difference image. The image difference can be described by formula (3).

$$\mathrm{Dif}(x, y) = \left|I_{\mathrm{ori}}(x, y) - I_{\mathrm{rep}}(x, y),\right. \tag{3}$$

where $I_{\mathrm{ori}}(x, y)$ is the original image, $I_{\mathrm{rep}}(x, y)$ the repaired image, and $|\bullet|$ the absolute value sign.

Since the defect area in the difference image may not be apparent, several enhancement operations are required, as shown in Figure 6. Conventional filtering methods are not used for denoising because they may blur the edges and details of the target. Rather, the threshold method is used to denoise the image directly, based on the background of the difference image. While filtering out the noise, the details of the segmented target can be preserved. The threshold method can be described by formula (4).

$$I_{\mathrm{deno}}(x, y) = \begin{cases} \mathrm{Dif}(x, y), & \text{if } \mathrm{Dif}(x, y) \geq \mathrm{th}, \\ 0, & \text{else}. \end{cases} \tag{4}$$

where th is a threshold for denoising.

Then, the brightness and contrast of the difference images are enhanced by a linear transformation, as in formula (5).

$$Y = \alpha X + \beta, \tag{5}$$

where X represents the pixel value of a certain point in the original image and Y represents the pixel value of the corresponding position after transformation. The contrast of the image can be adjusted using α, and the brightness of the image can be changed using β.

Finally, the OTSU algorithm is used for binarization to obtain the required mask image, which is the final segmentation result.

4. Training Process and Model Optimization

4.1. Training Process. In the sample-synthesizing module, we obtain the image group consisting of the defect image, repaired image, and mask image. Only the first two were used in the training of the repair model. In the GAN training, an alternate iteration method is adopted for model training. First, the G network was trained, following which the D network was trained. The training of the D network also requires the output of the G network in the previous round of gradient backpropagation as input. Figure 7 shows the training process for the G and D networks.

For the G network, the defect samples are input into the generation model to generate a fake repair image, and then, the discriminant model is used to obtain a score. The expected repair image generated is sufficiently real; therefore, this score will form an error with the true label "1." Meanwhile, an error is formed between the false and true repair images generated. The aforementioned two errors are combined to form the loss function of the G network, and the parameters of the G network can be updated by the gradient backpropagation through the loss function.

For the training process of the D network, a score was obtained after the true repaired image was inputted into the discriminant model. The D network is expected to be able to accurately distinguish between true and false repaired images. Therefore, the discriminant score of the true repaired image and the true label "1" form an error. Similarly, the score of the false repaired image and the false label "0" form an error. The average of the two errors constitutes the loss function of the D network.

The role of the D network is to interfere with the generation model, that is, the score of the true repaired image tends to the true label "1" and the score of the false repaired image tends to the false label "0." This contradicts the

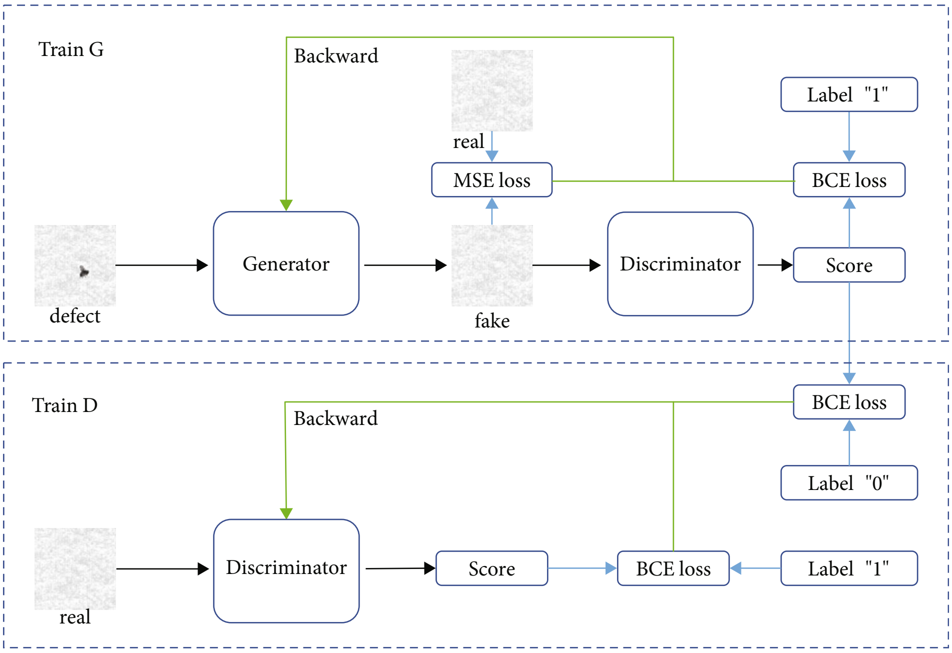


Figure 7: Training process of defect repair model.

expectation of the G network that the score of the fake repaired image tends to the true label "1," which is the antagonism of the GAN model. In an ideal situation, when the scores obtained after the true and false repaired images entering the discriminant model are all close to 0.5, it means that the discriminant model is unable to distinguish between the true and false repaired images. This means that the sample generated by the generation model has become the data of the real sample distributed. At this time, the model reaches an ideal balance.

4.2. Objective Function. In the training of the adversarial segmentation network, there are four errors: the discriminant error and the generation error of the G network and two discriminant errors of the D network. Therefore, four loss functions were included in the error analysis. For the G network, there was an error between the false repaired image and the true repaired image. The mean square error (MSE) was used for evaluation. In addition, there is an error between the score of the fake repaired image and the true label "1." This is a binary classification problem. Binary cross entropy (BCE) was used to calculate the loss. Similarly, in the D network, both errors were binary classification problems and BCE was used to calculate the loss.

First, we observe the composition of the MSE loss function, as shown in formula (6). An additional sample number average compared to the Euclidean distance formula can be described as the expected value of the square of the difference between the true value and the estimated value. The MSE loss of the G network can be simply described by formula (7).

$$\mathrm{MSE}(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{6}$$

where y_i and $\hat{y}_i$ are the true and estimated values, respectively.

$$\mathrm{MSE\ loss} = E\,[y - G(x)]^2, \tag{7}$$

where x and y are the defect and true repaired samples, respectively.

The calculation of the BCE loss function is described in formula (8), where $\hat{y}_i$ is the evaluation value of the sample and y_i is the label of the binary classification, which is 0 or 1.

$$\mathrm{BCE}(\hat{y}_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i)], \tag{8}$$

$$\begin{aligned} \mathrm{BCE}(\hat{y}_i, 1) &= \frac{1}{n}\sum_{i=1}^{n}[-\log \hat{y}_i], \quad (\text{if label}\, y_i = 1), \\ \mathrm{BCE}(\hat{y}_i, 0) &= \frac{1}{n}\sum_{i=1}^{n}[-\log(1 - \hat{y}_i)], \quad (\text{if label}\, y_i = 0\,). \end{aligned} \tag{9}$$

It can be concluded that the objective function of the D network can be expressed as

$$\min \quad V(D) = \frac{1}{2}E[-\log(D(y))] + \frac{1}{2}E[-\log(1 - D(G(x)))]. \tag{10}$$

The objective function of the G network can be described as

$$\min \quad V(G) = (\lambda - 1)E[y - G(x)]^2 + \lambda E[-\log(D(G(x)))], \tag{11}$$

Then, to unify the formula, the objective function of the D network can be changed to

$$\max \quad V(D) = \frac{1}{2}E[-\log(1 - D(y))] + \frac{1}{2}E[-\log(D(G(x)))]. \tag{12}$$

That is, the final objective function of the trained model can be described as

$$\min_{G} \max_{D} \quad V(G, D) = (\lambda - 1)E[y - G(x)]^2 + \lambda(E[-\log(1 - D(y))] + E[-\log(D(G(x)))]). \tag{13}$$

4.3. Model Optimization. Aiming at the fabric sample characteristics, especially weak and small targets, some optimizations were performed on the model.

(1) As mentioned earlier, the model in this study adopts the largest depooling layer in SegNet because, compared to deconvolution, the amounts of calculation and space occupation are less. Meanwhile, fewer coding and decoding layers are used but can retain more details to repair the image and significantly reduce the storage space occupied by the model. The downsampling process uses only eight layers of convolution and three layers of maximum pooling. Meanwhile, the multiclass SoftMax output layer was removed, making the output come directly after convolution. This simplifies the model and can meet the needs of an industrially embedded system

(2) Instance normalization (IN) is used instead of regular batch normalization (BN) in the defect repair network because IN is suitable for repairing defect images to normal images in the generator model. Since the result of image generation mainly depends on a certain image instance, using IN not only accelerates the model convergence but also maintains the independence between each image instance

(3) This model uses LeakyReLU as the activation function. For a regular ReLU activation function, when the input value is negative, the output value would be zero. Since the training goal is to obtain a repaired image, which has a similar value range as the input defect image, it needs to be activated as a negative value when the input is negative. Therefore, choosing LeakyReLU as the activation function not only solves the problem that ReLU can easily lead to necrosis but also ensures that the information is not completely lost when the input is negative. Therefore, the defect repair model can generate a better repair result

5. Performance Analysis

5.1. Experimental Setup. The experimental environment used was PyTorch1.3.1, Windows 10 system, CUDA 10.1, GPU: GTX 1050ti, and cuDNN 7.0. The sample resolution was 128 × 128 pixels.

Table 1: Experimental setup.

Parameter	Setting
Batch size	32
Training epoch	100
Initial learning rate	0.001
Momentum	0.5/0.999
Learning rate	0.0001/0.00001

The main parameter settings of the experiment are shown in Table 1.

All data in each epoch went through the network. In the training process, this experiment used the Adam optimizer, which combines the advantages of the RMSProp and AdaGrad optimization algorithms. In this experiment, the initial learning rate set for the Adam optimizer was 0.001 and the momentum values of the first-order moment and second-order moment estimation were 0.5 and 0.999, respectively. Meanwhile, a multistep learning rate decay strategy (MultiStepLR) was set. In the experiment, the learning rate was 0.0001 when the epoch was 30 and the learning rate was 0.00001 when the epoch was 60. The advantage of this setting is that the loss of the model can be rapidly reduced in the early stage and can gradually reach the optimum in the later stage.

Image processing after defect repair was implemented using the OpenCV method; the denoising threshold was set to 19, linear transformation process used the convertScaleAbs method, and α and β parameters were set to 5 and 0, respectively.

5.2. Datasets. The experiments in this study used the following three datasets:

5.2.1. Enterprise Dataset. The fabric defect samples in the experiment originated from the image acquisition equipment on the enterprise assembly line. In the production process, high-speed cameras are used to monitor product quality.

There were 4360 original samples, and the original image resolution was 371 × 257 pixels. After removing duplicate and invalid samples, there were only a total of 90 samples and these were labeled for defects. Then, we used image rotation, flip, transpose, and other operations to enlarge the image set and obtain seven new forms of defect images. In the process of transforming the defect image, the label corresponding to the image is also expanded so that there is no need to label the new defect image one by one.

We then adopted the method described in Section 3, to quickly obtain a large number of defect samples that were close to the original defect sample distribution, where the "sliding cutting" method was used with a sliding step of 20 and cutting resolution of 128 × 128 pixels.

5.2.2. AITEX Dataset. The AITEX dataset [57] is composed of 245 images of 4096 × 256 pixels with seven different fabric structures. There are 140 nondefect images in the database

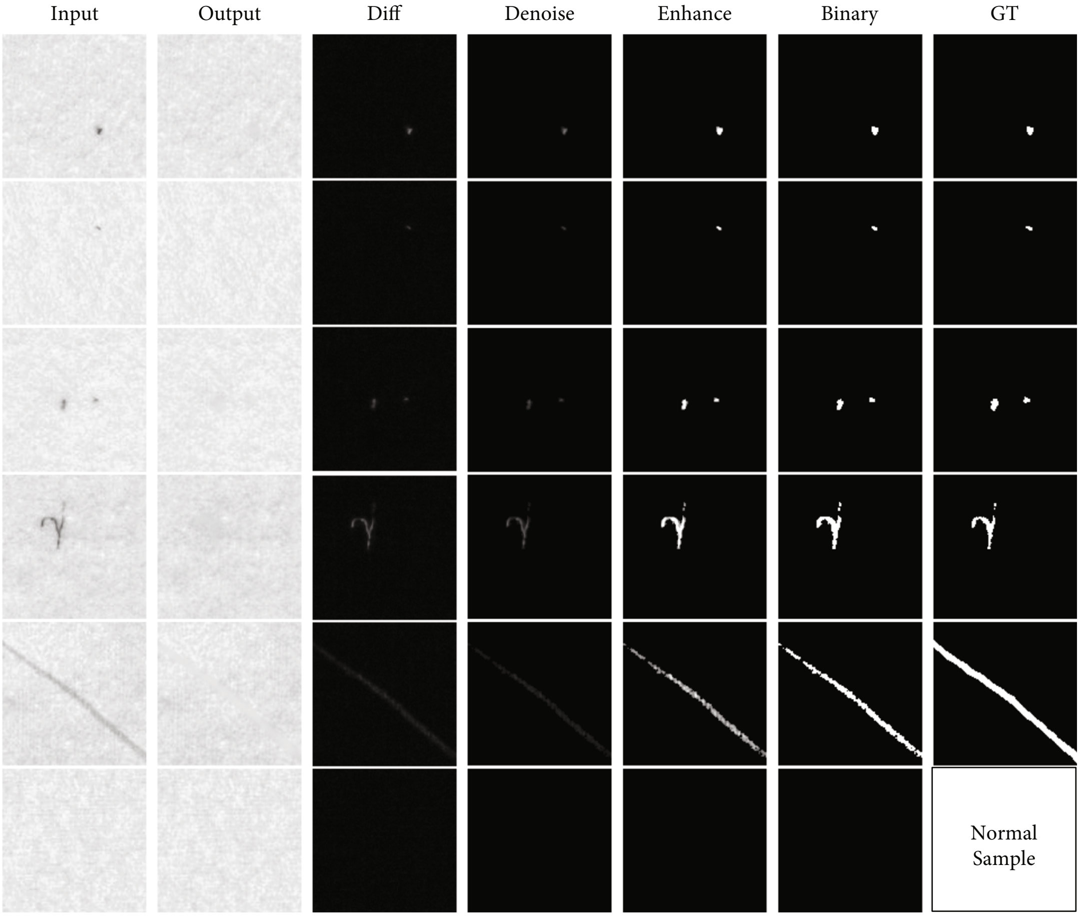


Figure 8: Segmentation results of enterprise data (the first five rows are defect samples, and the last row lists normal samples).

and 105 images of 12 different types of fabric defects that are common in the textile industry.

5.2.3. Expanded Dataset. Because the defect samples are actually small, the existing defects and types of defects are very limited. Therefore, we created artificial defect samples. Such defects did not appear in the training set and were, therefore, used to test whether our defect segmentation model was effective.

5.3. Evaluation Metrics. Our model is evaluated using several metrics, such as Pixel Acc and intersection over union (IoU). Pixel Acc represents the ratio of the number of correctly classified pixels to the total number of pixels in the segmentation image, including correctly classified background points. The IoU measures the similarity between the segmentation result and the ground truth, as shown in formula (14). Each pixel in the segmentation result is divided into four types, that is, true positive (TP) (the number of defect pixels that are correctly divided into the defect area by the model), false positive (FP) (the number of background pixels that are incorrectly divided into the defect area by the model), false negative (FN) (the number of defect pixels that are incorrectly divided into the background area by the model), and true negative (TN) (the number of background pixels that are correctly divided into the background area by the model).

$$\text{IoU} = \frac{TP}{TP + FN + FP}. \tag{14}$$

We used the mean absolute error (MAE) to evaluate the average pixel error after image repair, as in formula (15).

$$\text{MAE} = \frac{\sum_{i=1}^{n} |A_i - C_i|}{n}, \tag{15}$$

where A_i is the original value of the ith pixel, C_i is the repaired value of the ith pixel, and n is the total number of pixels in an image.

In addition, we used the structural Ssmilarity Iidex (SSIM) [58] to analyze the quality of image restoration.

5.4. Test Results

5.4.1. Enterprise Fabric Samples. Through the sample-synthesizing module, we obtained 19606 pairs of artificial defect samples. A total of 10240 pairs of samples were used as the training set, and 320 pairs were used as the validation set to observe the effect that training had on the model. The

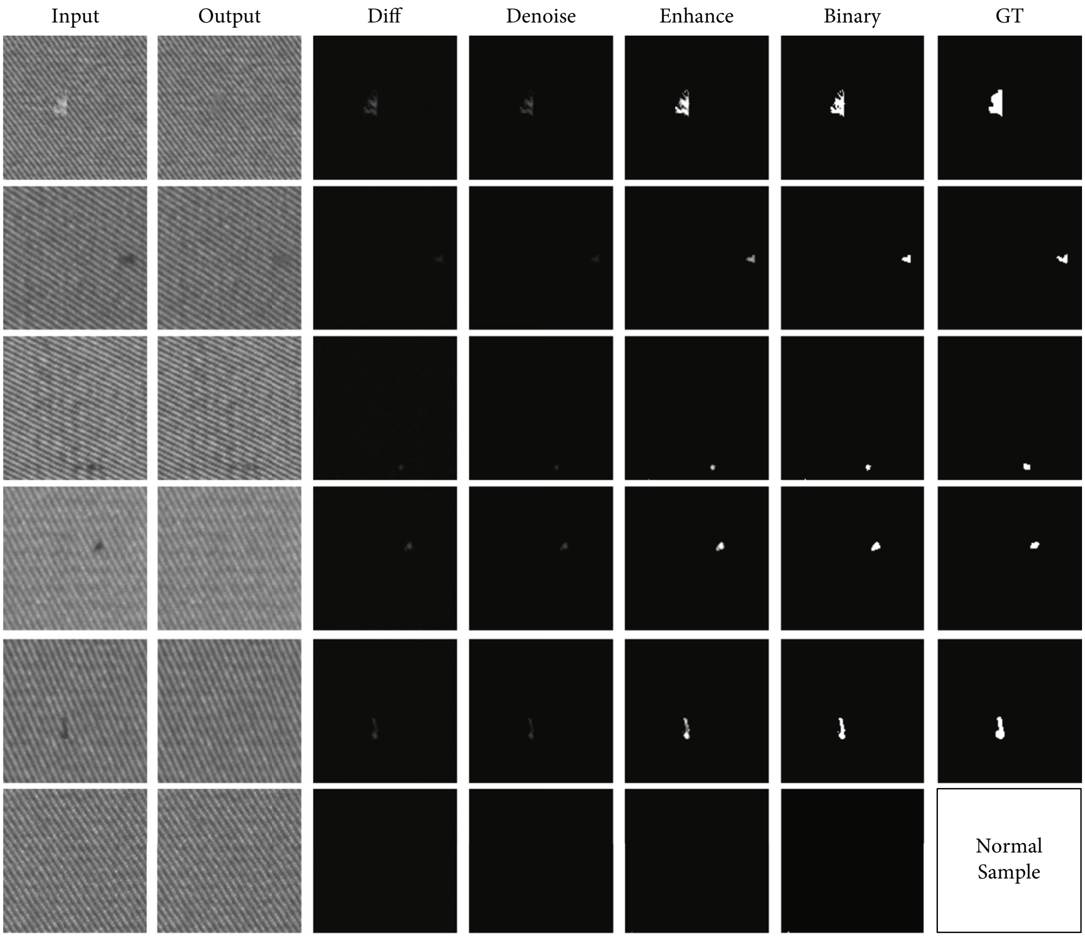


Figure 9: Segmentation results of AITEX samples (the first five rows are defect samples, and the last row lists normal samples).

Table 2: A comparison of segmentation results of different models.

	Pixel Acc	IoU
FCN [23]	0.9944	0.5811
U-Net [26]	0.9959	0.6400
SegNet [29]	0.9963	0.6811
FCNGAN	0.9952	0.5876
U-NetGAN	0.9951	0.6367
SegNetGAN	0.9968	0.7034
Ours	0.9968	0.7784

test set used the original 720 defect samples and 6480 normal samples.

In the fabric defect samples, the optimal model-generated error (MSE error) in the validation set was only 0.00021. The Pixel Acc was 99.68%, and IoU accuracy was 77.84%. Figure 8 shows the segmentation results for the samples.

5.4.2. AITEX Dataset. First, the large-sized samples were cropped to obtain 128 × 128 pixel samples. The sample preprocessing method and training hyperparameter settings were the same as those of the previous fabric defect sample set.

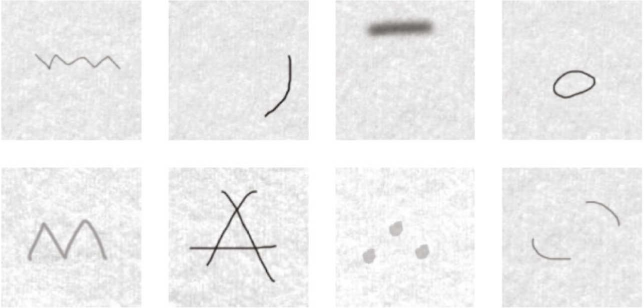


Figure 10: Defect samples by artificial construction.

The optimal model-generated error (MSE error) on the validation set was 0.00056, the Pixel Acc of defect segmentation was 99.94%, and the IoU score was 77.85%. Figure 9 illustrates the segmentation results for the AITEX samples. The model in this study also has a good effect on this type of model with a more complex background in terms of segmentation accuracy. Table 2 lists the experimental results for several samples in this model.

5.4.3. Extended Dataset. There would be some undetected and excluded defects in the training because defects of an

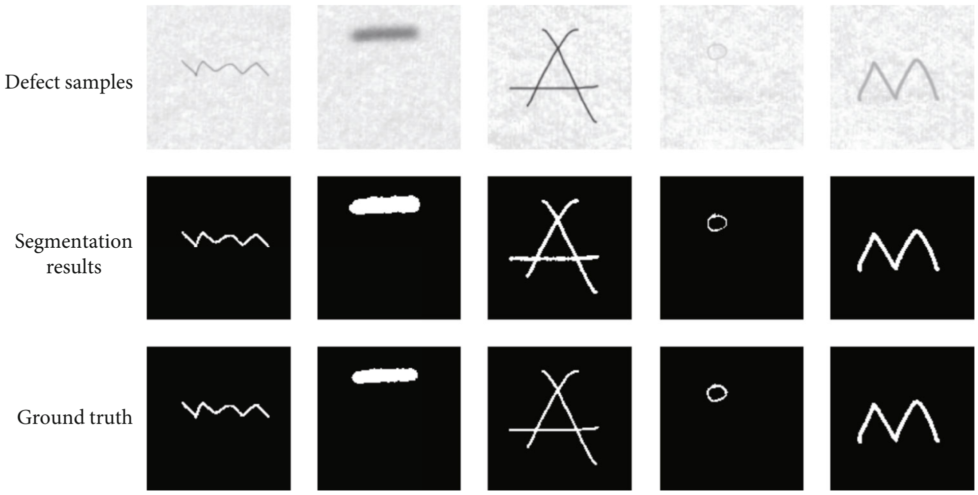


Figure 11: Segmentation results of artificially defected samples.

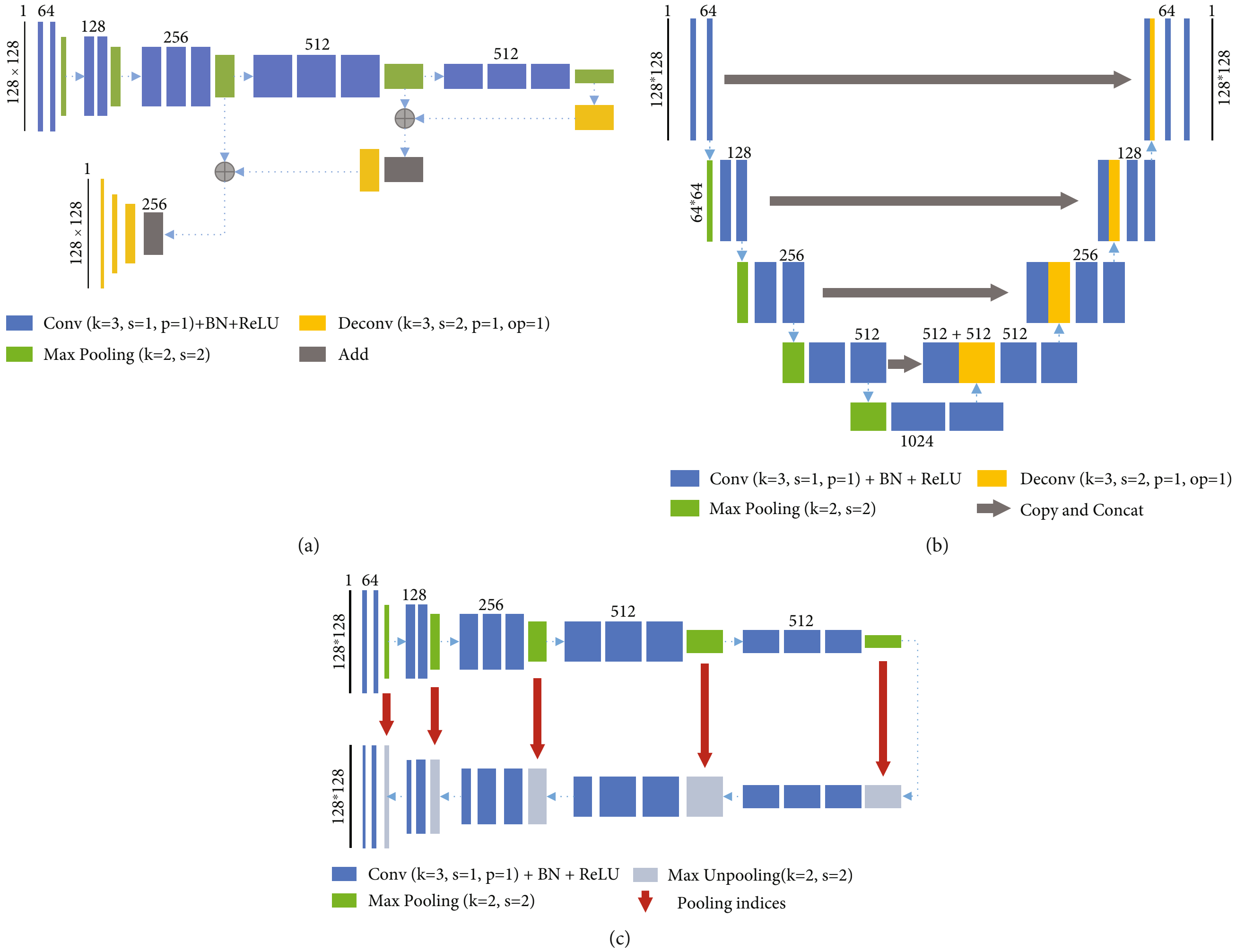


Figure 12: Three models in Table 2. (a) FCN, (b) U-Net, and (c) SegNet.

unknown type may be present in the production process. Several new types of defect samples were artificially constructed to test the robustness of our proposed model, as shown in Figure 10. These types of defects did not appear in the previous training samples and test samples. Examples of such defects are large-area defects and long-line defects.

After testing the extended sample set, the results showed that the segmentation effect of our previous model was

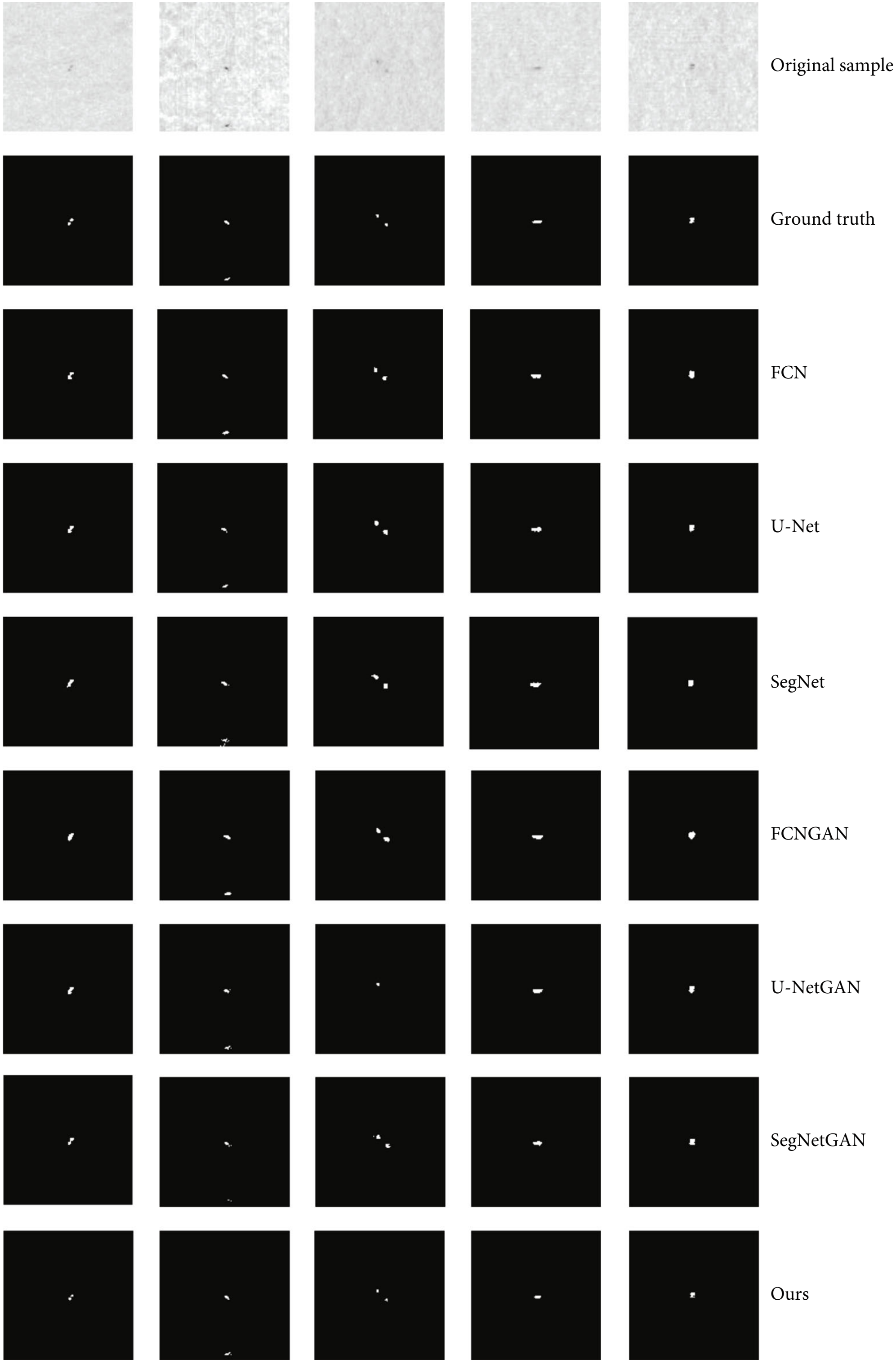


Figure 13: Segmentation results of weak and small-defect samples.

excellent. Figure 11 illustrates examples of the artificial defect segmentation results. In the extended set of 64 samples, the Pixel Acc of segmentation reached 99.3% and the IoU score reached 73.6%. The results achieved the segmentation accuracy of the existing defect samples.

5.5. *Analysis*

5.5.1. Comparative Experiments. To compare the performance of the different models, we implemented six other models, as shown in Table 2.

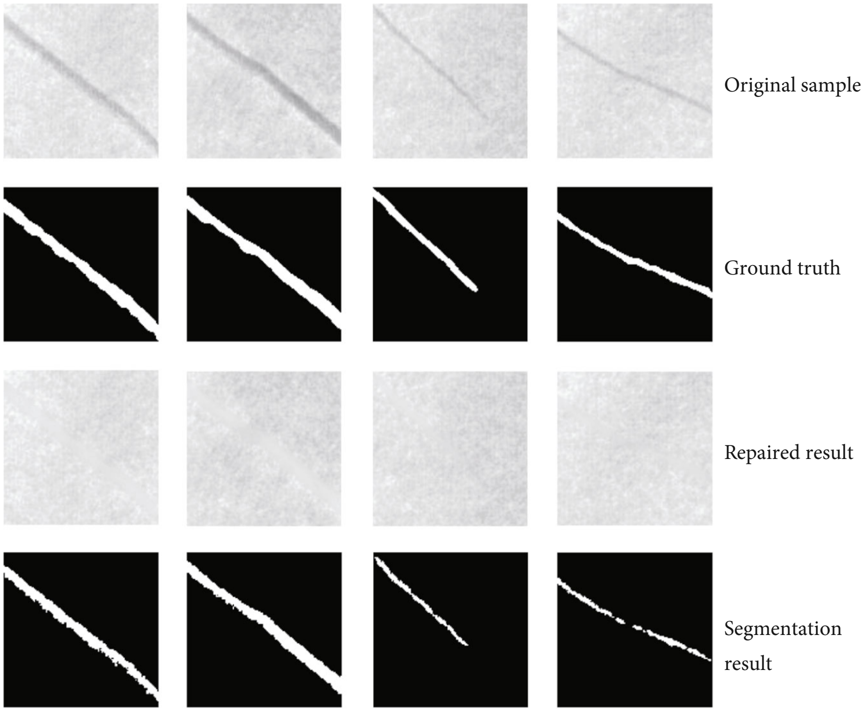


Figure 14: Segmentation results of uneven background samples.

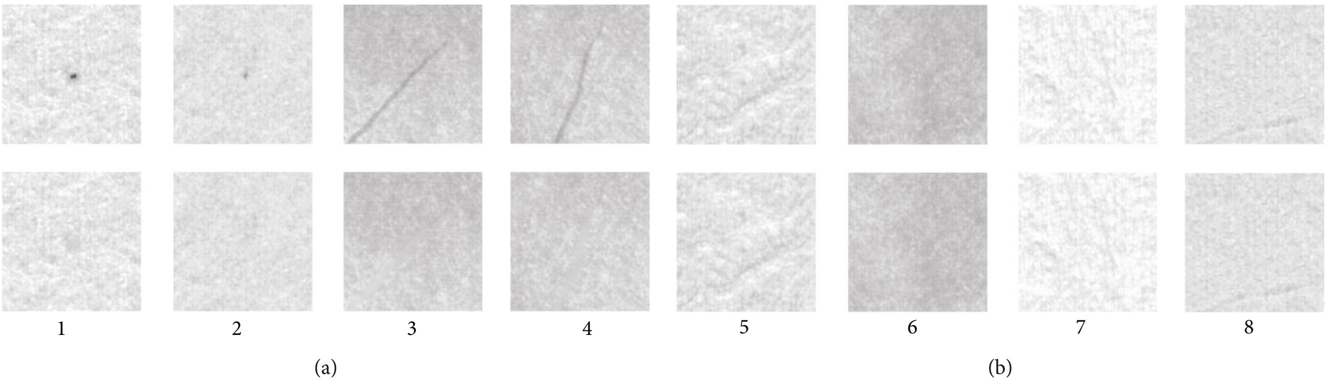


Figure 15: Repair results. (a) Repair results of the defective sample (the first row lists the original samples, and the second row lists the repaired results). (b) Repair results of the normal sample (the first row lists the original samples, and the second row lists the repaired results).

Table 3: MAE of repaired results in Figure 15.

Sample	1	2	3	4	5	6	7	8
MAE	1.22	0.87	2.18	1.83	0.98	1.65	1.28	0.85

Table 4: SSIM results in Figure 15.

Sample	1	2	3	4	5	6	7	8
SSIM	0.981	0.986	0.941	0.953	0.989	0.976	0.981	0.989

Table 5: A comparison of model size and FPS for various architectures.

	Model size (MB)	FPS
FCN [23]	89.2	132
U-Net [26]	131.7	69
SegNet [29]	112.4	109
FCNGAN	89.2	131
U-NetGAN	131.7	69
SegNetGAN	112.4	106
Ours	14.4	107

The FCN, U-Net, and SegNet models are described in [23, 26, 29], respectively. The specific implementation details of the three models are presented in Figure 12.

FCNGAN, U-NetGAN, and SegNetGAN in Table 2 indicate a model obtained by training based on the GAN mechanism, where FCN, U-Net, and SegNet, respectively, are used as the G network and the D network is composed of a six-layer convolutional network. The D-network model is shown in Figure 5. The training parameters of the models listed in Table 2 are consistent with those listed in Table 1.

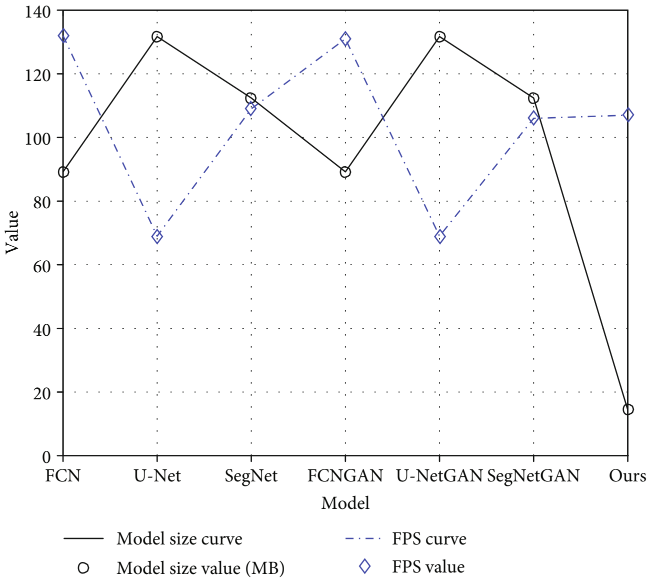


Figure 16: Comparison of the model size and FPS for various architectures.

Table 6: Segmentation results of different resolution.

Resolution	Pixel Acc	IoU	FPS
256 × 256	0.9985	0.7754	37
128 × 128	0.9968	0.7784	107

The Pixel Acc of the above models exceeded 0.99. The IoU indicator of the abovementioned model widened the gap, and the gap between high and low reached approximately 20%.

Among the first three segmentation models, SegNet has more advantages and the three segmentation evaluation indicators are better than the other two models. After the GAN training mechanism was introduced into the three segmentation models, the segmentation performance of FCNGAN and SegNetGAN was improved but U-NetGAN did not. Through comparison, it was found that the GAN training mechanism did not significantly improve the performance of the three semantic segmentation models. The model proposed herein achieved the best experimental results, and the segmentation performance evaluation index was better than the other six models. In this experiment, many of the sample defect areas were weak and small targets; therefore, the fluctuation of the segmentation results had a greater impact on the IoU but the IoU reached 0.7784, which is 7.5% higher than the best result of 0.7034 in the other six models.

5.5.2. Segmentation Effects of Weak and Small-Defect Samples. We compared the segmentation results of each model for weak and small-defect samples. As shown in Figure 13, the first row contains five weak and small-defect samples and the second row is the ground truth. Upon comparison, it was found that the first sample on the left contained two very small defects situated very close. The model segmentation result is closest to the ground truth, which separates two small defects.

In another example, there were two small defects in the third sample. The segmentation result of the U-NetGAN model misses the defect, and the defect segmentation results of the other models are enlarged. The segmentation results of the model proposed in this study are the most accurate.

5.5.3. Samples with Uneven Background. To test the ability of the model proposed herein to repair defect samples, we selected some defect samples with uneven backgrounds for testing. The test results show that the proposed model can effectively segment the defects. Since our model uses an image difference algorithm, the defect area that is very similar to the background may not be continuous in the segmentation results, as illustrated in Figure 14.

5.5.4. Analysis of the Sample Repair Effect. This study provides representative samples for analyzing the repair results of our model. In Figure 15(a), sample 1 has an apparent flaw, sample 2 has a weak and small defect, and samples 3 and 4 have long stripe defects. These four samples in Figure 15(b) illustrate normal samples with different backgrounds and textures.

Figure 15(a) illustrates the result of repairing the four defect samples. From the results of the repair, the flaws assumingly disappeared. We used the MAE to evaluate the average pixel error after image repair, as shown in Table 3. From the MAE results, the pixel error of the repaired image was less than three. Since samples 3 and 4 had long strips of flaws with larger areas, the MAE was also larger.

Figure 15(b) illustrates the results of normal sample image restoration. The results generated by our model are nearly identical to those of the original images. The MAE results showed that the average pixel error was less than two. The results showed that the repair effect of our model was excellent.

Moreover, we used SSIM to analyze the quality of image restoration. Table 4 presents the SSIM results, which show that the similarity between the original and repaired samples was generally high. In the sample in Figure 15(a) with flaws, samples 1 and 2 have a minimal effect on the similarity because the flaws are small, whereas, in samples 3 and 4, the similarity decreases owing to the larger area of the flaws. For the normal sample in Figure 15(b), the similarity was high. This indicates that the repair quality of our model is good.

5.5.5. Model Size Comparison. To verify whether the model can meet real-time requirements, we tested the segmentation speed of seven different models. The research index is the number of cotton samples (frames per second (FPS)) that the model can process in one second. The experimental results are presented in Table 5 and Figure 16. The model size represents the size of the saved model file. The model proposed in this article occupies a small space, only 14.4 MB, which is easy to be embedded in industrial equipment.

The first three models follow the probability that the smaller the model, the higher the FPS value. This is because the model size and computing speed do not necessarily show an anticorrelation. The size of the model directly represents the number of parameters of the model, but the speed of the model calculation is not only related to the number of parameters but also affected by the structure of the model. The processing speeds of the seven models can meet real-time requirements. Since our model is calculated using a GPU, the process of repairing the network to obtain the repaired image is very fast: the processing of 7200 samples takes only approximately 56 s (equal to 128 FPS). As for the image processing operations after repairing the network, the calculation time is only slightly increased based on the OpenCV calculation on the CPU. This results in a decrease in the overall FPS but still achieves good real-time performance.

Considering that the resolution of the test samples has an impact on the FPS, we used a sample set with a resolution of 256 × 256 pixels to test the model again; this included 720 defect samples and 6480 normal samples. These test samples did not come from an enlarged 128 × 128-pixel image but were cut directly from the original cotton cloth sample. The test results are presented in Table 6. The test results show that the FPS decreased owing to the increase in sample resolution; however, the real-time requirements can still be reached.

6. Conclusion

In this study, a lightweight system composed of three modules was designed to solve the segmentation problem of fabric defects, particularly for weak and small-defect targets. We used a GAN model based on the repair mechanism, which is lightweight and has good defect segmentation ability. The results of testing corporate samples and samples from a public database show that the model proposed in this study has good segmentation effects and can achieve real-time performance, thus demonstrating its application value in IIoT and industrial production lines.

In the future, we will focus on few-shot and unsupervised learning. In addition, further improvements in real-time performance are worth studying.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the project of the National Natural Science Foundation of China (Grant 61772562 and Grant 61976226) and the Hubei Provincial Natural Science Foundation of China for distinguished young scholars (Grant 2017CFA043). The authors thank Mr. Zheng He for the original data and Dr. Jingang Yu for the constructive discussions and suggestions.

References

[1] S. Wan, S. Ding, and C. Chen, “Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles,” *Pattern Recognition*, vol. 121, article 108146, 2022.

[2] J. Yang, C. Wang, B. Jiang, H. Song, and Q. Meng, “Visual perception enabled industry intelligence: state of the art, challenges and prospects,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 2204–2219, 2021.

[3] A. Kumar, “Computer-vision-based fabric defect detection: a survey,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 348–363, 2008.

[4] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, “Fabric defect detection systems and methods—a systematic literature review,” *Optik*, vol. 127, no. 24, pp. 11960–11973, 2016.

[5] A.-A. Tulbure, A.-A. Tulbure, and E.-H. Dulf, “A review on modern defect detection models using DCNNs – deep convolutional neural networks,” *Journal of Advanced Research*, vol. 35, pp. 33–48, 2022.

[6] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, “Automated fabric defect detection—A review,” *Image and Vision Computing*, vol. 29, no. 7, pp. 442–458, 2011.

[7] F. S. Cohen, Z. Fan, and S. Attali, “Automated inspection of textile fabrics using textural models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 803–808, 1991.

[8] C.-H. Chan and G. K. H. Pang, “Fabric defect detection by Fourier analysis,” *IEEE Transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, 2000.

[9] L. Tong, W. K. Wong, and C. K. Kwong, “Differential evolution-based optimal Gabor filter model for fabric inspection,” *Neurocomputing*, vol. 173, pp. 1386–1401, 2016.

[10] Z. Zhan, J. Zhou, and B. Xu, “Fabric defect classification using prototypical network of few-shot learning algorithm,” *Computers in Industry*, vol. 138, article 103628, pp. 1–11, 2022.

[11] X. Zheng, S. Zheng, Y. Kong, and J. Chen, “Recent advances in surface defect inspection of industrial products using deep learning techniques,” *The International Journal of Advanced Manufacturing Technology*, vol. 113, pp. 35–58, 2021.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[13] Z. Zhao, B. Li, R. Dong, and P. Zhao, “A surface defect detection method based on positive samples,” in *Lecture Notes in Computer Science Pac. Rim International Conference on Artificial Intelligence*, pp. 473–481, Springer, Cham, 2018.

[14] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, “An edge traffic flow detection scheme based on deep learning in an intelligent transportation system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2020.

[15] D.-M. Tsai and C.-H. Chiang, “Automatic band selection for wavelet reconstruction in the application of defect detection,” *Image and Vision Computing*, vol. 21, no. 5, pp. 413–431, 2003.

[16] H. Y. T. Ngan, G. K. H. Pang, S. P. Yung, and M. K. Ng, “Wavelet based methods on patterned fabric defect detection,” *Pattern Recognition*, vol. 38, no. 4, pp. 559–576, 2005.

[17] H. Zhang, J. Ma, J. Jing, and P. Li, “Fabric defect detection using l0 gradient minimization and fuzzy C-means,” *Applied Sciences*, vol. 9, no. 17, pp. 3506–3516, 2019.

[18] Q. Wang, J. Jing, L. Zhang, and X. Wang, “Denim defect detection based on optimal Gabor filter,” *Laser & Optoelectronics Proggress*, vol. 55, no. 7, article 071501, 2018.

[19] D. Yapi, M. S. Allili, and N. Baaziz, “Automatic fabric defect detection using learning-based local textural distributions in the Contourlet domain,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1014–1026, 2018.

[20] X. Kang and E. Zhang, “A universal defect detection approach for various types of fabrics based on the elo-rating algorithm of the integral image,” *Textile Research Journal*, vol. 89, no. 21–22, pp. 4766–4793, 2019.

[21] M. T. Habib, S. B. Shuvo, M. S. Uddin, and F. Ahmed, “Automated textile defect classification by Bayesian classifier based on statistical features,” in *2016 International Workshop on Computational Intelligence (IWCI)*, pp. 101–105, Dhaka, Bangladesh, 2016.

[22] Y. Jiang, G. Tong, H. Yin, and N. Xiong, “A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters,” *IEEE Access*, vol. 7, pp. 118310–118321, 2019.

[23] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, USA, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 1137–1149, 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[26] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, 2016.

[28] 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*, pp. 1925–1934, Honolulu, HI, USA, 2017.

[29] 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.

[30] L. C. 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, no. 4, pp. 834–848, 2018.

[31] Y. Li, W. Zhao, and J. Pan, “Deformable patterned fabric defect detection with Fisher criterion-based deep learning,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, 2017.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, Munich⊠ Germany, 2018.

[33] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, “A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning,” *Future Generation Computer Systems*, vol. 100, pp. 316–324, 2019.

[34] Y. Zhao, K. Hao, H. He, X. Tang, and B. Wei, “A visual long-short-term memory based integrated CNN model for fabric defect image classification,” *Neurocomputing*, vol. 380, pp. 259–270, 2020.

[35] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: scalable and efficient object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10778–10787, Seattle, WA, USA, 2020.

[36] H. Wang, D. Zhang, S. Ding, Z. Gao, J. Feng, and S. Wan, “Rib segmentation algorithm for X-ray image based on unpaired sample augmentation and multi-scale network,” *Neural Computing and Applications*, pp. 1–15, 2021.

[37] R. Ren, T. Hung, and K. C. Tan, “A generic deep-learning-based approach for automated surface inspection,” *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 929–940, 2018.

[38] D. Tabernik, S. Šela, J. Skvařc, and D. Sko, “Segmentation-based deep-learning approach for surface-defect detection,” *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.

[39] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues,” *Knowledge-Based Systems*, vol. 189, article 105124, 2020.

[40] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional GANs,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, Salt Lake City, UT, USA, 2018.

[41] P. Wang and X. Bai, “Thermal infrared pedestrian segmentation based on conditional GAN,” *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6007–6021, 2019.

[42] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *Proceedings of the European conference on computer vision (ECCV)*, vol. 11213, pp. 186–201, Munich⊠ Germany, 2018.

[43] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, Long Beach, CA, USA, 2019.

[44] B. J. B. Rani, "Survey on applying GAN for anomaly detection," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Coimbatore, India, 2020.

[45] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," in *International Conference on Information Processing in Medical Imaging. Lecture Notes in Computer Science*, vol. 10265, pp. 146–157, Springer, Cham, 2017.

[46] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-Supervised Anomaly Detection Via Adversarial Training," in *Asian Conference on Computer Vision*, pp. 622–637, Springer, Cham, 2019.

[47] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: one-class novelty detection using GANs with constrained latent representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019*, pp. 2893–2901, Long Beach, CA, USA, 2019.

[48] P. C. Ngo, A. A. Winarto, C. K. Kou, S. Park, F. Akram, and H. K. Lee, "Fence GAN: Towards better anomaly detection," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 141–148, Portland, OR, USA, 2019.

[49] J. Wang, W. Huang, S. Wang, P. Dai, and Q. Li, "LRGAN: visual anomaly detection using GAN with locality-preferred recoding," *Journal of Visual Communication and Image Representation*, vol. 79, pp. 103201–103208, 2021.

[50] S. Nema, A. Dudhane, S. Murala, and S. Naidu, "RescueNet: an unpaired GAN for brain tumor segmentation," *Biomedical Signal Processing and Control*, vol. 55, pp. 101641–101648, 2020.

[51] B. Murugesan, K. Sarveswaran, S. M. Shankaranarayana, K. Ram, and M. Sivaprakasam, "A context based deep learning approach for unbalanced medical image segmentation," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 1949–1953, Iowa City, IA, USA, 2020.

[52] J. Liu, C. Wang, H. Su, B. Du, and D. Tao, "Multistage GAN for fabric defect detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 3388–3400, 2019.

[53] Y. Huang, J. Jing, and Z. Wang, "Fabric defect segmentation method based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[54] C. H. Yeh, C. H. Lin, L. W. Kang et al., "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–15, 2021.

[55] Y. Liu, X. Y. Zhang, J. W. Bian, L. Zhang, and M. M. Cheng, "SAMNet: stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 3804–3814, 2021.

[56] Z. Cao, X. Xu, B. Hu, and M. Zhou, "Rapid detection of blind roads and crosswalks by using a lightweight semantic segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6188–6197, 2021.

[57] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno, "A public fabric database for defect detection methods and results," *Autex Research Journal*, vol. 19, no. 4, pp. 363–374, 2019.

[58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2845446, 12 pages
https://doi.org/10.1155/2022/2845446

Research Article

Detection of Botnet Attacks against Industrial IoT Systems by Multilayer Deep Learning Approaches

Mohammed Mudassir,[1] **Devrim Unal**,[2] **Mohammad Hammoudeh**,[3] **and Farag Azzedin**[3]

[1]*Department of Mechanical and Industrial Engineering, Qatar University, PO Box 2713, Doha, Qatar*
[2]*KINDI Center for Computing Research, Qatar University, PO Box 2713, Doha, Qatar*
[3]*Information & Computer Science Department, King Fahd University of Petroleum & Minerals, Saudi Arabia*

Correspondence should be addressed to Mohammad Hammoudeh; m.hammoudeh@kfupm.edu.sa

Received 15 January 2022; Revised 17 March 2022; Accepted 11 April 2022; Published 17 May 2022

Academic Editor: Mohammad R Khosravi

The publication of this article was funded by Qatar National Library.

Industry 4.0 is the next revolution in manufacturing technology that is going to change the production and distribution of goods and services within the following decade. Powered by different enabling technologies that are also being developed simultaneously, it has the potential to create radical changes in our societies such as by giving rise to highly-integrated smart cities. The Industrial Internet of Things (IIoT) is one of the main areas of development for Industry 4.0. These IIoT devices are used in mission-critical sectors such as the manufacturing industry, power generation, and healthcare management. However, smart factories and cities can only function when threats to cyber security, data privacy, and information integrity are properly managed. In this regard, securing IIoT devices and their networks is vital to preserving data and privacy. The use of artificial intelligence is an enabler for more secure IIoT systems. In this study, we propose high-performing deep learning models for the classification of botnet attacks that commonly affect IIoT devices and networks. Evaluation of results shows that deep learning models such as the artificial neural network (ANN), the long short-term memory (LSTM), and the gated recurrent unit (GRU) can successfully be used for classifications of IIoT malware attacks with an accuracy of up to 99%.

1. Introduction

The Industrial Internet of Things (IIoT) is the latest technological development in manufacturing and production that is being adopted rapidly all over the world. The IIoT is a part of the more general Internet of Things (IoT) network, which is characterized by its ability to connect billions of devices, appliances, sensors, equipment, and systems and to enable communication among these connected objects or "Things." The IoT market is expected to grow rapidly within the next decade, and its market value is estimated to be at least USD 2 trillion. The bulk of the objects in IIoT is low-powered devices with limited resources (such as battery and processing power). They need support systems for data analytics and security. From industrial equipment to home appliances, most electric-powered devices are becoming smart, interactive, and connected. IIoT is integral to creating cyberphysical systems (CPS) where physical processes are sensed, monitored, controlled, and commanded by humans or computer systems over the Internet. The IIoT is more focused on industrial applications such as smart factories, smart manufacturing, and Industry 4.0. IIoT is often integrated with other IoT networks and applications such as smart cities, smart transportation, smart grids, smart agriculture, smart healthcare, and other smart things. While the definition of Industry 4.0 varies, most notably it is the latest technological revolution in manufacturing that is at least supported by IIoT, CPS, and 5G networks, 3D printing, augmented and virtual reality, simulation, smart contracts, and sustainability measures [1].

As much of the enabling technologies for IIoT are based on IoT, they share a lot of similarities when it comes to security, privacy, and integrity. Although IIoT-enabled devices bring convenience to individuals and companies, this may come at the expense of their privacy [2–4]. Since IIoT devices are equipped with hardware and software that can potentially track user behavior, it is a necessity to design policies and technical solutions to ensure that the privacy, safety, and freedom of the users are always preserved. With numerous devices being connected to the Internet every day, IIoT opens up a broad platform for multifaceted cyberattacks. Common concerns related to IIoT devices include data theft, loss of privacy, and the possibility of abuse through unauthorized access that can take over control of the devices [5]. Many researchers are working on mitigating the various security problems related to IIoT devices and networks. Some of the regular attacks encountered in IIoT devices and networks include the distributed denial of service (DDoS) attacks over different communication protocols, data theft through keylogging and exfiltration, tracking through fingerprinting, and scanning for open ports over the network [6]. Many of these attacks on IIoT devices and networks are performed through botnets [7]. A botnet consists of several Internet-connected devices where each of the devices runs one or more bots. As the botnet infects other IIoT devices, the network of infected devices grows to make the botnet more computationally powerful and carry out larger attacks [8].

Furthermore, the vast applications of IIoT in critical industries and businesses have made them prone to cybercrimes where malicious agents try to override the security systems [9]. The risks involved in the potential overtake of the IIoT devices are enormous. The hazards involved in hacking include the theft of confidential information, the privacy of the public, and in some cases cyberattacks that can even result in loss of lives such as sabotage of medical equipment. For IIoT systems within Industry 4.0, it can mean disruption of production and services, stealing trade secrets, and leakage of confidential business data, all of which could lead to huge financial losses [10]. Hence, it is very important to provide layers of security over the IIoT devices to prevent any loss of data. In recent times, the number of IIoT attacks, especially the attacks carried out by the botnets, has increased substantially. Since there are many types of attacks over various protocols and devices, it becomes increasingly difficult to secure the IIoT devices and networks. Machine learning and deep learning have recently started to gain grounds for malware detection to help with this problem.

Botnets are assumed to be the biggest threats to IIoT networks. A Gartner report estimates that by 2025, the number of IoT devices will reach 50 billion [11]. This vast IoT network is a lucrative target for malicious agents. Many intrusion detection products and services are currently available in the market that offers various levels of protection against IIoT devices. However, there are new threats that emerge every day, and it is important to search for detection methods that are comprehensive, intelligent, and adaptive. Recent advances in machine learning and deep learning show promising results in the classification of attacks [12]. The superiority of deep learning models compared to conventional methods of detection is that they can learn from unstructured data without supervision [13]. Consequently, attacks that are new or can avoid signature-based methods can still be detected by deep learning-based models.

One of the shortcomings to using ML models for classifying malware and network traffic is that the ML model often fail to correctly identify classes that are minority in the train set. [14, 15] used a number of sampling techniques such as oversampling, undersampling, and others for improving the identification of the minority classes.

In this paper, we present deep learning models for the classification of malicious packets originating from IIoT devices. Our results show that deep learning models trained on balanced dataset can give a highly accurate classification of malware data with good precision and recall.

2. Background and Related Work

IIoT devices are often connected to the network and are controlled remotely through a user interface [16]. All of the IIoT devices are based on four characteristics which include a feedback mechanism, a few communication protocols, a control system, and some security layers. The signal to control the system is sent through the interface to the controlling device. The IIoT devices operate based on the signal received and send the feedback back to the interface. This feedback is sent through the sensors placed within the devices. These sensors convert the physical data into electronic signals and send it to the interface through the control systems [17].

2.1. Industry 4.0. Industry 4.0 is the next evolution in manufacturing processes. It is highly integrated across all levels of operation. Figure 1 shows an overview of the Industry 4.0 ecosystem. It is supported by IIoT that allows connectivity between all devices, sensors, machines, and operators. Industry 4.0 allows a high level of autonomy through smart factories. From production to quality control to final delivery of the product, little human intervention is required. Product defects can be identified using computer vision. Additive manufacturing can produce complex designs while reducing material wastage. The operators can be informed of the production processes through wearables. For example, the 3D printer could send a notification to the operator's smartwatch once the fabrication is complete.

Due to the large number of devices, protocols, and systems present in the IIoT network, it is a lucrative target for malware and botnets. For instance, if malware infects the 3D printer, it could alter the design, change the print parameters, and cause damage to the product. Due to the nature of 3D printing, some defects introduced by malware may not be readily noticeable and this could create a hazard for the end-user of the product [10].

Industry 4.0 can also utilize smart contracts and blockchain to help with preserving the integrity of the systems, processes, and operations. For instance, a product design can be cryptographically signed and verified with blockchain to preserve its integrity [18].

2.2. Architecture. The systems behind the functioning IIoT devices are quite complex and based on different kinds of

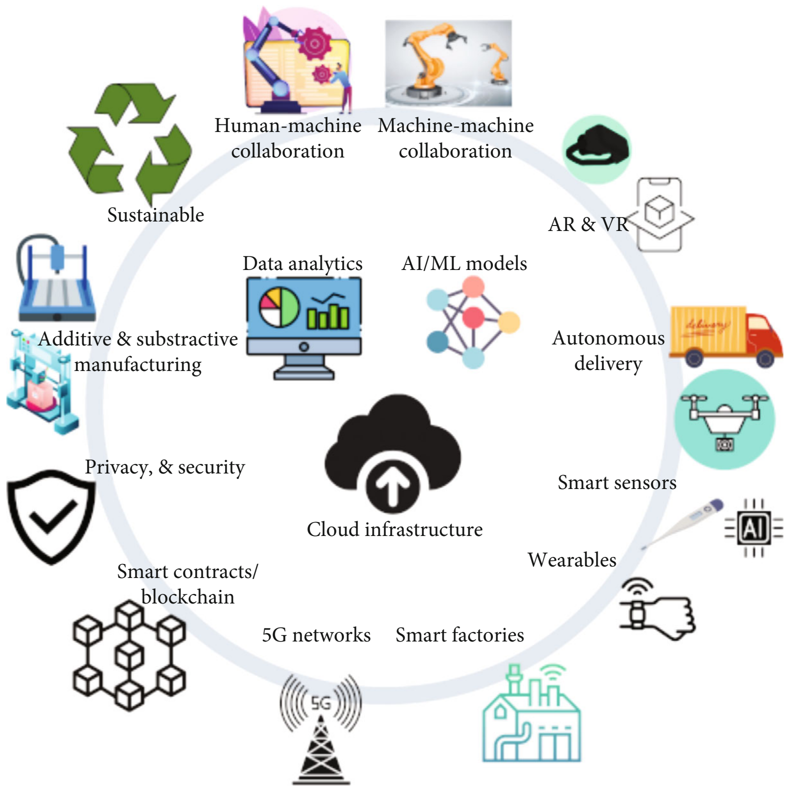


Figure 1: Overview of Industry 4.0 ecosystem. All devices are connected and integrated with IIoT. The data is transmitted to the cloud for analysis. Threats can be detected using ML-based detection methods.

layers. Depending on the model, the IIoT architecture may be based on three or five layers. The three-layer model includes a perception layer, a network layer, and an application layer. Additionally, most of the recent IIoT systems are based on the Service-Oriented Architecture (SOA) architecture [19].

The perception layer is a layer that is based on the hardware objects of the IIoT system, and hence, it is also called the object layer. This consists of physical sensors and measures the parameters controlled by the system. This data perceived by the physical sensor is then converted into the electronic signal by the electronic circuits and then transmitted to the interface through the network layer.

The network layer transmits the data from the device to the interface controlled by the user. It also transfers the data or the set values input by the user to the device. This layer is also called the transport layer. It is this layer that is most prone to hacking and intrusion and must be provided with protective systems to protect the device from any external control. The layer must be provided with the methods to prevent the intrusion. The network layer is based on connection protocols which are done using any wireless communication methods like NFC, Bluetooth, and Wi-Fi technology.

The application layer varies from service to service. This is the main interface that is available to the end-user through which he enters the commands and asks the device to perform accordingly. This layer must also be provided with security measures to protect the device from intervention from an outside source.

2.3. Attacks. The cyberattacks into the IIoT-based devices are of many types based on which type of layer they are attacking and the severity of the attack. These attacks make the IIoT solutions vulnerable and the main hurdle in the widespread use of the systems. The types of cyberattacks into the IIoT devices include [20] denial of service (DoS) attack, flooding, blackhole attack, Sybil attack, clone attack, and sinkhole attack among various others and combination thereof.

DoS attack is the one in which the application layer, which is the user interface, is no longer in control of the legitimate user. This attack is through the communication protocol followed by the system, which is either Bluetooth [21], Wi-Fi, or NFC technology. This attack also affects the hardware devices as well. DoS attack performed at a large scale through botnets is called distributed denial of service (DDoS) attack [22, 23]. Flooding is the one in which the cyber hackers take control of the interface over the network and show its presence by displaying the “Hello” message over the interface. A blackhole attack is one in which the route of the connection is changed, and the user is unable to access the device from the connection source [24]. Sybil attack is the one in which the multiple connection routes are created and the original information which is to be transmitted is corrupted. In the clone attack, similar connection routes are generated by the attackers which causes the data which is transmitted to be lost and get corrupted. A sinkhole attack is the one in which the original connection node acts as a sink and attracts and corrupts the surrounding connection nodes. Yavuz et al. proposed highly-scalable deep learning methods for the detection of IIoT routing attacks with high accuracy and precision results on decreased rank, hello-flood, and version number modification attacks [25].

2.4. Communication Protocols. The communication protocols followed by the IIoT devices are often insecure and unreliable. Therefore, there is a need for security layers to prevent any intrusion in the communication system. The communication protocols followed by IIoT devices include [26] IPv6, 6LoWPAN, User Datagram Protocol (UDP), Quick UDP Connection (QUIC), Datagram Transport Layer (DTLS), CNN (Content-Centric Networking), and Constrained Application Protocol (CoAP).

IPv6 communication technology has become standardized over the past few years because of its universality and ease of use. IPv6 is better than the other communication protocols as it provides a higher speed for the data packet transmission. In this protocol, the data which is to be transmitted does not need to be passed to network-address translators (NAT) as compared to the other protocols such as IPv4.

6LoWPAN communication protocol further reduces the data packet size by compressing it, and hence, this makes the data transmission faster and more reliable. This communication protocol has a working range of 2.4 GHz frequency range with the transfer rate of as fast as 250 kilobytes per second.

UDP provides a simpler communication protocol between the device and the interface due to its lightweight, which reduces the lag between the data communication. It is mostly used for live communication such as live processing of process-plant parameters as it has less overhead. QUIC is the more advanced version of UDP. As the name suggests, it is faster than a conventional UDP connection and hence more reliable. It also allows multiple complex connections between the two nodes, and hence, data can be transmitted faster.

DTLS communication protocol is used where private connections are required as this allows the data transmission privately without any external influence.

CNN communication protocol allows the data-centric transmission between the device and the user interface without any external noise. This is the most effective and reliable protocol for the accurate transmission of data.

CoAP communication protocol is used for application-specific purposes. It uses the HTTP server for the communication between the device and the interface. HTTP server with the URL provides the Web access to the communication, and hence, the interface is easy to understand and has a vast atmosphere.

2.5. Intrusion Detection. The intrusion detection system is based on the same concept as the working of the IIoT devices. The intrusion detection can be placed as a separate layer on the top of the IIoT architecture or it can be embedded into the application and connection layer of the IIoT architecture. The main concept behind the working of intrusion detection is that it assigns unique identifiers to the data nodes emerging from a specific network. All the data nodes which have different identifiers, not recognized by the system, are rejected, and the user is informed about the breaches.

There are many intrusion detection methods to spot any malicious activity on IIoT devices. Some of the most common intrusion detection methods which are used commonly by businesses and industries [27] include detection based on signatures, anomaly-based detection, detection based on specifications, detection using machine learning, deep learning, and a combination of approaches.

A signature-based detection system is used for the communication and data packets transmitted through the connection layer. These detection systems detect any abnormality in the data packets transmitted through the network and give an alert based on the data. This is a very operative and fast method to detect any intrusion within the system. This method works based on the attack signatures identified to it based on the past data. It investigates the past for the signatures of the data which caused the intrusion and looks for the same intrusion signatures in the future and notifies if it happens again. One of the drawbacks of this method is that it cannot detect any new intrusion occurring in the future as the system will not identify the signature as the intrusion [24]. The method is based on machine learning and statistical tools, and therefore, to apply the system on any device, the system must be fed with the previously known intrusion signatures, to begin with, and it continuously learns the new intrusions based on the inputs provided to it by the user. The algorithm can also be modified to detect any new data packet as an intrusion. This is more stringent and will also detect the new data input by the user as an intrusion. Some researchers have also modified the system, and instead of detecting the data packets, they have made the system recognize the energy consumed by the specific signature. This method is more reliable, stringent, and fast as compared to the previous one [26].

The drawback of the signature-based detection approach is that it cannot detect new intrusions into the system. Anomaly-based detection mitigates this problem as this is based on the anomaly or irregular data packets instead of the signatures. Any new data packet trying to enter the system that does not match with the regular attributes will be detected as an anomaly. This will make the system more secure, but the users must keep their data consistent so that the user data is not itself corrupted [28]. This method is also effective in detecting sinkhole attacks. If the data packets taken in by the system are large as compared to the normal usage, the system will detect this as an anomaly and will inform the user about the intrusion [29].

The specification-based intrusion detection method is based on the instructions provided to the system, and the system data packets will follow the instructions. This set of instructions will prevent any data packet not following the instructions as an anomaly. This method is effective for the DoS attacks in which the user is prevented from controlling the application. This approach is very much dependent on the specification set for the data packets.

Machine learning finds various applications in the field of intrusion detection. The applications are programmed to learn through past intrusions. This is possible through the machine learning application. If any similar intrusion is done again on the system, the program stops it immediately and informs the user about it [30]. Deep learning is different from machine learning in the sense that machine learning consists of a single algorithm that enables the machine to learn from past instances. In contrary to that, deep learning is a part of machine learning and consists of many layers of algorithms which are called ANN (artificial neural network) [31]. In intrusion detection, especially for IIoT devices and networks, unsupervised deep learning is not heavily dependent on past intrusions. If provided unstructured data, it can function well to detect any future intrusions into the layers of the IIoT. Deep learning is like the functioning of the human brain. Deep learning-based intrusion detection algorithms identify the differences between the required data packets and the intrusions by themselves and based on their learnings, preventing the intrusions from happening in the future [32, 33].

Shafiq et al. [34] used four different machine learning classifiers (random forest, support vector machine, decision tree, and Naive Bayes) for IoT botnet attack classification using the dataset developed by Koroniotis [35]. Their reported accuracy was higher than 99% for classifying some of the selected attacks. All models performed well with over 98% accuracy across all attack classes. Within healthcare IoT, [36] built a testbed that monitors the patients' biometrics and collects network flow metrics for providing them treatment and medical diagnostics and used different machine learning methods for training and testing against the dataset which included man-in-the-middle cyberattacks.

To make intrusion detection more advanced, a combination of the abovementioned intrusion detection methods is used. Each method has its specific features, and hence, to protect the system from multiple cyberattacks, a combination of the methods can be used. This approach provides more stringent protection as compared to the individual approaches.

3. Methodology

A structured and labeled dataset of IIoT botnet attack data is used for training the machine learning models. The machine learning models are developed in Python 3.8 using Keras, Tensorflow, and Scikit-Learn libraries. The data is scaled before training used the machine learning models. The overview of the methodology is shown in Figure 2.

3.1. Dataset. The IIoT botnet attack dataset was developed by [35]. It consists of several types of attacks including DoS, DDoS, theft, and reconnaissance. The DoS and DDoS attacks contain 3 different protocols such as the HTTP, TCP, and UDP. Theft includes keylogging and stealing data. Reconnaissance includes fingerprinting of the operating system and scan of open ports. Overall, the attacks can be categorized into 10 different classes as shown in Table 1. There are about 37 features. The complete labeled dataset is about 16 gigabytes. 5% of this dataset is considered for this study. Nevertheless, this smaller sampled dataset contains approximately 3.6 million records. The dataset is randomly sorted into two sets— training and testing. 80% of the data is allocated to the training set while the remaining 20% is allocated to the validation set. Figure 3 shows the labels of the attacks and their frequencies in the original dataset. The imbalance is present in the original dataset where the classes of attacks are imbalanced. This affects the deep learning models as they need sufficient training data in recognizing the attacks appropriately. Although some techniques such as oversampling from classes with fewer samples can be used in some instances, we kept the statistics of the sampled dataset to accurately reflect the original dataset, as this also represents the real-life attack scenarios, with some types of attacks being more frequent than others. Table 1 shows the number of records for each of the classes of attacks. Figure 4 shows that the sampled dataset is representative of the original dataset in terms of the attack classes being proportional compared against the total number of records in each dataset.

3.2. Imbalance Correction. The dataset created by [35] suffers from heavy class imbalance. This affected the performance of the deep learning models in correctly identifying the threats in multiclass classification. To improve the model performance, a balanced dataset is created using the techniques suggested by [15]. The Python imbalanced-learn module has been used for undersampling the majority class to create a balanced dataset with equal number of cases from each class [37].

3.3. ML Models. Three different kinds of deep learning models are used for this study: the artificial neural network (ANN), gated recurrent unit (GRU), and long short-term memory (LSTM).

3.3.1. Artificial Neural Network. ANN is commonly used for classification problems in supervised learning. The ANN consists of an input layer, an output layer, and several hidden layers which consist of neurons. The hyperparameters are tuned manually for optimal performance as shown in Table 2. The loss function is *categorical_crossentropy* which is used for multiclass classification problems along with the *accuracy* metric. The rectified linear unit (ReLu) is used for activations in all the layers except the output layer which uses *softmax* to give the multiclass outputs. The hidden layers and the number of hidden layers can be tuned manually to have better performance. The optimizer is *adam*, which is a gradient-based optimizer that is popular in machine learning problems for its fast convergence. A batch size of 64 and an epoch of 200 were used for training the models. The initial learning rate was set at 0.001 and adapted to lower rates as the training progressed over several epochs. The training set is scaled using *RobustScaler* as it improves the performance of the ANN.

3.3.2. Long Short-Term Memory. LSTM network is a state-of-the-art recurrent neural network that can learn from both long- and short-term dependencies and is more robust to the vanishing gradient problem in deep-layered networks. This deep learning algorithm is quite robust for modeling time-dependent data [38]. Since IIoT devices transfer data through packets over some time, the attack features can be considered time-dependent. For example, during a DDoS attack, the IIoT traffic might experience higher latency. These would result in a longer duration for data transfer,

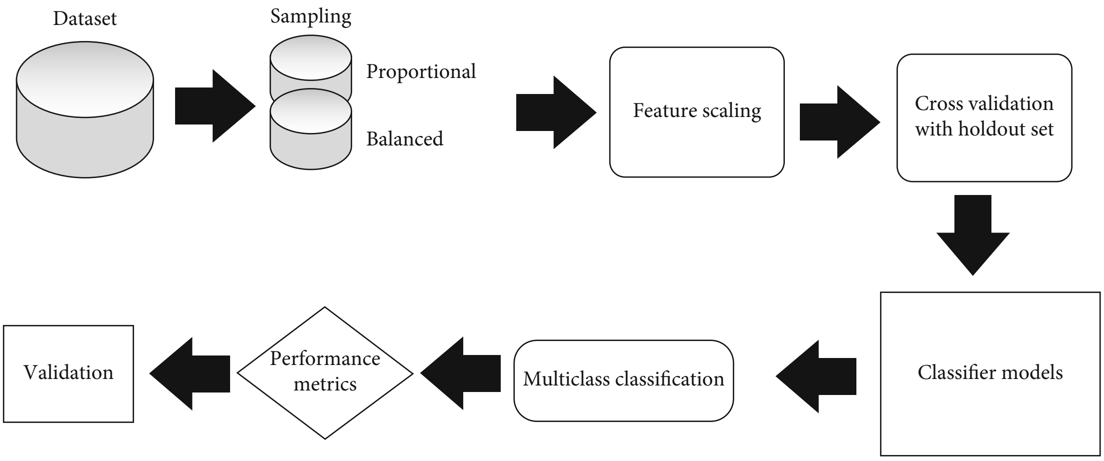


Figure 2: Overview of the methodology.

Table 1: Records of different attack types contained within the sampled dataset.

ID	Category	Frequency	Percent
0	Normal	477	0.01
1	dos_http	1485	0.04
2	dos_tcp	615800	16.79
3	dos_udp	1032975	28.16
4	ddos_http	989	0.03
5	ddos_tcp	977380	26.64
6	ddos_udp	948255	25.85
7	rcn_fngrprnt	17914	0.49
8	rcn_scan	73168	1.99
9	theft_data	6	1.6×10^{-4}
10	theft_keylog	73	1.99×10^{-3}
Total		3668522	100%

and the attack might be picked up by a well-trained LSTM model. The LSTM block, which is analogous to the neuron of the ANN, has three gates. These gates—forget (f), input (i), and output (o) gates—are represented by sigmoid functions. In the LSTM block, C_{t-1} is the cell state or memory from the previous block. X_t is the vector input, C_t is the cell state of the present block, h_{t-1} is the previous block output, and h_t is the output of the current block. Element-wise Hadamard product is performed at the ⊗ junction. Likewise, the element-wise summation is done at the + junction. The LSTM gates and memory equations are given by (1) to (6). The features are scaled using a min-max scaler before training. Table 3 shows the LSTM model hyperparameters used for training our models.

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{1}$$

where f_t is the activation vector of the forget gate, σ_g is the sigmoid function, W and U are the weight matrices, and b is the bias vector.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \tag{2}$$

where i_t is the activation vector of the input or update gate.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \tag{3}$$

where o_t is the activation vector of the output gate.

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c), \tag{4}$$

where the activation vector of the cell input is given by c_t, and σ_h is the hyperbolic tangent (tanh) function.

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t, \tag{5}$$

where c_t is the cell state vector.

$$h_t = o_t \otimes \sigma_h(c_t), \tag{6}$$

where h_t is the output vector of the LSTM block or the hidden state vector.

3.3.3. Gated Recurrent Unit. GRU is a recurrent neural network that is very similar to the LSTM yet simpler in its design. Instead of the 3 gates that LSTM utilizes, the GRU uses 2 gates: update and reset gates. It also does not have a separate cell state or memory. Instead, it uses the hidden state for transferring information. The update gate serves the function of both forget and input gates in that it decides what new information to consider and what information to forget. The reset gate is used for controlling the amount of past information to forget. Table 4 shows the hyperparameters used for training our GRU model.

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z), \tag{7}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r), \tag{8}$$

$$\widehat{h}_t = \phi_h\left(W_h x_t + U_h\left(r_t \otimes h_{t_1}\right) + b_h\right), \tag{9}$$

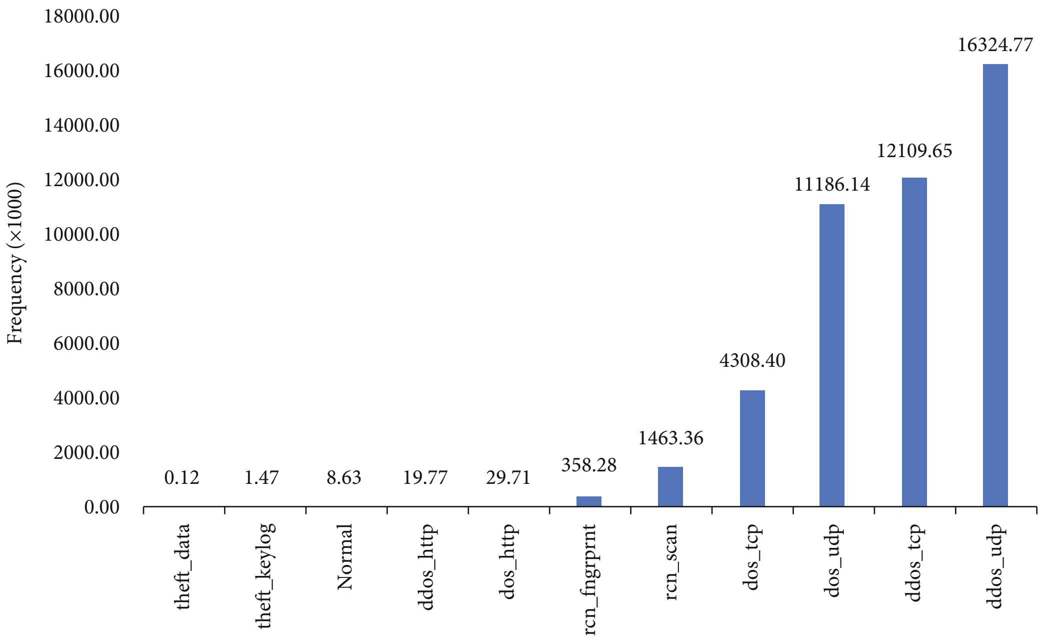


Figure 3: Bar chart of the original dataset.

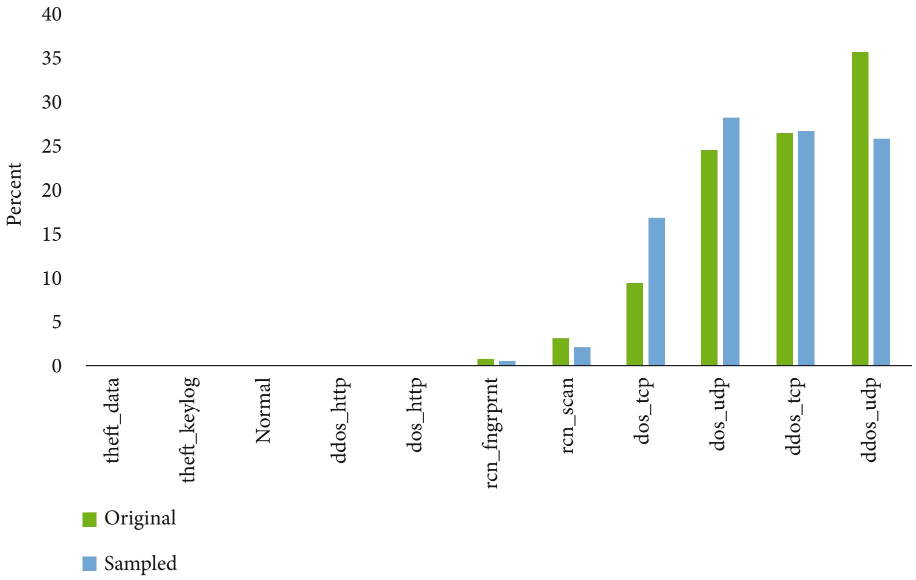


Figure 4: Comparison between the original dataset and the sampled dataset shows that the sampled dataset represents the original dataset in terms of proportionality.

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \widehat{h}_t, \quad (10)$$

where σ_g is the sigmoid function, ϕ_h is the hyperbolic tangent, x_t is the input vector, h_t is the output vector, $\widehat{h}_t$ is the candidate activation vector, z_t is the update gate vector, r_t is the reset gate vector, W and U are the parameter matrices, and b is the bias vector.

4. Results and Discussion

With the rapid growth of IIoT devices, it has become imperative to develop secure systems that can mitigate attacks against IIoT networks. Botnet attacks are regularly targeted towards these networks and devices for stealing data, denying legitimate users from accessing services, and invading user privacy. Traditional signature-based malware detection is not sufficient to protect against these threats. There were some previous studies such as [34, 35] which applied classical learning methods for botnet detection, such as decision trees, naive Bayes, and SVM. However, these models are not suitable for training on large amounts of data.

The deep learning classification models can be evaluated using different performance indicators (PI). The indicators are accuracy (11), F-1 score (12), and area under the receiver

Table 2: Settings of the ANN.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
Hidden layers	2
Neurons per hidden layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

Table 3: Settings of the LSTM.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
LSTM layers	2
LSTM blocks per layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

Table 4: Settings of the GRU.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
GRU layers	2
GRU blocks per layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

operating characteristic curve (AUC-ROC). These PI are based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The most commonly reported PI is accuracy. However, as the records of attack classes are imbalanced, F1-score may provide better insight. F1-score close to 1 indicates that the model performs well in both precision (13) and recall (14). AUC-ROC score indicates how good the model is in differentiating between the true positives and the true negatives. AUC-ROC score of 0.5 means that the model does not discriminate between classes. AUC-ROC score closer to 1 indicates that the model is good at making a distinction between classes, while scores less than 0.5 suggest that the model performs worse than a random classification [39]. Cohen's Kappa (k) is another classification metric that can be used to compare the test set classifications against the predicted set classifications. The k indicates the level of agreement between these two sets by a number between -1 and 1 with 1 being in perfect agreement. k of 0 implies that there is no agreement between the two sets despite having some probability, and a k value of -1 implies that the agreement is arbitrarily worse than random. The k is given by (15).

Table 5: Average performance scores of ML classification models.

Metrics	ANN	GRU	LSTM
Accuracy %	99	98	98
AUC-ROC score	0.85	0.83	0.84
Precision	0.98	0.99	0.98
Recall	0.99	0.98	0.98
F1-score	0.98	0.98	0.98
Cohen's Kappa (k)	0.98	0.98	0.98

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

where precision and recall are given by (13) and (14), respectively.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{14}$$

$$k = \frac{\rho_0 - \rho_e}{1 - \rho_e}, \tag{15}$$

where ρ_0 is the observed agreement similar to (11), and ρ_e is the probability of agreement by chance calculated from the classes present in the dataset.

Table 5 summarizes the average performance of the deep learning models concerning the various classification metrics. On average, the three deep learning models performed well with ANN reporting 99% accuracy, and both LSTM and GRU reporting 98%. In terms of AUC-ROC score, ANN scored 0.85, followed by LSTM with 0.84, and GRU with 0.83. GRU reported the highest precision of 0.99, and ANN and LSTM both reported 0.98. ANN reported the highest recall with 0.99, and LSTM and GRU both reported 0.98. All three models reported the same F1-score of 0.98. The three models also reported the same Cohen's Kappa of 0.98.

Tables 6, 7, and 8 show the performance of the deep learning models with respect to each attack type. When using the proportionally sampled dataset, all the models could not identify *theft_data* correctly in the test set. However, upon inspection, the predicted classification *theft_data* was misclassified as another type of attack and not as normal

Table 6: Precision of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	1	1	1	1	1	1
dos_http	0	1	1	1	0.93	1
dos_tcp	0.99	1	0.94	1	0.94	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.98	0	0.98	0.46	0.97
ddos_tcp	1	1	0.98	1	0.99	1
ddos_udp	1	1	0.99	1	1	1
rcn_fngrprnt	0	0.97	1	1	1	1
rcn_scan	0.64	0.98	1	1	1	1
theft_data	0	0.87	0	0.89	0	0.88
theft_keylog	1	1	0	0.97	0	0.98

^{B}Balanced dataset.

Table 7: Recall of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	0.14	0.84	0.7	0.97	0.72	0.98
dos_http	0	0.96	0.05	0.97	0.18	0.99
dos_tcp	0.98	1	0.97	1	0.98	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.89	0	0.91	0.09	0.96
ddos_tcp	0.98	1	0.96	1	0.96	1
ddos_udp	1	1	1	1	1	1
rcn_fngrprnt	0	0.97	0.99	1	1	1
rcn_scan	1	1	1	1	1	1
theft_data	0	0.88	0	0.85	0	0.84
theft_keylog	0.19	0.92	0	0.93	0	0.91

^{B}Balanced dataset.

Table 8: F1-score of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	0.25	0.91	0.82	0.98	0.84	0.99
dos_http	0	0.98	0.10	0.98	0.30	0.99
dos_tcp	0.98	1	0.95	1	0.96	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.93	0	0.94	0.15	0.96
ddos_tcp	0.99	1	0.97	1	0.97	1
ddos_udp	1	1	0.99	1	1	1
rcn_fngrprnt	0	0.97	0.99	1	1	1
rcn_scan	0.78	0.99	1	1	1	1
theft_data	0	0.87	0	0.87	0	0.86
theft_keylog	0.32	0.96	0	0.95	0	0.94

^{B}Balanced dataset.

traffic. This poor performance with regard to *theft_data* can be attributed to the low number of records for this attack class in the sampled dataset as shown in Table 1. ANN is the only model to identify *theft_key* with a precision of 1, recall of 0.19, and F1-score of 0.32. All three models identified the *rcn_scan* with ANN reported F1-score of 0.78 and both LSTM and GRU reporting F1-Score of 1. ANN could not correctly classify *rcn_fngrprnt*, while both LSTM and

Table 9: Performance of ML and DL models compared.

Works	Models and performance
Shafiq et al. [34]	Models have accuracy ≥99%. Models include decision tree, random forest, SVM, and naive Bayes.
Koroniotis et al. [35]	SVM, LSTM, and RNN models have accuracy ≥98%.
Our work	Deep LSTM, GRU, and ANN models perform with accuracy of 98% – 99%.

GRU were able to classify it with F1-score of 0.99 and 1, respectively. All models classified *ddos_udp* correctly with F1-score of 1. For *ddos_tcp*, ANN has had F1-score of 0.99, and LSTM and GRU both have received 0.97. GRU somewhat classified the *ddos_http* with an F1-score of 0.16, where both ANN and LSTM failed to classify it correctly. *dos_udp* was correctly classified by all 3 models. *dos_tcp* was classified by ANN with F1-score of 0.98, LSTM with F1-score of 0.95, and GRU with F1-score of 0.96. ANN could not correctly classify *dos_http*, whereas LSTM and GRU classified it with F1-score of 0.1 and 0.3, respectively. Lastly, all models classified the normal traffic well in precision; however, the recall performance dropped with ANN scoring 0.14, and LSTM and GRU scoring 0.7 and 0.72, respectively. When using the balanced dataset with equal samples from each of the classes, the results showed significant improvement in terms of precision, recall, and F1-score.

Compared to previous works that used ML and DL models, our models have performed well with accuracy ≥ 98% as shown in Table 9. The cited works differ in the use of different types of models and feature selection methods.

Deep learning models are preferred over classical (such as linear models and shallow ANN) machine learning models for big data since classical models take a significantly longer time to train on them. IIoT systems generate huge amounts of data in short periods, because of a large number of deployed IIoT devices. Considering Table 5 and Tables 6 to 8, it can be seen that deep learning models are promising in classifying IIoT attacks and can be potentially used for securing the IIoT network against previously unknown threats, thus protecting zero-day attacks.

In this work, two types of deep learning models are used for classifying the IIoT botnet attacks: the deeply connected neuron-based ANN and the recurrent neural network-based LSTM and GRU. All three models performed well in the selected performance measures across different attack types. The ANN had the highest average accuracy of 99% although it misclassified some attacks into the wrong category. LSTM and GRU are almost similar in performance; however, GRU performed slightly better in classifying some of the attacks such as *ddos_http* and *dos_http*. The poor performance of the models in precision and recall of identifying minority classes was fixed by balancing the dataset with equal size of classes. As for Industry 4.0, training deep learning models is computationally expensive. Thus, it may need to be optimized for deploying on IIoT systems and networks.

5. Conclusion

In this work, three different types of deep learning-based models—LSTM, GRU, and ANN—have been used for classifying ten different IIoT botnet attacks covering various communication protocols and devices. All the models are shown to have high performance with more than 98% classification accuracy. The implication of this study is that deep learning models can be used for IIoT malware detection especially within the context of novel threats that often elude the conventional methods. While the deep learning models may fail to identify minority classes, this can be fixed or improved by training the models on balanced dataset. Undersampling the majority classes have helped in correcting the imbalance in this case.

As the smart factories become more connected, the threats to people's data and privacy increase through sophisticated malware attacks and botnets. Deep learning models can be used for protecting these devices and networks by identifying the threats. The main advantage of these models is that they perform better as they learn from the big data produced by the billions of IIoT connected devices. In future works, areas of research that could be explored further include federated learning for IIoT networks as well as novel approaches to share threat analytics between devices and networks. Furthermore, different types of IoT datasets can be merged together to create a comprehensive IoT system dataset that can be used for training ML and DL models and provide security using federated learning and edge computing. For instance, healtcare IoT dataset [36] can be merged with IIoT datasets to extend the range and variety attacks on IIoT systems.

Data Availability

Previously reported IoT Botnet attack data (Bot-IoT dataset) were used to support this study and are available at: https://research.unsw.edu.au/projects/bot-iot-dataset. These prior studies (and datasets) are cited at relevant places within the text as reference [35].

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] E. Oztemel and S. Gursev, "Literature review of industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, no. 1, pp. 127–182, 2020.

[2] L. Zhou, K.-H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial Internet of Things: an overview of approaches to safeguarding endpoints," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, 2018.

[3] W. Ren, X. Tong, J. Du et al., "Privacy-preserving using homomorphic encryption in mobile IoT systems," *Computer Communications*, vol. 165, pp. 105–111, 2021.

[4] A. A. Abd El-Latif, B. Abd-El-Atty, S. E. Venegas-Andraca et al., "Providing end-to-end security using quantum walks in IoT networks," *IEEE Access*, vol. 8, pp. 92687–92696, 2020.

[5] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of Internet of Things," *Computer Science Review*, vol. 38, article 100312, 2020.

[6] G. De La Torre Parra, P. Rad, K.-K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," *Journal of Network and Computer Applications*, vol. 163, article 102662, 2020.

[7] I. Ali, A. I. A. Ahmed, A. Almogren et al., "Systematic literature review on IoT-based botnet attack," *IEEE Access*, vol. 8, pp. 212220–212232, 2020.

[8] A. Marzano, D. Alexander, O. Fonseca et al., "The evolution of bashlite and mirai iot botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, June 2018.

[9] R. Kour, "Cybersecurity issues and challenges in Industry 4.0," in *Applications and Challenges of Maintenance and Safety Engineering in Industry 4.0*, pp. 84–101, IGI Global, 2020.

[10] J. Prinsloo, S. Sinha, and B. von Solms, "A review of Industry 4.0 manufacturing process security risks," *Applied Sciences*, vol. 9, no. 23, p. 5105, 2019.

[11] S. Dange and M. Chatterjee, "IoT Botnet: the largest threat to the IoT Network," in *Data Communication and Networks. Advances in Intelligent Systems and Computing, vol 1049*, L. C. Jain, G. A. Tsihrintzis, V. E. Balas, and D. K. Sharma, Eds., pp. 137–157, Springer, Singapore, 2020.

[12] L. Gupta, T. Salman, A. Ghubaish, D. Unal, A. K. Al-Ali, and R. Jain, "Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach," *Applied Soft Computing*, vol. 118, p. 108439, 2022.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[14] D. Krishnan and P. Babu, "Imbalanced classification for botnet detection in Internet of Things," in *Next Generation of Internet of Things. Lecture Notes in Networks and Systems, vol 201*, R. Kumar, B. K. Mishra, and P. K. Pattnaik, Eds., pp. 595–605, Springer, Singapore, 2021.

[15] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*, vol. 8, no. 1, p. 6, 2021.

[16] M. Hammoudeh, J. Pimlott, S. Belguith et al., "Network traffic analysis for threats detection in the Internet of Things," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 40–45, 2020.

[17] S. H. Jafier, "Utilizing feature selection techniques in intrusion detection system for Internet of Things," in *ICFNDS '18: Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pp. 1–3, New York, New York, USA, June 2018.

[18] Q. Wang, X. Zhu, Y. Ni, G. Li, and H. Zhu, "Blockchain for the IoT and in dustrial IoT: a review," *Internet of Things*, vol. 10, article 100081, 2020.

[19] M. Hammoudeh, G. Epiphaniou, S. Belguith et al., "A service-oriented approach for sensing in the internet of things: intelligent transportation systems and privacy use cases," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15753–15761, 2020.

[20] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for Internet of Things based on suppressed fuzzy clustering," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018.

[21] M. Zubair, D. Unal, A. Al-Ali, and A. Shikfa, "Exploiting bluetooth vulnerabilities in e-health IoT devices," in *ICFNDS '19: Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pp. 1–7, New York, NY, USA, July 2019.

[22] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, pp. 283–294, 2019.

[23] I. Ghafir, V. Prenosil, M. Hammoudeh et al., "BotDet: a system for real time botnet command and control traffic detection," *IEEE Access*, vol. 6, pp. 38947–38958, 2018.

[24] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 570–577, 2014.

[25] F. Y. Yavuz, D. Unal, and E. Gul, "Deep learning for detection of routing attacks in the Internet of Things," *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, 2018.

[26] E. Anthi, L. Williams, and P. Burnap, "Pulse: an adaptive intrusion detection for the internet of things," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, p. 4, London, UK, 2018.

[27] S. Madhawa, P. Balakrishnan, and U.-m. Arumugam, "Data driven intrusion detection system for software defined networking enabled industrial Internet of Things," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1289–1300, 2018.

[28] F. Yulong, Z. Yan, J. Cao, O. Kone, and X. Cao, "An automata based intrusion detection method for Internet of Things," *Mobile Information Systems*, vol. 2017, Article ID 1750637, 13 pages, 2017.

[29] A. Tabassum, A. Erbad, and M. Guizani, "A survey on recent approaches in intrusion detection system in IoTs," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1190–1197, Tangier, Morocco, June 2019.

[30] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019.

[31] C. M. Liu, Y. Zhang, R. Chen, L. X. Xiao, and J. D. Zhang, "Research on intrusion detection for the Internet of Things based on clone selection principle," *Advanced Materials Research*, vol. 562-564, pp. 1982–1985, 2012.

[32] R. Chen, C. M. Liu, and C. Chen, "An artificial immune-based distributed intrusion detection model for the Internet of Things," *Advanced Materials Research*, vol. 366, pp. 165–168, 2011.

[33] H. Naeem and A. A. Bin-Salem, "A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images," *Applied Soft Computing*, vol. 113, article 107918, 2021.

[34] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT

network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[35] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[36] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: a comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[37] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[38] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach," *Neural Computing and Applications*, pp. 1–15, 2020.

[39] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315-1316, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8425506, 11 pages
https://doi.org/10.1155/2022/8425506

Research Article

A Quasistraight Line Routing Protocol for Square Grid-Based Wireless Sensor Networks

Md. Ajij,[1] Sanjoy Pratihar,[2] Ashish Kumar Luhach,[3] and Diptendu Sinha Roy[1]

[1] *National Institute of Technology Meghalaya, Shillong, India*
[2] *Indian Institute of Information Technology Kalyani, Kalyani, India*
[3] *PNG University of Technology, Morobe, Papua New Guinea*

Correspondence should be addressed to Ashish Kumar Luhach; ashish.kumar@pnguot.ac.pg

Received 27 January 2022; Revised 18 March 2022; Accepted 23 March 2022; Published 11 May 2022

Academic Editor: Mohammad R Khosravi

Sensor nodes in a wireless sensor network (WSN) are both energy-constrained and vulnerable to faults and disasters. Communication between the sensor nodes is generally hop-by-hop, and the nodes are distributed throughout the area to be covered. Broadcast-based routing protocols are not preferable in sensor networks since broadcasting is considered costly in terms of battery power consumption. In this paper, a digital quasistraight line segment- (DQSS-) based approach is employed for the detection of quasistraight line segments, i.e., for quasistraight path finding between WSN sensors arranged in a square grid. Comparative results show that the method is comparable with the best-known straight line finding algorithm in terms of path lengths and computation time. Moreover, the proposed method is capable of avoiding dead nodes by updating DQSS parameters dynamically during path finding. Hence, the proposed method is promising to be used in WSN square grids as a quasistraight line routing protocol.

1. Introduction

Wireless sensor networks (WSNs) are extensively used in monitoring environments, surveillance equipments, intelligent home appliances operated remotely, patient care systems, etc. In WSN, the sensor nodes establish the path for communication from the sender to the receiver. This path making process should be carried out with limited resources. The performance of WSN is generally affected by many factors. These affecting factors are bandwidth, mobility, scalability, data aggregation, power consumption, etc. Because nodes have limited power sources, the minimization of power consumption is a vital issue in WSN, and this defines the performance of WSN [1]. Maximum energy is consumed by the sensor nodes in the communication processes. Routing protocols should be robust and straightforward, ensuring less energy consumption. Because of the limited resources of WSN nodes, the routing protocols must support the extended lifespan of the nodes [2]. Therefore, many protocols have been proposed highlighting the minimization of energy consumption.

Different protocols have been developed for WSNs according to the different prerequisites of uses and a large number of WSNs types. Numerous studies have attempted to analyze and classify these routing protocols according to different parameters that have been published. WSN routing protocol can be classified based on (a) application type, (b) delivery mode, (c) network architecture, (d) initiator of communication, (e) path establishment (route discovery), (f) network topology, (g) protocol operation, (h) next hop selection, and (i) latency-aware energy-efficient routing. The main goal of WSN is to establish a path consisting of the WSN nodes which will be reliable and energy efficient [3]. Energy consumption in routing is mainly due to finding

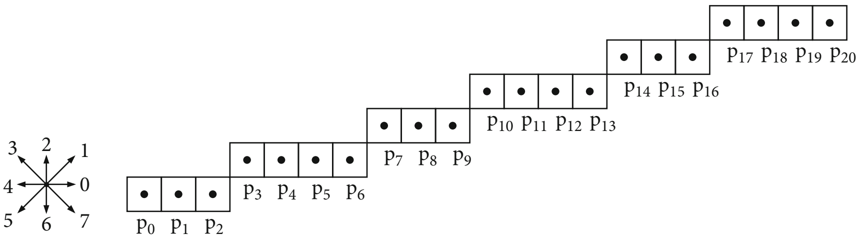


Figure 1: An example of a digital quasistraight line segment with singular code $s = 1$, nonsingular code $n = 0$, and the two run lengths (number of points in the run) l_1 and l_2 are 3 and 4, respectively.

neighbors for communications and necessary small computations. So, usually, the routing algorithms focus on how to compute the shortest path from the source node to the destination for quick transmission. A quick shortest path finding from the source node to sink may reduce the network congestion also.

Most of the traditional routing protocols cannot avoid the construction of curved (nonstraight) paths for data transmission. As a result, many multidirectional communications will lead to wastage of energy. Furthermore, the nonstraight paths normally contain more nodes than the straight paths, which leads to higher energy consumption [4]. So it is worthy of finding out a straight-line route (path).

Another serious problem is the recovery of failure nodes in the WSN environment. Node failure may be because of many reasons. The most crucial reasons for node failure in a wireless sensor network (WSN) [5] are (a) fabrication process problems, (b) environmental factors, (c) battery power depletion, and (d) enemy attacks. Node failure is a common issue in WSN, and this affects the connectivity in a network, which degrades the quality of communication [6]. In WSN, a connected network is desired for smooth communication. Hence, restoring connectivity is always given importance. Connectivity restoration is normally done by replacing dead nodes with other unused nodes [7]. This replacement mechanism should be robust, and the computational overhead must be taken care of as high-cost computations reduce the battery life [8–10]. WSNs and ad hoc networks are also vulnerable to faults, often disasters, and, owing to this very nature, are expected to fail and subsequently recover from such scenarios with minimal extraneous support [11, 12]. Energy optimal WSN operations have been studied extensively over the years, and the topologies and management strategies vary drastically with WSN use cases and applications [13, 14]. Various routing protocols have been studied, each with its own set of pros and cons. It has been well understood that traditional distributed routing involving broadcasts or those employing geographic information via GPS modules are not suited due to excessive battery drainage [4, 15–17]. This has paved the path for probabilistic routing such as gossip [18] and random routing [19, 20]. However, such probabilistic routing techniques are unsuited for WSNs with a considerable number of nodes as they cannot guarantee straight line paths, thus cannot ensure minimum distance, and are hence suboptimal in terms of energy expended while routing.

Table 1: Freeman's chain code for the line segment shown in Figure 1.

p_0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0

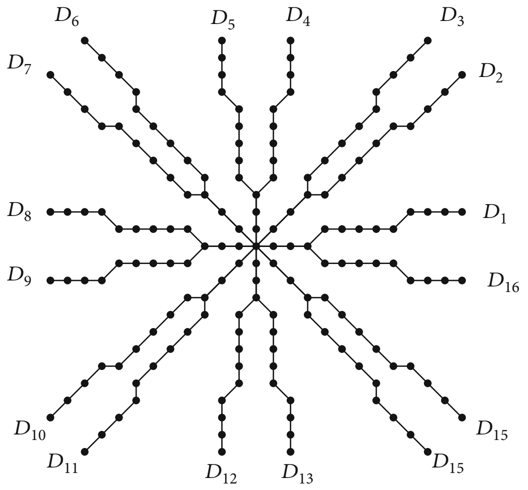


Figure 2: The sixteen directional DQSSs (representative segments with $l1 = 4$, $l2 = 5$).

1.1. Straight Line Routing. The random walk-based protocol is extensively used in WSN. Gossip [21] and rumor [20] are two well-known random walk-based routing protocols. Gossip concentrates on multicast, which suffers from power limitations and a high rate of wireless channel failure. In the rumor routing (RR) protocol, each node must maintain its list of neighbors. For propagation of a message, the node adds its list of neighbors to that message. Also, the message may keep track of all the nodes that this message has passed through. The node can decide a neighboring node to be the next node in the path. The next node must not have received the message earlier, and this way, it may prevent the route from growing in the backward direction. Rumor routing may show spiraling problems and energy is wasted in maintaining the records of visited nodes.

Chou et al. [22] proposed a routing protocol based on a random walk and straight line routing (SLR), intending to

Table 2: Direction understanding based on coordinates of the endpoints.

X-coordinate sign	Y-coordinate sign	$\|dy\| < \|dx\|$	$\|dy\| < \left\|\frac{dx}{2}\right\|$	$\|dx\| < \left\|\frac{dy}{2}\right\|$	No. of steps	Direction
Positive	Positive	Yes	Yes	No	$\|dy\| + 1$	D_1
Positive	Positive	Yes	No	No	$\|dx\| - \|dy\| + 1$	D_2
Positive	Positive	No	No	No	$\|dy\| - \|dx\| + 1$	D_3
Positive	Positive	No	No	Yes	$\|dx\| + 1$	D_4
Negative	Positive	No	No	Yes	$\|dx\| + 1$	D_5
Negative	Positive	No	No	No	$\|dy\| - \|dx\| + 1$	D_6
Negative	Positive	Yes	No	No	$\|dx\| - \|dy\| + 1$	D_7
Negative	Positive	Yes	Yes	No	$\|dy\| + 1$	D_8
Negative	Negative	Yes	Yes	No	$\|dy\| + 1$	D_9
Negative	Negative	Yes	No	No	$\|dx\| - \|dy\| + 1$	D_{10}
Negative	Negative	No	No	No	$\|dy\| - \|dx\| + 1$	D_{11}
Negative	Negative	No	No	Yes	$\|dx\| + 1$	D_{12}
Positive	Negative	No	No	Yes	$\|dx\| + 1$	D_{13}
Positive	Negative	No	No	No	$\|dy\| - \|dx\| + 1$	D_{14}
Positive	Negative	Yes	No	No	$\|dx\| - \|dy\| + 1$	D_{15}
Positive	Negative	Yes	Yes	No	$\|dy\| + 1$	D_{16}

```
Input: Geographical grid location of Source node (S) and Destination node (D)
Output: N: A quasistraight path from $S$ to $D$.
  Translate the source node($(S(i, j))$from $(i, j)$ to $(0, 0)$;
  Translate the destination node $(D(u, v))$ accordingly;
  Check x-coordinate sign, y-coordinate sign, if $|dy| < |dx|$, if $|dy| < |x/2|$, if $|dx|<|y/2|$;
  Find the applicable row in Table 2 and get directional codes: s (singular code) and n (nonsingular code); See. Fig. 2
  Find the number of steps $S_{count}$ and direction of DSS;
  Break the $S_{count}$ in into two integers;
$S_{count} = k + m$;(where $0 \le |k| - |m| \le 1$);
  Find the two run-lengths $l_1$ and $l_2$using the following criteria:
$S_{length} \longleftarrow \max(|dx|, |dy|)$;
$S_{length} \le k \times l_1 + m \times l_2$; Select $l_1$ and $l_2$when it shows minimum difference between $S_{length}$ and $(k \times l_1 + m \times l_2)$.
  N=Find-Path$(l_1, l_2, s, n)$;
```

Algorithm 1: DQSS-based routing algorithm.

extend the route as straight as possible. The central idea of the SLR protocol was creating the routing path hop-by-hop. In each hop, the next node is selected so that it lies on the extended straight path approximately. Liu et al. [4] proposed a new protocol based on straight line routing. The rumor routing (RR) protocol also solves the spiral problem. The basic idea of discovering the straight path was to find the angle using radio signals. Many routing protocols are specially designed to enhance the classic RR protocol. For example, DRR [23], IDRR [24], SDRR [25], and ZRR [26] have been proposed to solve the spiral problem. Improved sensor node localization technique is proposed by Phoemphon et al. [27], where the positions of the anchor nodes form a straight or nearly straight line. Banimelhem et al. [28] proposed a principal component analysis- (PCA-) based efficient path generation algorithm.

1.2. Faulty Node. Numerous strategies have been proposed for node deployment, which is often divided into two categories: random deployments and deterministic (grid-based) deployments [29]. Nodes are randomly eliminated and managed during ad hoc deportation in a stochastic deployment. When deploying on a grid, the nodes are arranged according to the angles of the grid points, which leads to greater accuracy in overall management. The physical

```
$p_{curr} \longleftarrow S$ ; $N \longleftarrow \varnothing$ ; Run⟵odd;index⟵1
while$p_{curr..x} \neq$ D.x A N D$p_{curr.y} \neq$ D.y do
      $N = N \cup p_{curr}$;
      if Run = odd then
            limit⟵$l_1$
      else
            limit⟵$l_2$
      while index ≤ limit do
            $p_{curr} \longleftarrow$ Point in the direction n from $p_{curr}$;
          $N = N \cup p_{curr}$;
          index⟵index+1
    if Run=odd then
          Run⟵even;
    if Run = even then
          Run⟵odd;
    $p_{curr} \longleftarrow$ Point in the direction s from $p_{curr}$;
    index⟵1;
if $p_{curr} \neq D$ then
Extend vertically or horizontally from $p_{curr}$ to D and condiser the points in N
return N;
```

Procedure 1: Find path (l_1, l_2, s, n).

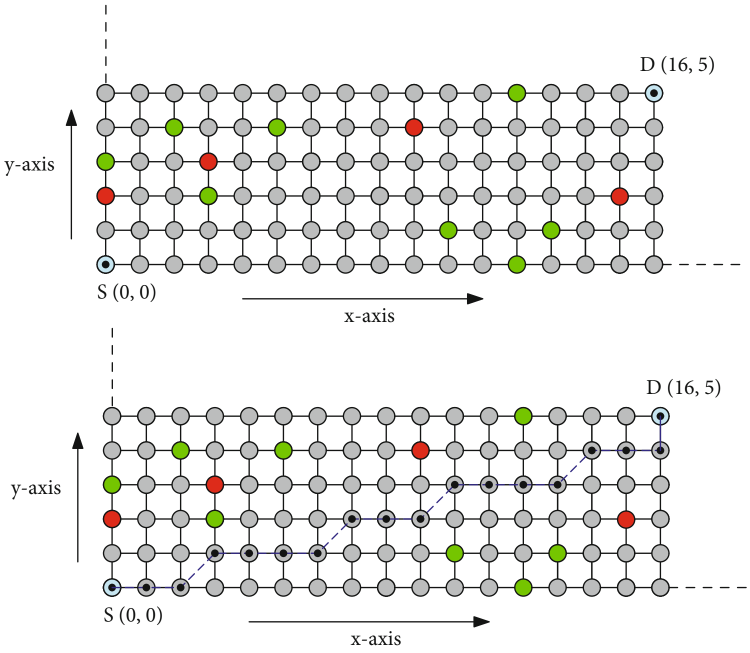


Figure 3: Demonstration of working of the algorithm; DQSS direction is $D1$, $n = 0$, $s = 1$, and $l_1, l_2 = 3, 4$; green nodes: active, gray nodes: sleeping, and red nodes: dead (the path does not go through any dead node).

positioning of sensor devices is better understood in grid-based deployment.

Many works exist to detect and analyze faulty nodes, and a few of them are listed below. Guo et al. [30] propose a sequence-based mechanism for detecting defective nodes. An algorithm for identification of fault node, based on a statistical z-score function, is proposed in [31], where all sensor nodes deliberately send information to the central node, and the root node analyzes the data to identify the fault. Asim et al. [32] provide an architecture for the management of faults in wireless sensor networks. They proposed that the entire network can be partitioned into the virtual lattice of cells and subsequently perform fault detection and recovery locally with the least energy utilization. A genetic algorithm-(GA-) based technique was proposed by Rajeswari and Neduncheliyan [33].

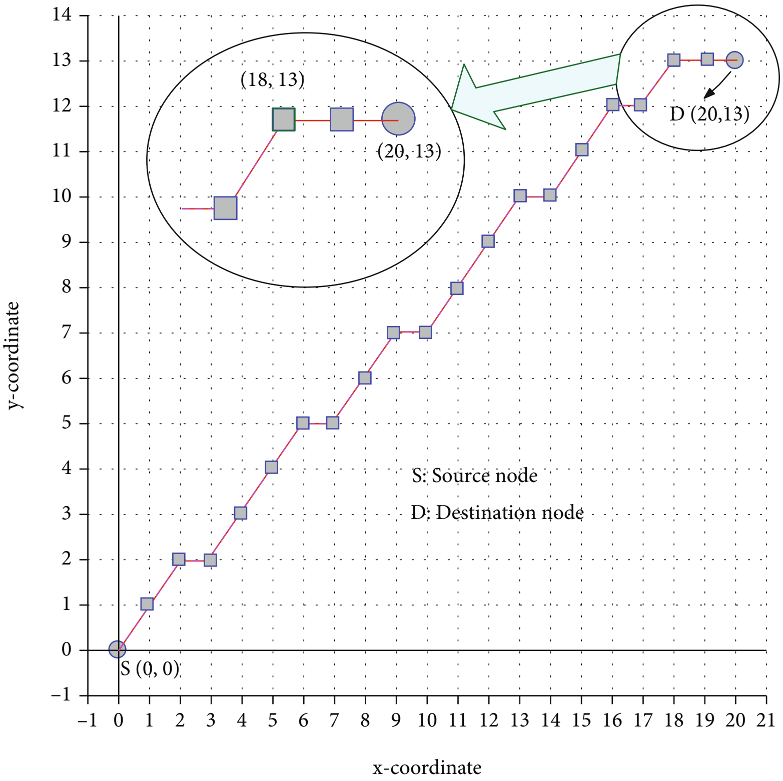


FIGURE 4: Demonstration of working of the algorithm. $n = 1$, $s = 0$, $l_1 = 3$, and $l_2 = 4$.

2. Our Contributions

In this work, we have proposed a novel path finding method based on quasistraight line fitting focusing on the grid-based deployment of sensor nodes. Moreover, we have proposed a protocol for path making and avoiding faulty nodes in a square grid of sensor nodes during path making. The protocol establishes a quasistraight line routing protocol for a node to node communication, involving a minimum possible number of sensors. We assume that there will be a few dead nodes in the sensor grid (mostly they are either live or sleeping). This proposed path making is fast and dynamic, and avoiding dead nodes does not incur extra communication costs.

3. Digital Quasistraight Line Segment (DQSS)

The structural view of rectangular grid-based wireless sensor network and points in digital space are indistinguishable. In our proposed work, our objective is to fit quasistraight digital line segments in the rectangular grid to find out the shortest path between two endpoints (source and destination) in WSN. The shortest distance between two points is indeed a straight line. In grid-based WSN, digital straight line will be suitable to explore the shortest route from sender to receiver.

Characterizations of digital straight lines have been given in many ways till date [34, 35]. Moreover, many algorithms exist to verify whether a given thin arc is digitally straight or not. Freeman introduced the chain code-based technique for representing 8 connected arcs and lines as a sequence of straight pieces [36, 37]. A chain code sequence (representing a digital curve) should possess the following properties if it represents a digital straight line segment (DSS) [34].

TABLE 3: Information stored at each sensor node.

Path ID $\{1, 2, 3, 4\}$	Run ID odd/even	Run limit l_1/l_2	Node ID in $\{1 \cdots l_1\}$ or $\{1 \cdots l_2\}$	n	s

(i) (R1) The runs have at most two directions, differing by 45°, and for one of these directions, the run length must be 1

(ii) (R2) The runs can have only two lengths, which are consecutive integers

(iii) (R3) One of the runs can occur only once at a time

(iv) (R4) For the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers

In this proposed work, we characterize a straight line segment as the chain code sequence: $n^p s n^q s n^p \cdots$, where n is nonsingular code (the code occurs consecutively multiple times) and s is singular code (occurs singly in between nonsingular codes' runs). Code values, n and s, are consecutive integer differing by 45°. In our consideration, the nonsingular

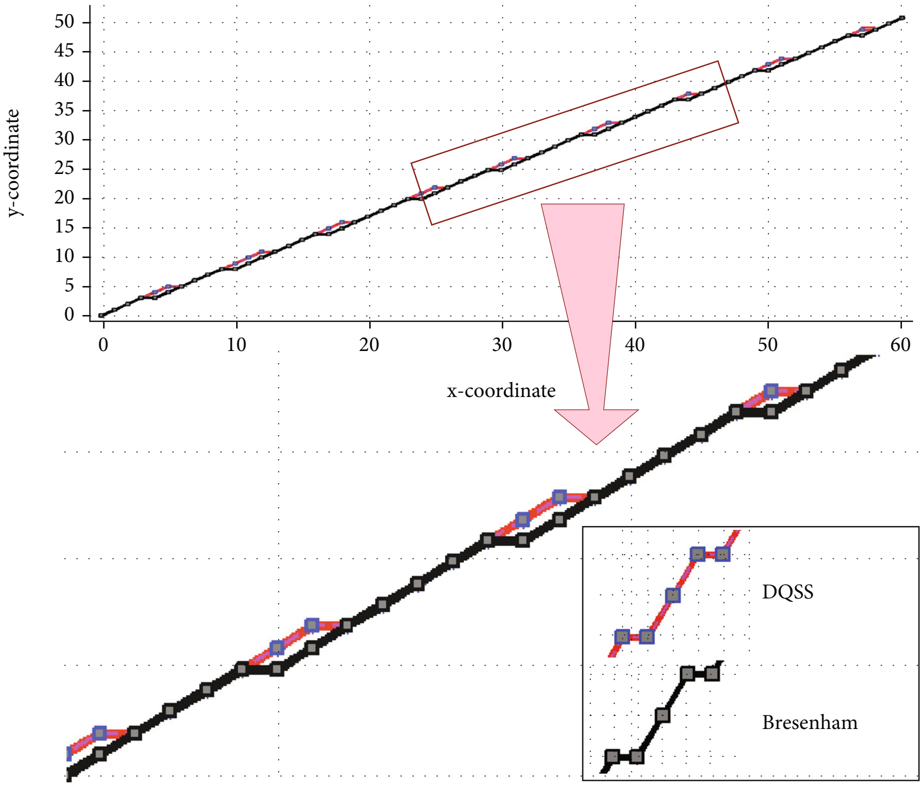


Figure 5: Differences in line segments: DQSS and Bresenham's line.

run lengths, p and q, are consecutive integers. Property $R1$ and $R2$ hold in our cases. On property $R3$, we are specific, because in our method, none of the runs occurs in runs. Instead, both the run lengths repeat alternately. So, it is evident that we may not always reach the destination point D. Whenever we reach the row or the column of the destination point in the grid, we use a horizontal or vertical stretch from that point to the destination point (D). Hence, we refer to the digital straight line segments obtained by us as digital quasistraight line segments (DQSS).

An example of a DQSS is shown in Figure 1, and the corresponding chain code is shown in Table 1. In Freeman's chain code-based properties, the mentioned run length refers to the length of a continuous sequence of the nonsingular code (in the chain code sequence). In our discussion, we have used it as the number of points in the continuous sequence in single direction. In the example shown in Figure 1, as per Freeman's definition, the two run lengths are $l_1 = 2$ and $l_2 = 3$, singular code $s = 1$, and nonsingular code $n = 0$. For the same example, we are considering the two run lengths $l_1 = 3$ and $l_2 = 4$.

3.1. Sixteen Directional DQSSs and Selection of DQSS.

A digital quasistraight line segment will fall into one of the sixteen directional clusters as shown in Figure 2. Given the two endpoints $S(x_1, y_1)$ and $D(x_2, y_2)$ of a segment, the direction can be determined in Table 2. The singular and nonsingular codes concerning the various directions are as follows: $D1$: $s = 1$, $n = 0$; $D2$: $s = 0$, $n = 1$; $D3$: $s = 2$, $n = 1$; $D4$: $s = 1$, $n = 2$; $D5$: $s = 3$, $n = 2$; $D6$: $s = 2$, $n = 3$; $D7$: $s = 4$, $n = 3$; $D8$: $s = 3$, $n = 4$; $D9$: $s = 5$, $n = 4$; $D10$: $s = 4$, $n = 5$; $D11$: $s = 6$, $n = 5$; $D12$: $s = 5$, $n = 6$; $D13$: $s = 7$, $n = 6$; $D14$: $s = 6$, $n = 7$; $D15$: $s = 0$, $n = 7$; and $D16$: $s = 7$, $n = 0$. Next, we find the number of steps, S_{count}, in the straight line segment, using Table 2 and do the followings to estimate the run lengths l_1 and l_2.

(i) Break the number of steps, S_{count}, into two integers such that, $S_{\text{count}} = k + m$, where $0 \leq |k| - |m| \leq 1$

(ii) If the DQSS has run lengths, l_1 and l_2 then check the following criteria: $S_{\text{length}} \leq k \times l_1 + m \times l_2$ and $S_{\text{length}} \longleftarrow \max(|\text{dx}|, |\text{dy}|)$

(iii) Select the DQSS which has minimum difference between S_{length} and $(k \times l_1 + m \times l_2)$

Our proposed DQSS-based quasistraight line finding method is shown in Algorithm 1. The algorithm selects the grid points or nodes to show the DQSS connectivity from the source node ($S_{i,j}$) to the destination node ($D_{u,v}$). The proposed algorithm selects and activates the nodes lying on the selected DQSS by maintaining the proper direction of the DQSS (following the properties as stated earlier), i.e., using the values l_1 and l_2 alternately starting from the source S and using the singular and nonsingular codes s and n as applicable. To start the path, we start with

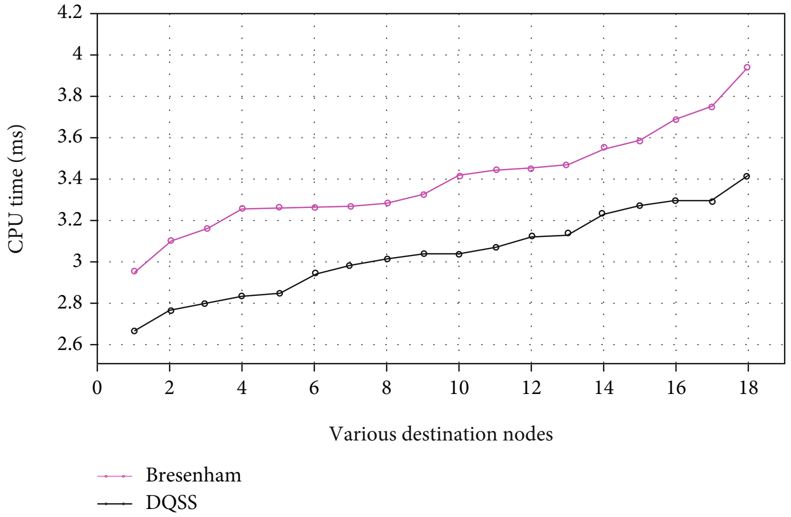


Figure 6: CPU time: DQSS vs. Bresenham's line (on 18 different line segments as shown in Table 4).

the smaller run length at source S. The selection of l_1, l_2, s, and n is shown in Algorithm 1 and the path making is shown in procedure FIND PATH (Procedure 1) of Algorithm 1.

4. Demonstration of the Proposed Algorithm

An example has been shown in Figure 3 to demonstrate the working of the proposed algorithm. Here in this example, the DQSS is to be fit in between $S(0, 0)$ and $D(16, 5)$. We find that the direction D_1 is applicable for this example. The number of stairs or steps, S_{count}, is $|\text{dy}| + 1$, i.e., 6. We find the possible values of k and m as 3 and 3. As, S_{length} is 16 here, we find that l_1 and l_2 can be set as 3 and 4, respectively, following the criteria: $S_{\text{length}} \leq k \times \text{l}_1 + m \times l_2$. Hence, we start path finding from S using $n = 0$, $s = 1$, $l_1 = 3$, and $l_2 = 4$. As shown in the procedure of Algorithm 1, p_{curr} is the current point during path making. We extend from the current point using nonsingular code's run lengths as applicable. Stairs are created using the jumps because of the application of the singular code, and we gradually proceed towards the destination point D. If the current point p_{curr} reaches either the row (when $p_{\text{curr}}.y = D.y$) or the column (when $p_{\text{curr}}.x = D.x$) of the destination point in this process, we stretch horizontally or vertically towards D from that current point p_{curr}. An example has been shown in Figure 4. In this example, when p_{curr} reaches $(18, 13)$, the y values of p_{curr} and D become equal. Hence, we stretch from $(18, 13)$ to $(20, 13)$.

5. The Protocol Using DQSS

Our proposed method works on a regular rectangular grid [38], where sensor nodes are positioned at grid intersection points. Our objective is to find the shortest quasistraight line path from the sender to the receiver. Sensor nodes are classified as given below:

(i) Active nodes: the nodes which are active in data transmissions

(ii) Sleeping nodes: initially, the nodes are sleeping and become activated based on requests

(iii) Dead nodes: dead nodes do not work in any condition as they are not in working condition; the dead nodes may be replaced with sleeping nodes

Wireless sensor nodes may be fixed nodes or mobile nodes. But in our case, we assume that the mobility of nodes is very less, and during movement, the nodes communicate with a core positioned at the grid points. So, virtually, the grid points are always the sensor nodes' locations. If a node is not active but lying on the detected straight line, then either it is a sleeping node or a dead node. If it is a sleeping node, the state of the node is changed from sleeping to active. If it is a dead node, then it does not respond to path making requests, and it is avoided reaching the destination. It must be noted that a dead node can be avoided by updating the run length limits, i.e., by preponing or postponing the application of the singular code. It is true that because of this preponing or postponing of the singular code, some runs may have run lengths other than l_1 or l_2. But, the length minimization constraint is maintained.

5.1. Sending and Receiving at Sensor Nodes. The starting point sensor initiates the path finding by sending a request to the prospective next sensor as per the codes and run lengths. We assume that the sensor nodes are equipped with local processors and storage registers to store their tagged information. If the next sensor responds to the previous, it is marked into the path, and the process continues until the destination is reached. The information which are tagged with each sensor are primarily path ID, run ID, run limit, node ID, n, and s as shown in Table 3. Here, path ID is the ID of the connecting straight line path. We assume that a sensor node can be part of four paths at most. Every path has several runs of codes. These runs are differentiated as

Table 4: Various line segments and the corresponding CPU time (in milliseconds).

	L1: dx = 10, dy = 6	L2: dx = 20, dy = 12	L3: dx = 30, dy = 18
DQSS	2.661	2.766	2.801
Bresenham	2.947	3.102	3.158
	L4: dx = 40, dy = 24	L5: dx = 50, dy = 30	L6: dx = 60, dy = 36
DQSS	2.833	2.843	2.941
Bresenham	3.250	3.260	3.259
	L7: dx = 70, dy = 42	L8: dx = 80, dy = 48	L9: dx = 90, dy = 54
DQSS	2.978	3.014	3.038
Bresenham	3.264	3.280	3.321
	L10: dx = 100, dy = 60	L11: dx = 120, dy = 72	L12: dx = 140, dy = 84
DQSS	3.039	3.063	3.119
Bresenham	3.406	3.440	3.446
	L13: dx = 160, dy = 96	L14: dx = 180, dy = 108	L15: dx = 200, dy = 120
DQSS	3.13	3.227	3.264
Bresenham	3.466	3.546	3.581
	L16: dx = 250, dy = 150	L17: dx = 300, dy = 180	L18: dx = 500, dy = 300
DQSS	3.288	3.288	3.409
Bresenham	3.682	3.748	3.939

odd and even. The initial run ID is odd, followed by an even ID run, followed by an odd ID run, and so on. If the run ID is odd, the sensor node is part of a run with length equal to l_1. If the run ID is even, the sensor node is part of a run with length equal to l_2. Node ID denotes the index of the node within the run. For example, if node ID is i and run ID is odd, then the current sensor is i-th node of a run whose length is l_1. Here, n and s are nonsingular and singular codes. All these pieces of information are stored and processed by the local processor. Using these values, we decide the next node at each sensor. For example, if the current node is l_1-th node in an odd run, then the next prospective node in the straight line lies in direction s from the current point. The current point is the latest point decided to be in the straight line segment.

For the DQSS example shown in Figure 3, the triplet (<RunID>,<RunLimit>,<NodeID >) values (at sensor nodes) starting from the source node are as follows: $(\mathrm{odd}, 3, 1)$, $(\mathrm{odd}, 3, 2)$, $(\mathrm{odd}, 3, 3)$, $(\mathrm{even}, 4, 1)$, $(\mathrm{even}, 4, 2)$, $(\mathrm{even}, 4, 3)$, $(\mathrm{even}, 4, 4)$, $(\mathrm{odd}, 3, 1)$, $(\mathrm{odd}, 3, 2)$, $(\mathrm{odd}, 3, 3)$, $(\mathrm{even}, 4, 1)$, $(\mathrm{even}, 4, 2)$, $(\mathrm{even}, 4, 3)$, $(\mathrm{even}, 4, 4)$, $(\mathrm{odd}, 3, 1)$, $(\mathrm{odd}, 3, 2)$, and $(\mathrm{odd}, 3, 3)$. Whenever a sensor node gets the node ID equal to run limit, it flips the run ID (odd to even or even to odd) and selects the next node in direction s from the current point.

The decision-making process is very lightweight as no other arithmetic or complex computations are involved. The nodes check the index limits at every node and forward the incremented index information, whereas in Bresenham's line drawing algorithm, we need to compute the decision parameter's value at each pixel position to decide the next pixel [39]. In Bresenham's algorithm, the decision parameter is updated by involving addition, multiplication, and comparison operations. In contrast, our method computes the necessary parameters once and only uses increment and comparison operations in the loop. Figure 5 shows a comparison of the DQSS line with the Bresenham's line. Also, a comparison between DQSS and Bresenham's line algorithm on the CPU times for various line segments is shown in Figure 6 and Table 4.

5.2. Dead Node Avoidance. We assume that significantly fewer dead nodes will be present in the grid. During the making of the straight line path, at some point, if a dead node appears as the following selection, we wish to avoid it. This is done by increasing or decreasing the current nonsingular run length (run limit). We have the following two cases.

(i) The current node p_{curr} is a dead node, and it is the first node of a run (reached using singular code s from the previous point). The run limit of the current run is increased by 1. An example is shown in Figure 7

(ii) The current node p_{curr} is a dead node, and it is any node other than the first node of a run (reached using nonsingular code n from the previous point). The run limit of the current run is reset by index – 1 (index points the current node p_{curr}. Hence, p_{curr} is avoided by applying a move using the singular code on the previous node of p_{curr}. An example is shown in Figure 8

5.3. Energy Consumption. In WSN, the energy consumed is the sum total of energy consumed by individual nodes (see

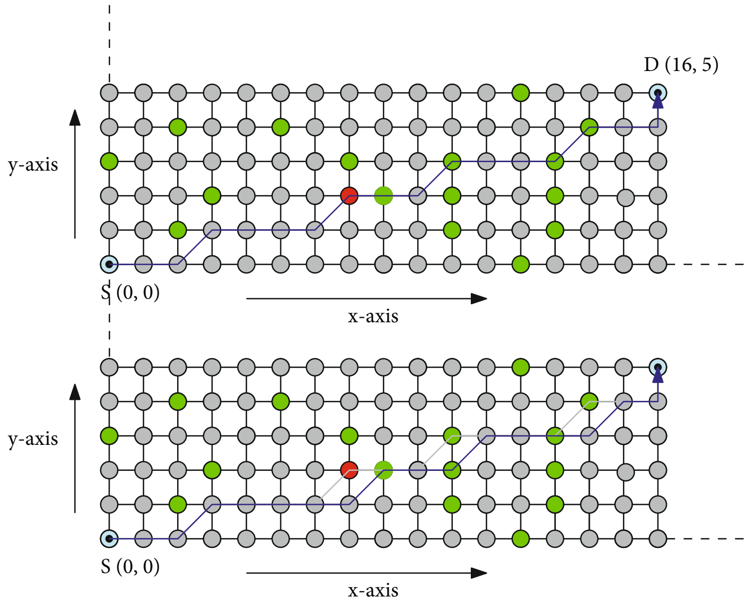


FIGURE 7: Dead node avoidance by increasing the current run length limit; the length of the first even run is increased by 1.

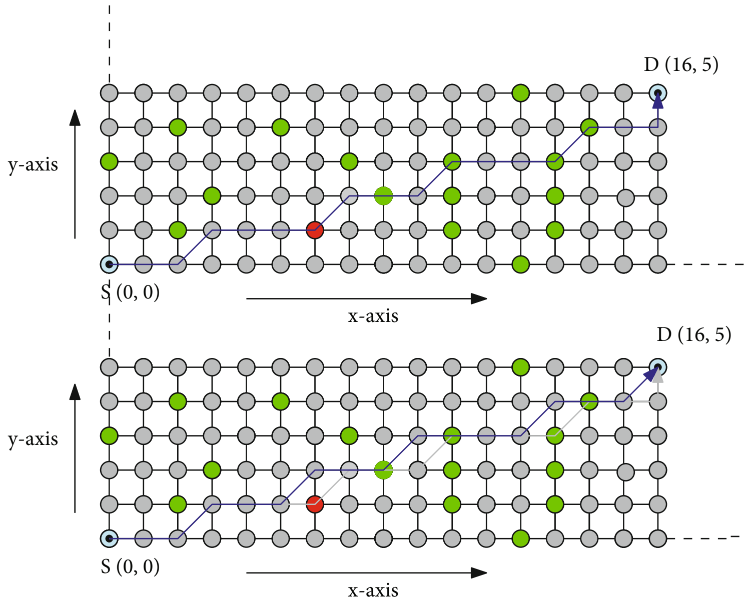


FIGURE 8: Dead node avoidance by decreasing the current run length limit; limit is reduced to 3 from 4.

Equation (1)) [40, 41]. Energy consumed by a node comprises of energy for transmitting packets (E_t), that for receiving packets (E_r), and consumptions because of sleeping (E_s).

$$E_{\text{Total}} = \sum_{i=1}^{n} E_i, \quad (1)$$

where $E_i = E_t + E_r + E_s$.

For E_r and E_s, most of the network simulators use standard values. However, E_t depends on various factors. Most prominent of which includes packet size (l) and distance between nodes (d). Hence, E_t may be expressed using the formula shown in

$$E_t = l * E_{\text{bit}} + l * \lambda * d^2, \quad (2)$$

where λ is medium constant.

Our proposed algorithm focuses on minimizing the path length between the two given nodes by finding a quasistraight line segment between the two nodes. Minimization of the path length ensures minimization of the energy consumption.

6. Conclusion

This paper proposes a novel quasistraight line routing protocol based on quasistraight line fitting, which is derived from Freeman's chain code-based straightness properties. The proposed algorithm focuses on the grid-based deployment of sensor nodes in WSN. If the constructed path attempts to go through a dead node, the path is modified so that the length minimization constraint is maintained with minimum deviation. The method has been compared with a standard straight line finding algorithm, and the results show its applicability.

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] F. Bouakkaz and M. Derdour, "Maximizing WSN life using power efficient grid-chain routing protocol (PEGCP)," *Wireless Personal Communications*, vol. 117, no. 2, pp. 1007–1023, 2021.

[2] R. Zagrouba and A. Kardi, "Comparative study of energy efficient routing techniques in wireless sensor networks," *Information*, vol. 12, no. 1, p. 42, 2021.

[3] J. Yan, M. Zhou, and Z. Ding, "Recent advances in energy-efficient routing protocols for wireless sensor networks: a review," *IEEE Access*, vol. 4, pp. 5673–5686, 2016.

[4] H.-H. Liu, S. Jia-Jang, and C.-F. Chou, "On energy-efficient straight-line routing protocol for wireless sensor networks," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2374–2382, 2017.

[5] S. Umamaheswari, W. S. Antony, and A. Joe, "Detection and correction of node failures in wireless sensor networks," in *International Conference on Advanced Computing and Communication Systems (ICACCS), volume 1*, pp. 1479–1483, Coimbatore, India, 2021.

[6] R. N. Jadoon, A. A. Awan, M. A. Khan, W. Zhou, and A. Shahzad, "An Efficient Nodes Failure Recovery Management Algorithm for Mobile Sensor Networks," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1749467, p. 14, 2020.

[7] K. Mahmood, M. K. Saeed, S. Ali, S. Zaman, A. Al Awady, and M. Saqib, "Smart node relocation (snr) and connectivity restoration mechanism for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–19, 2021.

[8] H. Yetgin, K. T. K. Cheung, M. el-Hajjar, and L. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[9] A. A. Aziz, Y. Ahmet Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[10] F. Engmann, F. A. Katsriku, J.-D. Abdulai, K. S. Adu-Manu, and F. K. Banaseka, "Prolonging the lifetime of wireless sensor networks: a review of current techniques," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8035065, 23 pages, 2018.

[11] I. Benkhelifa, N. Nouali-Taboudjemat, and S. Moussaoui, "Disaster management projects using wireless sensor networks: an overview," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 605–610, Victoria, BC, Canada, 2014.

[12] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A survey on multihop ad hoc networks for disaster response scenarios," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, Article ID 647037, 2015.

[13] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[14] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.

[15] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[16] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, New Orleans, LA, USA, 1999.

[17] R. S. Battula and O. S. Khanna, "Geographic routing protocols for wireless sensor networks: a review," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 12, pp. 39–42, 2013.

[18] E. Ahvar, S. Ahvar, G. M. Lee, and N. Crespi, "An energy-aware routing protocol for query-based applications in wireless sensor networks," *The Scientific World Journal*, vol. 2014, Article ID 359897, 9 pages, 2014.

[19] B. Blywis, M. Güneş, F. Juraschek, and S. Hofmann, "Gossip routing in wireless mesh networks," in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1572–1577, Istanbul, Turkey, 2010.

[20] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 22–31, Atlanta, Georgia, USA, 2002.

[21] M.-J. Lin, K. Marzullo, and S. Masini, *Gossip Versus Deterministic Flooding: Low Message Overhead and High Reliability for Broadcasting on Small Networks*, Technical report, Department of Computer Science and Engineering, University of California, San Diego, USA, 1999.

[22] C.-F. Chou, S. Jia-Jang, and C.-Y. Chen, "Straight line routing for wireless sensor networks," in *10th IEEE Symposium on Computers and Communications (ISCC'05)*, pp. 110–115, Murcia, Spain, 2005.

[23] H. Shokrzadeh, A. T. Haghighat, F. Tashtarian, and A. Nayebi, "Directional rumor routing in wireless sensor networks," in *2007 3rd IEEE/IFIP International Conference in Central Asia on Internet*, pp. 1–5, Tashkent, Uzbekistan, 2007.

[24] H. Shokrzadeh, M. Mashaiekhi, and A. Nayebi, "Improving directional rumor routing in wireless sensor networks," in

2007 Innovations in Information Technologies (IIT), pp. 108–112, Dubai, United Arab Emirates, 2007.

[25] S. Hamid, A. M. Rahmani, A. T. Haghighat, and N. Forouzideh, "SDRR: serial directional rumor routing in wireless sensor networks," in *2010 International Conference on Networking and Information Technology*, pp. 75–79, Manila, Philippines, 2010.

[26] T. Banka, G. Tandon, and A. P. Jayasumana, "Zonal rumor routing for wireless sensor networks," in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II, volume 2*, pp. 562–567, Las Vegas, NV, USA, 2005.

[27] S. Phoemphon, C. So-In, and N. Leelathakul, "Improved distance estimation with node selection localization and particle swarm optimization for obstacle-aware wireless sensor networks," *Expert Systems with Applications*, vol. 175, article 114773, 2021.

[28] O. Banimelhem, E. Taqieddin, and I. Shatnawi, "An efficient path generation algorithm using principle component analysis for mobile sinks in wireless sensor networks," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 69, 2021.

[29] M. A. Fadi, E. A. Ashraf, S. H. Hossam, and A. I. Mohamed, "Deploying faulttolerant grid-based wireless sensor networks for environmental applications," in *IEEE Local Computer Network Conference*, pp. 715–722, Denver, CO, USA, 2010.

[30] S. Guo, Z. Zhong, and T. He, "Find: faulty node detection for wireless sensor networks," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pp. 253–266, Berkeley, California, 2009.

[31] R. R. Panda, B. S. Gouda, and T. Panigrahi, "Efficient fault node detection algorithm for wireless sensor networks," in *2014 International Conference on High Performance Computing and Applications (ICHPCA)*, pp. 1–5, Bhubaneswar, India, 2014.

[32] M. Asim, H. Mokhtar, and M. Merabti, "A fault management architecture for wireless sensor network," in *2008 International Wireless Communications and Mobile Computing Conference*, pp. 779–785, Crete, Greece, 2008.

[33] K. Rajeswari and S. Neduncheliyan, "Genetic algorithm based fault tolerant clustering in wireless sensor network," *IET Communications*, vol. 11, no. 12, pp. 1927–1932, 2017.

[34] A. Rosenfeld, "Digital straight line segments," *IEEE Transactions on Computers*, vol. C-23, no. 12, pp. 1264–1269, 1974.

[35] P. Bhowmick and B. B. Bhattacharya, "Fast polygonal approximation of digital curves using relaxed straightness properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1590–1602, 2007.

[36] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, 1961.

[37] H. Freeman and L. S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Transactions on Computers*, vol. -C-26, no. 3, pp. 297–303, 1977.

[38] G. Ramamurthy, T. JagannadhaSwamy, and A. Jain, "Cost and energy efficient distributed computation: wireless sensor networks on uniform grid," in *2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET)*, pp. 1–5, Hyderabad, India, 2021.

[39] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[40] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, "A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 1115–1127, 2018.

[41] S. Verma, Y. Kawamoto, and N. Kato, "Energy-efficient group paging mechanism for qos constrained mobile IoT devices over LTE-A Pro networks under 5G," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9187–9199, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4336647, 9 pages
https://doi.org/10.1155/2022/4336647

Research Article

Energy-Efficient UART Design on FPGA Using Dynamic Voltage Scaling for Green Communication in Industrial Sector

D. Haripriya,[1] **Keshav Kumar**,[2] **Anurag Shrivastava**,[3] **Hamza Mohammed Ridha Al-Khafaji**,[4] **Vishal Moyal**,[5] **and Sitesh Kumar Singh**[6]

[1]*Department of ECE, SRM Institute of Science and Technology, Ramapuram Campus, Chennai 600089, India*
[2]*University Institute of Computing, Chandigarh University, Punjab, India*
[3]*Department of Electronics and Communication Engineering, Lakshmi Narain College of Technology and Science, Indore, 453111 Madhya Pradesh, India*
[4]*Biomedical Engineering Department, Al-Mustaqbal University College, 51001 Hillah, Babil, Iraq*
[5]*Department of Electrical Engineering, SVKMs Institute of Technology, Dhule, M.S 424002, India*
[6]*Department of Civil Engineering, Wollega University, Nekemte, Oromia, Ethiopia*

Correspondence should be addressed to Sitesh Kumar Singh; sitesh@wollegauniversity.edu.et

Received 19 February 2022; Accepted 29 March 2022; Published 5 May 2022

Academic Editor: Mohammad R Khosravi

In the present scheme of the world, the problem of shortage of power is seen across the world which can be a vulnerability to various communication securities. The scope of proposed research is that it is a step towards completing green communication technology concepts. In order to improve energy efficiency in communication networks, we designed UART using different nanometers of FPGA, which consumes the least amount of energy. This shortage is happening because of expanding of industries across the world and the rapid growth of the population. Therefore, to save the power for our upcoming generation, the globe is moving towards the concept and ideas of green communication and power-/energy-efficient gadget. In this work, a power-efficient universal asynchronous receiver transmitter (UART) is implemented on 28 nm Artix-7 field-programmable gate array (FPGA). The objective of this work is to reduce the power utilization of UART with the FPGA device in industries. To do this, the same authors have used voltage scaling techniques and compared the results with the existing FPGA works.

1. Introduction

In recent times, it has been observed that the whole globe is suffering from one serious problem which is power deficiency. This is happening all over the globe due to the vast increase in the population as well as industrialization. Therefore, to save power for our upcoming generation, the whole world is going towards the concept of energy-/power-efficient gadgets and green communication technology. The "green communication" refers to methods for conserving energy resources for future generations without affecting current generation use. As a result, UART may be useful in developing green communication concepts. Our research work is a step towards fulfilling the designs of green communication technologies. The green communication enables totally better idea of working, interacting, and cooperating, allowing corporations to go further while reducing pollution, greenhouse gas emissions, and power usage. Many organizations are reluctant to make the switch due to the high initial expenditures. We created UART utilizing various nanometers FPGA, which consumes the least amount of energy, in order to minimize energy usage in communication networks. UART is an abbreviation for universal asynchronous receiver transmitter. UART has a frequency of 1 GHZ, a responsibility cycle of 50%, and a time period of 1 ns. The responsibility cycle of a signal is the amount of time it is used. The power and duty cycle relationship = (PW/T) 100, where D is the responsibility cycle, PW is the pulse width, and T is the signal's time period. In UART, data is sent at a particular frequency called Baud rate. In the UART time technique, the data is sampled when the baud rates of the receiver and transmitter are properly aligned. In the process of data transfer, the data is sent in an irregular way via UART connection. That is, no clock signal is necessary to transfer data from UART A to UART B. As a result, UART may be useful in developing green communication concepts. The hardware circuit of UART device is associated with microcontrollers, laptops, and CPU of a computer. Sometimes it can be dedicated to an integrated circuit (IC). Despite a lot of new communicating ideas, UART communication is most preferred for serial communication. This is because UART devices are easily integrable, and it only uses two wires to perform the serial communication, which is given in Figure 1.

The data transfer in UART takes place in the form of packets; i.e., the UART sends data to another UART in the form of packets of bits. Since the communication of data in UART transfers data in an asynchronous manner, that is, for sending data from UART A to UART B, no clock signal is required. Therefore, UART can be beneficial for promoting the ideas of green communication. If we compare the proposed method with existing methods in this research, voltage scaling is used to calculate power, and the findings of the study are compared to previous methodologies. It has been found that researchers have employed a variety of strategies to minimize power consumption in previous studies, yet consumption can still be lowered. The existing works done on UART implementation on FPGA to promote the ideas of green communication are explained in Section 2.

The present article has been planned into seven sections. Section 1 describes the introduction of UART with green communication. Section 2 puts light on related work. The implementation setup and methodology have been mentioned in Section 3. Section 4 described the thermal properties for different voltage values. The power calculation of UART has been discussed in Section 5. Finally, Section 6 portrays the conclusion and possible future works based on the proposed framework.

1.1. Field-Programmable Gate Array (FPGA). Unlike the other microcontrollers, FPGAs are also those gadgets that are composed up of semiconductor materials [2–4]. These devices work similarly to the other microcontrollers but have one distinguished property which makes FPGA more convenient than the other microcontrollers. The major and the most important advantage of using FPGAs is these [5] can be reprogrammed after its manufacturing. The feature of being reprogrammed makes FPGAs handier and convenient to be used than the other microcontrollers [6, 7]. The major building blocks of FPGAs are look-up tables (LUTs), configurable logic blocks (CLBs), flip-flops, input/output (I/O) devices, memory devices, and buffers [8]. The building components of FPGAs are shown in Figure 2. FPGA devices are used for performing the green communication too. Green communication is the techniques in which we tend to save the energy resources for our future generation without compromising the use of present generation. The FPGA version of green computing model of UART is shown in Figure 3.

2. Literature Work

The authors created an energy-efficient instruction register for integrating green communication on Virtex 4, Virtex 5, and Virtex 6 FPGAs [9]. As a result, while much work has been done to incorporate the ideas of green communication and energy-/power-efficient devices for future generations on CU with various FPGAs, no work has been done to implement the CU circuit on Kintex-7 ultrascale FPGA, so in this work, the CU circuit is being designed on Kintex-7 ultrascale FPGA to promote green communication techniques.

To provide a high-performance FIFO for the CPU while reducing power usage, Saxena et al. [10] employed voltages and frequency scaling techniques to create FPGA-based FIFO architecture. They altered frequency from 20 M Hz to 250 MHz while keeping the voltage constant at 2.3 volt, while for the other experiments, they maintained the frequency constant while varying voltages from 1 volt to 2.3 volt. They concluded that the voltage scaling reduces power usage by 95.79 percent, whereas frequency scaling reduces power consumption by 4.38 percent.

Kumar et al. [1] proposed a Spartan-3 and Spartan-6 field-programmable gate arrays that are used to create a low-power transceiver (FPGA). As a transceiver, a universal asynchronous receiver transmitter (UART) device is employed. The power analysis findings are aimed on Spartan-3 and Spartan-6 FPGAs, and the implementation of UART is achievable with EDA tools named Xilinx 14.1. By altering the voltage supply, the change of different power of chips built on FPGA is noticed, for example, input/output (I/O) power consumption, leakage power absorption, signal power utilization, logic power utilization, and the use of complete power. This study examines how voltage changes affect the power consumption of the UART on Spartan-3 and Spartan-6 FPGA devices. Spartan-6 is proven to be more power efficient as the voltage supply is increased.

In the current international situation, the global energy crisis is a very serious concern. The energy crisis in India, as well as a scarcity of natural resources such as crude oil, coal, and other minerals, has an impact on the country's economy [11]. Global demand for energy has risen dramatically as a result of population increase and industrial development. So, in order to save energy, we are creating a UART with FPGAs that uses less power. The universal asynchronous receiver transmitter, or UART, is a serial data transfer device. For data transfer, just two wires are required in UART. Not only that, but there are no clock signals needed to run UART. When the voltage is at its maximum, the UART creates less noise and interference, allowing the signal to travel further [12, 13]. The writers created an electronic control unit (ECU) on FPGA to control the vehicle's system. For parallel work, the reduced instruction set computer (RISC) machine (ARM) processor is utilized in conjunction with FPGA [14].

Kumar and Pandey [14] used stub series terminated logic (SSTL) IO with three distinct FPGAs with varying nanometer (nm) gate sizes: 28 nm SP AR TAN-7, 20 nm KINTEX-7 ultrascale, and 16 nm ZYNQ ultrascale+. The model was created and implemented using the VIV ADO

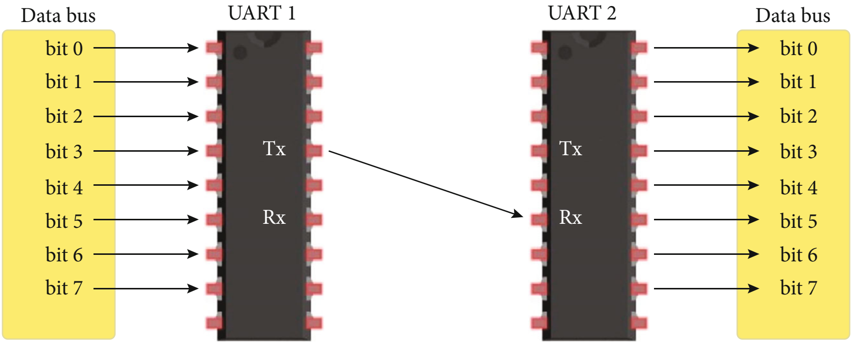


Figure 1: Serial communication in UART [1].

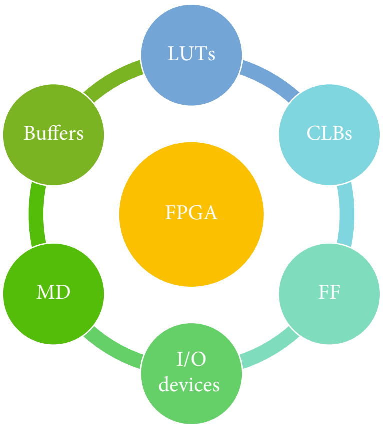


Figure 2: Building blocks of FPGAs.

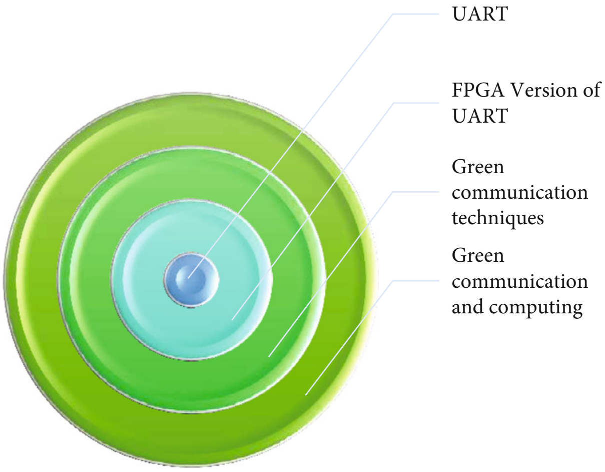

UART
FPGA Version of UART
Green communication techniques
Green communication and computing

Figure 3: Green computing model of UART.

ISE tool. According to the power study, the 16 nm ZYNQ ultrascale+ requires the most power for operation with SSTL18 I IO, while the 28 nm SP AR TAN-7 requires the least power for operation with SSTL135 IO, and the 20 nm KINTEX-7 ultrascale sits in the middle of both of these devices.

Pandey et al. [5] proposed that a power-efficient control unit (CU) is designed and implemented on the Kintex-7 ultrascale FPGA. The VIVADO HLx design suite is used to simulate and analyze the control unit. The energy consumed of the control unit is examined for various frequency values, and it is discovered that as the frequency grows, so does the total power consumption. As a result, the control unit is better suited to operate at low frequencies in order to reduce power consumption. In addition, lowering the device operating frequency of the control unit from 5 GHz to 100 MHz reduces the overall power usage by 36%.

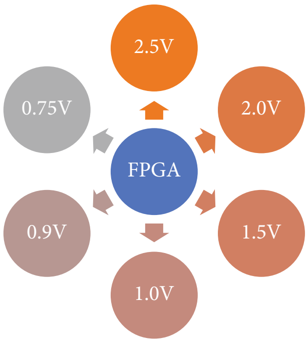


Figure 4: Voltage range for power calculation.

3. Implementation Setup and Methodology

The implementation and simulation of UART protocol with FPGA are done on XILINX ISE design suite [5, 15]. The results of power consumption of UART are observed for various input voltage ranging from 2.5 V to 0.75 V which is shown in Figure 4. The power calculation is done by X power analyzer tool [16, 17].

4. Thermal Properties for Different Voltage Values

The three thermal possessions are related to FPGA which are termed such as

(i) Effective thermal resistance to air (effective TJA) (°C/W). It shows how the power is distributed to ambient air. For all the value of voltage it is 3.3˚C/W [18–20]

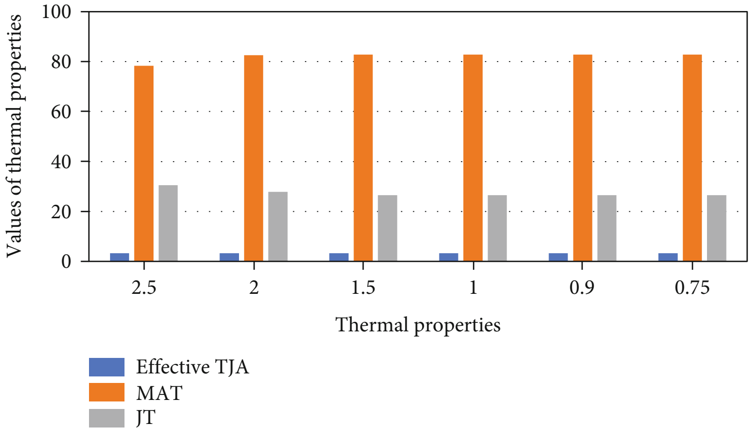


Figure 5: Thermal properties for different voltage values.

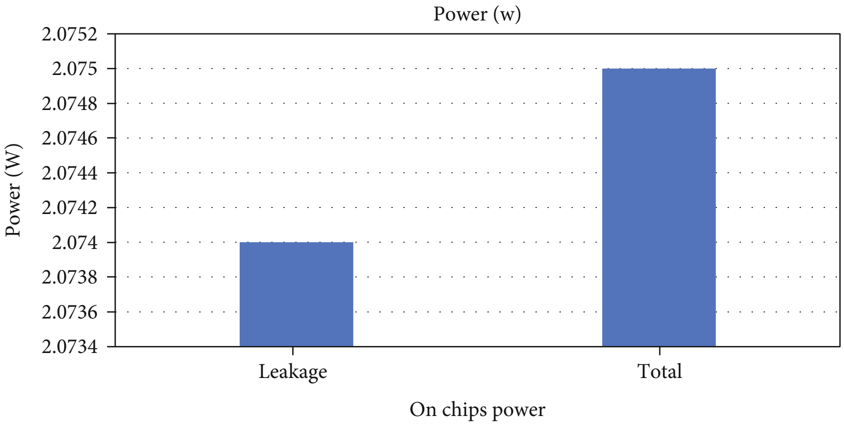


Figure 6: TP for the voltage of 2.5 V.

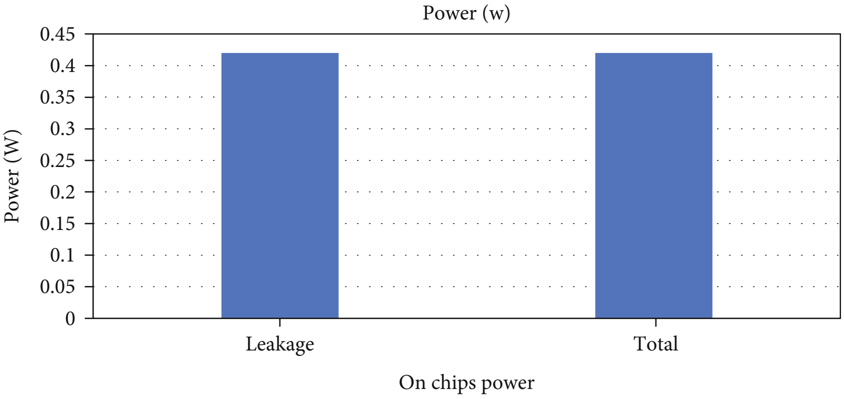


Figure 7: TP for the voltage of 2.0 V.

(ii) Maximum (max) ambient temperature (MAT) (°C). Under operating conditions, it is expressed as the temperature around the FPGA [21–23]

(iii) Junction temperature (JT) (°C). It is called as the operational temperature of the FPGA [5, 24]. It is the aggregate total on chips power, effective TJA, and MAT [5]

The thermal properties with all the voltage range of values for UART protocol for Artix-7 FPGA are represented in Figure 5.

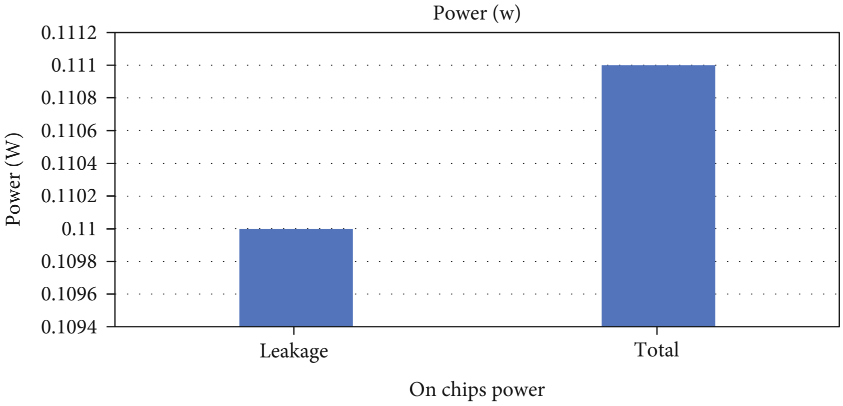


Figure 8: TP for the voltage of 1.5 V.

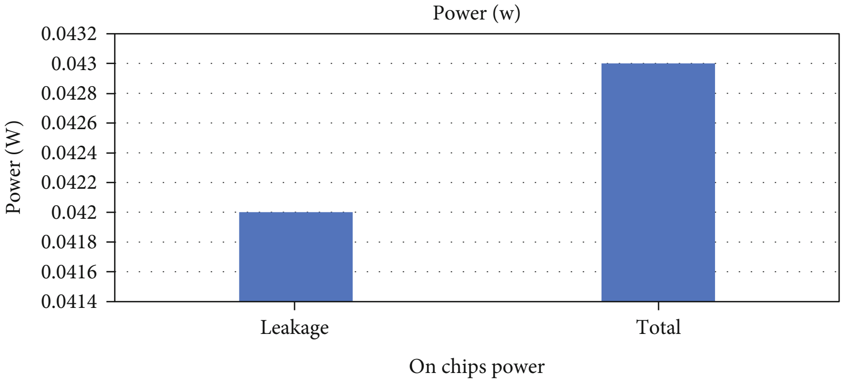


Figure 9: TP for the voltage of 1.0 V.

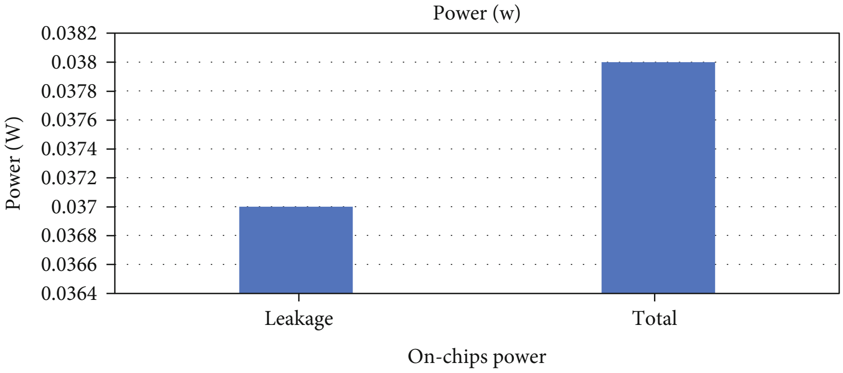


Figure 10: TP for the voltage of 0.9 V.

5. Power Calculation of UART

The total power (TP) dissipation of UART protocol on FPGA device is the sum up of the dynamic power (DP) of the device and the static power (SP) of the device [25, 26]. Although there are a large number of innovative communication concepts, serial communication via UART is still the most popular. This is due to the ease with which UART devices may be integrated and the fact that serial communication is accomplished with only two wires. The dynamic power is the power calculated when there is any switching in the device, whereas the static power is the steady-state power of the device. In a FPGA device the clock, logic, IO, and signal power are the device static power, whereas the leakage power is the device dynamic power [3, 4]. Whenever the transmission rates of the transmitter and the receiver are suitably aligned, the data is sampled using the UART time approach. Microcontrollers, laptops, and a computer's CPU are all connected to the physical circuit of a UART device. Sometimes, it can be dedicated to an integrated circuit (IC).

$$\mathrm{TP} = \mathrm{DP} + \mathrm{Sp}. \tag{1}$$

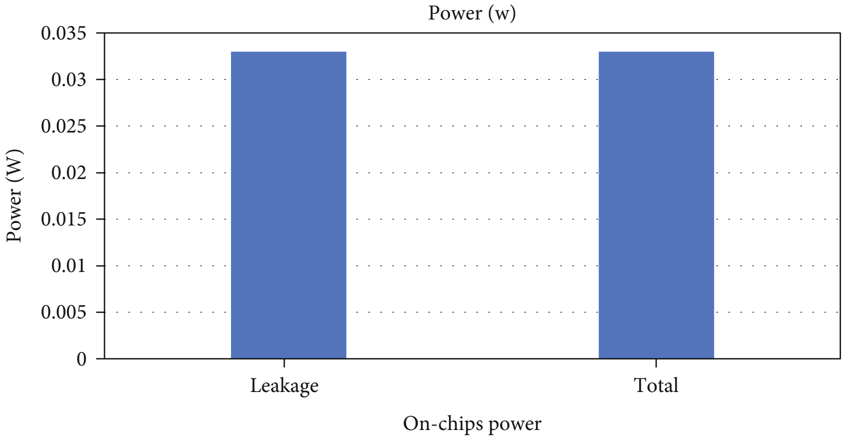


Figure 11: TP for the voltage of 0.75 V.

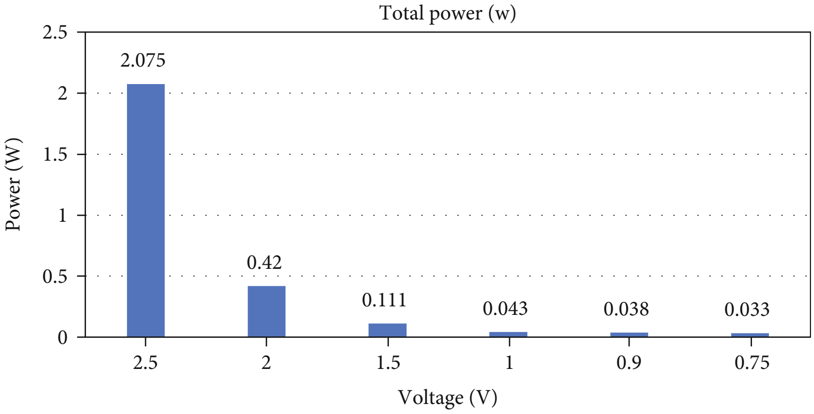


Figure 12: TP for all voltage values.

Table 1: Comparative power analysis.

S. no	Reference	FPGA	Power (W)
1.	[9]	Virtex 6	17.226
2.	[10]	Virtex 6	2.244
3.	[1]	Virtex 6	1.293
4.	[11]	Virtex 6	45.334
5.	[12]	Kintex 7	1.804
6.	[13]	Virtex 4	0.177
7.	[14]	Virtex 6	1.407
8.	[27]	Spartan 6	0.296
9.	[28]	Spartan 6	0.297
10.	[4]	Spartan 3	0.080
11.	This work	Artix 7	0.033

5.1. Power Analysis for 2.5 V Voltage. When the voltage is set to 2.5 V for the power calculation, then, there is no SP consumption for the FPGA device; that is, the SP is 0.00 W. On the other hand, the DP, which is the leakage power consumption, is 2.074 W. Hence, the TP of the UART for 2.5 V is 2.075 W. The TP for the voltage of 2.5 V is shown in Figure 6.

5.2. Power Analysis for 2.0 V Voltage. When the voltage is set to 2.0 V, the TP consumption of the device becomes 0.420 W. For 2.0 V voltage, the leakage power is 0.420 W. There is no consumption of SP for the device at this level of voltage. The TP at this voltage value is equivalent to the leakage power of the FPGA. The power consumption for 2.0 V voltage is represented in Figure 7.

5.3. Power Analysis for 1.5 V Voltage. For the voltage value of 1.5 V, the device DP is 0.110 W, and there is no power utilization of SP. Hence, the TP for this voltage becomes similar to the DP. The TP utilization for this voltage is 0.111 W. The power consumption for this value of voltage is described in Figure 8.

5.4. Power Analysis for 1.0 V Voltage. When the voltage is regulated to 1.0 V for the power calculation, then, there is no SP consumption for the FPGA device that is the SP is 0.00 W. On the other hand, the DP, which is the leakage power consumption, is 0.042 W. Hence, the TP of the UART for 2.5 V is 0.043 W. The TP for the voltage of 1.0 V is shown in Figure 9.

5.5. Power Analysis for 0.9 V Voltage. When the voltage is tweaked to 0.9 V, the TP consumption of the device becomes

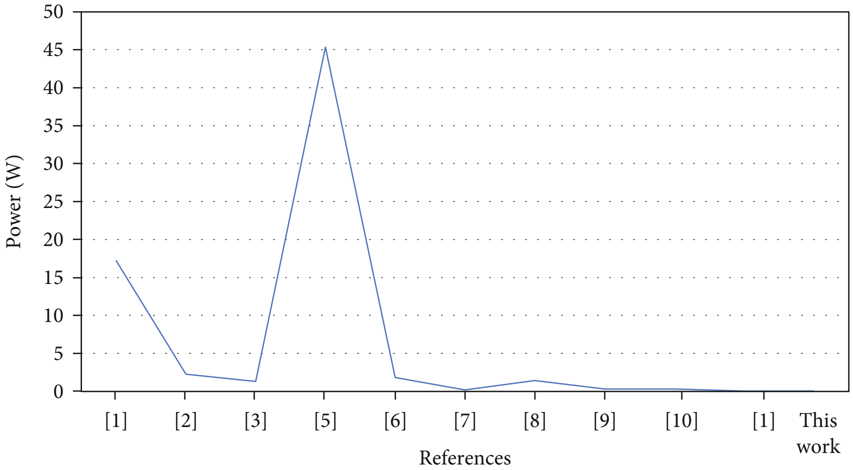


Figure 13: Total power comparisons.

0.038 W. For 0.9 V voltage, the leakage power is 0.037 W. There is no consumption of SP for the device at this level of voltage. The TP at this voltage value is equivalent to the leakage power of the FPGA. The power consumption for 0.9 V voltage is represented in Figure 10.

5.6. Power Analysis for 0.75 V Voltage. When the voltage is regulated to 0.75 V for the power calculation, then, there is no SP consumption for the FPGA device; that is, the SP is 0.00 W. On the other hand, the DP, which is the leakage power consumption, is 0.033 W. Hence, the TP of the UART for 0.75 V is 0.033 W. The TP for the voltage of 0.75 V is shown in Figure 11.

By analyzing the power, it can be seen that as the value of voltage drops, the power consumption gets decreased. The power consumption is higher for 2.5 V voltage and lower for 0.75 V voltage. The TP consumption for all the value of voltages is represented in Figure 12.

5.7. Comparative Power Analysis. From the related work section, it is observed that a lot of work has been done by the researchers to optimize the power consumption of the UART protocol. The voltage scaling method is used to calculate power, and the findings of the study are compared to previous methodologies. It has been found that researchers have employed a variety of strategies to minimize power consumption in previous studies, yet consumption can still be lowered. In this section, we have compared our best results with the existing work in recent times. In this work, we have found that the power consumption of UART is optimized when the input supplied voltage is 0.75 V. Of all the rest of the values of voltages, the power consumption is higher as it is explained in Section 4. The comparative power analysis of our work with the other existing work is described in Table 1.

From Table 1, it can be seen that in [9] using the capacitance scaling technique, the TP consumption is 17.226 W. When the thermal characteristics are adjusted in [10], the power usage is 2.244 W. On the Virtex-6 FPGA, TP consumption reaches 1.2936 W in [1]. By adjusting the capacitance at the output load in [11], the power dissipation is increased to 45.334 W. [12] uses multiple IO standards to reduce the TP consumption on the Kintex-7 FPGA to 1.804 W. The power dissipation in [13] is 0.177 W due to the utilization of numerous FPGAs with varied nanoscale technologies. The power consumption of UART is 1.407 W in [14] when various IO standards are used. On the Spartan-6 FPGA, the UART power reaches 0.296 W in [27]. Using the frequency scaling technique, [28] determined that the power usage of the UART is 0.297 W. In [4] with the idea of voltage scaling, the power consumption of UART is 0.080 W on Spartan-3 FPGA. But in our work, the power of UART reaches to 0.033 W, by applying the voltage scaling approach on 28 nm Artix-7 FPGA. The comparison of the total power consumption of our proposed method with the existing techniques is shown in Figure 13.

The problem of energy shortage is affecting the entire planet. This is occurring as a result of massive population and industry expansion throughout the world. As a result, the entire globe is attempting to embrace green communication technology and power/energy saving gadgets. This project is only focused on these technologies. The dynamic power is computed when the device is switched on; meanwhile, the static power is determined when the device is in its stable state. At 2.0 V voltage, there is no use of SP by the gadget. At this voltage level, the TP is equal to the FPGA's leakage power. When the voltage is controlled at 1.0 V for power calculations, the FPGA device consumes no SP, resulting in an SP of 0.00 W. When the voltage is increased to 0.9 V, the device's TP consumption drops to 0.038 W. The leakage power at 0.9 V voltage is 0.037 W. When looking at the power, it can be seen that when the voltage value decreases, the power consumption increases.

6. Conclusion and Future Scope

In the work introduced in this research, we have implemented UART on 28 nm Artix-7 FPGA for green communication. The analysis and simulation are implemented on XILINX design suite, and the power calculation is done through X power analyzer. The purpose of this research is to lower the power usage of UART in companies using an FPGA device. The scientists employed voltage scaling techniques to accomplish this and compared the findings to previous FPGA work. In this work, the power calculation is done by scaling voltage, and the results of the analysis are compared with the existing techniques. It is observed that in the existing works, researchers have used a lot of different techniques to reduce the power consumption, but the consumption can be reduced up to 0.080 W of power in [1]. In the other work, the power consumption is relatively more than the power consumption in [1]. From comparing our results with [1], it is found that in our proposed design, the power consumption is reduced up to 58.75%. The implementation of UART can be done on the upcoming advanced ultrascale and ultrascale+ FPGAs in future. Later these designs can be converted into ASIC design which is handier and portable than FPGAs.

Data Availability

The data shall be made available on request.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This research work is self-funded.

References

[1] K. Kumar, B. Pandey, A. K. Pandit, Y. A. El-Ebiary, S. A. Mjlae, and S. Bamansoor, “Design of low power transceiver on Spartan-3 and Spartan-6 FPGA,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12S2, pp. 27–30, 2019.

[2] V. Jagota, M. Luthra, J. Bhola, A. Sharma, and M. Shabaz, “A secure energy-aware game theory (SEGaT) mechanism for coordination in WSANs,” *International journal of swarm intelligence research*, vol. 13, no. 2, pp. 1–16, 2022.

[3] K. Kumar, K. R. Ramkumar, and A. Kaur, “A design implementation and comparative analysis of advanced encryption standard (AES) algorithm on FPGA,” in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 182–185, Noida, India, 2020.

[4] A. Shrivastava, A. Rizwan, N. S. Kumar et al., “VLSI implementation of green computing control unit on Zynq FPGA for green communication,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 4655400, 10 pages, 2021.

[5] B. Pandey, K. Kumar, A. Batool, and S. Ahmad, “Implementation of power-efficient control unit on ultra-scale FPGA for green communication,” *3C Tecnología*, vol. 10, no. 1, pp. 93–105, 2021.

[6] R. Hartenstein, “Basics of reconfigurable computing,” in *Designing Embedded Processors*, pp. 451–501, Springer, Dordrecht, 2007.

[7] G. V. Bharadwaj, A. V. Krishna, M. S. Krishna, and T. Akhil, *Novel Technique on Channel Security using UART*, 2014.

[8] T. Kumar, B. Pandey, T. Das, and B. S. Chowdhry, “Mobile DDR IO standard based high performance energy efficient portable ALU design on FPGA,” *Wireless Personal Communications*, vol. 76, no. 3, pp. 569–578, 2014.

[9] M. T. Siddiquee, K. Kumar, P. Pandey, and A. Kumar, “Energy efficient instruction register for green communication,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 2S2, 2019.

[10] A. Saxena, A. Bhatt, P. Gautam, P. Verma, and C. Patel, “High performance FIFO design for processor through voltage scaling technique,” *Indian Journal of Science and Technology*, vol. 9, no. 46, 2016.

[11] D. Nandy, “Energy crisis of India: in search of new alternatives,” *Journal of Business & Financial Affairs*, vol. 5, no. 4, pp. 1–6, 2016.

[12] K. Kumar, A. Kaur, S. N. Panda, and B. Pandey, “Effect of different nanometer technology-based FPGA on energy efficient UART design,” in *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 1–4, Bhopal, India, 2018.

[13] K. Kumar, A. Kaur, B. Pandey, and S. N. Panda, “Low power UART design using different nanometer technology-based FPGA,” in *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 1–3, Bhopal, India, 2018.

[14] J. Pérez, M. Alcázar, J. M. Velasco, J. A. Cabrera, and J. J. Castillo, “Low-cost FPGA-based electronic control unit for vehicle control systems,” *Sensors*, vol. 19, no. 8, p. 1834, 2019.

[15] T. Kumar, B. Pandey, S. H. Mussavi, and N. Zaman, “CTHS based energy efficient thermal aware image ALU design on FPGA,” *Wireless Personal Communications*, vol. 85, no. 3, pp. 671–696, 2015.

[16] T. Das, B. Pandey, M. A. Rahman, and T. Kumar, “SSTL based green image ALU design on different FPGA,” in *2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, pp. 146–150, Chennai, India, 2013.

[17] A. Shrivastava and S. K. Sharma, “Various arbitration algorithm for on-chip (AMBA) shared bus multi-processor SoC,” in *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS-2016) organized by MNIT*, pp. 1–7, Bhopal, India, 2016.

[18] “AMBA AXI bus verification technique,” *International Journal of Applied Engineering Research*, vol. 10, no. 24, pp. 44178–44182, 2015.

[19] “Reliable routing architecture and algorithm for network-on-Chip,” *Journal of Electronic Design Technology*, vol. 6, no. 3, pp. 40–48, 2015.

[20] D. S. Ushakov, I. I. Haiovyi, I. M. Minich, and K. D. Didenko, “Transnational players in tourism: regional features of functioning,” *Geojournal of Tourism and Geosites*, vol. 32, no. 4, pp. 1425–1432, 2020.

[21] A. K. Singh, A. Shrivastava, and G. S. Tomar, “Design and implementation of high performance AHB reconfigurable arbiter for onchip bus architecture,” in *IEEE International Conference on Communication Systems and Network Technologies, organized by SMVDU*, pp. 455–459, Katra, India, 2011.

[22] A. Shrivastavastava, G. S. Tomar, and K. K. Kalra, “Efficient design and performance analysis for AMBA bus architecture

based system-on-chip," in *IEEE International Conference on Computational Intelligence and Communication Systems organized by R.G.P.V*, pp. 656–660, Bhopal, India, 2010.

[23] D. M. Pham and S. M. Aziz, "FlexiS—a flexible sensor node platform for the internet of things," *Sensors*, vol. 21, no. 15, p. 5154, 2021.

[24] K. Kumar, A. Kaur, and K. R. Ramkumar, "Effective data transmission with UART on Kintex-7 FPGA," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 492–497, Bhimtal, India, 2020.

[25] M. Shabaz and U. Garg, "Shabaz–Urvashi link prediction (SULP): a novel approach to predict future friends in a social network," *Journal of Creative Communications*, vol. 16, no. 1, pp. 27–44, 2021.

[26] P. Jindal, A. Kaushik, and K. Kumar, "Design and implementation of advanced encryption standard algorithm on 7th series field programmable gate array," in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, pp. 1–3, Chennai, India, 2020.

[27] B. Pandey and R. Kumar, "Low voltage DCI based low power VLSI circuit implementation on FPGA," in *2013 IEEE Conference on Information & Communication Technologies*, pp. 128–131, Thuckalay, India, 2013.

[28] A. Kumar, B. Pandey, D. A. Hussain, M. A. Rahman, V. Jain, and A. Bahanasse, "Low voltage complementary metal oxide semiconductor based energy efficient UART design on Spartan-6 FPGA," in *2019 11th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 84–87, Honolulu, HI, USA, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4327380, 15 pages
https://doi.org/10.1155/2022/4327380

Research Article

Federated Reinforcement Learning-Based UAV Swarm System for Aerial Remote Sensing

Woonghee Lee

Department of Applied Artificial Intelligence, Hansung University, Republic of Korea

Correspondence should be addressed to Woonghee Lee; whlee@hansung.ac.kr

Received 10 January 2022; Revised 25 February 2022; Accepted 23 March 2022; Published 29 April 2022

Academic Editor: Mohammad R Khosravi

In recent years, due to the development of technologies for unmanned aerial vehicles (UAVs), also known as drones, UAVs have developed rapidly. Because of UAVs' high mobility and computational capability, UAVs have a wide range of applications in Industrial Internet of Things (IIoT), such as infrastructure inspection, rescue, exploration, and surveillance. To accomplish such missions, it is more proper and efficient to utilize multiple UAVs in a swarm, rather than a single UAV. However, it is difficult for an operator to understand and control numerous UAVs in different situations, so UAVs require the significant level of autonomy. Artificial intelligence (AI) has become the most promising combination with UAVs to ensure the high autonomy of UAVs by establishing swarm intelligence (SI). However, existing learning methods for building SI require continuous information sharing among UAVs, which incurs repeated data exchanges. Thus, such techniques are not suitable for constructing SI in the UAV swarm, in which communication resources are not readily available on unstable UAV networks. To overcome this limitation, in this paper, we propose the federated reinforcement learning- (FRL-) based UAV swarm system for aerial remote sensing. The proposed system applies reinforcement learning (RL) to UAV clusters to establish the SI in the UAV system. Furthermore, by combining federated learning (FL) with RL, the proposed system constructs the more reliable and robust SI for UAV systems. We conducted diverse evaluations, and the results show that the proposed system outperforms the existing centralized RL-based system and is more suited for UAV swarms from a variety of perspectives.

1. Introduction

These days, the performance of the hardware and software needed for computing and artificial intelligence (AI) has become remarkably advanced, so AI is being used in a wide variety of fields including Industrial Internet of Things (IIoT). In particular, the development of deep learning has allowed computers to perform various complex operations previously performed only by humans. Unsupervised learning is used in many areas by developing from supervised learning with tagging data, and reinforcement learning (RL), in which machines learn by themselves, has already surpassed people in many areas. Since the development of deep Q network (DQN) by Google DeepMind [1], RL has been applied to Atari Games in 2015 [2], Go in 2016 [3], and StarCraft II in 2019 [4], and many studies have drawn attention to solving various problems in IIoT.

Unmanned aerial vehicles (UAVs), also known as drones, are useful in that they can be put into difficult environments for people to perform the given missions. Thus, they are used in various applications in IIoT, such as infrastructure inspection, traffic patrol, rescue, exploration, environmental monitoring, remote sensing, and surveillance [5]. To accomplish such missions, UAVs are controlled by radio from a remote controller or are self-judged by a system that has already been designed by an operator. However, it is difficult for the operator to clearly understand the situation in which UAVs exist over long distances and to control the UAVs' behaviors elaborately. In addition, it is impossible to come up with all the countermeasures for various unpredictable situations. Moreover, in recent years, a number of UAVs, rather than a single UAV, are simultaneously utilized in a cluster to perform more diverse missions of IIoT more efficiently, but it is hard to control all of these UAVs in a centralized manner. Thus, UAVs require the significant level of autonomy and should have the ability to perform tasks in unexpected situations without human intervention.

To ensure the sufficiently high autonomy of UAV, a number of studies were conducted to enable UAV clusters to perform common missions more efficiently and intelligently by utilizing AI algorithms. However, despite a lot of interest in AI, the collaboration with swarm intelligence (SI) in IIoT has not been considered deeply. It is because that it is not easy to satisfy the concept of SI systems in which each object has to decide on an action based on local and partial information obtained from its own environment.

RL is performed by an agent repeating an action based on a state in a given environment and maximizing a reward. Therefore, even for learning with the same goal of a certain application, the results of the learning can be substantially different as the environment changes. In addition, the action is chosen stochastically, so different results can be produced each time even in the same environment. Thus, even if the same learning is performed, it can result in biased results depending on the agent, which increases difficulty in establishing swarm intelligence in IIoT. To overcome this, many studies on multiagent RL have been proposed, and the studies simultaneously utilize multiple agents to perform RL. However, such methods require sharing information of agents, which incurs continuous data exchanges. Thus, it is not easy for them to be applied to the environments such as UAV systems in IIoT, in which communication resources are not readily available on unstable UAV networks.

Federated learning (FL) is a new approach to training machine learning (ML) models that decentralizes the training process, and it was first introduced in the paper published by Google [6]. In FL, each agent receives an initial common global model, which is not trained, from a server, and each agent performs independent learning. After that, the server collects the trained local models, creates a global model, and returns it back to the agents. These operations are repeated to achieve a fully trained global model. By using FL, each agent has an advantage in terms of communication resources in that it does not need to repeatedly share the data required for learning. Fusing FL with RL allows multiple agents to compose the global and unbiased model based on many agents' diverse actions in different environments without exchanging data for learning. Thus, due to these advantages, federated reinforcement learning (FRL) is suited for UAV swarms in IIoT, but only few studies have yet been applied to UAV systems.

Motivated by the fact described above, in this paper, we propose the FRL-based UAV swarm system for aerial remote sensing. To show the application of our proposed system, we take a gas detection as an application example and propose the FRL-based gas sensing system using UAV swarm. However, since the proposed system is not designed to be specialized in specific applications, the system can be applied to any IIoT applications using UAVs.

To summarize the contributions of this paper:

(i) We propose the FRL-based UAV system that outperforms the existing centralized RL-based system

(ii) We establish the swarm intelligence in UAV system by applying RL to UAV clusters

(iii) By combining FL with RL, we construct the more reliable and suitable swarm intelligence for UAV systems

(iv) We conducted diverse performance evaluations considering various factors to analyze the proposed system from a variety of perspectives

The remainder of this paper is organized as follows. In Section 2, we introduce related work and describe our research's novelties and advantages against the related work. We describe preliminary knowledge related to our research in Section 3. After that, in Section 4, we explain our proposed system and give detailed explanations about the learning algorithm and implementation. In Section 5, we describe the experiments and performance evaluation results. Finally, Section 6 concludes this paper with explaining remarks and future directions.

2. Related Work

In this section, we firstly introduce several researches which tried to apply RL or FL to UAV systems. Then, we describe some studies focusing on utilizing FRL for various systems in IIoT. After that, we explain our research's novelties and advantages in comparison with the relevant studies.

Several studies have been conducted that present a variety of techniques using RL to perform path planning tasks or address some of the subtasks. Pham et al. proposed a deep reinforcement learning (DRL) algorithm which enables UAVs to learn their paths autonomously and to pass through changing environments without collisions [7]. Lin et al. proposed a combination of DRL and long short-term memory (LSTM) [8] network that allows UAVs to interact with their surroundings directly and continuously [9]. Lilicrap et al. proposed an improved deep deterministic policy gradient (DDPG) [10] algorithm for object avoidance and target tracking [11]. The proposed algorithm uses reward functions and penalty actions to achieve smoother trajectories. Koch et al. investigated the performance and accuracy of the inner control loop providing attitude control when using autonomous flight control systems trained with various RL algorithms [12].

Using traditional DL-enabled approaches, data needs to be transmitted and stored at a central server. This can be a significant problem because it generates massive network communication overhead to send raw data to centralized entities, which can lead to network usage and energy inefficiency of UAVs. The transferred data can also include personal data such as location and identity of UAVs that can directly affect privacy issues. As a solution, FL was introduced for privacy and low communication overhead. Considering the advantages of FL, FL is much better suited for many UAV-enabled wireless applications in IIoT than the existing DL methods [13], so some researches tried to apply FL to UAV systems in IIoT. Chhikara et al. proposed an FL algorithm within a drone swarm that collects air quality data using built-in sensors [14]. Using the proposed scheme, a UAV swarm composes the SI to find the area with the highest air quality index value effectively. Awada et al. introduced an FL-based orchestration framework for a federated aerial edge computing system [15]. The authors proposed a federated multioutput linear regression model to estimate multitask resource requirements and execution time to find the optimal drone deployment.

FRL, the combination of FL and RL, is a relatively recently proposed technique, and a few researches tried to apply FRL to applications of IIoT. Lim et al. proposed an FRL architecture to allow multiple RL agents to learn optimal control policy on their own IoT devices of the same type but with slightly different dynamics [16]. Abdel-Aziz et al. proposed a RL-based cooperative perception framework and introduced an FRL approach to speed up the training process across vehicles [17]. Xu et al. proposed a multiagent FL-based incentive mechanism to capture the stationarity approximation and learn the allocation policies efficiently [18]. Xue et al. proposed an FRL framework which extracts the knowledge from electronic medical records across all edge nodes to help clinicians make proper treatment decisions [19].

This paper has novelty and advantages compared to the related studies. As explained before, a few researches utilized FRL for applications of IIoT, but among them, there are few studies that tried to apply FRL to UAV systems. However, in this paper, we propose the FRL-based UAV system for aerial remote sensing. We establish the SI in UAV system by applying RL to UAV clusters. Furthermore, by combining FL with RL, we constructed the more reliable and suitable SI for UAV systems.

3. Preliminary

This section describes preliminary knowledge related to our research. We first explain DRL, and then give a description of FL and FRL.

3.1. Deep Reinforcement Learning. RL is a mathematical framework for experience-driven autonomous learning [20], and the main base of RL is learning through interaction with environments [21]. In RL, the agent observes state, s_t, in the environment at time t. The state is statistics containing the information, such as sensor values and the agent's position, and it is necessary for the agent to select the action. In a given state, the policy returns an action, and the agent takes the selected action. After that, the state transitions to the new state, s_{t+1}, and the agent gets the reward, r_t, from the environment as feedback. The best order of action is determined by the rewards provided by the environment, and the optimal policy is one that maximizes the reward expected in the environment. Thus, using RL algorithms, the agent tries to learn a policy that maximizes expected returns.

DRL was introduced to accelerate the development of RL [22], and DRL uses neural networks to deliver innovative ways to obtain more optimal policy [1]. DL allows RL to deal with intractable decision-making problems in high-dimensional states and environments [2]. There are a variety of DRL algorithms, such as DQN, DDPG, proximal policy optimization (PPO) [23], trust region policy optimization (TRPO) [24], soft actor-critical (SAC) [25], and asynchronous advantage actor-critic (A3C) [26].

3.2. Federated Learning. Without data, model learning cannot be performed. Data often exists in the form of data islands, and the direct solution is to process the data in a centralized manner, requiring training data to be concentrated on the same server. FL shifts the focus of research on ML with data islands. In comparison to centralized learning methods, FL belonging to distributed learning methods allows individual devices in different locations to collaborate with others to learn ML models. The concept of FL was introduced by Google in 2016 and first applied to Google keyboards for joint learning on multiple Android phones [27]. Given that FL can be applied to all edge devices in IoT, there is the potential to revolutionize various IIoT areas, such as healthcare, transportation, and finance [28].

FL offers new research directions on AI in IIoT, and FL provides a new way of learning to build a personalized model without exchanging raw data. With the advancement of computing technologies, the computing resources of IoT devices have become more powerful. Training for AI is also gradually moving from central servers to edge devices. FL provides a privacy mechanism that can effectively use the computing resources of the device to train the model, preventing the leakage of personal information during data transmission. In various areas, numbers of wireless devices exist and there are a large amounts of valuable data, so FL can take full advantage of them. FL is the collection of training information from distributed devices to learn the model, and it includes the following basic steps [29, 30]. Firstly, the server sends the initial model to all of the devices, and then, each device trains its own local model using local data. After that, the devices send local model parameters back to the server, and the model parameters are aggregated into the global model. The aggregated global model is delivered to the devices again, and the above procedures are repeated.

3.3. Federated Reinforcement Learning. The combination of RL and FL was first studied in [31]. Unlike traditional FL, the authors proposed a new FL framework based on RL [2, 20, 32], i.e., FRL. In the study, the authors demonstrated that the FRL approach can take full advantage of the joint observations in the environment and perform better than simple DQNs with partial observations in the same environment. FRL was also applied to autonomous driving, and all participant agents perform steering control actions with knowledge learned by others, even when acting in very different environments [33]. In robot system control, FRL was used to fuse robot agent models and communicate experience effectively using prior knowledge and quickly adapting to new environments [34]. However, there are few studies that applied FRL to UAV systems.

4. System Design and Implementation

In this section, we explain the details of our proposed system and implementation. We first explain the concept of the

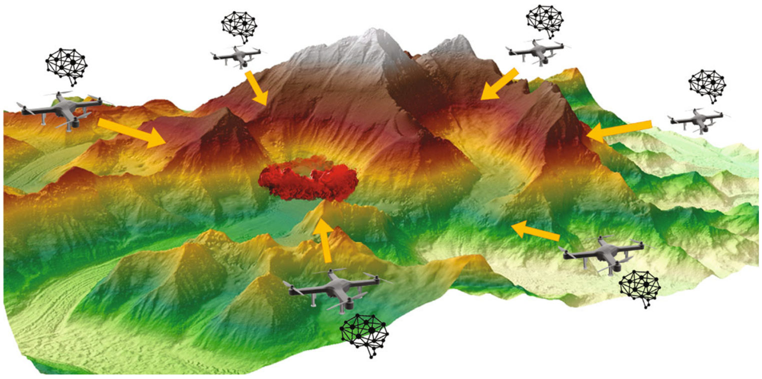


Figure 1: The application concept of the proposed system.

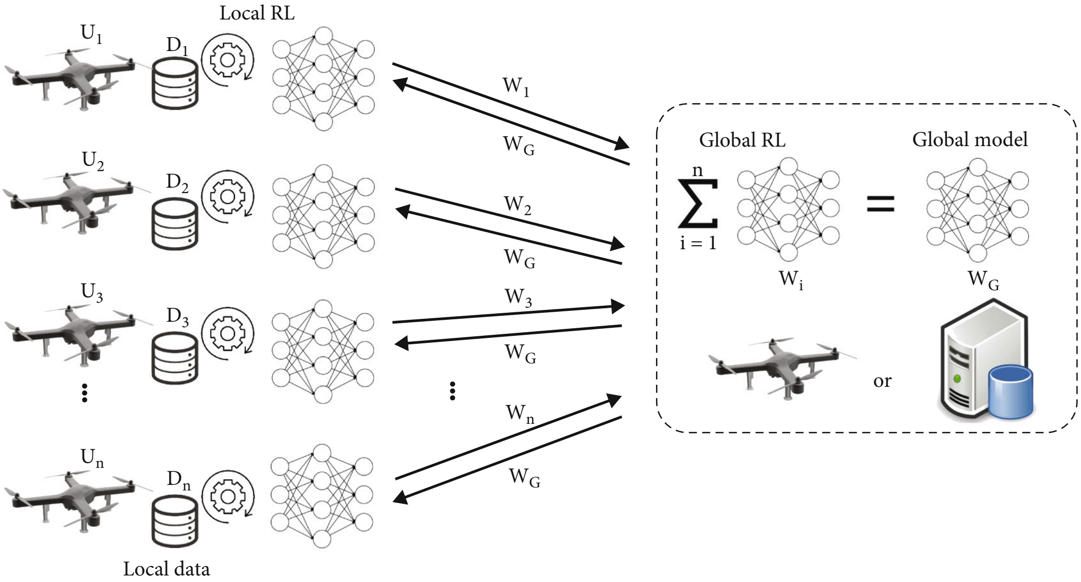


Figure 2: The overall operations of FRL in the proposed system.

proposed system. Then, we give descriptions of our FRL system, the RL algorithm used in the system, and the environment constructed for learning. After that, we describe the system implementation.

4.1. System Concept. We propose the FRL-based UAV swarm system for aerial remote sensing. As we mentioned before, to show the application of our proposed system, we take gas sensing as an application example, and Figure 1 shows the proposed system's application concept. Initially, a UAV swarm consisting of multiple UAVs is arranged in an area where a gas source is expected to exist. In this situation, the mission of the UAV swarm is to find the origin of the gas source, marked as red smoke in the figure, with avoiding collisions not only between UAVs but also with other obstacles, such as tall trees. The UAVs continually move without any predetermined guidance or programmed function. At the same time, they repeatedly perform local learning based on their own actions and data collected from gas sensors and ranging sensors, such as LiDAR or radar. After that, the UAVs share only their locally trained models with each other periodically. During the mission, the UAVs repeat such moving, learning, and occasional sharing to build SI.

4.2. Federated Reinforcement Learning System. In the proposed system, the neural network of UAVs is trained using FRL, and Figure 2 shows the overall learning procedures in the system. To explain the FRL operations in our system, we assumes n UAVs, $U_1, \ldots, U_n$ with their own data $D_1, \ldots, D_n$. The proposed FRL scheme includes the following main steps. First, a server (a ground control system) or a header UAV in our system sends initial global models to all UAVs, and each UAV trains their own local model using local information including states, actions, and rewards. We will describe the detailed explanation about the learning algorithm in Section 4.3. The UAVs send the local model parameters, $W_1, \ldots, W_n$, back to the server, and then, the server aggregates the model parameters into the global model as follows:

$$W_G = \sum_{n=1}^{n} W_i. \tag{1}$$

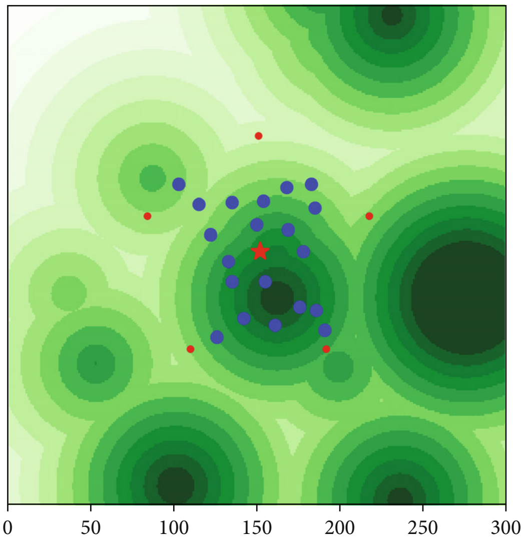


Figure 3: An example of map used for the proposed FRL system.

The global model's parameters, W_G, are distributed back to the UAVs and the above procedures are repeated until the global model is sufficiently trained.

4.3. Reinforcement Learning Algorithm. This subsection describes the RL algorithm used in our proposed system. The PPO algorithm is based on the actor-critic concept and utilizes two separate networks [23]. The actor network determines the agent's optimal behavior, whereas the critic network evaluates policies and trains the actor using rewards. The PPO algorithm was inspired by the TRPO algorithm [24], and the PPO algorithm provides a more direct approach to implementing and coordinating tasks for learning. Compared to TRPO, PPO is also known to provide simpler and better performance in many applications in IIoT [35]. The UAV system prefers algorithms requiring a small amount of computation, so PPO is suitable for various tasks performed by UAVs [5]. In fact, many studies used PPO as the RL algorithm for UAV systems, and many results have shown that PPO is superior to other algorithms in various aspects [5]. For these reasons, we chose PPO as the RL algorithm of our FRL system.

We describe the detailed explanation about the learning algorithm for the proposed system with reference to [16, 23, 34]. In training, an agent observes a state, s_t, in the environment at time step t. The actor model, π_θ, with its model parameters, θ, is used to determine an action, a_t, to be taken in the given state, s_t. The agent takes the selected action, the state transitions to the new state, s_{t+1}, and the agent gets the reward, r_{t+1}. For every time step, the agent stores the trajectory segment, $<s_t, \cdot a_t, \cdot r_{t+1}, \cdot s_{t+1} >$ in the trajectory memory. The critic model, V_μ, with its model parameters, μ, evaluates whether the action led the agent to a better state, and the critic model's feedback is used to optimize the actor model. Whenever a determined number of steps proceed, based on the PPO algorithm, the gradients for the optimization of the actor and critic models are calculated using the trajectory segments in the trajectory memory. The objective function, L^{PG}, in a general policy gradient RL is as follows:

```
Input: sensing information, distance information
Output: state
state = zeros($n_{state}$)
Calculating state values regarding sensing:
$s_{sum} \longleftarrow 0$
for each sensing value, $s$, in sensing value set, $S$ do.
    $s_{sum} \longleftarrow s_{sum} + s$
end for.
$s_{average} \longleftarrow s_{sum}/n_{sensor.}$
$s_{max} \longleftarrow 0$
for $i$, $s$ in enumerate($S$) do
    state[$i$] $\longleftarrow s - s_{average}$
    if $s_{max} <$ abs(state[$i$]) then $s_{max} \longleftarrow$ abs(state[$i$])
end for
for $i$ in range($n_{sensor}$) do
    state[$i$] $\longleftarrow$ state[$i$]/$s_{max}$
Calculating state values regarding distance:
$o \longleftarrow$ NearestObj ($O$)
dist $\longleftarrow$ CalDist ($o$)
    if dist $\leq$ size$_{uav}$ then state[$-4$] $\longleftarrow -1$
else state[$-4$] $\longleftarrow 1$
$\vec{o} \longleftarrow$ CalVec ($o$)
state[$-3$] $\longleftarrow \vec{o}_x$
state[$-2$] $\longleftarrow \vec{o}_y$
state[$-1$] $\longleftarrow \vec{o}_z$
return state
```

Algorithm 1: Algorithm for getting the state.

Table 1: Variables used for getting the state.

Notation	Description
state	Set of state values
n_{state}	Number of state values in state
S	Set of sensing values
s	Each sensing value in S
s_{sum}	Sum of sensing values in S
$s_{average}$	Average of sensing values in S
s_{max}	The maximum of the absolute values of S
n_{sensor}	Number of sensors attached onto the UAV
O	Set of nearby objects
o	The nearest object
dist	The distance to o
$\vec{o}$	Normalized vector to o

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta (a_t | s_t) \hat{A}_t \right], \tag{2}$$

where $\hat{\mathbb{E}}_t[\cdots]$ means the empirical average over a finite batch of samples and $\hat{A}_t$ is an estimator of the advantage function at timestep t. Utilizing the generalized advantage

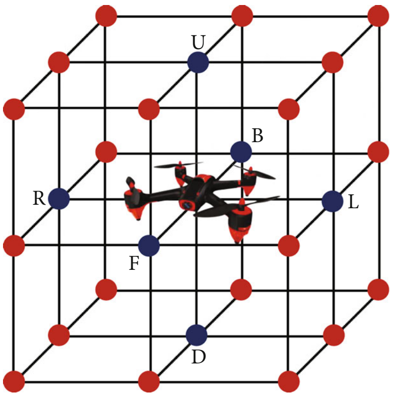


Figure 4: The sensors' position and possible movement actions of UAV in the proposed FRL.

```
Input: action, sensing information, distance information
Output: reward
reward ⟵ 0
  Detecting a collision with any other objects:
$o$ ⟵ NearestObj ($O$)
if $o \neq t$ then
    $\text{dist}_{\text{obj}}$ ⟵ CalDist ($o$) then
    if $\text{dist}_{\text{obj}} \leq \text{size}_{\text{uav}}$ then
        reward ⟵ −2
        return reward
    end if
end if
  When the agent reaches the target:
if $o == t$ then
    $\text{dist}_t$ ⟵ CalDist ($t$)
    if $\text{dist}_t \leq \text{th}_{\text{succ}}$ then
        if action == 'staying' then
            reward ⟵ 1
            return reward
        end if
    end if
end if
  Calculating the reward in the other cases:
$\vec{t}_x$ ⟵ $s_{\text{right}} - s_{\text{left}}$
$\vec{t}_y$ ⟵ $s_{\text{front}} - s_{\text{back}}$
$\vec{t}_z$ ⟵ $s_{\text{up}} - s_{\text{down}}$
$\vec{t}$ ⟵ $v_t/\|v_t\|$
$\vec{a}$ ⟵ NorVec(action)
reward ⟵ InnerProd($\vec{d}$, $\vec{a}$) abs(reward) < $\text{val}_{\text{clip}}$
if reward ≥ 0 then
    if reward ⟵ $\text{val}_{\text{clip}}$
    else reward ⟵ $-\text{val}_{\text{clip}}$
end if
return reward
```

Algorithm 2: Algorithm for determining the reward value.

Table 2: Variables used for determining the reward.

Notation	Description
O	Set of nearby objects
o	Nearest object
t	Target object
dist_{obj}	Distance to o
size_{uav}	Radius size of UAV
dist_t	Distance to the target
th_{succ}	Threshold of distance to the target where the agent is deemed to arrive at the target
$\vec{t}$	Normalized vector to target
$\vec{a}$	Normalized vector of action
val_{clip}	Clip value for determining reward

Table 3: Hyperparameters and values used for learning.

Hyperparameter	Value
Actor network dimension	16∗256∗256∗256∗5
Critic network dimension	16∗256∗256∗256∗5
Minibatch size	5
Number of epochs	4
Learning rate	0.0003
Horizon value	20
Generalized advantage estimator	0.95
Discount factor gamma	0.99
Clipping parameter	0.2
Value function coefficient	0.5
Optimizer algorithm	Adam

estimator (GAE) [36], $\widehat{A}_t$ can be calculated as follows:

$$\widehat{A}_t = \delta_t^V + (\gamma\lambda)\delta_{t+1}^V + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}^V, \tag{3}$$

where γ is the discount factor ($\gamma \in [0, 1]$), λ is the GAE parameter ($\lambda \in [0, 1]$), T is the size of mini-batch samples, and $\delta_t^V = r_t + \gamma V_\mu(s_{t+1}) - V_\mu(s_t)$. The objective function, L^V, is as follows:

$$L^V(\mu) = \widehat{\mathbb{E}}_t\left[\left|\widehat{V}_\mu^{\text{target}}(s_t) - V_\mu(s_t)\right|\right], \tag{4}$$

where $\widehat{V}_\mu^{\text{target}}$ is the target value of time-difference error (TD-error), and $\widehat{V}_\mu^{\text{target}}(s_t) = r_{t+1} + \gamma V_\mu(s_{t+1})$. Using a stochastic gradient descent (SGD) algorithm (i.e., Adam optimization [37]), the parameters of V_μ are updated as follows:

$$\mu = \mu - \eta_\mu \nabla L^V(\mu), \tag{5}$$

where η_μ is the learning rate for the critic model optimization.

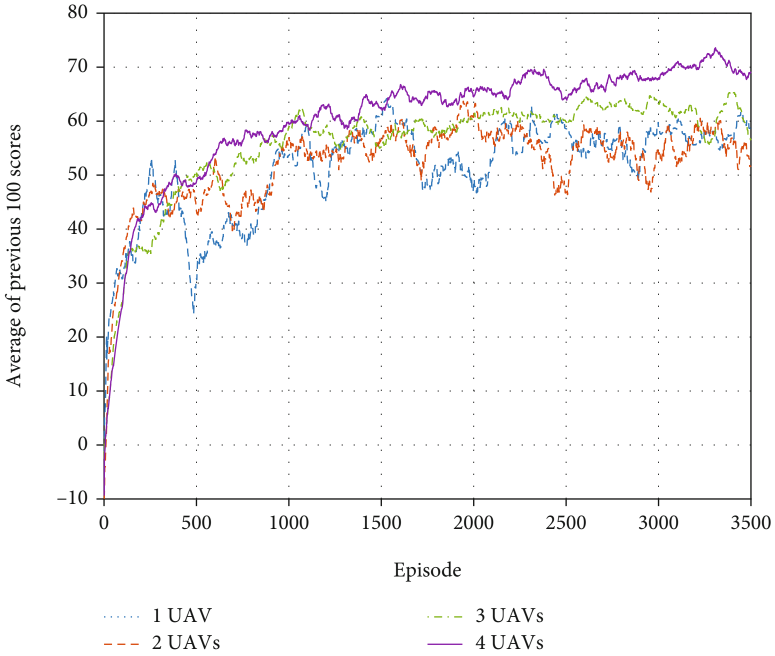


Figure 5: The average of the score values as the episode goes by.

In the actor model of TRPO, the importance sampling is used to obtain the expectation of samples gathered from the old policy, $\pi_{\theta_{\text{old}}}$, under the new policy, π_θ. The TRPO algorithm maximizes the surrogate objective function, L^{CPI}, presented in

$$L^{\text{CPI}}(\theta) = \hat{\mathrm{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t = \hat{\mathrm{E}}_t\left[R_t(\theta)\hat{A}_t\right]\right] \tag{6}$$

where CPI refers to conservative policy iteration [38] and $R_t(\theta)$ denotes the probability ratio. The TRPO algorithm optimizes L^{CPI} subject to the constraint on the amount of the policy update as follows:

$$\hat{\mathrm{E}}_t\left[KL\left[\pi_{\theta_{\text{old}}}(\cdot\,|s_t), \pi_\theta(\cdot\,|s_t)\right]\right] \leq \delta, \tag{7}$$

where KL refers to the Kullback-Leibler divergence [39]. As we explained before, the PPO algorithm was inspired by the TRPO algorithm, and the objective function of PPO, L^{CLIP}, is as follows:

$$L^{\text{CLIP}}(\theta) = \hat{\mathrm{E}}_t\left[\min\left(R_t(\theta), \text{clip}(R_t(\theta), 1-\varepsilon, 1+\varepsilon)\right)\hat{A}_t\right], \tag{8}$$

where ε is the clipping parameter. The parameters of π_θ are updated by the SGD algorithm with the gradient, ∇L^{CLIP}, as follows:

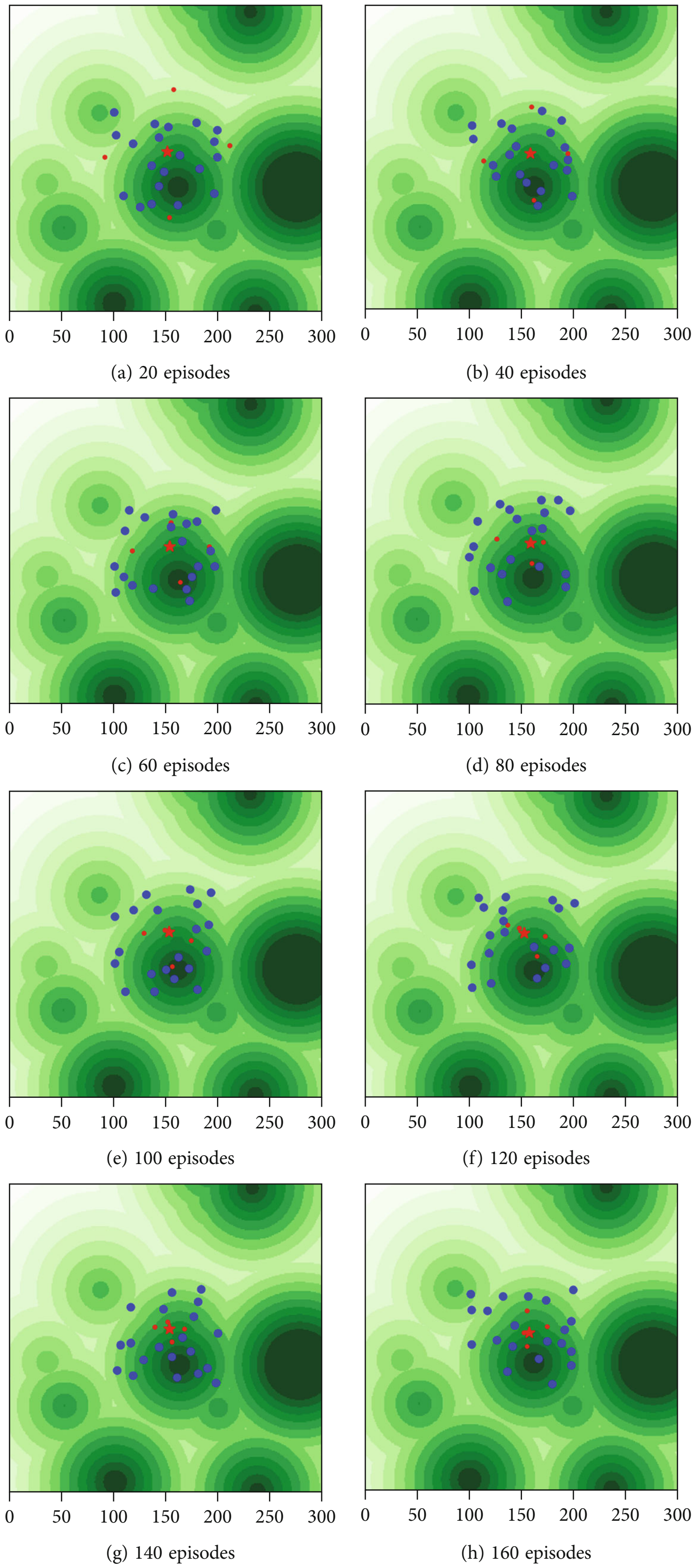
$$\theta = \theta - \eta_\theta \nabla L^{\text{CLIP}}(\theta), \tag{9}$$

where η_θ is the learning rate for the actor model optimization.

Using the above algorithm, each agent in our system performs RL repeatedly, and the agents send the updated model parameters to the server periodically as we explained in Section 4.2.

4.4. Environment. The agents continually interact with the environment while performing learning, so it is important to construct an appropriate environment for proper learning. We constructed the environment for agents to perform learning well to accomplish the mission described in Section 4.1. This subsection provides a detailed description of the environment, especially about map, state, action, and reward.

4.4.1. Map. Figure 3 shows an example of map which is used for FRL of the proposed system. In the map, green circle lines mean contour lines. In other words, an area marked as darker green means a higher area. Red and blue dots represent UAVs and obstacles, respectively. The red star in the middle means the gas source that the UAVs should find. We set the map to change every a certain period so that the UAVs can experience various environments. At each change, both the position of the obstacles and the height of the terrain change. The UAVs are initially placed evenly between UAVs outside a certain range from the gas source since it is efficient and reasonable to spread them as much as possible. The obstacle is assumed to be a very tall object, such as a transmission tower, so that the UAVs cannot avoid the obstacle by flying higher but should move horizontally to avoid the obstacle. Considering collisions not only with obstacles but also between UAVs, if a UAV gets closer to another object than a certain distance, it is considered as a collision.



(a) 20 episodes

(b) 40 episodes

(c) 60 episodes

(d) 80 episodes

(e) 100 episodes

(f) 120 episodes

(g) 140 episodes

(h) 160 episodes

Figure 6: Final position of UAVs as the episode goes by.

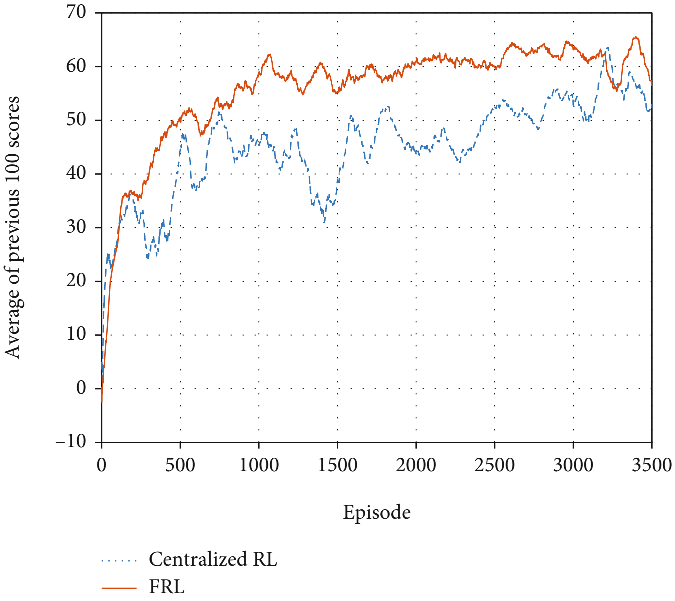


Figure 7: The learning performance comparison between centralized RL- and FRL-based systems.

4.4.2. State. In order for an agent to take an appropriate action, the state should consist of appropriate values. Algorithm 1 shows the pseudocode for getting the state, and Table 1 lists the variables used in Algorithm 1. The state is composed of two value sets, one set regarding sensing values and the other set containing distance information, so the algorithm for obtaining state is also composed of two parts.

Lines 2 to 14 in Algorithm 1 are relevant to calculating state values regarding sensing. The UAV has multiple sensors, gas sensors in our application example scenario, and blue dots in Figure 4 present the position of sensors attached onto the UAV. Each sensor continuously collects sensor data. In real-world environments, there is always noise in sensor values obtained from real sensors. Therefore, in order to consider noise in a real environment, we added different Gaussian noise to sensor values. We will give the detailed explanations about the noise values and the performance evaluation considering the sensor noise in Section 5.3. Using the sensor data, the agent finds the sum of the collected values and calculates the average of them. After that, the agent subtracts the mean value from each sensor value, and in this process, the agent finds and memorizes the maximum absolute value of the result values. The agent performs normalization using this maximum value, and the agent takes these final results as values of the state's first set.

Lines 15 to 22 in Algorithm 1 are relevant to the state's second value set, state values regarding distance information. First, the agent finds the nearest object, a UAV or an obstacle, from the agent, and then calculates the distance to the object. If the distance is smaller than the size of the UAV, there is a collision, so -1 is stored in the state, and if not, 1 is stored. After that, the agent calculates the normalized vector directed towards the nearest object, and the values of x, y, and z axes of the vector are stored in the state.

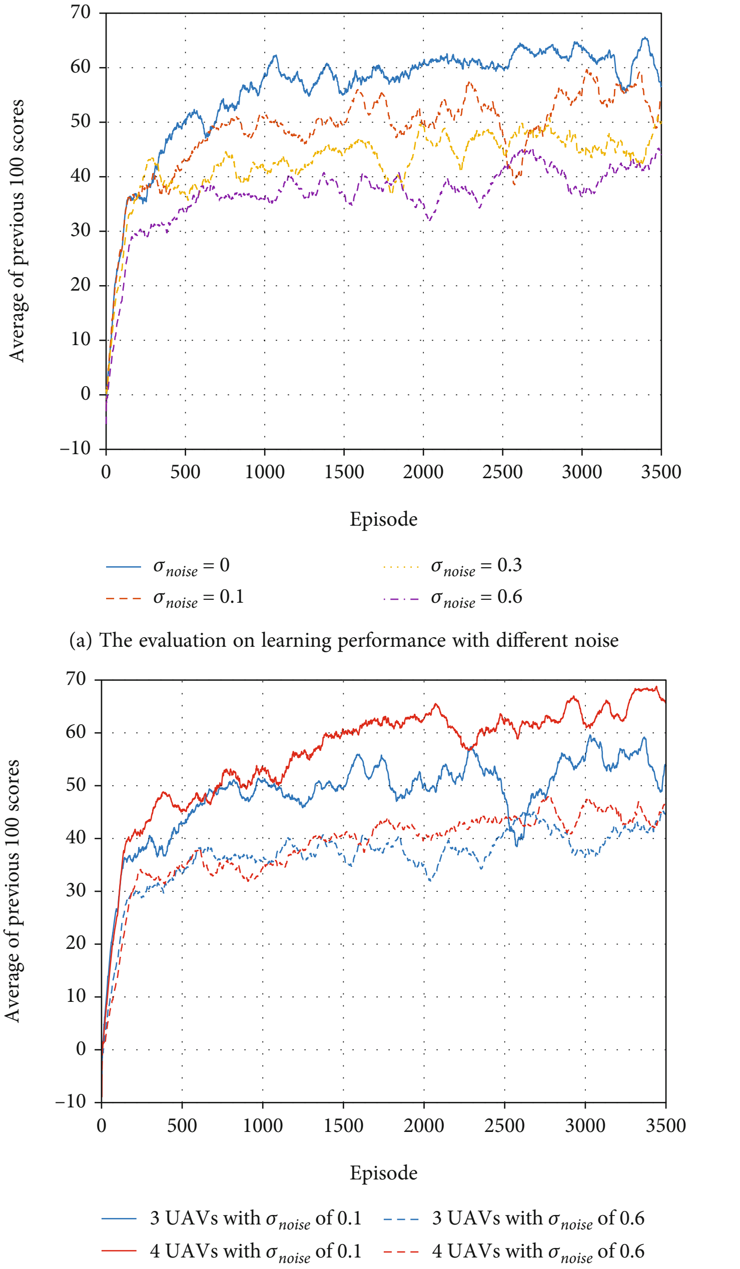
4.4.3. Action. Figure 4 shows the movement actions that the UAV can choose. UAVs in real world can move in more diverse directions, but in order to reduce the complexity of learning, we assumed that UAVs can perform only 27 actions, moving in 26 directions and staying. Red and blue dots in the figure indicate the 26 directions, and blue dots also show the position of sensors attached onto the UAV as explained before.

4.4.4. Reward. An appropriate reward should be given for an agent to perform well in learning. Algorithm 2 shows the detailed process of determining the reward value, and Table 2 lists the variables used in Algorithm 2.

The UAV should not collide with other UAVs or obstacles while moving. Lines 2 to 9 in Algorithm 2 are relevant to detecting a collision with any other objects. First, the agent finds the nearest object among nearby objects. If the nearest object is not the target, the agent calculates the distance to the object. If the distance is less than the radius of UAV, in other words, if a collision occurs, the reward is set to -2 to train the agent not to do such action causing the collision in the future.

If the agent arrives at the target, it is reasonable for the agent to be located there without moving, and lines 10 to 18 in Algorithm 2 are relevant to this case. Firstly, the agent calculates the distance to the target. When the distance is shorter than the determined distance for judging whether the agent arrives at the target, if the agent takes the action of staying there, the agent gains 1 as compensation.

In the other cases, the agent calculates the reward, and lines 19 to 28 in Algorithm 2 are relevant to these cases. The principle of determining the reward is that the better the agent moves in the direction of the target, the larger the reward the agent receives. The shorter the distance



(a) The evaluation on learning performance with different noise

(b) The impact of packet loss depending on the number of UAVs participating in learning

Figure 8: The performance evaluation considering sensor noise.

between the sensor and the target is, the larger or smaller the sensing value is, depending on the characteristics of sensors. In the case of gas sensor, the shorter the distance, the larger the sensing value [40]. Therefore, a larger sensing value means that the sensor is closer to the target. The agent obtains a normalized vector, on x, y, and z axes, directed toward the target using values of sensors marked with blue circles in Figure 4. After that, the agent calculates the normalized vector for the action and obtains the inner product of the two vectors. If the absolute value of the reward is too small, learning may not be performed well, so the reward is adjusted based on the clipping value.

4.5. Implementation. As explained in Section 4.3, we used PPO as the RL algorithm, and we implemented the RL model of the proposed system by using the PyTorch library [41] with reference to [42]. Table 3 shows hyperparameters used in the algorithm. By adding FL to the RL model, we constructed the FRL model with reference to [43]. We implemented the FRL system on Ubuntu 20.04 LTS using a desktop with AMD Ryzen™ 7 5800X and 32 GB RAM. For faster learning, we trained the learning model by using NVIDIA's compute unified device architecture (CUDA) on the NVIDIA GeForce RTX 3070 8 GB GDDR6 PCI Express 4.0 graphic card. In addition, we constructed a map,

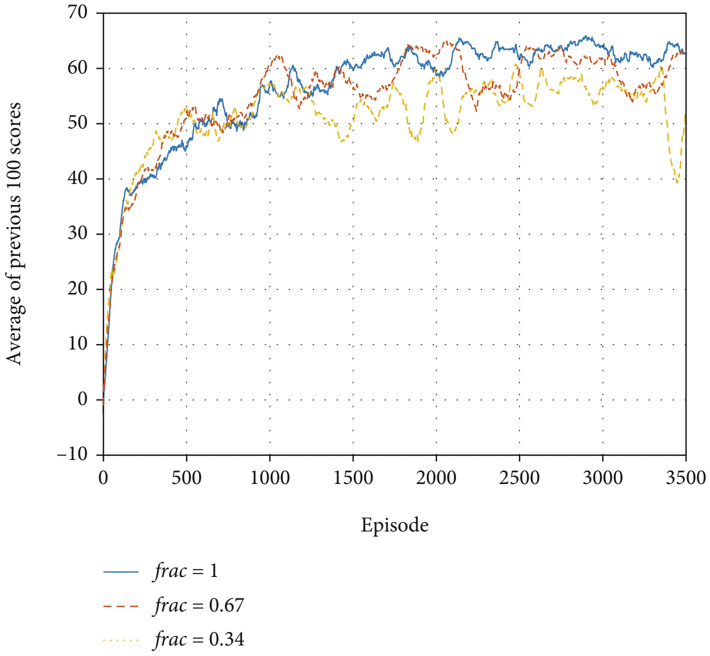


Figure 9: The learning performance depending on the participation ratio.

explained in Section 4.4.1, referring to the 2D Gaussian grid map introduced in [44].

5. Performance Evaluation

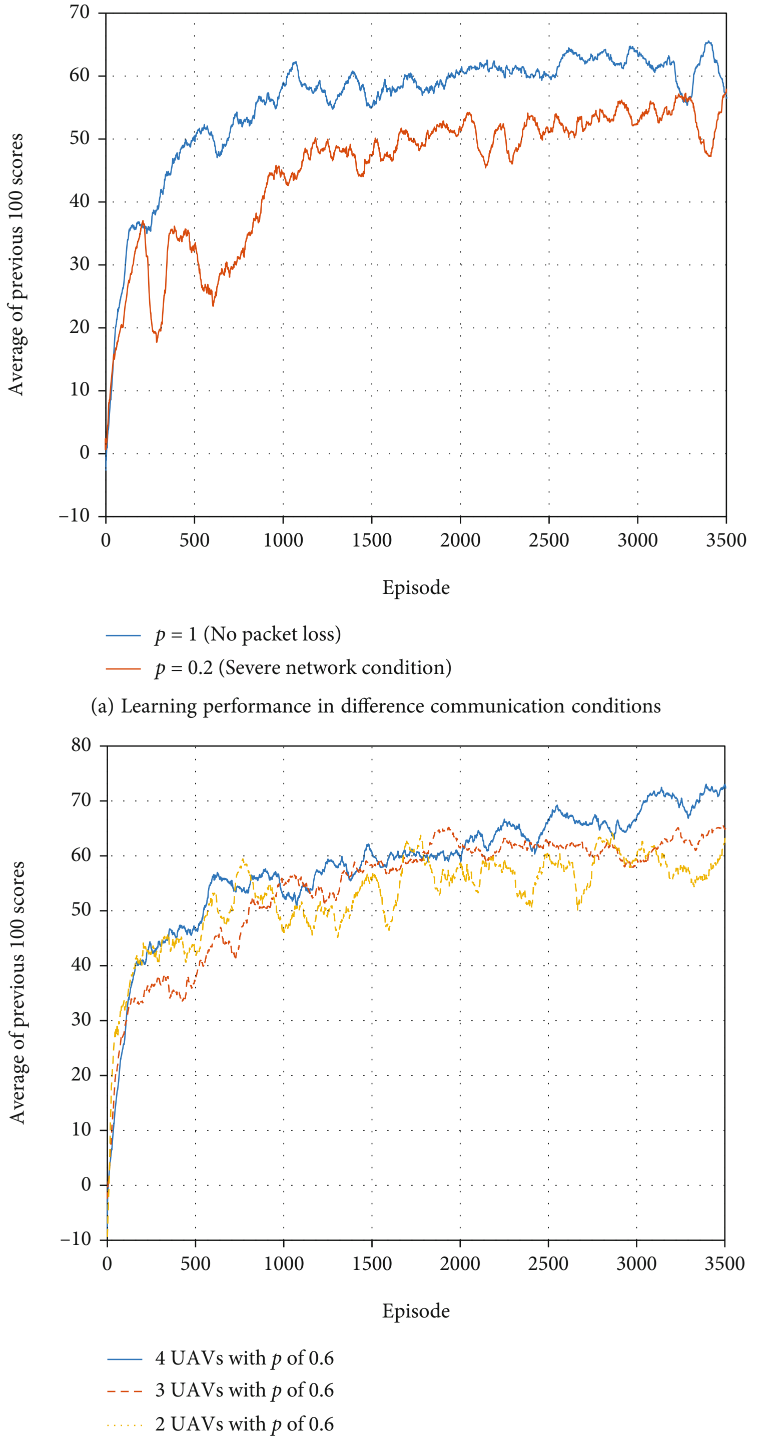
In this section, we explain the various experiments and evaluation results. We first explain the performance evaluation of the proposed FRL system and then show the result of performance comparison between RL- and FRL-based systems. After that, we describe diverse evaluations considering various factors, such as sensor noise, participation ratio, packet loss, and duplication sending.

5.1. Evaluation on Learning Performance. An episode is a unit of learning, and each episode ends after a determined number of steps proceed. To evaluate the learning performance, we recorded the sum of the reward values gained by the agent in the episode as the score of the episode, and we investigated the sum of scores from the last 100 episodes. We conducted the evaluation by varying the number of agents, and the four lines with different colors in Figure 5 show the results. As shown in the figure, the average of the score values increases as the episode goes by, which means that the agent performed the mission well as the learning was repeated. The average value continues to increase up to about 3000 episodes and reaches the saturation point. In terms of the number of agents, the result shows that the more agents participate in learning, the better the learning performance is. In other words, the average score increases higher and the range of fluctuation is smaller in cases where the more agents participate in learning. This is because the more UAVs learn together, the more diverse experiences are collected, which not only makes learning better but also causes unbiased learning to be performed. However, it is not easy for many UAVs to continuously send raw data to centralized entities, which can lead to massive communication overhead and energy inefficiency of UAV systems. Thus, our FRL-based system is suited for UAV swarms because FRL has an advantage in terms of communication resources in that it does not need to repeatedly share the raw data for learning.

As shown in Figure 5, the learning progresses rapidly in the early stage. To analyze this in more detail, Figure 6 shows the final positions of UAVs every 20 episodes. In the figure, after only 20 episodes, in other words, when the sufficient learning was not performed, the UAVs could not find the target. However, as the episode went by, the more UAVs moved closer to the target, which means that the learning was performed well.

5.2. Performance Comparison between RL- and FRL-Based Systems. In existing RL approaches, it is common to collect data and perform learning in a centralized manner. In UAV systems, it is not easy to continuously send all raw data to the central entity in real time, so the learning can be performed by transferring data to the server after the flight of all UAVs is over. We compared the results of learnings performed using such centralized RL-based method and our FRL-based method. As shown in Figure 7, the FRL-based method performed learning better and reached the saturation point faster than the centralized RL-based method. The reason for this result is that the FRL-based method does not require raw data transmission so that learnings can be performed more frequently, resulting that agents can be trained faster and more stably.

5.3. Learning Performance considering Noise. As explained in Section 4.4.2, there is always noise in sensor values obtained from real sensors. Therefore, to evaluate the performance



(a) Learning performance in difference communication conditions

(b) The impact of packet loss depending on the number of UAVs participating in learning

Figure 10: The performance evaluation considering packet loss.

considering noise, we analyzed the learning performance by adding a different Gaussian noise of $\mathcal{N}(\mu, \sigma^2)$ to sensor values. We performed FRL with 3 agents by using the zero mean and different variance values from 0 to 0.6 with reference to the values obtained from real gas sensors [40]. As shown in Figure 8(a), the higher the noise, the lower the learning performance. However, even when there was noise, a certain level of learning was sufficiently performed. Thus, this result shows that the proposed FRL system can be utilized in a real environment with noise.

As shown in the result above, the noise degrades the learning performance. However, as the number of UAVs increases, the more experience the UAVs have and share, which mitigates the degradation caused by noise. As shown

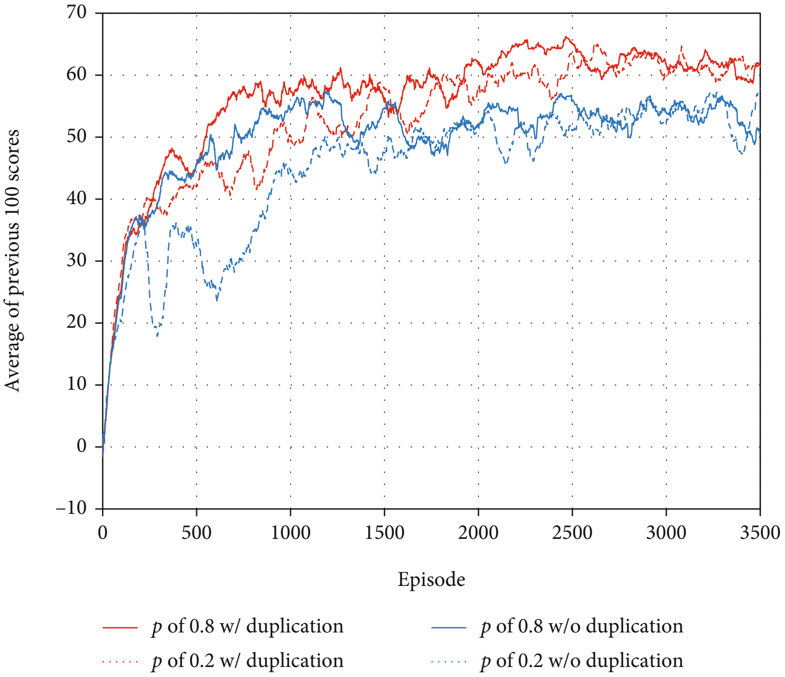


Figure 11: The learning performance depending on the use of the duplication sending technique.

in Figure 8(b), when there was little noise, the more UAVs participated in learning together, the better the learning performance. Similarly, even when there was severe noise, the fluctuation was smaller when more UAVs participated in learning although the overall learning performance was relatively low. In summary, using the FRL-based system, the more UAVs participate in building SI, the impact of noise can be alleviated as well as the overall performance can be improved.

5.4. Performance Evaluation considering Participation Ratio. In real situations, all devices may not be always able to participate in learning on all rounds due to diverse causes, such as the situation of devices, communication, and network problems. Therefore, in the actual FL, the ratio of devices participating in learning is determined, some of the devices are chosen every round according to the participation ratio, and the selected devices participate in learning. Thus, we evaluated the performance by changing the participation ratio in the learning, and Figure 9 shows the result. Naturally, the higher the participation ratio, the more stable and better performance, but for this to occur, many devices should participate in the learning of every round. As shown in the figure, even when 0.67 was selected as the participation ratio, there is the little degradation in performance compared to the case with the participation ratio of 1. Thus, this result shows that it is possible to obtain not only efficient learning but also acceptable performance by using the proper participation ratio.

5.5. Performance Evaluation considering Packet Loss. In UAV systems, due to the high mobility of UAV and continuous changes in network topology, wireless data communications are frequently unstable, which can lead to packet loss. When packet loss occurs or communication situation is poor, some of trained local models cannot be transferred. In consideration of this situation, we evaluated the performance by changing the packet loss probability, and Figure 10 shows the result. Figure 10(a) shows learning performance in cases where there was no packet loss in a stable communication situation and where a lot of packet losses occurred due to poor network condition. In the case of severe network condition, since the packet loss occurred frequently, the trained models could not be transferred well, so learning was performed unstably at the beginning of learning. However, as shown in Figure 10(b), the learning performance can be improved if more agents participate in learning even when the communication situation is unstable. In conclusion, if FRL is utilized in a UAV system composed of a number of UAVs, it is possible to perform learning even in poor communication situations.

5.6. Performance Evaluation considering Duplication Sending. These days, it is not difficult for UAVs to transmit packets through multiple paths by leveraging multiple interfaces simultaneously. In our previous work [45], to improve the reliability and stability of controlling UAVs, we proposed a scheme that selectively duplicates only important packets and then transfers the originals and copies of them through different paths. Such technique and other similar ones can increase the success rate of transmitting trained models, which in turn improves learning performance. Figure 11 shows the results of learning performance depending on the use of the technique when the packet transmission probability is 0.8 or 0.2. As shown in the result, we can get better learning performance when using the duplication sending technique. This means that the reliable communication and network in UAV systems are critical for improving the FRL system's learning performance to build SI.

6. Conclusion

Nowadays, UAVs are widely used in various fields of IIoT due to the many advantages of the UAVs. In order to carry out today's complicated and complex missions, it is more appropriate and efficient to use multiple UAVs together, so many people utilize UAVs in the form of swarm. However, it is not easy to control multiple UAVs from a distance at the same time. Thus, UAVs are required to have the high autonomy, and AI is the most promising technique to provide the intelligence to UAVs. However, to secure SI using existing techniques, raw data should be continuously exchanged between UAVs, which is not suitable for UAV systems operating on unstable networks. Motivated by the fact described above, in this paper, we proposed the novel FRL-based UAV swarm system for aerial remote sensing. The proposed system utilizes RL to ensure the high autonomy of UAVs, and moreover, the system combines FL with RL to construct the more reliable and robust SI for UAV swarms. Through the performance evaluations, we showed that the proposed system outperformed the existing centralized RL-based system. Furthermore, we conducted various analyses considering the diverse factors, such as sensor noise, participation ratio, packet loss, and duplication sending, and the results proved that our proposed system is more suited for UAV swarms from a variety of perspectives.

We have several directions as future work. We will implement our FRL algorithm on UAV devices and apply the proposed system to UAV systems in a real environment. In order to do this, we will construct the more complex state and devise the more sophisticated reward algorithm. In addition, we plan to elaborate our system to include additional techniques, such as more efficient model exchange and adaptive participation ratio, which results in the better SI development.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request and with permission of funders.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1092939).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," 2013, arXiv preprint arXiv: 1312.5602.

[2] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[4] O. Vinyals, I. Babuschkin, W. M. Czarnecki et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[5] A. T. Azar, A. Koubaa, N. Ali Mohamed et al., "Drone deep reinforcement learning: a review," *Electronics*, vol. 10, no. 9, 2021.

[6] B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-efficient learning of deep networks from decentralized data," *Artificial intelligence and statistics. PMLR*, pp. 1273–1282, 2017.

[7] H. X. Pham, H. M. La, D. Feil-Seifer, and L. V. Nguyen, "Autonomous uav navigation using reinforcement learning," 2018, arXiv preprint arXiv: 1801.05086.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of UAV flight formation communication with priority: a deep reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 892–903, 2020.

[10] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, arXiv preprint arXiv: 1509.02971.

[11] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064–29074, 2020.

[12] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.

[13] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53841–53849, 2020.

[14] P. Chhikara, R. Tekchandani, N. Kumar, M. Guizani, and M. M. Hassan, "Federated learning and autonomous UAVs for hazardous zone detection and AQI prediction in IoT environment," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15456–15467, 2021.

[15] U. Awada, J. Zhang, S. Chen, and S. Li, *Air-to-Air Collaborative Learning: A Multi-Task Orchestration in Federated Aerial Computing*, 2021.

[16] H. K. Lim, J. B. Kim, J. S. Heo, and Y. H. Han, "Federated reinforcement learning for training control policies on multiple IoT devices," *Sensors*, vol. 20, no. 5, p. 1359, 2020.

[17] M. K. Abdel-Aziz, C. Perfecto, S. Samarakoon, M. Bennis, and W. Saad, "Vehicular cooperative perception through action branching and federated reinforcement learning," arXiv preprint arXiv: 2012.03414 2020.

[18] M. Xu, J. Peng, B. Gupta et al., "Multi-agent federated reinforcement learning for secure incentive mechanism in

intelligent cyber-physical systems," *IEEE Internet of Things Journal*, 2021.

[19] Z. Xue, P. Zhou, Z. Xu et al., "A resource-constrained and privacy-preserving edge-computing-enabled clinical decision system: a federated reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9122–9138, 2021.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2018.

[21] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," 2017, arXiv Preprint arXiv: 1708.05866.

[22] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 10, 2009.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, arXiv preprint arXiv: 1707.06347.

[24] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, PMLR, 2015.

[25] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[26] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a GPU," 2016, arXiv preprint arXiv: 1611.06256.

[27] J. Konecny, H. B. McMahan, D. Ramage, and P. Richtarik, "Federated optimization: distributed machine learning for on-device intelligence," 2016, arXiv preprint arXiv: 1610.02527.

[28] P. M. F. Mammen, "Learning: opportunities and challenges," 2021, arXiv Preprint arXiv: 2101.05428.

[29] H. B. McMahan, E. Moore, and D. Ramage, "Federated learning of deep networks using model averaging," 2016, arXiv preprint arXiv: 1602.05629.

[30] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, article 106775, 2021.

[31] H. H. Zhuo, W. Feng, Q. Xu, Q. Yang, and Y. Lin, "Federated reinforcement learning," 2019, arXiv preprint arXiv: 1901.08277.

[32] J. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine, "Self-consistent trajectory autoencoder: hierarchical reinforcement learning with trajectory embeddings," in *International Conference on Machine Learning*, pp. 1009–1018, PMLR, 2018.

[33] X. Liang, Y. Liu, T. Chen, M. Liu, and Q. Yang, "Federated transfer reinforcement learning for autonomous driving," 2019, arXiv preprint arXiv: 1910.06001.

[34] B. Liu, L. Wang, and M. Liu, "Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4555–4562, 2019.

[35] M. Chen, H. K. Lam, Q. Shi, and B. Xiao, "Reinforcement learning-based control of nonlinear systems using Lyapunov stability concept and fuzzy reward scheme," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, pp. 2059–2063, 2019.

[36] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, arXiv preprint arXiv: 1506.02438.

[37] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, arXiv preprint arXiv: 1412.6980.

[38] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *In Proc. 19th International Conference on Machine Learning*, Citeseer, 2002.

[39] S. Kullback, *Information Theory and Statistics*, Courier Corporation, 1997.

[40] S. Lee, S. Park, and H. Kim, "Gas detection-based swarming: deterministic approach and deep reinforcement learning approach," *The Journal of Supercomputing in submission.*.

[41] A. Paszke, S. Gross, F. Massa et al., "Pytorch: an imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.

[42] P. Tabor, *Youtube-Code-Repository*, 2020, https://github.com/philtabor/Youtube-Code-Repository/tree/master/ReinforcementLearning/PolicyGradient/PPO/torch.

[43] A. R. Jadhav, *Federated-Learning (PyTorch)*, 2021, https://github.com/AshwinRJ/Federated-Learning-PyTorch.

[44] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "PythonRobotics: a Python code collection of robotics algorithms," 2018, arXiv preprint arXiv:1808.10703.

[45] W. Lee, J. Y. Lee, H. Joo, and H. Kim, "An MPTCP-based transmission scheme for improving the control stability of unmanned aerial vehicles," *Sensors*, vol. 21, no. 8, p. 2791, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4197178, 12 pages
https://doi.org/10.1155/2022/4197178

Research Article

Research on Underwater Target Recognition Technology Based on Neural Network

Zhiguang Guan, Chenglong Hou, Siqi Zhou, and Ziyi Guo

Shandong Provincial Engineering Lab of Traffic Construction Equipment and Intelligent Control, Shandong Jiaotong University, Jinan, 250357 Shandong, China

Correspondence should be addressed to Zhiguang Guan; guanzhiguang@sdjtu.edu.cn

Received 9 December 2021; Revised 25 January 2022; Accepted 23 March 2022; Published 14 April 2022

Academic Editor: Shunmei Meng

At present, the underwater environment required by the seafood aquaculture industry is very bad, and the fishing operation is completed artificially. In this environment, the use of machine fishing instead of artificial fishing is the development trend in the future. By comparing the characteristics of different algorithms, the multiscale Retinex algorithm (autoMSRCR) is selected to deal with image color skew, blur, atomization, and other problems. Labelimg software is used to annotate underwater targets in the image and make data sets. Of these, 20% are used as test sets, 70% as training sets, and 10% as verification sets. The target detection network of You Only Look Once Version4 (YOLOv4) based on convolutional neural networks (CNN) is adopted in this paper. The main feature extraction network adopts CSPDarknet53 structure, and the feature fusion network adopts SSP, and PANet network carries out sampling and convolution operations. The prediction output of extracted features is carried out through YoloHead network. After training the recognition model of the training sets, the detection effect is obtained by testing the data of the test sets. The identification accuracy of sea cucumber and sea urchin is 90.8% and 87.76%, respectively. Experiments show that the target detection network model can accurately identify the specified underwater organisms in the underwater environment.

1. Introduction

In China, offshore seafood aquaculture, sea cucumbers, and sea urchins grow at the bottom of the seawater. In particular, sea cucumbers and sea urchins live on reefs of 12-13 meters underwater or artificial reefs. When the temperature is lower than 0°C or higher than 20°C, sea cucumbers will enter the seabed or dormancy. The depth and the presence of rocks make fishing operations extremely difficult. At present, divers can only go into the sea to fish seafood. "Humans cannot work underwater for a long time because of their body structure and the way they breathe. Fishing divers are prone to decompression sickness and rheumatoid arthritis" [1]. Hence, the high fees paid to divers result in huge fishing costs. At present, there is no suitable robot to replace artificial underwater operations in seafood aquaculture. The fishing robot can replace the artificial long-term dangerous operation and reduce the risk and cost of fishing [2]. "With the rapid development of digital image processing and computer technology, neural network technology is becoming more and more mature in the field of computer. The neural network model has the advantages of self-learning ability, strong adaptability, and high robustness and is especially suitable for classification and recognition problems" [3–5].

Currently, popular target detection algorithms based on deep learning can be divided into two categories: one-stage network and two-stage network [6]. One-stage network is much faster in detection speed than two-stage network, but lower in detection accuracy.

Two-stage network firstly carries out region proposal (RP) for the input images and then classifies them through CNN. Representative algorithms include R-CNN, SPP-Net,

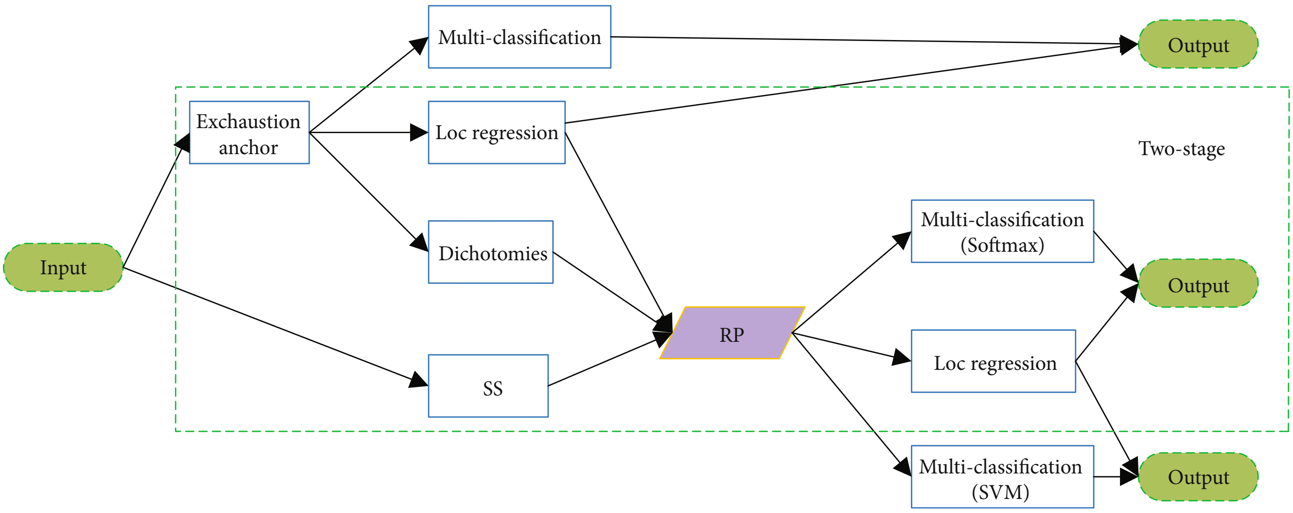


Figure 1: Two-stage network structure.

Fast R-CNN, Faster R-CNN, and R-FCN [7, 8]. The network structure is shown in Figure 1.

One-stage network input does not use the RP generation prior-box but directly extracts features from CNN to predict object classification and location. Representative algorithms include OverFeat, YOLOv1, YOLOv2, YOLOv3, YOLOv4, SSD, and RetinaNet [9]. The network structure is shown in Figure 2.

The target detection network based on deep learning is applied to the underwater robot. The main problem is the accurate recognition of seafood in a shallow sea environment. The specific steps are as follows:

(1) Manufacturing of data sets: according to the changes of underwater environment, the image is preprocessed to enhance the feature information and distinguish the training sets, test sets, and verification sets. Labelimg is used to mark the data for network training

(2) Targeting recognition: building the framework of YOLOv4, inputting the processed training sets, and getting the trained model

(3) Adjusting model parameters: using the test sets to test, then adjusting the learning rate and the network model of data processing way, and letting the model accuracy and speed realize optimality

(4) Realizing recognition: building a platform on the existing underwater fishing robot to realize the combination of algorithm and model and verifying the performance and reliability of the algorithm

2. Data Collection and Production

The data sets and network structure are the main factors influencing on the detection accuracy in the target detection algorithm based on neural network. Having large data sets is the premise of training and optimizing high performance network model.

2.1. Data Collection and Processing. Since it is underwater real-time detection, the authenticity of the image will directly affect the robustness of the training model. Underwater image sets come from visual competitions, most of which are underwater aquaculture environments of sea cucumbers and sea urchins. The original data sets of the competition have four species. Two species of sea cucumbers and sea urchins are adopted and carried out manual labeling.

The complex physical environment changes of ocean make the underwater images by ocean optical and visual imaging system degraded greatly. There are some serious problems such as image color fatigue, low contrast, and blurred details. The severely degraded underwater images lack effective data and information for target recognition, so the recognition difficulty increases [10]. Therefore, it is necessary to preprocess the image using image enhancement technology and carry out feature extraction in CNN.

Retinex is based on the theory that the color of an object is determined by its reflection of light, not by the absolute value of reflected light intensity. The color of an object is not affected by the illumination uniformity and has consistency.

Multiscale Retinex algorithm formula is as follows:

$$\log R_i(x, y) = \log \sum_{k=1}^{k} W_k\{\log I_i(x, y) - \log [F_k(x, y) * I_i(x, y)]\}, \tag{1}$$

where K stands for the number of scales, which is usually 3. When $K = 1$, it is the single-scale Retinex algorithm. W_k stands for weighting coefficient. F_k stands for filter function. I_i stands for original input images. The i stands for RGB color channel.

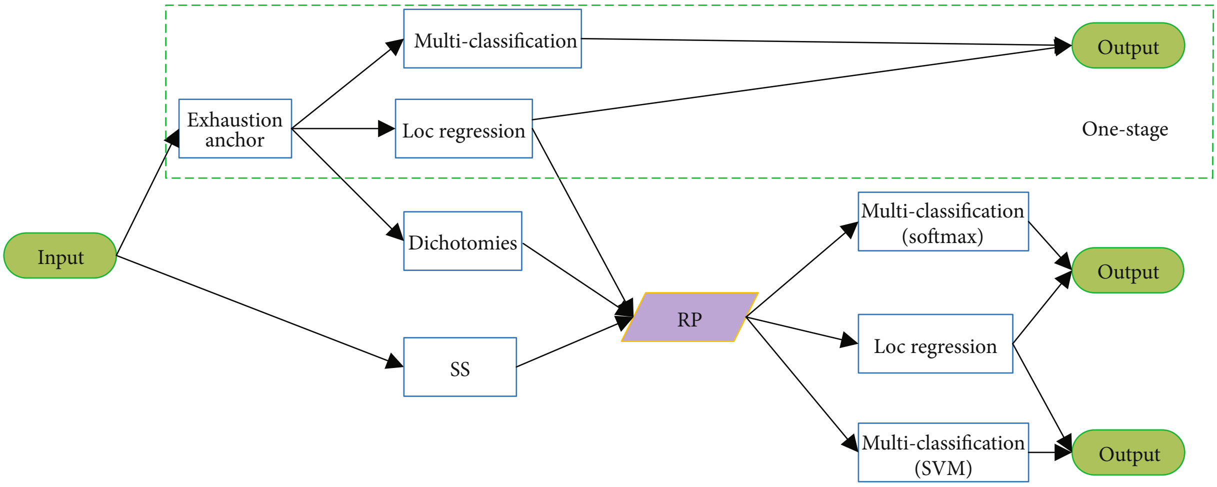


Figure 2: One-stage network structure.

Multiscale Rentinex algorithm with color balance is an improved algorithm of MSR. In MSR image, due to the increase of noise, local detail color distortion will be caused, and the overall visual effect will be worse. To solve this problem, color restoration factor is added to adjust the weight between the three-color channels in the original image. Thus, the information in the relatively dark area can be color adjusted to eliminate the defect of image color distortion [11]. The formula of MSRCR with color balance is as follows [12]:

$$R_{MSRCRi}(x, y) = C_i(x, y)R_{MSRi}(x, y), \tag{2}$$

where $C_i(x, y)$ stands for color recovery factor of channel i.

The MSRCR algorithm with automatic color levels removes the largest and smallest part of the MSRCR processing results according to a certain percentage. Then, the remaining middle part is quantified to 0-255, which can restore the image better than MSRCR [13]. The image preprocessing effect comparison is shown in Figure 3.

The image processed by autoMSRCR has the most obvious contrast, the better defogging effect, the most obvious local details, and the better effect. Therefore, autoMSRCR is selected as the image preprocessing algorithm.

2.2. Data Set Making. The research object are underwater image data, which are difficult to collect. The data sets in the online vision competition are adopted in the paper, which contains four species, namely, sea cucumber, sea urchin, coral reef, and seaweed. Only sea cucumbers and sea urchins are selected in the experiment. 1200 images are manually selected as the original data sets, but such small data sets will cause problems such as low precision and overfitting of the model in the training process. Therefore, data augmentation is used to expand the number of images in the data sets.

Mosaic data augmentation method is used in this paper. Mosaic date augmentation method takes four images and splices them together to form a new image. The process is to read four pictures randomly and then reverse the four pictures, zoom, gamut, and other changes. And according to the position of the top left, top right, bottom left, and bottom right, an image is spliced. And then, combine it into an image, which is shown in Figure 4.

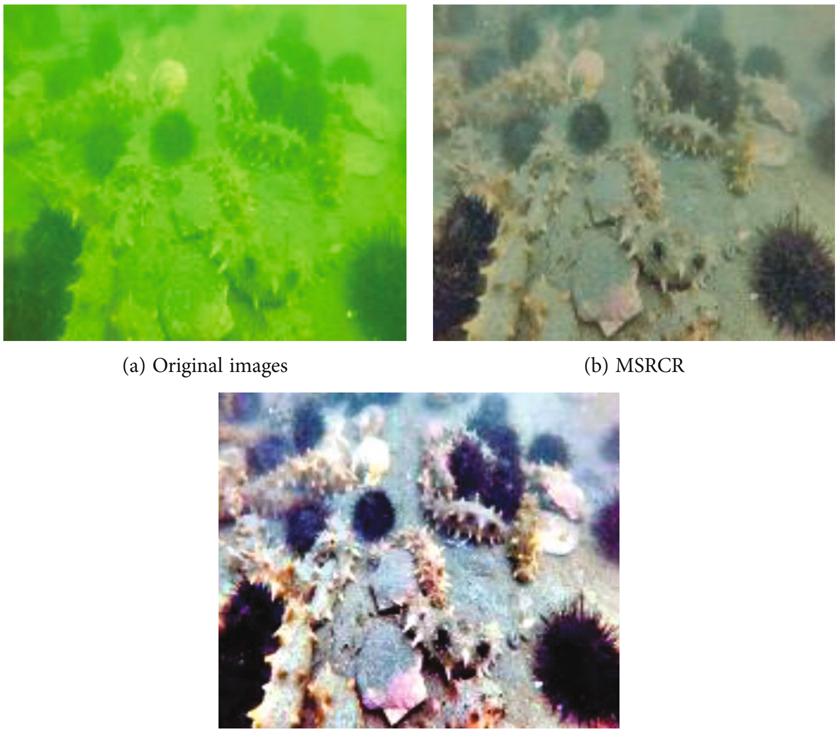
Object detection based on deep learning is a kind of supervised learning [14]. The feature of the target is extracted directly through the convolutional network for learning. Therefore, the position of the target in the image needs to be manually labeled, and the labeled information is converted into VOC2007 format. Labelimg is used in this experiment to select the target. The labeling annotation process is shown in Figure 5.

The annotated data sets are divided into training sets, verification sets, and test sets in proportion. To ensure a wide range of data coverage, the division principle is random. The ratio adopted in this experiment is as follows: training sets : verification sets : test sets is 7 : 1 : 2, that is 840 images of training sets (excluding augmented images), 120 images of verification sets, and 240 images of test sets.

3. Target Detection Algorithm Based on YOLOv4 Network

YOLOv4 network mainly consists of three parts: backbone network, neck network, and head network [15]. CSPDarknet53 is used as the backbone feature extraction network. Mish function is used as the activation function. SSP and PANet are used as the neck network, which can effectively separate the most significant features of context. In the head part, the YOLO Head is adopted as the feature utilization part to extract and convolved. The anchor frame system of RCNN is introduced to greatly improve the map. There is no regional sampling, so it performs well on the global information.

3.1. YOLOv4 Network Structure. The backbone network of YOLOv4 adopts CSPDarknet53 network structure with large residual edges. The image size used in this experiment is 416 × 416. It is input into the CSPDarknet53 network, and



(a) Original images

(b) MSRCR

(c) autoMSRCR

Figure 3: Image preprocessing effect comparison.

channels are added by using a single convolutional layer of Mish function. Among them, the Mish function is the activation function, which has the advantages of smooth gradient descent good effect. Meanwhile, the unboundedness of Mish function can avoid the saturation problem, which is used in this experiment. Then, feature extraction is performed through an 11-layer Resblock network with residual structure to generate 52×52 output I. At the same time, the output continues to extract features from the 8-layer Resblock network to produce output II with a size of 26×26. Output II also extracts features from the 4-layer Resblock network to produce output III with a size of 13×13.

In the neck network, the output III generated by the backbone network enters the SPP network structure. Output III is pooled at four different scales, and the pooled nucleus sizes are 13×13, 9×9, 5×5, and 1×1, respectively. Output II and output I are transmitted into PANet network, and the output through SPP structure is also transmitted into PANet network through a connection layer. Features are repeatedly extracted through up- and downsampling pyramid to achieve the best separation effect.

In the head network, YOLOv4 extracts three feature layers transmitted by the neck network and predicts the output through two-layer convolution. The overall network structure of YOLOv4 is shown in Figure 6.

3.2. Target Loss Function. The loss function of YOLOv4 can be divided into three parts: classification loss, confidence loss, and location loss [16]. The CIoU loss function is used in location loss to reflect the deviation between the real frame and the prediction frame, which is added the coverage area, center distance, and aspect ratio based on IoU loss function. The loss function is only calculated for positive samples. The formula are as follows [17]:

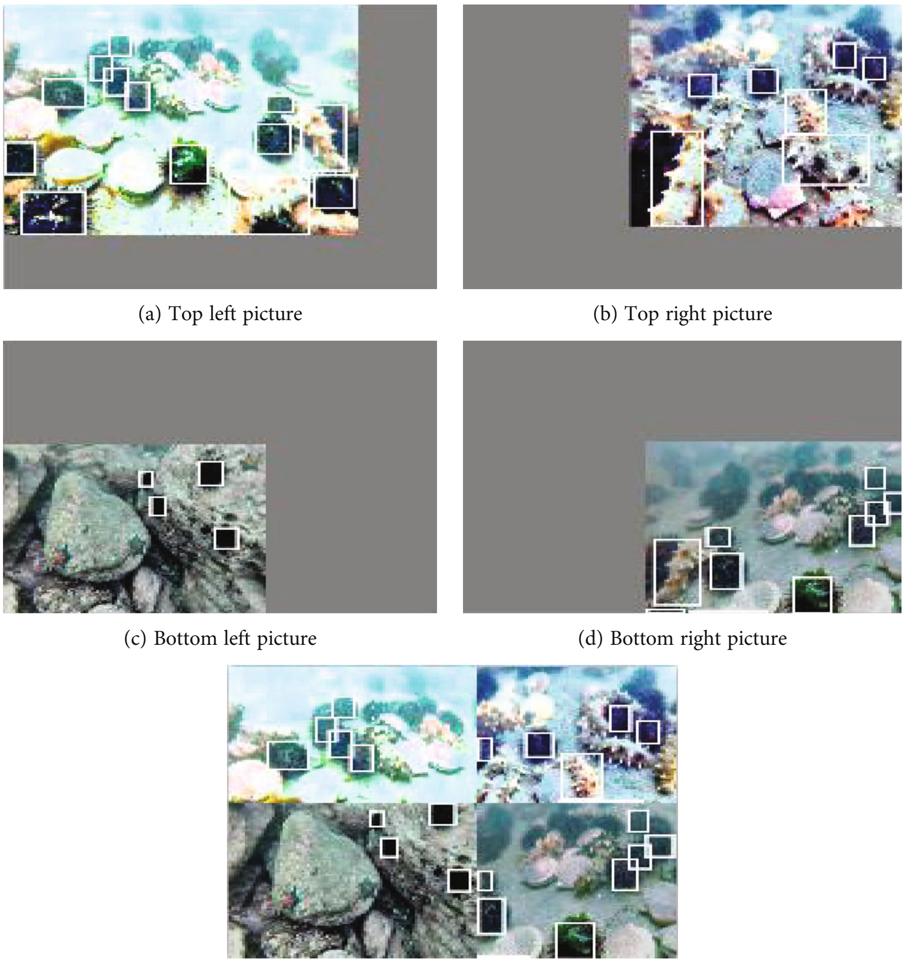
$$\begin{aligned} l_{CIoU} &= 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \partial v, \\ v &= \frac{4}{\pi^2}\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}\right)^2, \end{aligned} \tag{3}$$

where $\rho^2(b, b^{gt})$ stands for Euclidean distance between the real frame and the prediction frame, c stands for the diagonal length of the minimum enclosing rectangle between the real frame and the prediction frame, v stands for distance between the width ratio of the real frame and the prediction frame, if the width and height are similar, then v =0. ∂ stands for item weight. w and w^{gt} and h and h^{gt} stands for the width and height of prediction frame and real frame, respectively.

The classification loss adopts binary cross entropy loss, which is calculated only when the sample is positive.

Confidence loss is divided into two parts, target-oriented loss and target-free loss, which are calculated both in positive and negative samples. It is better as positive sample confidence is closer to 1, or negative sample confidence is closer to 0.

In the training process, through the random gradient descent method and back propagation, the loss value of the loss function is continuously reduced in the iterative training, the learning rate is constantly updated according to the loss value, and the model parameters are constantly



(a) Top left picture

(b) Top right picture

(c) Bottom left picture

(d) Bottom right picture

(e) Splice picture

Figure 4: Image enlargement effect.

Figure 5: Data sets annotation process.

adjusted. The learning rate is also constantly updated according to the loss value, which minimizes the deviation between the prediction frame and the real frame. Continuously improve the network category confidence, so as to achieve optimal network performance.

4. Experimental Training and Result Analysis

By identifying sea cucumber and sea urchin, YOLOv4 network parameters are set according to the above methods, and the model is obtained by training on GPU.

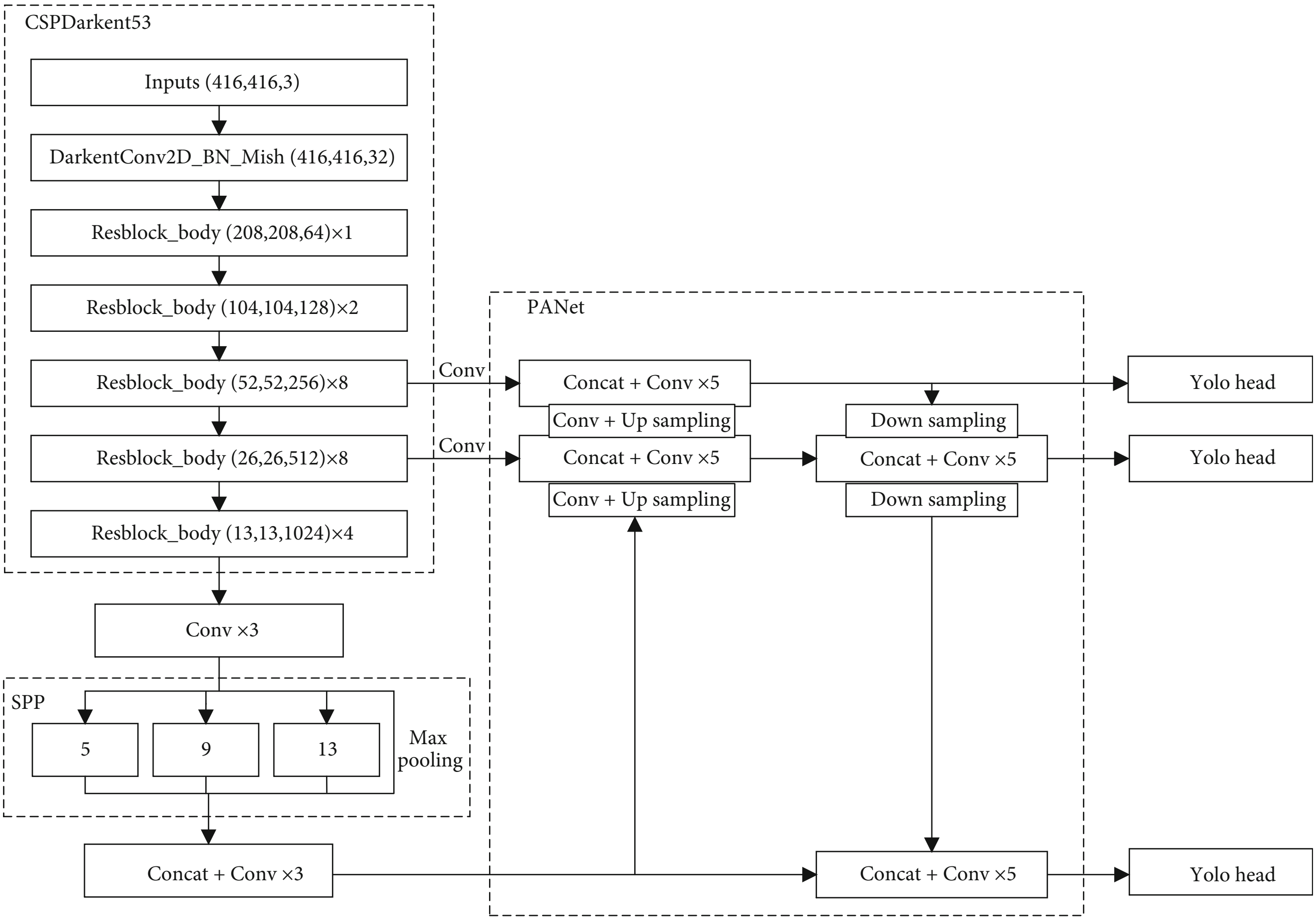


Figure 6: YOLOv4 network structure.

4.1. Experimental Platform and Parameter Design. Pytorch1.7 deep learning tool based on python3.8 is adopted, which supports a variety of classical neural network models. The system environment is Linux Ubuntu18.04. NIVIDIA CUDA11.0 version is adopted in GPU computing framework, and the corresponding neural network acceleration library cudnn is configured. The overall configuration is shown in Table 1.

The image autoMSRCR algorithm processing and TensorboardX and Tqdm library network model training process are realized using OpenCV-python library.

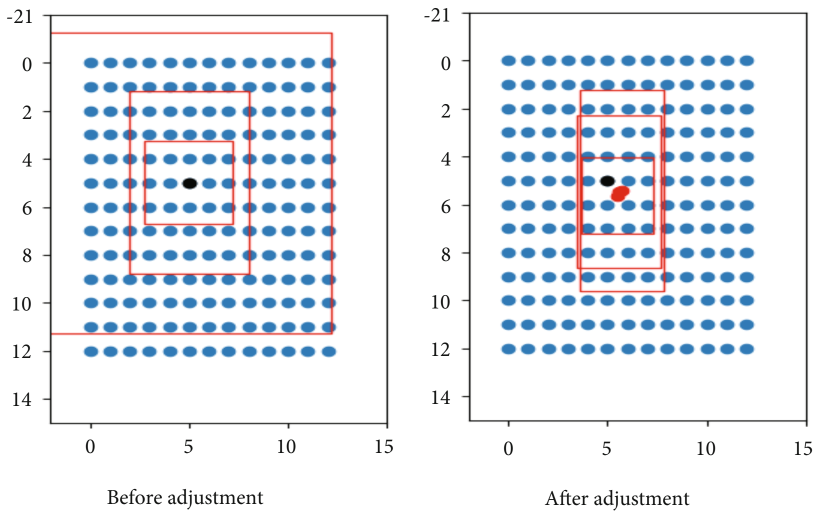
Many parameters are involved in the initialization process of network training. The selection of training parameters and training strategies has influence on the convergence result and detection performance of the network. The main parameters are as follows: training batch (Batch_size), total iteration times (Epoch), frozen iteration times (Freeze_epoch), thawed iteration times (Thaw_epoch), optimizer, initial learning rate (Base_LR), learning rate change strategy (Cosine_lr), and weight attenuation (Weight_decay).

The training batch is the number of samples selected in every training. The Batch_size directly affects the optimization degree and learning speed of the model. By setting Batch_size, GPU utilization is improved, and training time is reduced. The larger the Batch_size is, the more accurate the gradient calculation will be. Meanwhile, the number of iterations should be increased. The smaller Batch_size is, the less accurate the gradient calculation will be and the more obvious the oscillation will be.

Table 1: Experimental environment configuration.

Project	Parameter
Operating system	Linux Ubuntu18.04
CPU	Intel(R) Xeon(R) Gold 6130 CPU
GPU	Tesla V100-32GB
Video driver	CUDA 11.0
Software environment	OpenCV 3.4.1.15 Python3.8
Deep learning framework	Pytorch 1.7

In YOLOv4 network, the loss value of the model is calculated in the forward propagation process, and the gradient is calculated in the back propagation process. The selection of optimization algorithm will directly affect the training speed and accuracy of the model. Adaptive moment estimation is adopted to calculate and update the adaptive learning rate of each parameter. The learning rate determines the learning degree of each iteration and the updating speed of weights in



(a) Before adjustment

(b) After adjustment

Figure 7: Comparison before and after adjustment of the prior-box.

Table 2: Parameters setting table.

Parameter	Numerical value	Parameter	Numerical value
Batch_size	32	Optimizer	Adam
Epoch	500	Cosine_lr	TRUE
Freeze_epoch	100	lr	0.01
Thaw_epoch	400	Weight_decay	0

the whole training process [18, 19]. When the learning rate setting is too large, it is easy to cause overfitting, while if the learning rate is too small, it is easy to produce slow convergence rate and poor recognition effect after model training.

4.2. Training Methods. The training of neural network is to transmit the image data to the network for calculation and reverse update the weight parameters of each layer of network. The network can accurately extract and detect the target features, calibrate the position of the target object, and output the processed image.

In YOLO Head, there are 13 × 13, 26 × 26, and 52 × 52 different size outputs, and the image is converted into a corresponding number of grids. 3 prior-boxes are generated at each grid point. By sigmoid function, the prediction results are normalized so that the center point of the prior-box is in the grid and the size of the prior-box is adjusted. The comparison before and after adjustment of the prior-box is shown in Figure 7.

In order to quickly get the accurate position of the prior-box, the k-means clustering algorithm is used to precalculate the prior-box. The k-means clustering algorithm is a clustering algorithm based on statistics, which can quickly obtain the size of clustering center and prior-box without machine learning [20]. 9 clustering centers are divided and the clustering standard is IoU [21]. The central coordinates after clustering are as follows:

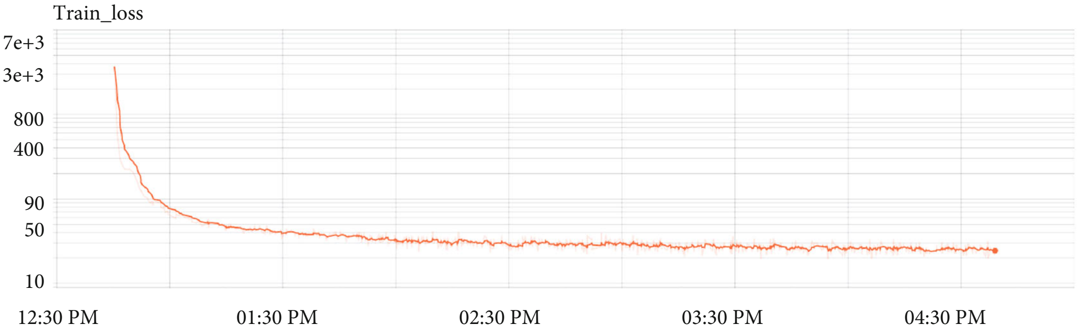
```
[20.8 19.41333333]
[31.2 31.89333333]
[34.08888889 59.57530864]
[39.86666667 133.5308642]
[46.8 45.19506173]
[50.26666667 71.90123457]
[62.97777778 97.58024691]
[88.97777778 140.72098765]
[92.44444444 68.81975309]
```

The method of freezing training is first adopted and then thawing training in model training. The principle is to freeze the weight parameters of common parts (such as backbone network) and train the remaining parameters through the weight files obtained in advance. More resources are allocated to the neck and head network for training, and then after a certain number of iterations, the training time and resource utilization are improved.

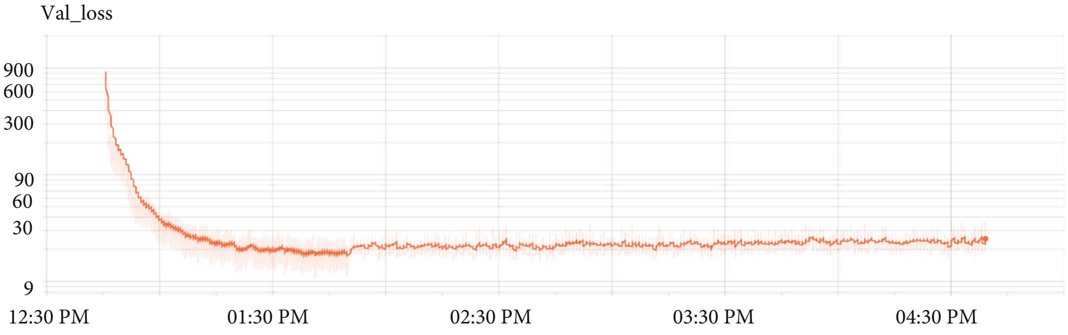
In the process of model optimization training, there may be several local optimal solutions besides the global optimal solution. In the training process of gradient descent algorithm, the model may fall into the local minimum and cannot be optimized again. The study rate improvement strategy is cosine annealing algorithm (hot restart algorithm) to further improve the study rate. The principle is that hot restart is turned on after a few iterations, and local minimum value is skipped by increasing the learning rate of the model and learning continues. When the model approaches the global minimum, the control learning rate becomes smaller to avoid overfitting. When the loss value tends to be stable, the position deviation between the prediction frame and the real frame reaches the minimum. Category confidence is the highest, and network performance is the best.

Network training parameters are shown in Table 2:

4.3. Training Results. According to the above parameter setting, VOC2007 data set pretraining model is used to train the labeled data sets. There are 1000 times of



(a) Training_loss



(b) Validation_loss

Figure 8: Loss variation diagram of training sets and validation sets.

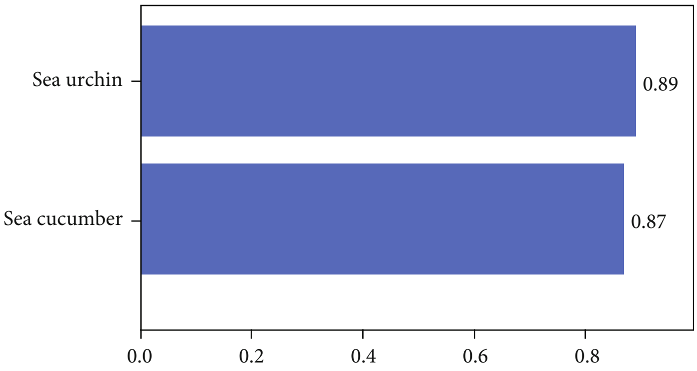


Figure 9: Average precision.

training where 100 iterations are freezing training and 900 iterations are thawing training. TensorboardX is used to record the Train_Loss value and Val_Loss value in the training process, which is shown in Figure 8.

It can be seen from the above figures that the loss value of the model is in a state of oscillation convergence during the training process. The loss value decreases with the increase of training times. The average accuracy of sea cucumber and sea urchin is shown in Figure 9.

The test is carried out on the test sets. After comparing the effects of each model on the test sets, it is found that the thawing training has the highest accuracy when the number of iterations is 805. Therefore, this model is selected as the final underwater sea cucumber and sea urchin recognition model, and the recognition effect is shown in Figure 10, where "green" color represents sea cucumber and "red" color represents sea urchin.

The sea cucumber and sea urchin recognition model is used to test the video at the rate of 11 frames per second on Windows system and 30 frames per second on Linux server. The video viewing rate is 24 frames per second, so running the model on the server can meet the real-time requirement.

The selected sea cucumber and sea urchin recognition model is tested on the test sets. Its accuracy, recall rate, comprehensive index ($F1$), and average accuracy are calculated to evaluate the model. The results are shown in Table 3.

$F1$ is a comprehensive index of precision and recall rate, which can be considered as the average effect. In general, the precision rate and recall rate affect and restrict each other. The calculation formula of precision and recall rate are as

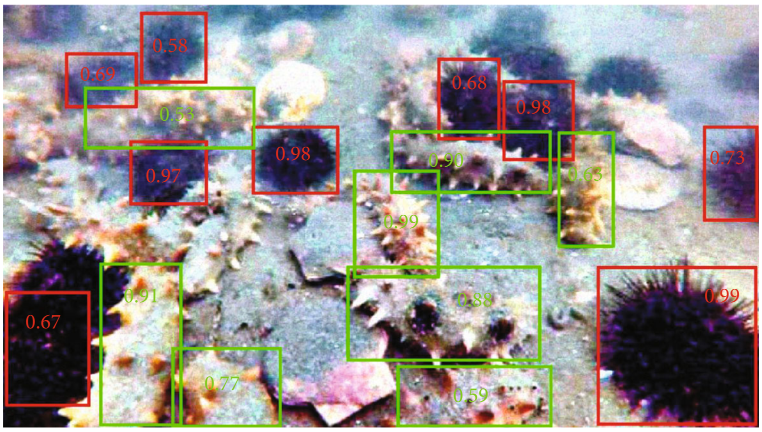


Figure 10: Recognition effect on the test sets.

Table 3: Parameters calculating.

Species	Identification accuracy (%)	Recall rate (%)	$F1$ (%)	Mean accuracy (%)
Sea cucumber	90.8	58.96	71	86.65
Sea urchin	87.76	76.14	82	89.31

follows [17]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

where TP means that the actual situation is the positive example, and the predicted result is the number of positive examples. FP means that the actual situation is the number of negative examples, and the predicted result is the number of positive examples. FN means that the actual results are positive examples, and the predicted results are the number of negative examples.

The formula of $F1$ is as follows:

$$F1 = \frac{2}{(1/\text{PRE}) + (1/\text{REC})}, \tag{5}$$

where PRE stands for precision rate and REC stands for recall rate.

Various data show that the sea cucumber and sea urchin recognition model has a good effect on the test sets and can detect the target regardless of whether the target contour is clear or not. However, in general, the robustness of the model needs to be improved. When the image is blurred, the target object cannot be detected and the training times are less.

4.4. Error Analysis. After YOLOv4 model training and detection, there are two types of errors. One is the error in model training, and another is the error of model detection.

The main error of the model in the course of training is overfitting. The model overfitting the characteristics of the training data performed well in the training sets and predicted and distinguished all the targets almost perfectly. But in the validation sets, the performance is average with poor generalization and low robustness. There is no way to accurate judgement if it is a target with a new sample. The main reason for this problem in model training is that the amount of data is too small. In the training, Train_Loss decreases continuously while Val_Loss increases gradually, as shown in Figure 11.

YOLOv4 can detect sea cucumber and sea urchin targets at different scales and different scenarios, but some detection problems may occur in some environments. The experimental results show that there are two kinds of detection problems in the model test sets: missed and false detection.

Missed detection means that the model misses one or several objects in the image during detection, resulting in incomplete detection. False detection refers to the identification of an object in an image that is not a sea cucumber or sea urchin as a sea cucumber or sea urchin [22]. The reasons for this problem on YOLOv4 are two aspects. One is the image preprocessing using automatic color recovery Retinex algorithm. The characteristics of recognized objects are blurred due to distortion and contrast imbalance after image processing. Furthermore, it is lost in the process of convolutional neural network and feature transfer, which leads to missed or false detection. Another is the inaccuracy of artificial data sets. In the manual annotation data sets, some fuzzy objects observed by human eyes are not marked. Therefore, the model does not learn the fuzzy object during learning, also resulting in missed or false detection.

In view of the problems of the above model, a solution is proposed: firstly, manually relabel the data sets, especially the target in the fuzzy region, so that the model can be learned in the training process. Secondly, the value of Batch_size and iteration epoch should be adjusted appropriately during training to make model learning more

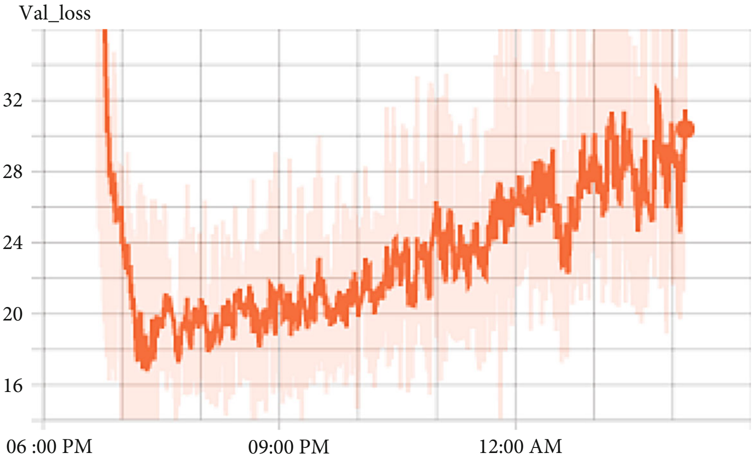
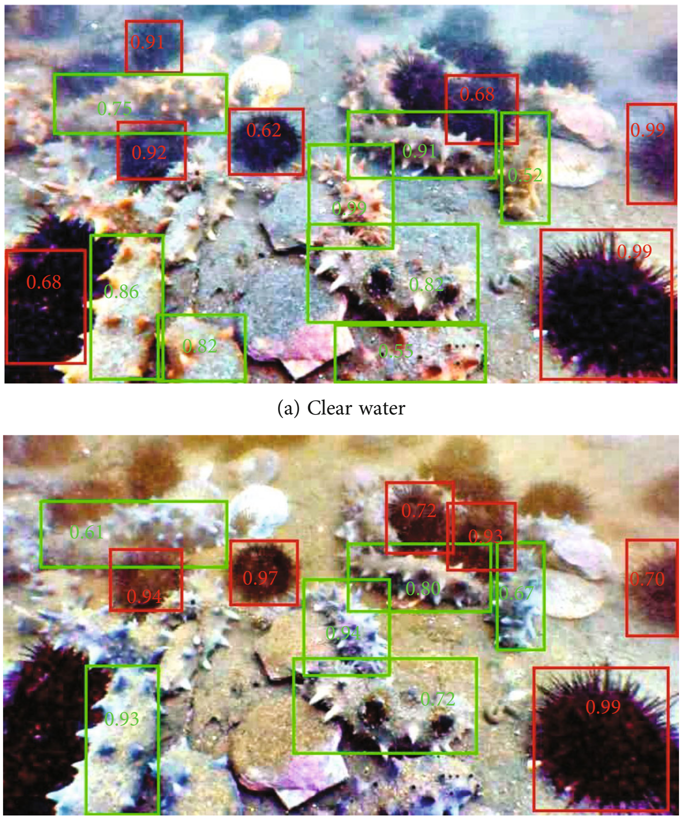


Figure 11: Overfitting loss changes.



(a) Clear water

(b) Muddy water

Figure 12: Recognition effect in clear and muddy environments.

sufficient. Furthermore, optimizing the autoMSRCR algorithm and adding penalty items reduce the distortion after image processing.

The self-developed underwater fishing robot will be arranged for launching experiments. The underwater operation of the robot is controlled by the control panel, and the underwater monitoring image and model detection results are displayed on the screen.

The recognition effect of the underwater robot in clear and muddy environments is shown in Figure 12, where "green" color represents sea cucumber and "red" color represents sea urchin.

The result shows that the detection effect is better in clear water. The image restored by autoMSRCR in the muddy water shows color distortion, which causes poor detection effect. Generally, the model has certain feasibility and reliability. In order to further improve the robustness and application level of the model, muddy water quality and data sets under different illumination conditions can be supplemented to train the model.

5. Conclusion

The YOLOv4 target detection platform is built by Linux Ubuntu 18.04 system, and target detection models of two species of sea cucumber and sea urchin are obtained through training. The main conclusions are as follows:

(1) *k*-means clustering algorithm is adopted to calculate the size and location coordinates of the prediction frame. The YOLOv4 underwater sea cucumber and sea urchin detection model is trained by using the learning rate optimization strategy of cosine annealing. The results of model training are reliable

(2) The data sets adopted this time is manual annotation data sets, and some fuzzy targets are not marked. The detection accuracy is reduced and there are some cases of missed detection

(3) The model can also be further optimized, such as model lightweight, which can improve the model detection rate per second and make the monitoring effect more smoothness

(4) Experiments show that the target detection network model adopted in the paper can accurately identify the specified underwater organisms, such as sea cucumber and sea urchin. But if the number of samples increases, the identification accuracy will also improve

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.

Acknowledgments

This work is supported by the Major Science and Technology Innovation Project of Shandong Province (2019JZZY020703), Science and Technology Support Plan for Youth Innovation in Universities of Shandong Province Colleges and Universities (2019KJB014), and Shandong Jiaotong University "Climbing" Research innovation Team Program (SDJTUC1805).

References

[1] Z. Wang, M. Lin, and C. Ban, "Research on hydrodynamics analysis and double loop integral sliding mode control of 4-joint underwater manipulator," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 728–733, Jeju, South Korea, 2017.

[2] C. Dai, M. Lin, Z. Guan, and Y. Liu, "Aquatic organism recognition using residual network with inner feature and kernel calibration module," *Computers and Electronics in Agriculture*, vol. 190, article 106366, pp. 1–13, 2021.

[3] S. Herzog, C. Tetzlaff, and F. Wrgtter, "Evolving artificial neural networks with feedback," *Neural Networks*, vol. 123, pp. 153–162, 2020.

[4] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral Image Classification with Convolutional Neural Network and Active Learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[5] X. Xiaolong, H. Li, X. Weijie, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: A survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[6] S. K. Patnaik, C. Narendra Babu, and M. Bhave, "Intelligent and adaptive web Data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 279–297, 2021.

[7] T. Zhao, F. Ye, Y. Ming, H. Liu, and S. Basodi, "A Survey on Algorithms for Intelligent Computing and Smart City Applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[8] F. Dai, P. Huang, X. Xiaolong, L. Qi, and M. Khosravi, "Spatio-Temporal Deep Learning Framework for Traffic Speed Forecasting in IoT," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 66–69, 2020.

[9] Z. Guan, Z. Dong, M. Lin, and J. Li, "Mechanical Analysis of Remotely Operated Vehicle," in *2018 4th International Conference on Control, Automation and Robotics*, pp. 446–450, Auckland, New Zealand, 2018.

[10] C. Dai, Z. Guan, and M. Lin, "Single low-light image enhancer using Taylor expansion and fully dynamic convolution," *Signal Processing*, vol. 189, article 108280, pp. 1–14, 2021.

[11] G. Hou, *Research on Underwater Image Enhancement and Object Recognition Algorithms*, Ocean University of China, 2015.

[12] X. Zhang, J. Qin, and Z. Jia, "Study on the Application of Multi-scale Retinex to Image Defogging Algorithm," *Journal of Xichang University (Natural Science Edition)*, vol. 35, no. 3, pp. 60–65, 2021.

[13] C. Dai, M. Lin, Z. Wang, D. Zhang, and Z. Guan, "Color Compensation Based on Bright Channel and Fusion for Underwater Image Enhancement," *Acta Optica Sinica*, vol. 38, no. 11, pp. 1–10, 2018.

[14] X. Xu, Z. Fang, J. Zhang et al., "Edge Content Caching with Deep Spatiotemporal Residual Network for IoV in Smart City," *ACM Transactions on Sensor Networks (TOSN)*, vol. 17, no. 3, pp. 1–33, 2021.

[15] A. L. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection. CVPR, Seattle USA," 2020, https://arxiv.org/abs/2004.10934.

[16] G. Ziyan, H. Huiyan, and H. Ligang, "Gesture Recognition Algorithm and Application Based on Improved YOLOV4," *Journal of North University of China (Natural Science Edition)*, vol. 42, no. 3, pp. 223–231, 2021.

[17] R. Shi, D. Jiang, and Q. Fang, "Aircraft target detection in remote sensing image based on YOLOv4," *Bulletin of Surveying and Mapping*, vol. S1, pp. 134–138, 2021.

[18] X. Xiaolong, Q. Huang, X. Yin, M. Abbasi, M. Khosravi, and L. Qi, "Intelligent Offloading for Collaborative Smart City Services in Edge Computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7919–7927, 2020.

[19] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang, and F. Zhou, "Access Control and Authorization in Smart Homes: A Survey," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 906–917, 2021.

[20] S. Xia, D. Peng, D. Meng et al., “Ball k-Means: Fast Adaptive Clustering with No Bounds,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, Piscataway, NJ, 2020.

[21] Z. H. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: faster and better learning for bounding box regression,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2020.

[22] X. Qiao, *Sea Cucumber Identification in real-time Based on Underwater Machine Vision Technique*, China Agricultural University, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3204407, 9 pages
https://doi.org/10.1155/2022/3204407

Research Article

Enhancement of Unmanned Aerial Vehicle Image with Shadow Removal Based on Optimized Retinex Algorithm

Wenfei Xi,[1,2] **Xiaoqing Zuo**,[1] **and Arun Kumar Sangaiah**[3]

[1]*Faculty of Land Resources Engineering, Kunming University of Science and Technology, Kunming 650093, China*
[2]*Faculty of Geography, Yunnan Normal University, Kunming, 650500 Yunnan, China*
[3]*Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, Taiwan*

Correspondence should be addressed to Xiaoqing Zuo; zxq@kust.edu.cn

Received 3 February 2022; Revised 27 February 2022; Accepted 23 March 2022; Published 14 April 2022

Academic Editor: Varun Menon

Images taken by UAVs have shadows due to terrain factors. The image pixel brightness of the shadow areas is compressed, and the information is deficient, which impacts the recognition of image information and thus limits the subsequent image application. Therefore, the shadow removal of the image is crucial. Image enhancement algorithm is capable of improving the whole and partial contrasts of images, highlighting detail information, and removing shadows. Three classical optical image enhancement algorithms are analyzed. The analysis results show that image would be enhanced excessively after the histogram equalization algorithm to the shadow image enhancement. The pixel brightness are compressed by the Mask homogenization algorithm enhancement and uneven brightness in some areas after the enhancement of the traditional Retinex algorithm. Using the Retinex enhancement algorithm, this study proposes a combination algorithm to remove the shadow of the UAV remote sensing image. The proposed algorithm integrates the Retinex algorithm with the two-dimensional (2D) gamma function to remove the brightness colour of the UAV image, so it is capable of removing the shadow area of the UAV image and correcting the uneven darkness attributed to the image enhancement. The acquired UAV image is used to perform the experiment, and it is integrated with the LOG algorithm to extract the enhanced image features. As indicated by the experimental results, the integrated algorithm is proved with better performance to remove the UAV image shadow. The shadow areas of the features cannot be extracted in the original image, but after using the new algorithm to remove the shadow, the ground edge features can be clearly extracted.

1. Introduction

Due to terrain factors, the image acquired by a UAV flying at low altitudes will have shadow areas. The pixel brightness of the shadow areas will be compressed, and the information will be lost, which will limit the subsequent application of the image. Image enhancement aims to highlight the useful information of images and eliminate or weaken the interference information. After image enhancement, the previous unclear or interesting features required to be highlighted are enhanced, and the image quality is improved, which is more consistent with the requirements of the visual requirements and image analysis and processing [1–3].

The spatial domain method and the frequency domain method are common image enhancement methods. Spatial image enhancement follows direct operation, which processes pixels for enhancement directly. After the enhancement, the gray level is distributed evenly in the image, thereby expanding the image contrast. Using template and image convolution operation will cause some features to be suppressed, or prominent, which enhances the visual effect of the image. Frequency domain image enhancement is considered a type of indirect operation. Before image enhancement, images should be transformed into the frequency domain space for filtering, and the enhanced image is obtained after inverse transform procession of frequency

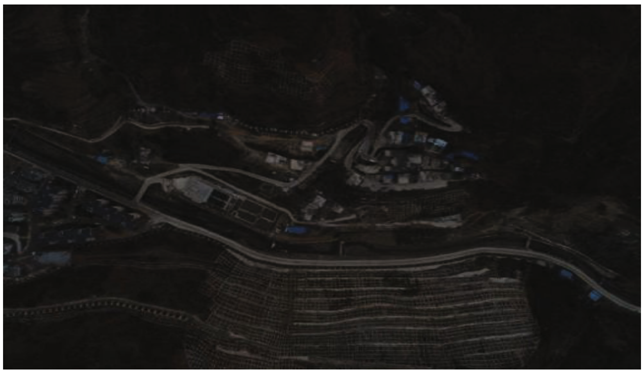


Figure 1: Original image of UAV.

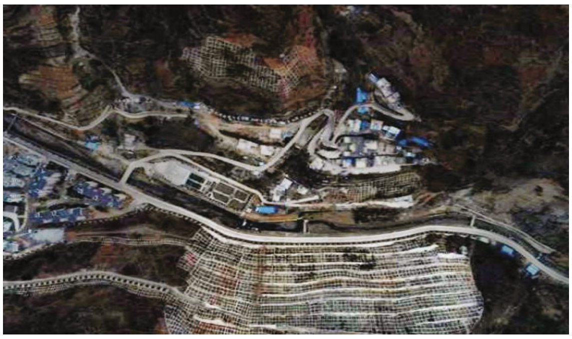


Figure 2: Histogram equalization-enhanced image.

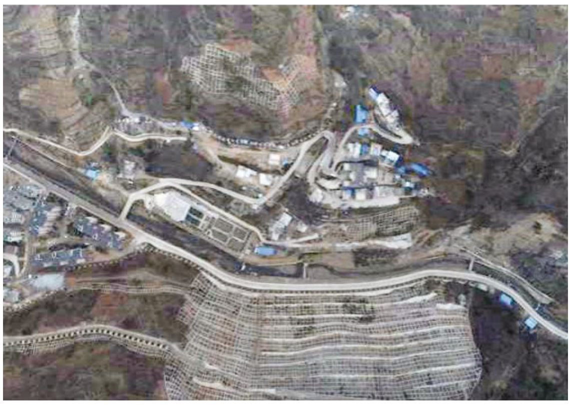


Figure 3: Retinex-enhanced image.

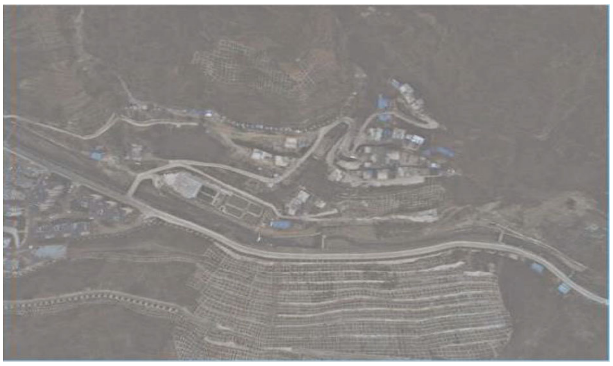


Figure 4: Mask-enhanced image with uniform light.

information. As the image edge feature belongs to high-frequency information and the image background pertains to low-frequency information, high-pass filtering or low-pass filtering can be exploited to sharpen the enhanced image.

In 1971, Land and Mc proposed an adaptive Retinex enhancement algorithm from three aspects (i.e., colour balance, edge area enhancement, and gray level change), and it has been extensively used in image enhancement [4]. Professor Pal et al. of India built four complex nonlinear functions with parameters and simulated the corresponding typical nonlinear mapping [5]. In accordance with the optimization criteria, 12 parameters were adjusted adaptively. As a result, the fuzzy image was enhanced, and the satisfactory results were achieved. Uncertainty, complexity, and instability are considered the characteristics of images. Anzueto-Rios et al. applied fuzzy set theory for image enhancement and achieved good results [6, 7]. Gu et al. used the Retinex algorithm for image enhancement processing to improve the recognition degree of image details [8]. This method is capable of estimating the brightness and reflectivity of the image and performing enhancement processing directly on the image, which can achieve a relatively ideal effect. Kou et al. proposed a gradient-domain guided filtering algorithm using deep convolutional network for image restoration and image superresolution enhancement and achieved effective results [9]. Liu et al. used adaptive segmentation to correct the gray scale inhomogeneity of the image [10]. Xiong et al. combined different image enhancement algorithms for adaptive parameters adjustment and then applied the adjusted parameters to different linear enhancement areas, which led to significant results [11]. Jiang et al. decomposed the original image into images in different regions and then distributed different radiation coefficients to the regions, so the illumination compensation in image enhancement could be solved [12]. Hu et al. proposed an algorithm for image brightness correction using the bilateral gamma function [13]. Mao et al. developed an adaptive bidirectional logarithmic change image enhancement algorithm [14]. Yu et al. built a defogging degradation model to enhance the shadow image. The built model could improve the brightness and visual effect of the image, whereas it could not effectively suppress the noise influence in darker areas [15]. Song et al. decomposed the image frequency and divided the image into high-frequency images and low-frequency images [16]. The Retinex algorithm was adopted to enhance the low-frequency band, which can improve the effect of insufficient illumination on the image, whereas this method cannot display the overall details of the image. Han optimized the existing gamma correction function and developed an adaptive gamma algorithm for image enhancement processing of panoramic images under low illumination. The developed algorithm is capable of improving the brightness of the image, suppressing the brighter areas in the image, and improving the detailed features of the image area [17]. Zhang et al. proposed a multiscale Retinex algorithm for colour protection and image enhancement. However, after the image was enhanced, the image illumination was not uniform, and the brightness area

Table 1: Comparison of image enhancement.

	Average gradient (MG)	Variance (S)	Information entropy (IE)
Original image	3.07	213.93	5.36
Histogram equalization-enhanced image	12.13	2697.10	7.36
Retinex-enhanced image	19.17	3636.40	7.67
Mask-enhanced image with uniform light	3.99	232.56	5.11

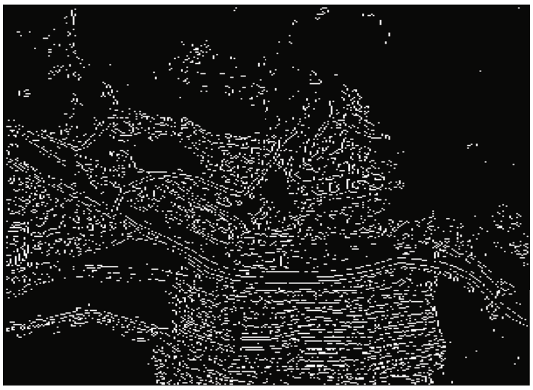


Figure 5: Extraction results of edge image enhancement.

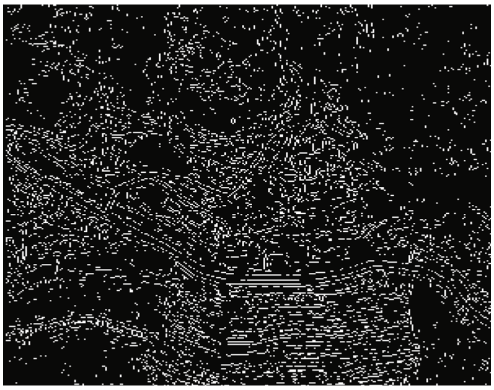


Figure 6: Extraction results of edge features of Retinex algorithm after features of original ground objects image.

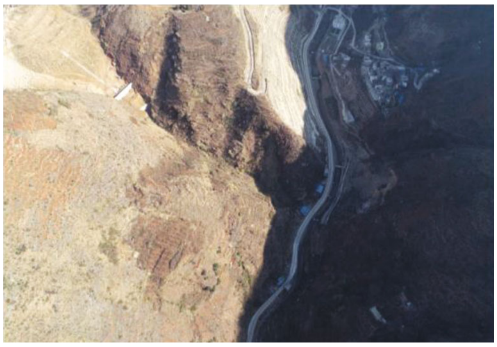


Figure 7: UAV shadow image.

turned out to be brighter, while the dark area was darker [18]. Li et al. adopted multiscale gradient domain-guided filter brightness enhancement algorithm and histogram adaptive brightness correction algorithm, which can be more suitable for colour image enhancement under weak light source [19].

The shadow removal method of UAV images in mountainous areas is investigated in this study. The identification of ground information in this shaded region is prone to error due to the occlusion of the mountain, the shadow of the UAV image, as well as the pixel brightness compressed and information gap of the image in the shadow areas during the image acquisition process for the drone. The UAV images were enhanced using the histogram equalization algorithm, the Retinex algorithm, and the Mask uniform light algorithm. Subsequently, the experimental results of the classical algorithm analysis were analysed and then compared using qualitative and quantitative methods. The shadow on the image could be removed using the Retinex algorithm, whereas the removal results were limited by partial problems (e.g., the uneven light and dark distribution of image, colour aberration phenomenon, and the image texture distorted to affect the visual effect). The Retinex algorithm was optimized to convert the RGB colour of the drone images to the HSV colour, and the optimized Retinex algorithm was used to enhance the brightness area after colour conversion and improve the image quality. The optimized Retinex algorithm can have high performance in the image shadow area removed, better image texture characteristics preserved, and the phenomenon of uneven image light and shade eliminated.

In the rest of this study, four general sections form the body of this study. In Section 2, the comparison of the UAV image enhancement algorithm is described. In Section 3, the employed intelligent methods are presented. Afterward, in Section 4, the obtained results are presented and discussed. Lastly, Section 5 is the conclusion giving a brief report of the results of this study.

2. Comparison of UAV Image Enhancement Methods

2.1. Comparative Analysis of Traditional Enhancement Algorithms. The grayscale frequency and brightness range of pixels in an image can be reacted by a histogram. Histogram equalization is performed using the cumulative function to correct the gray scale values, distribute the grayscale values uniformly, and enhance the image through nonlinear stretching. The pixel values are again matched for the stretched images. Thus, after the redistribution, the difference between the pixel values is not sharp, i.e., the original image is transformed into a well-distributed histogram by equalization the histogram [20].

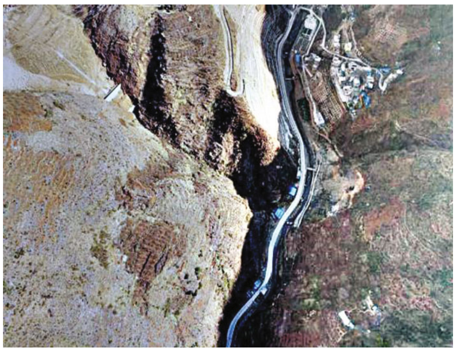


Figure 8: Shadow removal effect of Retinex algorithm.

The gray scale of an image can be considered a random variable of the interval [0,1], which can be represented using a probability density function. Assuming that the gray scale value of the image element is $r (0 \le r \le 1)$, the pixel gray level after the transformation is s, and $P_r(r)$, $P_s(s)$ are denoted as the probability density function of the random variable r and s, respectively, and the transformation function is $T(r)$; the equation is written as [21]:

$$\begin{cases} s = T(r), \\ P_s(s) = P_r(r)\left|\dfrac{dr}{ds}\right|. \end{cases} \tag{1}$$

A gray level s is generated for the respective transformed image element gray level r on the original image. The transformation function satisfies with that:

(1) It is single-valued and monotonically increasing in the interval$0 \le r \le 1$

(2) When $0 \le T(r) \le 1$, to ensure that the output gray scale has the same range as the input gray scale, it yields:

$$s = T(r) = \int_0^r P_r(w)dw, \tag{2}$$

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r P_r(w)dw\right] = P_r(r). \tag{3}$$

Substituting Equation (3) into Equation (1), it yields:

$$P_s(s) = P_r(r)\frac{dr}{ds} = P_r(r)\frac{1}{P_r(r)} = 1. \tag{4}$$

A continuous function transformation equation is presented above. When the digital image processing is used, if the digital image grayscale is ordered:

$$S_k = T(r_k) = \sum_{j=0}^{k} P_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, \tag{5}$$

where $k = 0, 1, 2, \cdots, L-1$; k represents the image gray scale; n is the total number of pixels; n_j is the number of pixels in the jth grayscale layer; $P_r(r_j)$ is probability density in the jth grayscale layer; $T(r_k)$ is the pixels state function in the kth grayscale layer; s_k is the final transformation result.

Retinex theory is a model first proposed by Edwin Land on how the human visual system regulates the colour and brightness of perceived objects, Retinex (abbreviation for Retina and Cortex) [4]. The Retinex algorithm can be balanced in the grayscale dynamic range compression, edge enhancement, and colour constant qualitative. It can achieve a balance in three aspects, i.e., gray scale dynamic range compression, edge enhancement, and colour identity, so it can adapt various images enhancement automatically. In essence, the Retinex algorithm is an image enhancement algorithm based on lighting compensation. Currently, the SSR (single-scale Retinex) algorithm and MSR (multiscale Retinex) algorithm have widespread applications for light compensation for close-range shooting images. Moreover, it also plays a good dodging effect for the long-distance remote sensing image, especially for the coloured images, which can maintain the colour information of the image while removing uniform brightness. The image can comprise luminance components and reflection volume, and the imaging model can be represented as [22]:

$$F(x, y) = R(x, y)I(x, y), \tag{6}$$

where $F(x, y)$ is denoted as an image; $R(x, y)$ represents the reflected light component. In addition, the magnitude of illumination intensity does not affect the reflected light component. $I(x, y)$ denotes the incident light component, which determines the image gray-scale dynamic range. Thus, according to the principle, if the luminance component affected by outdoor light intensity in the image can be estimated, followed by the removal of the luminance component. The final result is the reflected light component $R(x, y)$ of the object with the object's own reflection ability in the algorithm for image enhancement. The Retinex algorithm process can fall into the steps below.

The 1st step: images are represented in Equation (6), where $I(x, y)$ is the incident ray component and $R(x, y)$ is the reflected light component.

The 2nd step: the image is converted to a log domain.

$$\begin{aligned} f(x, y) &= \log\left[R(x, y)I(x, y)\right] = \log\left(I(x, y)\right) + \log\left(R(x, y)\right) \\ &= r(x, y) + i(x, y). \end{aligned} \tag{7}$$

The 3rd step: $i'(x, y)$ is obtained by filtering processing $i(x, y)$.

The 4th step: according to the above equation, the original image minus the incident light component to obtain the reflected light component $r(x, y)$.

$$r(x, y) = f(x, y) - i'(x, y). \tag{8}$$

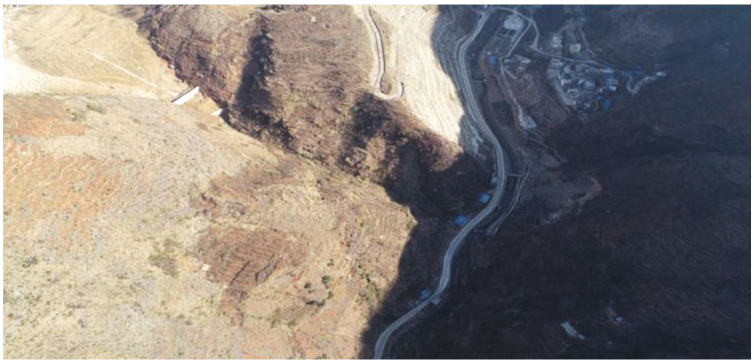


Figure 9: Original UAV image.

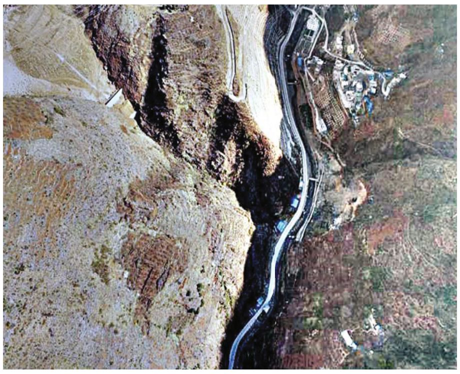


Figure 10: Shadow removal results of the Retinex algorithm.

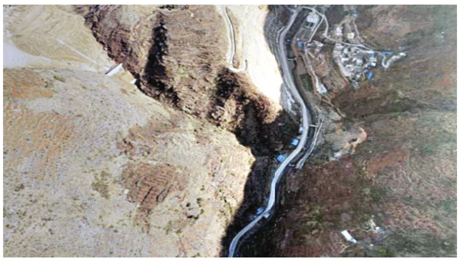


Figure 11: Image brightness correction result of the improved algorithm.

The 5th step: to take the antilog of $r(x, y)$ and finally get the enhanced image.

$$R(x, y) = \exp\ (r(x, y)). \tag{9}$$

The Mask algorithm is a common dodging method, mainly used for analysing the image from the frequency domain. However, it is considered that the intensity change is relatively slow, so the information reflecting the brightness change trend should be positioned in the low frequency part of the image. It is considered the interference part of the image if the part only reflects the changes in brightness without excessive image information. Instead, the information that reflects the scene reflection properties characteristics lies in the high frequency part unaffected by light, and it is the image with uniform brightness [23].

$$f(x, y) = r(x, y) + g(x, y). \tag{10}$$

In Equation (10), where images with uneven illumination are $f(x, y)$, the image after the dodging treatment is $r($ $x, y)$, expressing the background images during processing as $g(x, y)$. The original images and the acquired noise images are subtracted. Subsequently, the gray scale offset is added to the light equilibrium result. It can be expressed as:

$$r(x, y) = f(x, y) - g(x, y) + \text{offset}, \tag{11}$$

where the offset is constant since images processing aims to maintain the average brightness of the original image, so the average brightness of the original image is usually taken as offset. The contrast ratio of the image after the dodging treatment should be adjusted to improve the brightness value of the image.

The UAV is used to obtain the images of Longtoushan Town, Ludian County. On that basis, the enhanced effect on the UAV image of the above four methods is analysed. To facilitate the operation, the image size of the UAV is adjusted, with the adjusted size of 512 $*$ 512. Figure 1 illustrates the original image of the UAV. The histogram equalization enhancement algorithm, the Retinex enhancement algorithm, and the Mask uniform light enhancement algorithm are adopted to enhance the image, respectively, with the results presented in Figures 2–4.

According to the visual results, the image quality after enhancement is significantly improved, with buildings clearly visible, and the outline of the target object is defined clearly. Houses, roads, slope, and vegetation can be clearly displayed.

2.2. UAV Image Quality Evaluation Index. On the whole, the UAV image quality evaluation falls into two parts, i.e., subjective evaluation and objective evaluation. Subjective qualitative evaluation is also known as visual evaluation. In accordance with previous scales and evaluation standards and combined with existing experience, the quality of both images before and after processing is analysed, and the conclusions are drawn. The evaluation method primarily

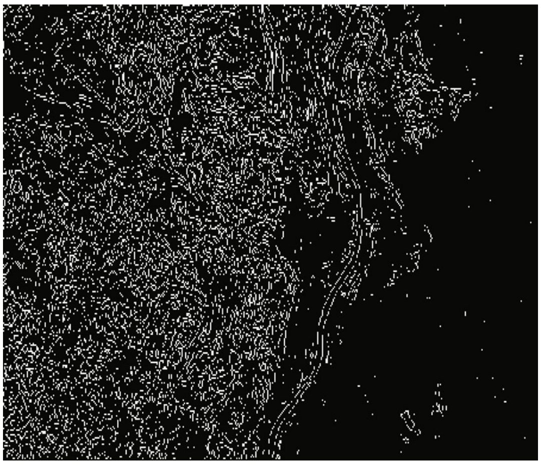


Figure 12: Original image feature extraction results.

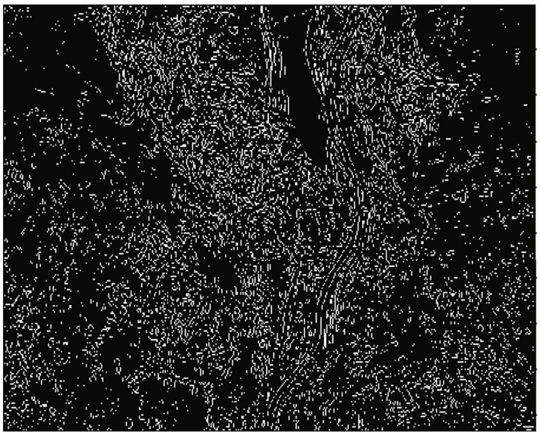


Figure 13: Image feature edge extraction results after brightness correction.

complies with experience, and it has low accuracy since different images and standards may produce different evaluation results.

Objective evaluation is evaluated using the mathematical model. Some properties of the image are quantitatively evaluated by quantifying the index factors and adopting some mathematical models. It is of important significance for image evaluation to evaluate the changes of these properties before and after image processing and build an evaluation system.

In brief, using subjective and objective quality evaluation comprehensively can evaluate the quality of image processing scientifically and reasonably. Furthermore, there are numerous objective evaluation indicators of image quality, in which variance, information entropy, and mean gradient are commonly used.

2.2.1. Variance. The number of image details can be compared by the gray variance, reflecting the discrete of the relative gray average of each image to a certain extent, which can be used to evaluate the size of image information volume. Scattered image gray scale distribution, large image contrast, and more information can be seen under a large variance. Conversely, small variance means low contrast and less image details. Therefore, the higher the image variance in the image comparison, the richer gray scale level, the higher the image quality will be, and vice versa.

$$S = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i, j) - u]^2, \tag{12}$$

where $M \times N$ represents the size of the image; $f(i, j)$ is the pixel value of the selected image; u represents the average value of image pixel.

2.2.2. Average Gradient. The image definition is mainly affected by weather conditions, UAV flight status, camera parameter setting, and other factors. The evaluation of the image definition is critical. Image definition is largely indicated in image blur, poor saturation, and tone difference. The common definition evaluation index is the average gradient, which can reflect subtle contrast changes in the image. The higher the average gradient, the better the image definition will be. In contrast, the smaller the average gradient, the worse the image definition will be. The mean gradient is formulated as:

$$\begin{aligned} MG = {} & \frac{1}{(M-1)(N-1)} \\ & \times \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\left(\frac{(f(i,j) - f(i+1,j))^2 + (f(i,j) - f(i,j+1))^2}{2}\right.}. \end{aligned} \tag{13}$$

In Equation (13), the gray value of the i line and the j list is $f(i, j)$, and the size of the image is $M \times N$.

2.2.3. Information Entropy. Information entropy, a measure of the amount of information in the image, is proportional to the amount of information contained in the image. The higher the information entropy, the more information the image will contain, i.e., the more detailed the image will be. Moreover, the smaller the information entropy, the less information the image will contain. The information entropy is calculated as:

$$\mathrm{IE}[f(x, y)] = -\sum_{i=1}^{\mathrm{num}} p_i \cdot \log p_i. \tag{14}$$

In Equation (14), the gray scale of image is num, and probability of the i gray level occurrence is p_i.

Based on the quantitative analysis, the average gradient value, variance, and information entropy are applied for image enhancement comparison [24]. The comparison results are listed in Table 1.

According to Table 1, the original image is compressed due to insufficient exposure, and the average gradient, variance, and information entropy of the image are relatively low. Three indicators have been significantly improved using the three kinds of classic enhancement algorithm for image enhancement. The analysis results show that the histogram equalization algorithm shows excessive enhancement after the shadow image enhancement and pixel brightness compression after the Mask homogenization algorithm enhancement; after the enhancement, *S* value, IE value, and MG of Retinex algorithm are higher than the others in three methods, thereby proving that the Retinex algorithm is the optimal. In the enhanced image, the ground objects and the texture of the landslide reinforcement and buildings can be seen clearly, and the vegetation texture on the mountain is relatively clear compared with that of the other three algorithms. According to the mentioned experiments, the Retinex algorithm is capable of achieving a better enhancement effect for the underexposed images.

The edge feature of the image contains considerable information regarding the image, and it can significantly indicate the reaction image clarity. The LOG (Gauss-Laplace Algorithm) operator comprises the Laplace edge detection algorithm and the Gaussian filtering algorithm, in which the Lapura operator primarily aims to highlight the edge and contour parts of the gray scale in the image and reduce the areas with slow changes in gray scale, and the Gaussian filtering is a linear smoothing filter suitable for eliminating Gaussian noise, with wide applications in image denoising. The Gaussian filtering function is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{15}$$

Convolution operations are performed between $G(x, y)$ and $f(x, y)$, and $I(x, y)$ is the smoothed image:

$$I(x, y) = G(x, y) \otimes f(x, y). \tag{16}$$

The Laplace operation is performed on the smooth images.

$$\begin{aligned} h(x, y) &= \nabla^2 I(x, y) = \nabla^2 (G(x, y) \otimes f(x, y)), \\ h(x, y) &= \nabla^2 G(x, y) \otimes f(x, y), \\ \mathrm{LOG}(x, y) &= \nabla^2 G(x, y) = \frac{1}{\pi\delta^4}\left[\frac{x^2 + y^2}{2\delta^2} - 1\right] e^{\left(x^2+y^2\right)/2\delta^2}. \end{aligned} \tag{17}$$

The LOG operator takes the form of $\nabla^2 G(x, y)$. Edge feature points are the zero intersection points between the template and the image convolution operations. The LOG template, with the typical size of 5×5, is presented as:

$$\begin{bmatrix} -2 & -4 & -4 & -4 & -2 \\ -4 & 0 & 8 & 0 & -4 \\ -4 & 8 & 24 & 8 & -4 \\ -4 & 0 & 8 & 0 & -4 \\ 2 & -4 & -4 & -4 & -2 \end{bmatrix}. \tag{18}$$

The LOG operator is used to extract the edge features of the ground objects after the optimized Retinex algorithm. Moreover, it is compared with the original image, and the result of the ground objects edge feature extraction is presented in Figures 5 and 6.

According to Figure 5, the edge features of the original image are extracted poorly, and a large area of empty appears in the mountain. In Figure 6, the Retinex algorithm can be applied for image enhancement, which can enhance the underexposure image under weak light source. This algorithm improves the image features recognition degree and extracts abundant edge features. Furthermore, considerable edge information is extracted from the mountain.

3. Optimized Retinex Algorithm for Shadow Removal

According to the previous experiment, the Retinex algorithm has the optimal enhancement, and this algorithm is adopted to remove the shadow area of the image. The original image is presented in Figure 7. The Retinex algorithm is used to remove the shadow region, and the result is shown in Figure 8.

According to Figure 8, if the algorithm is adopted to remove the shadow area, the shaded part of the image will be relatively clearer, whereas the image appears the phenomenon of halo artifacts. The bright area increases excessively, and the dark area enhancement is insufficient. The whole image appears uneven distributions of light and shade, colour distortion, and image texture distortion, which will also affect the after processing.

To solve the problems above, the original Retinex algorithm is improved, which converts the enhanced RGB colour into the HSV space for processing. This algorithm uses the existing 2D gamma algorithm for brightness image processing. Subsequently, reverse operation is performed for the corrected image to restore the UAV image.

3.1. HSV Colour Space.

It is difficult to ensure that each colour channel is enhanced or attenuated in the same proportion if the image enhancement algorithm is used to correct the colours in the three RGB channels, which leads to the colour distortion after enhancement. The calculation of the three channels simultaneously requires complex calculation. To perform targeted image brightness correction, this study chooses to correct colour images with uneven light in HSV colour space [25].

HSV model can reflect colour and saturation and conduct clustering calculation for various colours. It also can extract gray level and brightness information from the colours clustering, eliminate the effect of brightness and colour separation, and make the program more robust and have a better recognition effect than the RGB model. Chromaticity and saturation can reflect the colour type accurately, which are not very sensitive to the change of external lighting conditions. The colour conversion from

RGB to HSV is nonlinear [26].

$$h = \begin{cases} \text{unefined}, & \text{if } \max = \min, \\ 60^0 \times \dfrac{g-b}{\max - \min} + 0^0, & \text{if } \max = r \text{ and } g \geq b, \\ 60^0 \times \dfrac{g-b}{\max - \min} + 360^0, & \text{if } \max = r \text{ and } g < b, \\ 60^0 \times \dfrac{b-r}{\max - \min} + 120^0, & \text{if } \max = g, \\ 60^0 \times \dfrac{r-g}{\max - \min} + 240^0, & \text{if } \max = b, \end{cases} \tag{19}$$

$$s = \begin{cases} 0, & \text{if } \max = 0, \\ \dfrac{\max - \min}{\max} = 1 - \dfrac{\min}{\max}, & \text{otherwise}. \end{cases} \tag{20}$$

In Equations (19) and (20) above, the gray values of the three primary colours of the image are r, g, and b, and the hue, saturation, and value of the image are h, s, and v.

3.2. Two-Dimensional Gamma Function. To achieve uniform illumination image, 2D gamma function is used for colour correction, and parameters are adjusted according to the distribution characteristics of illumination components. For the input image, a 2D gamma function is constructed based on the extracted light component, and its expression is stated as follows [27].

$$O(x, y) = 255\left(\frac{F(x, y)}{255}\right)^{\gamma} \quad \gamma = \left(\frac{1}{2}\right)^{(I(x,y)-m)/m}. \tag{21}$$

In Equation (21), the brightness value of the corrected output image is $O(x, y)$, the parameter of brightness enhancement is γ, and m is the average brightness value of the illumination component.

When the light value of a pixel is lower than the average value, the 2D gamma function operation can improve the brightness value of the pixel. If the light value at a point (x, y) is 64, and the brightness value of the input image is 120, the corrected brightness value of the output image will be 149. Thus, the output image performance is improved when the original image illumination is too low. When the light of a pixel value is higher than the average, the brightness value at a certain point will be 120. Assuming that the light value at this point is 192, and the brightness value of the corrected output image is 108. As a result, the brightness of the image decreases when the light in the original image is too high.

4. Experiments and Analyses

We used the DJI Phantom 4 Pro UAV to capture images since its flight platform is more stable than the others and easy to operate. Its take-off weight was at 1388 g and had a maximum flight time of nearly 23 min. 1-inch CMOS effective pixels were 20 million, and its image resolution was up to 5472 × 3648. Moreover, the lens focus distance was 35 mm with the autofocus f/2.8-f/11 aperture. It had a maximum rise speed and a maximum flight height of 6 m/s and of 6000 m, respectively, as well as a GPS/GLONASS dual satellite positioning mode. The image of the acquired UAV was shaded by the occlusion of the mountain during the UAV flight.

The UAV image was loaded (Figure 9). First, the Retinex algorithm was used for image enhancement, and the results are presented in Figure 10. In the enhanced UAV image, the shadow region was significantly improved. The brightness of the image in the shadow region was significantly improved, and the object in the shadow region could be clearly identified. However, the enhanced image underwent texture distortion and colour distortion.

After the image enhancement, the colour transformation was performed, which converted the RGB colour to the HSV component. The image after colour correction of the inverse operation synthesized into a new image (Figure 11). After the 2D images underwent function correction, the brightness was uniform, and texture feature of the image was clear.

The gauss-Laplace operator was adopted to extract the edge features of the ground object surface after the correction of the image brightness. Moreover, the edge features of the original image were compared with those of the original image. The original image edge features are presented in Figure 12, and the corrected image edge features are illustrated in Figure 13.

According to Figure 12, no shadow existed in the original image, the texture was rich, and the edge features were relatively obvious. In the areas with shadows, there was emptiness in the extracted edge features, which revealed that the edge features were unclear and that the target object contour could not be extracted. According to Figure 13, the corrected UAV image could extract more obvious edge features from the original shadow area, which demonstrated that the method significantly impacted the removal of shadow areas and the correction of brightness.

5. Conclusions

When the UAV acquires the image, the acquired UAV image has underexposure and shadows due to the blocking of mountains and the influence of light. The image brightness value of the shaded region is compressed, and the information is lost. Vital information is contained in the image of the shadow region. To remove the shadow region of the image, an optimized Retinex algorithm is used to process the shadow. First, the Retinex algorithm is used for image enhancement, whereas the brightness of the image after enhancement is uneven, which reduces the quality of the image. To solve the problem above, the image can be converted from RGB colour to HSV space. The brightness of the HSV colour space is corrected by 2D gamma function, and the correction image receives inverse operation. Combined with the actual experiment for the UAV image, the optimized Retinex algorithm has a better effect in shadow area removal. After the shadow removal, the ground feature details of the shadow area of the optical image can be clearly identified, and the image quality improved. The algorithm works better on areas with lighter shadows, but less so with darker shadows. It is also the direction to be studied in the future.

Data Availability

(1).The (data type) data used to support the findings of this study are included within the article. (2) The (data type) data used to support the findings of this study are included within the "Figures information file(s)".

Conflicts of Interest

The authors declared that there is no conflict of interest in presenting this manuscript.

Acknowledgments

An amended opinion was presented by Arun Kumar Sangaiah. We acknowledge the support provided. We would like to express our gratitude to the reviewers for their constructive comments on improving the quality of this study. This work was supported by the Yunnan Provincial Science and Technology Department Fund (202001AT070059).

References

[1] M. Wang, J. Pan, S. Q. Chen, and H. Li, "A method of removing the uneven illumination phenomenon for optical remote sensing image," *IEEE Transactions*, vol. 5, pp. 3243–3246, 2005.

[2] K. Singh and R. Kapoor, "Image enhancement using exposure based sub image histogram equalization," *Pattern Recognition Letters*, vol. 36, no. 1, pp. 10–14, 2014.

[3] S. Ji, P. Dai, M. Lu, and Y. Zhang, "Simultaneous cloud detection and removal from bitemporal remote sensing images using cascade convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 732–748, 2021.

[4] E. H. Land and C. J. Mc, "Lightness and Retinex theory," *Journal of Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1971.

[5] S. K. Pal, D. B. Handari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognition Letters*, vol. 15, no. 3, pp. 261–271, 1994.

[6] A. Anzueto-Rios, J. A. Moreno-Cadenas, and F. Gomez-Castaneda, "Fuzzy technique for image enhancement using B-spline," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 347–349, Cancun, Mexico, 2009.

[7] T. Chen and X. Cheng, "Image segmentation based on fuzzy mathematical morphology and watershed algorithm," *Journal of Southwest University (Natural Science Edition)*, vol. 30, no. 3, pp. 142–145, 2008.

[8] Z. H. Gu, F. Li, and X. G. Lv, "A detail preserving variational model for image Retinex," *Applied Mathematical Modelling*, vol. 68, pp. 643–661, 2019.

[9] F. Kou, W. Chen, C. Wen, and Z. Li, "Gradient domain guided image filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4528–4539, 2015.

[10] Z. Liu, Z. H. Li, and D. W. Wang, "Adaptive adjusting for the image with pocket density," *Control Engineering of China*, vol. 10, no. 3, pp. 249–252, 2003.

[11] X. H. Xiong, Z. B. Qing, and Y. Chen, "Remote sensing image enhancement based on genetic optimization," *Acta Geodaetica et Cartographica Sinica*, vol. 33, no. 4, pp. 341–346, 2004.

[12] Y. X. Jiang, X. T. Wang, X. G. Xu, and H. Huang, "A method for image enhancement based on light compensation," *Acta Electronica Sinica*, vol. 37, no. 4, pp. 151–154, 2009.

[13] Y. Hu, T. T. Li, and L. S. Huang, "Brightness preserving image enhancement method based on bilateral gamma correction," *Computer Applications And Software*, vol. 36, no. 5, pp. 204–210, 2019.

[14] D. Y. Mao, Z. X. Xie, and X. Q. He, "Adaptive bilateral logarithm transformation with bandwidth preserving and low-illumination image enhancement," *Journal of Image and Graphics*, vol. 22, no. 10, pp. 1356–1363, 2017.

[15] C. Y. Yu, X. D. Xu, H. X. Lin, and Y. Xinyan, "Low-illumination image enhancement method based on a fog-degraded model," *Journal of Image and Graphics*, vol. 22, no. 9, pp. 1194–1205, 2017.

[16] R. X. Song, D. Li, and X. C. Wang, "Low illumination image enhancement algorithm based on HSI color space," *Journal of Graphics*, vol. 38, no. 2, pp. 217–223, 2017.

[17] P. F. Han, *Research on low-light panoramic image enhancement algorithm*, Xi AN University of Posts& Telecommunications, 2019.

[18] X. Zhang, W. Wang, and D. Xiao, "Improved image enhancement algorithm based on multi-scale Retinex with chromaticity preservation," *Computer Science*, vol. 45, no. 10, pp. 247–249, 2018.

[19] H. Li, R. Y. Wang, Z. X. Geng, and H. U. Haifeng, "Low-illumination image enhancement algorithm based on multi-scale gradient domain guided filtering," *Journal of Computer Applications*, vol. 39, no. 10, pp. 3046–3052, 2019.

[20] M. Tang, Y. S. Li, X. Li, and L. Bo, "Local enhancement method and its applications to UAV image matching," *Remote Sensing for Landand Resources*, vol. 25, no. 4, pp. 53–57, 2013.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Publishing House of Electronics Industry, Beijing, 2002.

[22] G. Orsini, G. Ramponi, P. Carrai, and R. Di Federico, "A modified Retinex for image contrast enhancement and dynamics control," *Image Processing*, vol. 14, no. 17, pp. 393–396, 2003.

[23] H. Wang, Y. Zhang, H. H. Shen, and Z. Jing-zhong, "Review of image enhancement algorithms," *Chinese Optics*, vol. 10, no. 4, pp. 438–448, 2017.

[24] Y. H. Cao, *Application of Image Shadow Removal Based on Retinex Methods*, Xi'AN University of Science And Techionlogy, 2017.

[25] H. Q. Yao, S. W. Yang, Z. J. Liu, X. W. Yang, and L. M. Zhang, "Shadow detection method of city large objects based on world-view-2image," *Science of Surveying and Mapping*, vol. 40, no. 10, pp. 110–113, 2015.

[26] X. S. Han, *Technology Research on Shadow Detection and Compensation in Optical Remote Sensing Images*, Information Engineering University, 2017.

[27] Q. C. Chu, H. B. Wang, and L. Tao, "Local adaptive gamma correction method," *Computer Engineering and Applications*, vol. 51, no. 7, pp. 189–193, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7283841, 10 pages
https://doi.org/10.1155/2022/7283841

Research Article

Research on the Mechanism of Influencing Factors of the Urban Road Traffic Operation State

Tianxiao Wang,[1,2] Hao Wu,[2] Qiang Li,[3] Shaoquan Ni,[1] Tinghui Qin,[2] Yifan Niu,[1] and Di Cao[4]

[1]School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 611756, China
[2]The Smart City Research Institute of China Electronics Technology Group Corporation, Shenzhen 518038, China
[3]Government Services Data Bureau of Shenzhen Municipality, China
[4]Zhejiang ZhongShang Technology Co. Ltd., Hangzhou 311215, China

Correspondence should be addressed to Qiang Li; 360779@qq.com

Received 8 January 2022; Revised 4 March 2022; Accepted 12 March 2022; Published 6 April 2022

Academic Editor: Shunmei Meng

Profound understanding of an interaction mechanism among influencing factors in the perspective of urban road traffic operation is of great significance for scientific and effective urban congestion management. In this paper, 6 indicators are proposed for the road traffic operation in the aspects of average speed of road sections, regional traffic index, and number of traffic incidents. Based on this, 4 major influencing factors and 12 measurable subinfluencing factors are proposed according to urban traffic supply and demand, and the SEM (structural equation model) is established to find out their interaction mechanism. And then, several sets of local traffic data collected in Shenzhen are used for model fitting, validation, and path analysis. The mathematical results show that all 6 indicators affiliated to the road traffic operation have a good explanation when it comes to the change of operation status. Among 4 latent variables in the traffic supply and demand aspect, the service equipment operation level and control equipment operation level can positively influence the road traffic operation status, while the urban traffic demand plays a negative role. Complex interactions among four latent variables are further pointed out. Finally, on the basis of the path coefficient relationship among the influencing factors, scientific suggestions and guidance are provided for urban road management control.

1. Introduction

In the current situation of increasingly severe road traffic congestion, comprehensively combining the influencing factors of urban road traffic operation status and understanding and mastering its change interaction mechanism will help people deeply understand the urban road traffic operation mechanism and make practical and effective management and control decisions.

There have been many studies on issues related to the operation status of urban road traffic. Those researches can be roughly divided into three categories: single road sections and intersections, local road networks, and urban macro road traffic. Lang and Fu [1] used indicators such as saturation and vehicle speed to analyze and evaluate the traffic operation status of local sections of expressways based on actual data; Chen et al. [2] proposed innovative traffic index calculation methods accurately calculated and described the street-level traffic operation characteristics of small areas; Wei and Ma [3] regarded urban road traffic as a giant system at the macro level and studied the evaluation system and method of its coordination degree from the two subsystems of supply and demand. In recent years, the research scope has gradually developed from a single level to a combination of macro and micro level. Bai et al. [4] established a comprehensive evaluation method of traffic operation status by looking for the three-level evaluation links of highway points, lines, and surfaces. By summarizing the existing research, most of them discuss the operation status of urban road traffic from the perspective of evaluation, and its role is limited to identifying the quality of road traffic operation status at different levels [5, 6]. Moreover, urban road traffic is a giant system, and traffic supply, demand, and operation status are closely linked and interact. However, existing research often considers them separately, so that it is impossible to accurately grasp the operating state of the traffic state from an overall perspective, it is impossible to conduct in-depth research on its evolution law, and it is impossible to provide specific and effective guidance for the formulation of traffic management and control strategies [7–10].

The contributions of this paper include three parts. Firstly, from the perspective of combination of micro and macro, three indicators and factors of traffic supply, traffic demand, and traffic operation status are sorted out. The second is to construct SEM, complete the model modification and evaluation, clarify their relationship through the SEM, explore the interaction mechanism, and then analyze the evolution law of the urban road traffic operation status. Thirdly, according to the analysis results of the model, references are provided for urban traffic managers to formulate feasible control strategies.

This paper is organized as follows. In Section 2, the evaluation index system is proposed in the aspect of road traffic operation status. The structural equation model (SEM) is established in Section 3, and confirmatory factor analysis, correction, and model evaluation are carried out. A conclusion is made in Section 4.

2. Evaluation Index System

The evaluation index system is proposed based on the road traffic operation status. On this basis, starting from the two general directions that affect road traffic operation status, traffic supply, and traffic demand, focusing on service equipment, management and control equipment, urban traffic demand, and surrounding traffic demand in the city, sort out and identify the influencing factors of road traffic operation status.

2.1. Road Traffic Operation Status. Existing studies analyze and measure the road traffic operation status based on the overall average speed of the road section, the average regional traffic index, and the average congestion time [11, 12]. However, the indexes such as the average speed of the whole section and the average traffic index of the whole region ignore the different importance of variable levels of sections and regions to the overall road traffic operation status of the city, resulting in the lack of accuracy of the calculation results. In addition, these indexes are limited to the characteristics of the vehicle itself. On the other hand, the variables that determine the status of road traffic should also take the impact of some incidents connected with safety issues into considerations [13].

Based on this, this paper selects the regional traffic index and the average speed of road sections as evaluation indicators and categorizes them into the average speed of key roads, the average speed of other roads, the traffic index of key areas, and the traffic index of other areas from the spatial perspective. In addition, the two types of indicators that cause accidental damage, including accident alarms and congestion alarms, are included in the evaluation index system [14, 15]. The selection of indicators is shown in Table 1.

2.2. Influencing Factors of Road Traffic Operation Status. From a macro perspective, real-time road traffic demand and road traffic supply are two key factors that affect the state of road traffic.

(1) Road traffic supply

The supply of urban road traffic [16] is generally composed of transportation infrastructure, that is, related services and control equipment, e.g., roads, vehicles, stations, transportation organizations, and services [17]. The road traffic supply can be subdivided into static supply and dynamic supply according to the speed and frequency of changes in the supply. The static supply index mainly covers the quantity of infrastructure, such as the length of roads at all levels and the number of control equipment, while the dynamic supply is reflected by some real-time data. These real-time data are mainly retrieved from urban road traffic facilities, which can be further divided into traffic service facilities and traffic management facilities according to their functions. The traffic service facilities include parking lots, sight guidance screens, etc., while the traffic management facilities mainly include traffic signs, traffic lines, physical isolation devices, traffic signal control equipment, traffic violation recognition and capturing equipment, etc. [18]. The traffic operation state is directly affected by the dynamic supply when the static supply of infrastructure and other facilities is determined.

Dynamic supply of road traffic mainly covers two aspects: service equipment operation level and control equipment operation level. Among them, service equipment includes parking, road traffic guidance, and networked coordinated control [19]. Control equipment includes road and intersection monitoring, signal light control, etc. The selection of specific indicators is shown in Table 2.

(2) Traffic demand

Real-time traffic demand is reflected by the flow of passenger and freight traffic moving on the road, which indicates strong spatial and temporal characteristics. In terms of spatial characteristics, passenger and freight traffic flows in the scope of intracity and the municipal boundary area are extracted [20].

Intracity traffic measures the total amount of real-time vehicle traffic in the intracity area; municipal boundary traffic measures the passenger and freight traffic in and out of the city and on highways around the city [21, 22]. Among them, the traffic volume entering and leaving the intracity area and volume of high-speed traffic are divided based on the spatial characteristics, while those intracity demands including the volume of expressway, arterial road, and other roads are considered in the aspects of temporal characteristics. The selection of specific indicators is shown in Table 3.

Table 1: Evaluation index system of road traffic operation status.

Object	Evaluation indicators
Road traffic operation status	Average speed of key roads Average speed of other roads Key area traffic index Other regional traffic index Number of accidents Number of congestion alert

3. Structural Equation Model (SEM)

3.1. In-Line Style. The structural equation model, also known as structural equation modeling, is a statistical method based on the covariance matrix of variables to analyze the relationship between variables. Compared with traditional statistical methods such as regression analysis, the structural equation model has the following advantages [23–25].

(1) Capable of processing multiple dependent variables simultaneously. In traditional multiple regression or path analysis, the coefficients of the dependent variables are calculated independently, which neglects the influences and relationships with other factors. The SEM considers multiple factors at the same time, and the analysis effect is more accurate

(2) Error tolerant between dependent and independent variables. Many social and psychological studies involve variables that cannot be accurately and directly measured, which are known as latent variables. When analyzing the relationship between latent variables, neither the independent nor the dependent variable can be accurately measured. When traditional regression models deal with such problems, they will only consider the error of the dependent variable. SEM analysis allows measurement errors in both independent and dependent variables and has advantages in analyzing the correlation between latent variables and other issues

(3) Simultaneous analysis of factor structures and relationships. In SEM analysis, the correlation between latent variables and their underlying factors is calculated at the same time. Compared with the traditional analysis method that calculates factor structures and relationships independently, SEM can comprehensively consider the interaction and role of all coexistence factors, appropriately adjust the factor structure, and obtain a more comprehensive correlation

(4) More complex factor relationships are considered. Traditional factor analysis is difficult to deal with models that have more complicated subordination relationships such as one factor subordinate to multiple factors or consider higher-order factors, while SEM can make up for this deficiency

(5) Able to evaluate the rationality of factor structure and affiliation. SEM can evaluate its rationality by calculating the overall fitting degree of different factor structures and affiliations to the same sample data, so as to obtain the closest relationship to the data

3.2. Model Structure. The SEM consists of two parts: measurement equation and structural equation [26]. The measurement equation describes the relationship between latent variables and indicators, while the structural equation describes the relationship between the latent variables. The measurement equation and structural model are shown in (1) and (2).

$$\begin{aligned} x &= \Lambda_x \xi + \delta, \\ y &= \Lambda_y \eta + \varepsilon, \end{aligned} \tag{1}$$

where x, y are the exogenous and endogenous index vectors, respectively; ξ, η are exogenous and endogenous latent variables, respectively; Λ_x, Λ_y are the relationships between exogenous indicators and exogenous latent variables and the relationships between endogenous indicators and endogenous latent variables, respectively; and δ, ε are the error terms of exogenous and endogenous indexes, respectively.

$$\eta = B\eta + \Gamma\xi + \xi, \tag{2}$$

where ξ, η are the exogenous and endogenous latent variables. Bare the relationship between the endogenous latent variables, Γ are the influence of the exogenous variables on the endogenous latent variables, and ξ are the structural equation residuals.

3.3. SEM Construction. The construction and analysis process of the SEM is shown in Figure 1. The process of SEM construction is as follows: (1) propose theoretical assumptions about subordination and correlation; (2) introduce the definition of related variables and classify them in different categories; (3) based on the proposed theoretical assumptions and variables, construct a SEM; (4) collect data; (5) perform confirmatory factor analysis and adjust the model based on the analysis results so as to meet the related tests of reliability and validity; (6) output the model evaluation results; and (7) perform path analysis and output the analysis results of the affiliation and correlation between the key variables.

The realization of the above process is based on AMOS, which is powerful visualization software for SEM analysis. Firstly, the conjecture model is constructed, and the visual conjecture model is constructed according to the theoretical hypothesis and variable determination. The elliptic variable represents the latent variable that is not easy to measure directly [27]. Rectangular variables represent measurement variables that measure latent variables; the circular variable represents the allowable error between the latent variable and the measured variable; directed edges represent the interaction between variables. Then, the model was tested and the data were input into the conjecture model for confirmatory factor analysis. The model was modified according to the modification suggestions given by AMOS, which mainly focused on the presence and direction of the interaction between variables and specifically modified the form of

Table 2: Evaluation index system for dynamic road traffic supply.

Object	Equipment	Evaluation indicators
Dynamic road traffic supply	Service equipment	Parking space vacancy rate
		Online rate of road traffic guidance screen
		Networked coordinated control of the ratio of intersections
	Control equipment	Intersection electric police online rate
		Semaphore online rate
		Online rate of road section monitor

Table 3: Evaluation index system for road traffic demand.

Object	Type	Evaluation indicators
Road traffic demand	Municipal boundary	Traffic volume entering the intracity area
		Traffic volume leaving the intracity area
		High-speed traffic volume around the city
	Intracity	Expressway traffic volume
		Arterial road traffic volume
		Other road traffic volume

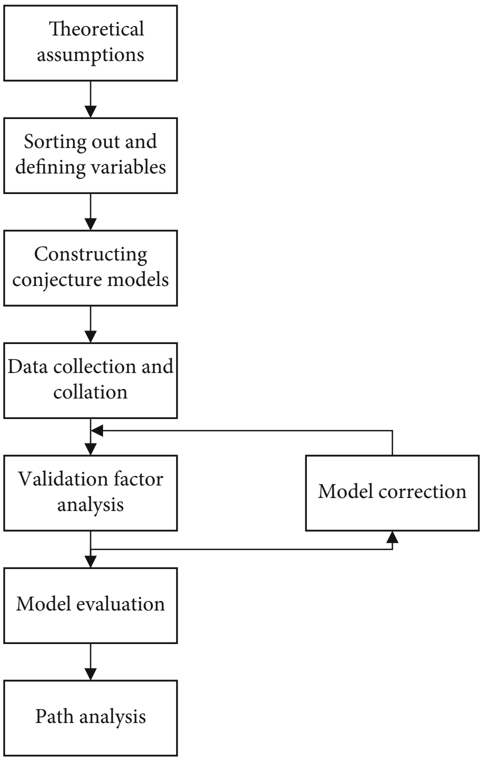

Theoretical assumptions
Sorting out and defining variables
Constructing conjecture models
Data collection and collation
Validation factor analysis
Model correction
Model evaluation
Path analysis

Figure 1: Flow chart of SEM construction.

directed edges. Finally, the path analysis is carried out based on the modified model [28].

3.3.1. *Model Establishment.* According to the evaluation indicators of road traffic operation status and related influencing factors constructed, combined with the relevant rules of the SEM structural equation model, the model variables are listed in Table 4.

There are 5 latent variables of road traffic operation status, service equipment operation level, management and control equipment operation level, municipal boundary demand, and intracity demand, which further contain 18 subordinate indicators that are taken as measured variables. Based on the relevant theoretical knowledge and the above variables, a hypothetical SEM is constructed as shown in Figure 2.

3.3.2. *Data Collection.* Sample data of the measured variables are required to be input into the SEM so as to perform confirmatory factor analysis and path analysis, and there are specific requirements for the sample data volume. Existing researches [29, 30] show that the difference between the sample data volume and the number of parameters to be estimated in the hypothetical model needs to be greater than 50. Thus, the number of samples between 100 and 200 is suitable for estimating the SEM.

This paper uses the real traffic data collected in Shenzhen from June 1, 2020, to June 7, 2020. The frequency of retrieving data is 1 hour, and the sample data volume for each measurement indicator is 168. The dataset is shown in Table 5.

SPSS software is used to standardize the data. Afterwards, AMOS software is used to draw a hypothetical model and bring in sample data to complete the model confirmatory factor analysis, correction, and evaluation.

3.3.3. *Confirmatory Factor Analysis, Correction, and Model Evaluation.* This paper uses AMOS software, brings in sample data to the constructed hypothetical SEM, and carries out confirmatory factor analysis. In this process, the maximum likelihood estimation method is used to fit the model [31–33], and according to the analysis results, the defects

Table 4: SEM variables for influencing factors of road traffic operation status.

Latent variable	Measured variable
Road traffic operation status	Average speed of key roads Y1 Average speed of other roads Y2 Key area traffic index Y3 Other area traffic index Y4 Number of accidents Y5 Number of congestion alarms Y6
Service equipment operation level	Parking space vacancy rate X1 Online rate of road traffic guidance screen X2 Network coordinated control intersection ratio X3
Control equipment operation level	Online rate of traffic lights X4 Online rate road section monitoring X5 Online rate of intersection electric police X6
Municipal boundary demand	Traffic volume entering the intracity area X7 Traffic volume leaving the intracity area X8 High-speed traffic volume around the city X9
Intracity demand	Expressway traffic volume X10 Arterial road traffic volume X11 Traffic volume on other roads X12

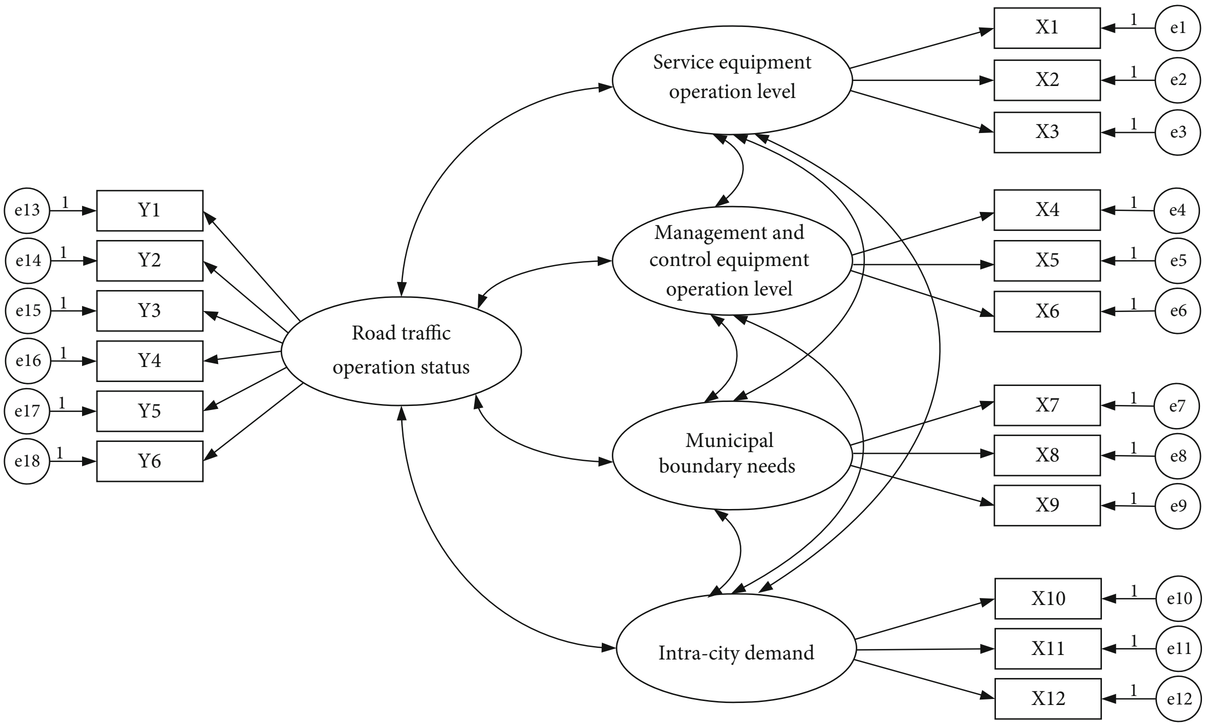


Figure 2: Hypothetical SEM.

of the hypothetical model are corrected, and the revised model structure is shown in Figure 3.

In the application of the SEM, after the hypothetical model is fitted with the sample data, the degree of fitness between the hypothetical model and the actual data must be tested through the corresponding structural equation fitness evaluation indicators [34, 35]. There are three commonly used evaluation indicators, namely, absolute fit index, value-added fit index, and parsimony fit index. Absolute fit index measures the degree of fitting between the constructed model and sample data [36–38]. Commonly used indicators include GFI (goodness-of-fit index), RMSEA (root mean square of approximate error), SRMR (root mean square of normalized residual error), etc. A value-added fit

Table 5: Traffic data in Shenzhen, China.

	X1	X2	X3	X4	X5	X12	Y1	Y2	Y3	Y4	Y5	Y6
1	0.510	0.906	0.962	0.604	0.827	2281.7	33.5	43.5	3.8	3.5	37	7
2	0.511	0.906	0.955	0.604	0.827	1400.7	33.5	43.5	3.8	3.5	14	1
3	0.512	0.906	0.955	0.604	0.827	932.0	33.5	43.5	3.8	3.5	12	0
4	0.512	0.906	0.932	0.604	0.827	877.0	33.5	43.5	3.8	3.5	9	0

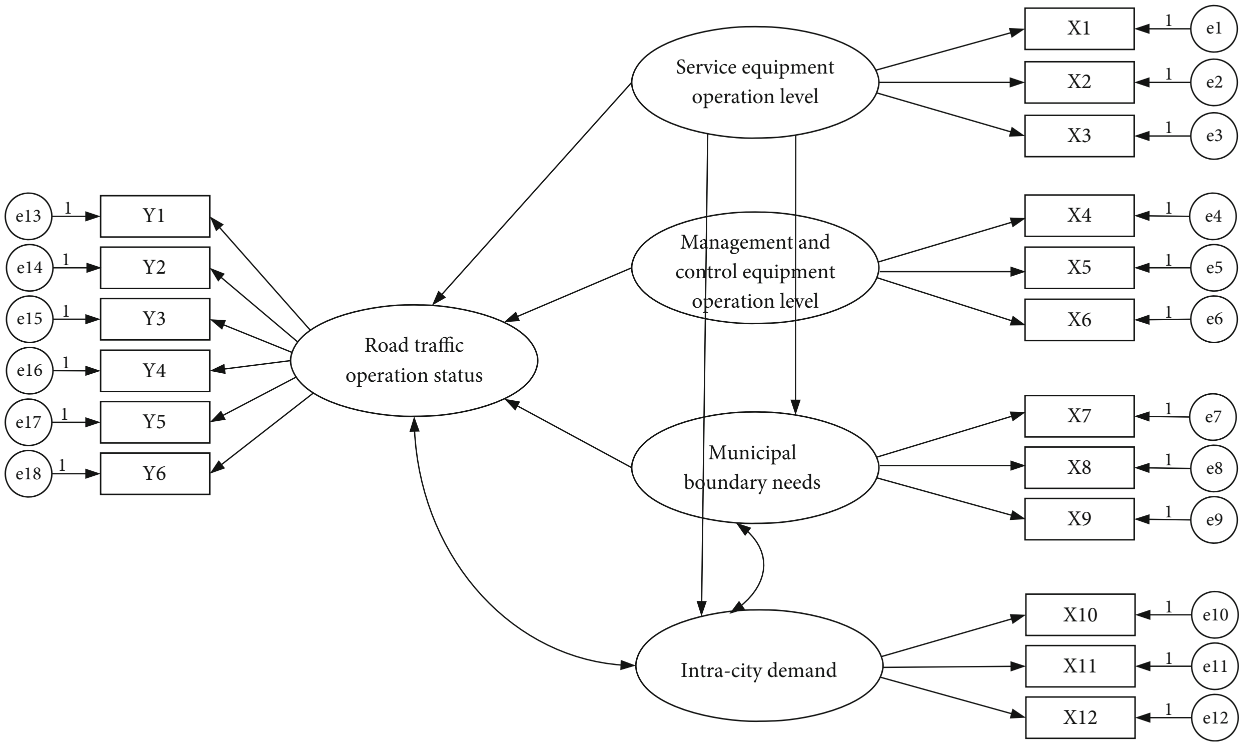


Figure 3: Revised SEM structure.

Table 6: Confirmation factor fitting results of the revised model.

Index system	Absolute fitness index	Value-added fitness index		Reduced fitness index		
Index	GFI	NFI	TLI	X^2 /DF	PNFI	PCFI
Adaptation value	>0.9	>0.9	>0.9	<5	>0.5	>0.5
Fitted value	0.908	0.913	0.905	4.6	0.609	0.62
Fit judgment	Yes	Yes	Yes	Yes	Yes	Yes

index, which is also called a relative fitness index, is a statistic obtained by comparing the theoretical model with the benchmark model which indicates the degree of improvement. Benchmark models are usually the models with the most limitations and the worst fitness. Commonly used indicators include NFI (normed fit index) [40], TFI (nonnormed fit index) [39], and CTI (comparative fit index) [40]. Parsimony fit is a kind of indicator derived from the previous two types of fit [42]. Based on the idea of parsimony ratio of df_t and df_n, which are the degree of freedom of the theoretical model and benchmark model, respectively, the commonly used indicators include the chi-square degree of freedom (X^2/DF), parsimony normed fit index (PNFI), and parsimony relative noncentrality index (PCFI). This paper uses 6 indicators including GFI, NFI, TLI, chi-square freedom ratio (X^2/DF), PNFI, and PCFI [43–45]. The fitting results are shown in Table 6. Table 6 shows that all indicators of the revised model meet the fitting standard.

In addition, it is necessary to carry out a variation extraction value (AVE) and combination reliability (CR) test for latent variables.

$$\mathrm{AVE} = \frac{\sum \lambda^2}{n} \tag{3}$$

$$\mathrm{CR} = \frac{\left(\sum \lambda\right)^2}{\left(\sum \lambda\right)^2 + \sum\left(1 - R^2\right)}, \tag{4}$$

where λ is the path coefficient, n is the number of measured variables, and R is the residual error [39].

Table 7: Combination reliability evaluation results.

Latent variable	Number of measured variables	AVE	Combination reliability
Service equipment operation level	3	0.6307	0.8364
Management and control equipment operation level	3	0.6221	0.8309
Municipal boundary demand	3	0.7525	0.8996
Intracity demand	3	0.7536	0.9014
Road traffic operation status	6	0.6615	0.9187

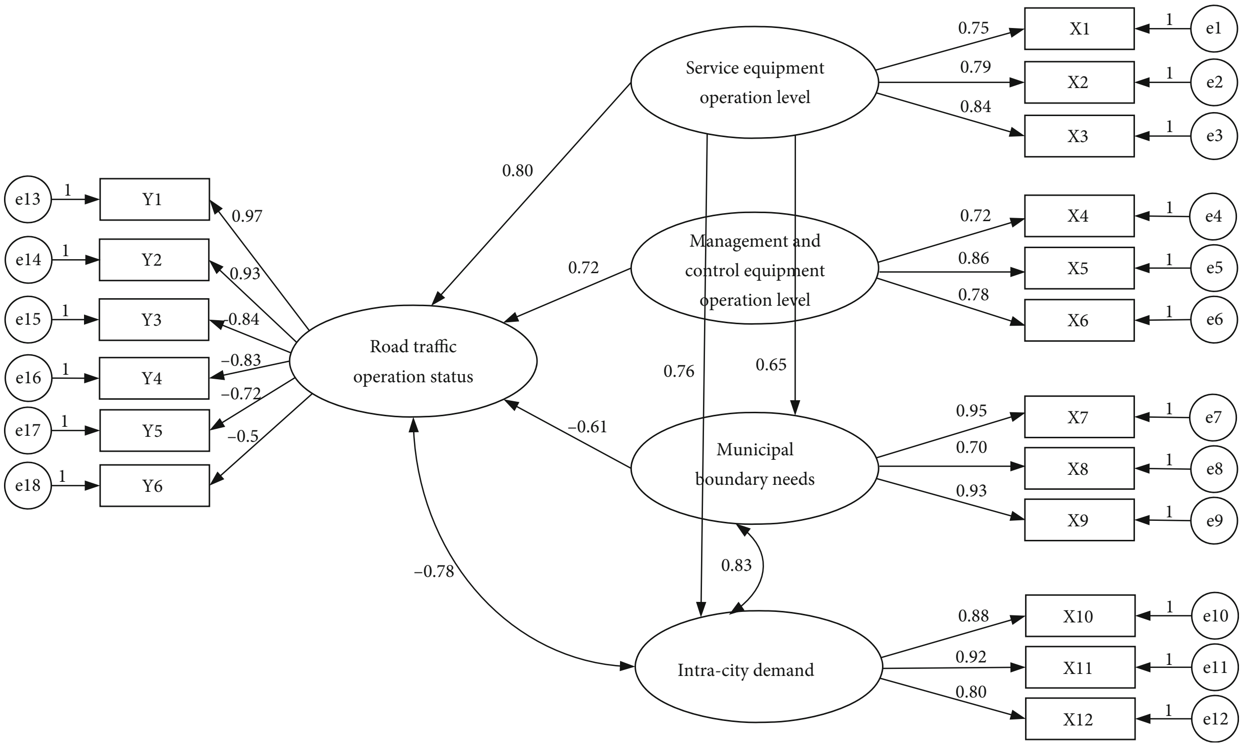


Figure 4: Path diagram of SEM.

Table 8: Standardized path coefficients among latent variables.

Latent variable *A*	Path direction	Latent variable *B*	Path coefficient
Service equipment operating level	→	Road traffic operating status	0.80
Management and control equipment operation level	→	Road traffic operation status	0.72
Municipal boundary demand	→	Road traffic operation status	-0.61
Intracity demand	←→	Road traffic operation status	-0.78
Service equipment operation level	→	Intracity demand	0.76
Intracity demands	→	Municipal boundary demand	0.65
Municipal boundary demand	←→	Intracity demands	0.83

Based on the calculation methods above, the results are shown in Table 7.

It indicates ideal convergence validity when the AVE is greater than 0.5 and the combination reliability is greater than 0.8. As shown in Table 6, the SEM has good performances in the two aspects. The above results show that the revised SEM of road traffic operation status meets the relevant requirements and standards, effective factor division measurement, good fit of the sample data, and reasonable structure. It can accurately reflect the relationship between relevant influencing factors in the perspective of the urban road traffic operation state.

3.3.4. Path Analysis. Based on the basic structure of the SEM of the road traffic operating state, the path analysis is carried out and the path diagram is shown in Figure 4.

The range of path coefficient of latent variables is $[-1,+1]$. The path coefficient approaching to +1 indicates a stronger

Table 9: Standard path coefficients of latent variables and key measurement variables.

Latent variable	Measured variable	Path coefficient
Road traffic operation status	Average speed of key roads Y1	0.97
	Average speed of other roads Y2	0.93
	Key area traffic index Y3	-0.84
	Other area traffic index Y4	-0.83
Service equipment operation level	Network coordinated control intersection ratio X3	0.84
Management and control equipment operation level	Online rate road section monitors X5	0.86
Municipal boundary demand	Volume of traffic entering the intracity area X7	0.95
	High-speed traffic volume around the city X9	0.93
Intracity demand	Expressway traffic volume X10	0.88
	Arterial road traffic volume X11	0.92
	Traffic volume on other roads X12	0.80

positive correlation between variables, while the path coefficient approaching to -1 indicates a stronger negative correlation. And the closer the coefficient is to 0, the weaker the correlation between the variables [46, 47]. The specific values are shown in Table 8. The analysis shows that for the road traffic operation status, the operation level of service equipment and control equipment at the traffic supply level have a positive influence on it, between which the operation level of service equipment has a relative higher influence. Thus, the high operation level of service equipment will positively improve the operation status of road traffic. At the traffic demand level, the intracity demand generated by the driving of vehicles on the roads in the city and municipal boundary has a negative impact on the status of road traffic, while the impact from the intracity area is relatively greater. In addition to the status of road traffic operation, there are also obvious correlations between traffic supply and demand related factors, which are concentrated in the promotion of the generation and change of municipal boundary and intracity road traffic demand by the operation level of service equipment. It can be seen that higher service equipment operation level will not only improve the traffic operation status but also stimulate more traffic demand. There is also a correlation between the municipal boundary and intracity area in the aspect of road traffic demand. The main reason is that the two are strongly related at the spatial level. The dynamic traffic flow changes within the intracity area and municipal boundary will make influences on each other.

The relationship between the latent variables and their key measurement variables (the absolute value of the path coefficient is greater than 0.8) is shown in Table 9. The analysis shows that the key points of the measured variables, the average speed and key points of other roads, and the traffic index of other regions have a strong ability to explain the road traffic operation status. It can be seen that the average speed and traffic index can be distinguished according to the spatial scope to be more detailed and accurate. Show the real situation of road traffic operation status. The networked coordinated control of the ratio of intersections and the online rate of road checkpoint monitoring have the strongest ability to explain the two types of latent variables related to traffic supply. Except for the traffic volume for leaving the city, the traffic volume on all levels and types of roads in other cities has a strong correlation with traffic demand.

3.3.5. Suggestions on Road Traffic Control Strategies. By combining the path coefficients between the latent variables and the latent variables and key measurement variables, the following road traffic-related management and control strategy are proposed.

(a) Using the form of spatial division to split the macroscopic average speed of the whole road section or the whole area traffic index and taking factors such as safety into consideration can describe the road traffic operation status in a more comprehensive and detailed manner

(b) The interaction mechanism between the related factors of road traffic operation status is complicated. In order to improve the road traffic operation status, it is necessary to comprehensively consider key factors related to traffic supply and demand at the same time and clarify the internal interaction relationship between supply and demand

(c) Improving the operation level of road traffic service equipment can directly promote the operation of road traffic, but it will also stimulate more traffic demand. The two types of promotion exist at the same time, and there is a certain balance between them

(d) Focus on improving the operation level of road management and control equipment or restrain the road traffic demand to a certain extent. For example, reducing the traffic volume of private cars can relatively more effectively improve the overall road traffic operation status

3.3.6. Limitations of the Model. Due to the limitation of datasets, the selected influencing factors can be still enriched, and the research samples can be expanded. Further studies concentrating on the interaction mechanism among related influencing factors can be carried out with enriched data considering cases in different environments and seasons.

4. Conclusion

In summary, this paper establishes SEM, which can conduct a comprehensive and in-depth analysis and discussion on the interaction mechanism among influencing factors of road traffic operation status from the data perspective. It can quantify their interactions, provide new ideas for a comprehensive and in-depth understanding of the changes in road traffic operation status, and provide scientific support for formulating targeted, practical, and effective management and control strategies aimed at improving road traffic operation status.

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] L. Haipeng and F. Jingjing, "On the traffic operation evaluation for basic road sections of an expressway," *Technology & Economy in Areas of Communications*, vol. 15, no. 3, pp. 95-96, 2013.

[2] C. Xi, S. Guohua, and Z. Xi, "Design and analysis of traffic indicator calculation method for residential districts," *Journal of Highway and Transportation Research and Development*, vol. 36, no. 7, pp. 136–142, 2019.

[3] W. Lianyu and M. Yongfeng, "Supply-demand coordination development of urban road traffic system," *Journal of Traffic and Transportation Engineering*, vol. 4, pp. 58–61, 2004.

[4] B. Hua, W. Jianjun, and J. Yimei, "Evaluation of highway network traffic operation state based on point, line, and area," *China Journal of Highway and Transport*, vol. 31, no. 11, pp. 197–204, 2018.

[5] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering.*, vol. 7, no. 2, pp. 766–775, 2020.

[6] X. Xiaolong, H. Li, X. Weijie, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[7] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications.*, vol. 38, no. 5, pp. 968–979, 2020.

[8] Q. He, C. Wang, G. Cui et al., "A Game-Theoretical Approach for MitigatingEdge DDoS Attack," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[9] C. Hu, W. Fan, E. Zeng et al., "Digital twin-assisted real-time traffic data prediction method for 5G-enabled Internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2811–2819, 2022.

[10] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 2, pp. 1–21, 2021.

[11] X. Jianmin, W. Jia, and S. Yanfang, "Comprehensive evaluation of urban road traffic operation status based on game theory-cloud model," *Journal of Guangxi Normal University: Natural Science Edition*, vol. 38, no. 4, pp. 1–10, 2020.

[12] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the Internet of things: Mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[13] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, 2021.

[14] L. Wang, X. Zhang, T. Wang et al., "Diversified and Scalable Service Recommendation With Accuracy Guarantee," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 5, pp. 1182–1193, 2021.

[15] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[16] C. B. Li, Q. Y. Zhao, R. X. Guo, and L. L. Tian, "Research on the measurements of the disequilibrium degree of the urban agglomeration traffic supply and demand," *Advanced Materials Research*, vol. 912-914, 2014.

[17] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks (TOSN)*, vol. 17, no. 3, pp. 1–33, 2021.

[18] J. Jin, X. Zhu, B. Wu, J. Zhang, and Y. Wang, "A dynamic and deadline-oriented road pricing mechanism for urban traffic management," *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 91–102, 2022.

[19] Y. Liu, Z. Song, X. Xiaolong et al., "Bidirectional GRU networks-based next POI category prediction for healthcare," *International Journal of Intelligent Systems*, pp. 1–21, 2021.

[20] K. Chandan, A. Seco, and A. Silva, "A real-time traffic signal control strategy under partially connected vehicle environment," *Promet Traffic & Transportation*, vol. 31, no. 1, pp. 61–73, 2019.

[21] H. Fujii, "Multi-agent based traffic simulation at merging section using coordinative behavior model," *Computer Modeling in Engineering and Sciences*, vol. 63, no. 3, 2010.

[22] X. Xia, F. Chen, Q. He et al., "Data, user and power allocations for caching in multi-access edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1144–1155, 2022.

[23] Q. Zhenghao, *Principles and Applications of Structural Equation Modelling*, 2009.

[24] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[25] L. Yuan, Q. He, S. Tan et al., "Coopedge: a decentralized blockchain-based platform for cooperative edge computing," in *Proceedings of the Web Conference 2021*, pp. 2245–2257, Vitual, 2021.

[26] A. A. Igolkina and M. G. Samsonova, “SEM: structural equation modeling in molecular biology,” *Biophysics*, vol. 63, no. 2, pp. 139–148, 2018.

[27] A. Amraeinia, Y. Zuo, and J. Zheng, “Interface modification of TiO_2 electron transport layer with PbCl 2 for perovskiote solar cells with carbon electrode,” *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 741–750, 2022.

[28] X. Xu, H. Tian, X. Zhang, L. Qi, Q. He, and W. Dou, “DisCOV: distributed COVID-19 detection on X-ray images with edge-cloud collaboration,” *IEEE Transactions on Services Computing*, 2022.

[29] Q. Wentao and K. Lingzhen, “Path analysis of PPP project performance based on SEM model,” *Journal of Wuhan University of Technology: Information & Management Engineering*, vol. 39, no. 2, pp. 202–207, 2017.

[30] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. Wang, “Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system,” *IEEE Internet of Things Journal*, 2021.

[31] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, and L. Qi, “Robust collaborative filtering recommendation with user-item-trust records,” *IEEE Transactions on Computational Social Systems*, pp. 1–11, 2021.

[32] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, “Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT,” *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[33] X. Zhou, W. Liang, K. Wang, and L. T. Yang, “Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 171–178, 2021.

[34] Z. Cai, Z. Xiong, X. Honghui, P. Wang, W. Li, and Y. Pan, “Generative adversarial networks,” *ACM Computing Surveys.*, vol. 54, no. 6, pp. 1–38, 2021.

[35] X. Shiyuan, X. Chen, and Y. He, “EVchain: an anonymous blockchain-based system for charging-connected electric vehicles,” *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 845–856, 2021.

[36] R. H. Hoyle, “Structural equation modeling: concepts, issues, and applications,” *The Statistician*, vol. 45, 1996.

[37] X. Zhou, X. Yang, J. Ma, and K. Wang, “Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT,” *IEEE Internet of Things Journal*, 2021.

[38] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[39] P. M. Bentler and D. G. Bonett, “Significance tests and goodness of fit in the analysis of covariance structures,” *Psychological Bulletin*, vol. 88, no. 3, pp. 588–606, 1980.

[40] P. M. Bentler, “Comparative fit indexes in structural models,” *Psychological Bulletin*, vol. 107, no. 2, pp. 238–246, 1990.

[41] R. P. McDonald and H. W. Marsh, “Choosing a multivariate model: noncentrality and goodness of fit,” *Psychological Bulletin*, vol. 107, no. 2, pp. 247–255, 1990.

[42] C. Yang, Y. Shao, J. Zhou, Y. Wang, L. Wang, and P. Du, “Mortgage behavior analysis of family farm land management right based on SEM model,” in *Fifth International Conference on Economic and Business Management*, China, 2020).

[43] F. Dai, P. Huang, X. Xu, L. Qi, and M. R. Khosravi, “Spatio-temporal deep learning framework for traffic speed forecasting in IoT,” *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 66–69, 2020.

[44] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaaidi, “Multivariate deep learning approach for electric vehicle speed forecasting,” *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.

[45] J. Zhou, K. Cao, X. Zhou, M. Chen, T. Wei, and H. Shiyan, “Throughput-conscious energy allocation and reliability-aware task assignment for renewable powered in-situ server systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 3, pp. 516–529, 2022.

[46] J. Zhou, J. Sun, M. Zhang, and Y. Ma, “Dependable scheduling for real-time workflows on cyber-physical cloud systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7820–7829, 2021.

[47] F. Wang, M. Zhu, and M. Wang, “6G-enabled short-term forecasting for large-scale traffic flow in massive IoT based on time-aware locality-sensitive hashing,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5321–5331, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8783380, 25 pages
https://doi.org/10.1155/2022/8783380

Research Article

An Intelligent SDN-Based Clustering Approach for Optimizing IoT Power Consumption in Smart Homes

Amin Nazari, Fazeleh Tavassolian, Mahdi Abbasi, Reza Mohammadi, and Parsa Yaryab

Department of Computer Engineering, Engineering Faculty, Bu-Ali Sina University, Hamedan, Iran

Correspondence should be addressed to Mahdi Abbasi; abbasi@basu.ac.ir

Received 28 January 2022; Revised 12 March 2022; Accepted 18 March 2022; Published 5 April 2022

Academic Editor: Shunmei Meng

As a novel technology, the Internet of Things (IoT) has many applications in diverse fields, especially in smart homes. IoT includes a variety of communication networks and technologies which facilitate communication between heterogeneous devices. One of the primary challenges of IoT is energy consumption. This paper introduces a new Software Defined Network-based (SDN-based) clustering approach using intelligent algorithms for energy conservation in IoT. The proposed method uses an evolutionary algorithm to identify the required number of clusters and ensures their distribution in the environment. A virtual network is also employed to ensure network coverage and the formation of balanced clusters. Clustering, steady, and routing are the main steps of the proposed method that the clustering step is done in SDN. By expanding the steady phase and leveraging energy-based greedy routing, the network's lifetime increases. After simulation in MATLAB, the proposed method is tested then the results are compared with other well-known algorithms. The evaluation results indicate that the proposed method has improved in terms of metrics such as energy consumption and network lifetime. The proposed approach improves energy consumption by 31%, 28%, 8% and 21% than FPA, MCFL, BEEG and NodeRanked respectively. The lifetime has been improved by 34% and 71% than BEEG and NodeRanked, respectively, and more than 100% for MCFL and FPA.

1. Introduction

In recent years, many researchers and leading technology companies have considered the application and advancement of the Internet of Things (IoT) in various fields, such as smart cities, smart homes, agriculture, and intelligent animal husbandry. IoT consists of numerous self-organized, tiny, and low-cost sensor nodes. IoT nodes, through their sensors, can collect and transmit data from the environment to the base station (sink) [1]. In addition, IoT is responsible for tracking network coverage issues and transferring monitoring results from the sink to the administrator [2, 3]. The monitoring center will derive valuable information using artificial intelligence, machine learning, and data mining algorithms from the collected data and provide it to administrators via mobile applications.

Wireless sensor networks (WSNs) are one of the most useful foundations for implementing IoT architecture. The IoT provides the ability to connect various things to the internet. IoT sensors can automatically track, process, and route data and allow various real-time applications. Also, they allow diverse real-time applications such as smart cities and smart homes to be developed [4]. WSN have been in many fields during the past two decades, such as habitat monitoring [5], battlefield target tracking [6], health monitoring [7], gas monitoring [8], and smart homes [9]. The advantages of these networks are combined in a smart home. A smart home can make numerous aspects of health, social, and emotional care more efficient and sustainable for its occupants [10].

Heterogeneous nodes and energy consumption are two primary limitations of the IoT. Flexible layered architecture is required, to deal with the heterogeneous nodes. Software-defined network (SDN) is a modern approach to growing network flexibility. The SDN network distinguishes the control plane from the data plane. This separation makes it possible to dynamically control the network, provide a better quality of service (QoS), and offer network management consistency and simplicity [11]. SDN is a promising approach in WSN and IoT, allowing controllers to be isolated from sensor nodes. The SDN controller decides about various network parts and defines the current input rules based on the information it receives from the network [4]. SDN enables network administrators to manage network services without lower-level details being assessed [12].

Besides, the inadequate battery capacity of the sensor nodes is one of the most critical limiting factors. So, limiting energy use, which directly affects the network lifetime, is one of the vital problems in these networks [13]. Clustering is a key and popular technique for prolonging the network lifetime and efficient network management. It provides a range of benefits, including reducing intracluster communication, balancing the traffic load, and improving its scalability [14].

This paper introduced an energy-efficient clustering-based routing method. Its primary focus is on balanced and distributed clustering, which helps to extend the network's lifetime. The following are the key features of the proposed approach:

(i) Provision of a framework for balanced clustering based on SDN architecture

(ii) Clustering based on multiobjective optimization algorithms

(iii) Use greedy distributed routing

The following is the paper's structure: Section 2 discusses the research literature. We have used a three-tier model for system modelling, which is reviewed in section 3. The proposed method includes phases of set-up, steady state, routing, and reclustering, which will be discussed in section 4. The simulation results are given in section 5. Finally, a review of the achievements and conclusions is provided in section 6.

2. Related Work

This section discusses the first various clustering methods then evaluates SDN-based approaches. Generally, clustering algorithms can be classified into the following categories:

(i) Hierarchical clustering algorithms

(ii) Virtual grid-based clustering algorithms (based on virtual grid)

(iii) Fuzzy logic-based clustering algorithms (based on Fuzzy logic)

(iv) Based on metaheuristic algorithms (metaheuristic-based clustering algorithms)

The key aim of hierarchical clustering is to preserve the energy levels of clusters by using multistep paths. The LEACH method is the most well-known hierarchical clustering method [15]. This method uses a random probability function to pick cluster-head nodes. ([16] and [17] attempt to enhance the LEACH method. In [18], each node is allocated a rank depending on the route's cost and the number of connections between nodes, and CHs are selected based on this rank. A routing approach for optimizing network lifetime and reducing data latency is proposed in [19]. The cluster-based routing protocol (CRPD) is presented in [20]. In CRPD, the energy efficiency has been enhanced by clustering and routing algorithms through periodic updating of the network topology. In [21], only nodes with an energy greater than 20% of the initial energy can be nominated as headers.

Grid-based methods use GPS or location detection algorithms to make nodes be aware of their geographical location [22]. An energy-based, scalable georouting was introduced in [23], where the last position of the sink is stored and updated in several intersecting nodes. The other nodes will locate the sink's last position by sending a message to the closest crossover node. The key emphasis of BEEG [24] is on getting the cluster size. The optimum size causes reduced transmission and energy consumption. In [25], the routing fixed-parameter tractable (RFPT) algorithm and the load balancing virtual grid are used for routing and less time complexity. It uses an FPT-approximate clustering algorithm and a supreme routing tree that optimally connects all nodes.

The high speed of inference in fuzzy logic makes it possible to use it in real-time environments [26]. The use of a fuzzy method to pick the CHs decreases computational complexity. A fuzzy-based approach to reducing battery consumption for WSN maintenance for smart homes is applied in [27]. In this approach, a fuzzy logic controller (FLC) sets the network nodes' sleep time dynamically. Multistage fuzzy clustering is implemented in [28]. The CHs are chosen based on residual energy, the number of neighbors, and the distance to the sink. A method named DUCF is introduced in [29]. In DUCF, parameters such as residual energy, base station distance, and node degree are considered fuzzy inputs. CH selection increases each round's energy consumption by using this method.

Metaheuristic algorithms can be used to routing and select CHs. A routing method based on an ant colony algorithm to prolong the network's lifetime was suggested in [30]. In [31], the flower pollination algorithm (FPA) was used to control the grid's energy level. A dynamic particle swarm optimization (PSO) clustering algorithm was used in [32] to determine the position of the CHs. Adaptive clustering based on node distribution makes cluster distribution more logical, effectively balancing grid energy consumption. A combination form of heuristic algorithm and K-means algorithm is implemented in [32]. It selects optimum clustering depending on the distance between the clusters, the distance from within the clusters, and the number of CHs. An improved bee algorithm is used to cluster creation. New clusters are created using based on the remained energy of the nodes. In [33], the authors presented a multiobjective cluster head selection algorithm for IoT networks in smart cities.

The introduction of SDN led to the development of several approaches using a centralized controller. The global knowledge of network devices like AUVs or sensors is accessible to the SDN-based controller [34] [35]. Under current conditions, the network can be managed effectively, and rules can be extended. In [36], DRL helps to monitor packets' flow in the WSN via SDN architecture, and the 2D and 3D convolution layers of two CNN models are used to evaluate. In [37], the Low-Power Network Routing Protocol (RPL) is introduced using the Multihop Clustering technique (MHC-RPL). In [38], the authors provide an SDN-based framework for Quality of Service-based routing on the Internet of Underwater Things. The primary approach of this method is to collect route information at the base station and select the best route based on the probability of packet loss and delay.

3. System Model

Today, the Internet of Things has resulted in the massive volume of data being produced. The use of traditional methods for processing this big data has encountered numerous problems. Cloud computing has been proposed, to address these issues. Many IoT applications, like smart city app, need latency-aware computations [39].

Processing IoT requests over the Cloud alone is not an efficient solution, especially for some time-sensitive applications. Because delays in data transfer and processing in the Cloud layer reduce system performance. Fog computing is introduced to address this issue. It acts as an intermediate layer between the Cloud layer and IoT devices. Fog devices can be placed close to IoT nodes, allowing latency to be noticeably reduced. Therefore, a three-tier architecture is proposed to better manage smart city applications.

Figure 1 shows the architecture of the system. Like [40], we use a three-tier architecture. In the first layer are IoT devices that communicate via wireless communications and, after collecting data through sensors, send data packets to the sink. Data is collected and processed in the sink. If more processing is required, the data is sent to the Fog layer. The Fog layer can be formed by one or more Fog domains. Tasks will be offloaded to the Cloud when more processing is required.

Here, our focus is on the first layer. The goal is to use the SDN architecture for optimal management in the first layer to minimize the need for processing in the Fog and Cloud layers.

Some instruments, such as cameras and mobile devices, are wired directly to AC power and do not have energy consumption restrictions. Others may be dispersed randomly into the environment or travel in the ambit, such as motion sensors, temperature, vibration, and sound. City power cannot, however, be used, and batteries are supplied. Equation (1) shows the energy consumption model [31].

$$E_{T_x}(l,d)=\begin{cases} l*E_{elec}+l*\varepsilon_{fs}*d^2, & \text{if } d\le d0 \\ l*E_{elec}+l*\varepsilon_{mp}*d^4, & \text{if } d>d0 \end{cases}, \tag{1}$$

where d_0 is obtained as Equation (2).

$$d_0=\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}. \tag{2}$$

That E_{T_x} represents the loss of transmitter and receiver energy, ε_{fs} is the booster energy in space, and ε_{mp} is the energy consumed by multipath emission. E_{Rx} is the energy to receive packets, obtained from Equation (3).

$$E_{Rx}(l)=L\times E_{DA}, \tag{3}$$

where E_{DA} is the energy of data aggregation. Therefore, Equation (4) denotes the total cost of transfer and receive.

$$E=E_{Tx}+E_{Rx}. \tag{4}$$

By combining smart homes, smart streets, and highways, a smart city will be created. IoT plays a vital role in these cities. This technology consists of numerous heterogeneous devices, which enhances the need for SDN. Therefore, in this paper, an SDN-based architecture is used.

4. The Proposed Method

The proposed protocol has four stages: set-up, steady state, routing, and reclustering. Each phase is divided into subsections. In the following, each of the phases is examined.

4.1. Set-Up Phase. First, the nodes transmit information such as position and energy level to the sink, and the network enters the setup phase. The set-up phase is divided into two parts: identifying the appropriate number of clusters and centralized clustering, which are performed in the SDN. After gathering sensor data, the sink utilizes the SDN to calculate the required number of clusters. The purpose of this subphase is to minimize energy usage. Because increasing the number of clusters increases energy consumption, a limited number of clusters ignores the benefits of clustering in addition to increasing energy consumption. The optimal number of clusters is determined using Equation (5) [41].

$$K_{opt}=\frac{\sqrt{N}}{\sqrt{2\pi}}\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}\frac{M}{d_{toBS}^2}}, \tag{5}$$

where K_{opt} is the number of clusters, N is the number of IoT devices, M is the area space, and $d_{toBS}$is the average distance between nodes and the sink, which is calculated according to Equation (6).

$$d_{toBS}=\int_A\sqrt{x^2+y^2}\frac{1}{A}dA=0.765\frac{M}{2}. \tag{6}$$

In the second stage of the set-up, the sink selects cluster heads according to the previous step information. The start-up process has a lot of computing and routing overhead. In SDN, the controllers, which provides a global view of the

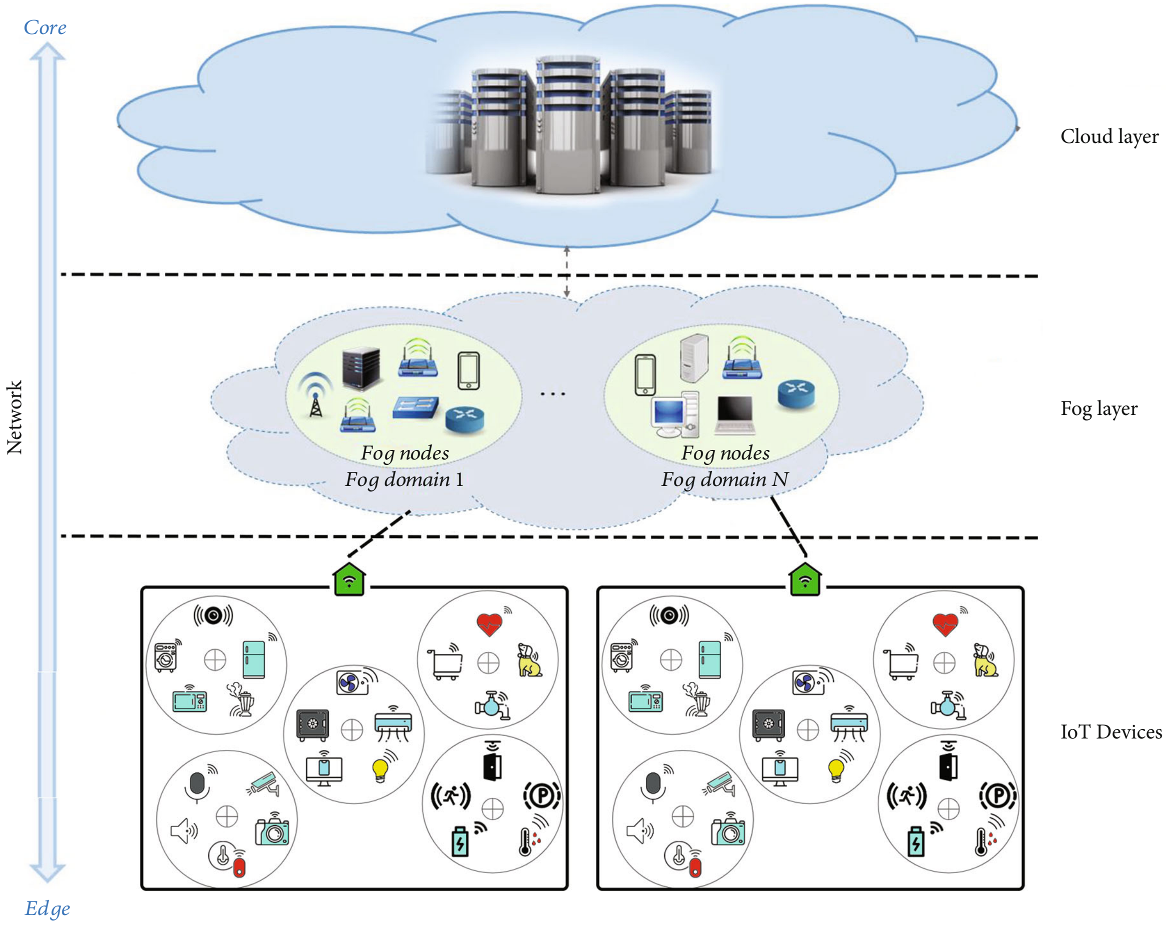


Figure 1: General system model.

whole network. To maintain a global network view, SDN controllers need to gather information from the whole of the network. To reduce traffic, this phase does periodically, not in every round. So, it is prevented from sending additional packets as much as possible. The reclustering process is then done locally to mitigate the transferring of packets. A genetic multiobjective algorithm considers multiple requirements for determining clusters. This approach considers parameters such as cluster centrality, balance, and distribution. The genetic algorithm runs in the sink and generates the virtual grid proportional to the number of clusters.

The virtual grid picks the initial CHs instead of random nodes. The virtual grid ensures that CHs are distributed throughout the environment. However, because the number of CHs is always smaller than the number of cells and the density of each cell's nodes is not equal with another one, the genetic algorithm chooses the final clusters. The objective function is defined to maintain the balance and centrality of clusters. Based on Equation (7), the general purpose is to minimize the square error value by having a given number of clusters. x and u_i denote the sensor's location and cluster centers, respectively.

$$F_{distance} = \min \sum_{i=1}^{k} \sum_{x \in S} x - u_i^{2}. \tag{7}$$

The number of cluster members is estimated to ensure that the clusters are balanced. The objective function, the disparity between the largest and smallest clusters, is minimized. Equation (8) ensures that the cluster balancing conditions are met.

$$F_{balance} = \min\left(\max\left(\sum_{k} |x \in u_k|\right) - \min\left(\sum_{k} |x \in u_k|\right)\right). \tag{8}$$

Consequently, the final objective function of the sum of Equations (7) and (8) is obtained as Equation (9).

$$\text{objective Function} : \min(\alpha \times F_{distance} + \beta \times F_{balance}). \tag{9}$$

Algorithm 1 shows the set-up phase algorithm. It is implemented in the sink and begins by gathering data from all IoT devices and calculates the number of clusters required. It then constructs a virtual grid and selects a candidate cluster head from each cell at random. This is necessary for cluster head distribution throughout the environment. Finally, the genetic algorithm chooses the final cluster heads to create balanced clusters.

```
Input: $IoT\ devices\ information, N, \varepsilon_{fs}, \varepsilon_{mp}, M, d\_toBS$
Output: $Clusters\ Heads$
Data:Sensors energy level, location
$K_{opt} = (\sqrt{N}/\sqrt{2\pi})\sqrt{\varepsilon_{fs}/\varepsilon_{mp}}(M/d^2_{toBS})$
Set genetic algorithm parameters
  Begin:
    Create an initial population
    $Evaluation\ individuals//based\ on\ objective\ function$
      do
        for ($\mathbf{i} = 0$ ; $\mathbf{i} \le |\mathbf{N_S}|/2$ ; $\mathbf{i}$++)
          $Select\ two\ parents$ in the $population$
Generate two offspring by crossover operation between two parents
          Insert two offspring into new generation list
        end for
          $Mutation\ some$ of $offspring$ in the $new\ generation\ list$
        $Evaluation\ individuals$
      $Select\ n\ top$ of the $list\ as\ new\ population$
      While$stop\ condition\ reached$
    End
$Cluster\ Heads = Best\ individuals$
```

Algorithm 1: Set-up phase algorithm.

4.2. Steady-State Phase. After selecting the CHs and sending the CHs information to all nodes, the non-CH nodes are connected to the nearest CH by receiving the header information and the form of the final clusters. Each CH assigns its cluster members a Time Division Media Access (TDMA-based) scheduler and notifies the nodes. Therefore, cluster members are active only in their time slots and send information that improves energy consumption.

At this phase, the nodes sense the environment and send the collected data to CH. Instead of sending several packets, CHs send a packet to the sink by aggregating data. CH may act as a header in the next round and does not need a recluster phase if CH energy is not below the threshold. Several steady phases are implemented, for one setup phase. So, the amount of energy saved in this method, assuming that j is the number of steady state with one setup phase, will be equal to Equation (10).

$$E_{Saved} = j \times \left(\sum_{\alpha \in N} \left(E_{elec} + \varepsilon_{fs} \times d^2\right) \times L + \sum_{\beta \in N} \left(E_{elec} + \varepsilon_{fs} \times d^4\right) \times L + N \times E_{Rx} \right), \tag{10}$$

where α and β are a subset of nodes that $\alpha \cup \beta = AllNodes$ and $\alpha \cap \beta = \varnothing$, when the energy of a CH is lower than the threshold, that CH selects the most appropriate node as its CH based on its member's information and transmits the new CH information to the others. The threshold value for each cluster is calculated as Equation (11).

$$E^k_{\min} = \gamma.\overline{E^k_r}. \tag{11}$$

That γ is a numerical value between (0, 1), and $\overline{E^k_r}$ is the average energy of the cluster member nodes. The second phase algorithm is shown in Algorithm 2. This algorithm is executed by the cluster head and determines the time of sending the data to the sink and the time of performing the clustering again.

4.3. Reclustering Phase. If CH energy is smaller than the threshold value, CH locally identifies one of the cluster members as CH in the next round. If the CH energy exceeds the threshold value, the CH will reach the routing phase directly. As the sensors' computing power is much smaller than that of the sink, Equation (12) is used (for determining the new cluster head) that does not require much computing power.

$$W_i = \frac{E_i * E^k_r}{\sum_{j \in K} \left| x_i - x_j \right|}. \tag{12}$$

W_i determines the fitness criteria of each node for CH, where

E_i is the energy of the node, E^k_r is the average energy of the nodes in the cluster, and the denominator determines the distance of each node to the candidate node. Algorithm 3 shows the reclustering algorithm. It performs reclustering locally, with the minimum number of packets possible.

4.4. Routing Phase. Each CH sends data through intermediate steps after collecting and aggregating data instead of directly sending it to its sink. The cluster obtains the value of the objective function given in Equation (4) for all adjacent clusters, utilizing information such as the neighbor's position and energy level. Consequently, neighbors with fewer distances from the sink and more energy from the other neighbors than the sink's distance are candidates for the next step. The routing phase is indicated in Algorithm 4. This algorithm is implemented in clusters' head and they send data to the sink in several steps using intermediate nodes.

```
while (true):
    foreach (Node in Nodes)
        Node sense area
        Node creates a data packet and send it to their own CH
    end foreach
    foreach (CH in Cluster Heads)
        CH aggregate data packets
        CH Routing Data Packets toward sink (See Alg.4)
        if (Energy_CH < E_Threshold)
            Run ReClustering algorithm(See Alg.3)
        end if
        if (number_of_dead_nodes == total_number_of_nodes)
            Break;
        end if
    end foreach
End while
```

Algorithm 2: Steady-state phase.

```
Input: Current Cluster Head, Cluster Members
Output: New Cluster Head
Begin
    $E_{max} = 0$;
    foreach (Node i in Cluster Members)
        $W_i = E_i * E_r^k / \sum_{jeK} |x_i - x_j|$
    end foreach
    foreach (Node in Cluster Members)
        if ($W_{Node} < W_{max}$)
            $NewCH = Node$
            $W_{max} = W_{Node}$
        end if
    end foreach
End
```

Algorithm 3: Reclustering phase.

5. Experimental Result

In this section, the proposed method evaluated in five different scenarios. And it compared with recent algorithms, such as NR_LEACH [15], BEEG (GRID) [24], Adaptive MCFL [28] and FPA [31]. NR LEACH is a hierarchical protocol, BEEG is chosen from network-based protocols, MCFL comes from fuzzy-based protocols, and FPA is a metaheuristic protocol. MATLAB software was used for evaluation. The parameters assessed and studied in this paper include the network's lifetime, number of alive nodes, and death of the first node and average energy consumption, the routing overhead, and the computational complexity.

5.1. Scenarios. The different algorithms for grids of (a) 100 × 100, (b) 150 × 150, (c) 200 × 200, (d) 250 × 250, and (e) 300 × 300 m^2 were assessed in the same conditions and with the simulation parameters presented in Table 1. The nodes are assumed to be altered in the environment randomly. The same scenarios are used for a fair assessment of the algorithms. Besides, CBR traffic is considered for all algorithms with the same transmission rate.

5.2. Energy Consumption. Energy consumption is one of the most important criteria of the Internet of Things. Figures 2(a)–2(e) show the result of the various algorithms for all scenario in average energy consumption. Finally, Figure 2(f) shows the average energy consumption of the various scenarios.

The lower the energy consumption, the higher the algorithm efficiency. The proposed algorithm runs the clustering phase locally in comparison to other methods. This local selection avoids the transfer of additional packets to the sink and reduces network traffic. The proposed solution retains its consistency with increasing grid size, whereas the FPA, MCFL, and Node-Ranked methods are not flexible. The BEEG is scalable and the Node-Ranked performs better in small networks than large ones.

The average energy consumption in all networks is shown in Figure 2(f). Generally, there is a probability that energy consumption will rise with an increasing number of nodes. However, in the 150 × 150 scale network, the three proposed approaches, BEEG, and Node-Ranking, indicate decreased energy consumption; this may be caused by a reduction in the sensor area's node density.

5.3. Death of the First Node. When different tasks are distributed among nodes, the first node is the last to die. The half-life node criterion is given in Figure 3 with the death of the first node. Figures 3(a)–3(e) show the criteria half-life node and first dead node for different scenarios. Figures 3(f) and 3(g) demonstrate the first dead node and the half-life node for all states, respectively.

Comparison of various methods indicates that the proposed approach works better than others. It uses the SDN architecture intelligently to select CHs and balance clusters. The current CH also handles the local selection of the next cluster head. As a result, selecting a specific node a CH-

```
Input: Neighbor Nodes(CHs), Sink Location, Energy Consumption Model
Output: Best Next Hop
$E_{\min} = \inf$
 Begin:
    foreach (neighbor in Neighbors)
        $d_n$ = distance CH to the neighbor
        if ($d_n < d_0$)
            $E_{cn} = l * E_{elec} + l * \varepsilon_{fs} * d_n^2$
        Else
            $E_{cn} = l * E_{elec} + l * \varepsilon_{fs} * d_n^4$
        end if
        $d_s$ = calculate distance neighbor to sink
        if ($d_s < d_0$)
            $E_{cs} = l * E_{elec} + l * \varepsilon_{fs} * d_s^2$
        Else
            $E_{cs} = l * E_{elec} + l * \varepsilon_{fs} * d_s^4$
        end if
        $EnergyCons = E_{cs} + E_{cn}$
        if ($EnergyCons < E_{\min}$)
            $E_{\min}$ = EnergyCons
            Best Next Hop = neighbor
    end if
    end foreach
 End
```

Algorithm 4: Routing phase.

Table 1: Simulation Parameters.

Parameter	Values(amounts)
The number of nodes	100, 150, 200, 250, 300
Sink position	District center
Pack size	4000 bits
Efs	10 × 10E − 14
Emp	0.0013 × 10E − 14
ADE	5 × 10E − 9
xTE	50 × 10E − 9
ERx	50 × 10E − 9
The initial energy of the node	0.5 J

node repeatedly is prevented. So, distinct network duties are distributed across all nodes.

Furthermore, the nodes with the most significant energy are used in routing, creating a shorter path to the sink. Considering the CHs energy, the longer the steady-state stage prevents the spread of additional packets. The results indicate that the proposed methods and BEEG are scalable. FPA clustering and MCFL need to send sensors data in the sink. The collection of this data requires a significant number of packets to be sent. Since delivery and reception of packets are the most critical factor in energy consumption, increasing traffic would waste energy resources. BEEG and Node-Ranked execute clustering locally and avoid additional packets from being sent. However, they cannot benefit from SDN.

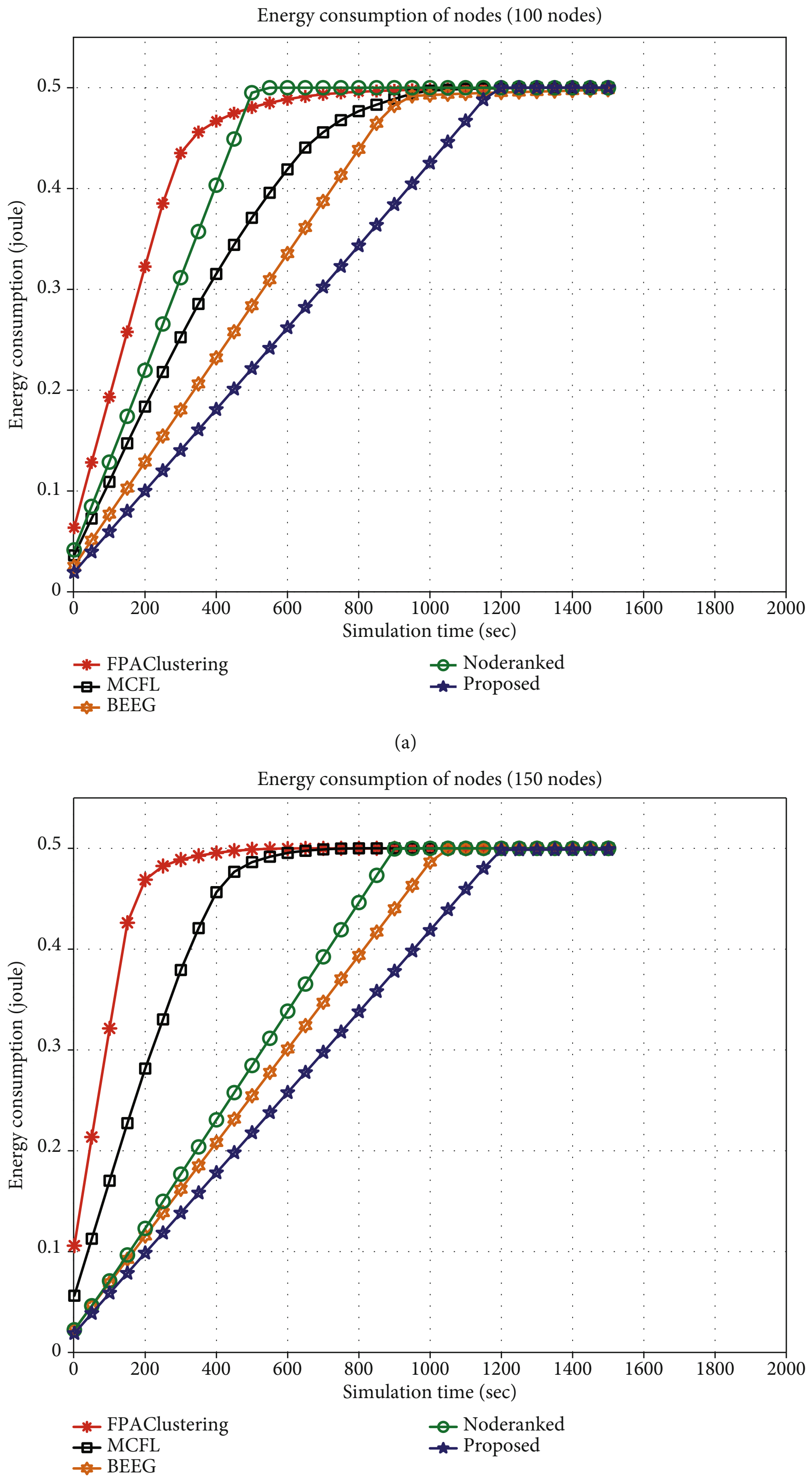
5.4. Network Lifetime. Increasing network lifetime is one of the most critical priorities of designing IoT. Figure 4 displays the network lifetime diagram with various methods. The death of 20% of nodes as a measure of network lifetime is regarded here.

As the nodes with the most energy are chosen in the reclustering phase, the network lifetime increases. The proposed method also ensures that clusters are balanced and that there are an adequate number of them.

5.5. Alive Nodes. The more active nodes, the more network coverage. IoT coverage is one of the qualities of service requirements (QoS). The number of packets increases in the network by increasing the number of nodes. Nodes lose a certain amount of energy to send and receive each packet. Thus, by increasing the number of packets, the nodes' energy reduces. Another benefit of clustering is the easier handling of the number of packets sent via data aggregation. Figure 5 indicates the number of live nodes for various scenarios.

The proposed method saves energy due to consideration of the energy criterion in the routing phase. In the final rounds, the network will practically lose its connection due to the intermediate nodes' death and will not have the requisite efficiency. Our approach distributes energy consumption among nodes by dividing the various tasks between them. BEEG, MCFL and FPA do not consider the distribution of tasks.

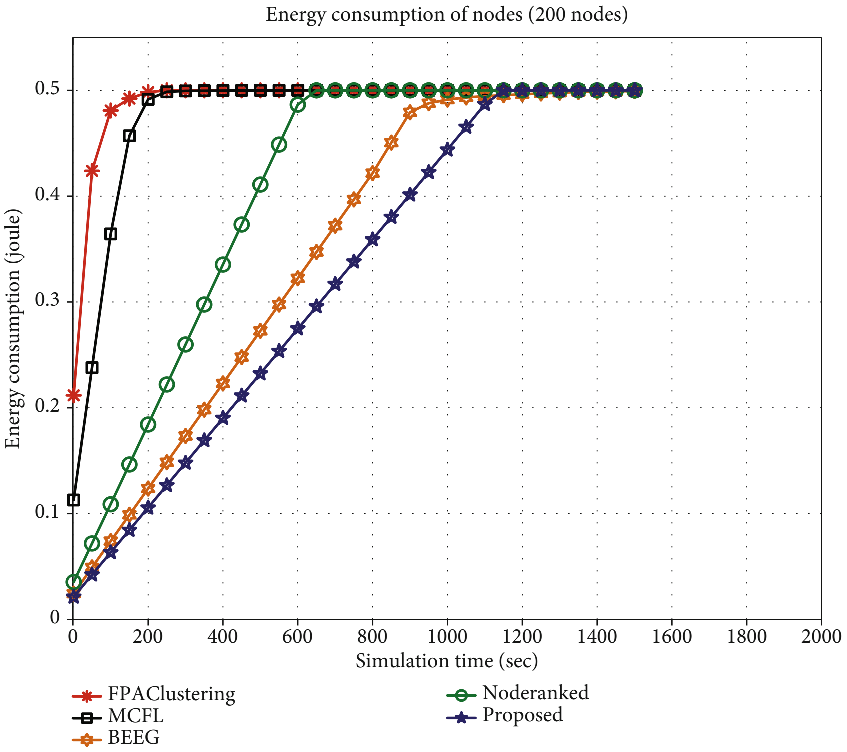
5.6. Routing Overhead. The routing overhead specifies how many routing packets for each data packet are sent. The lower the value, the better because it indicates that fewer



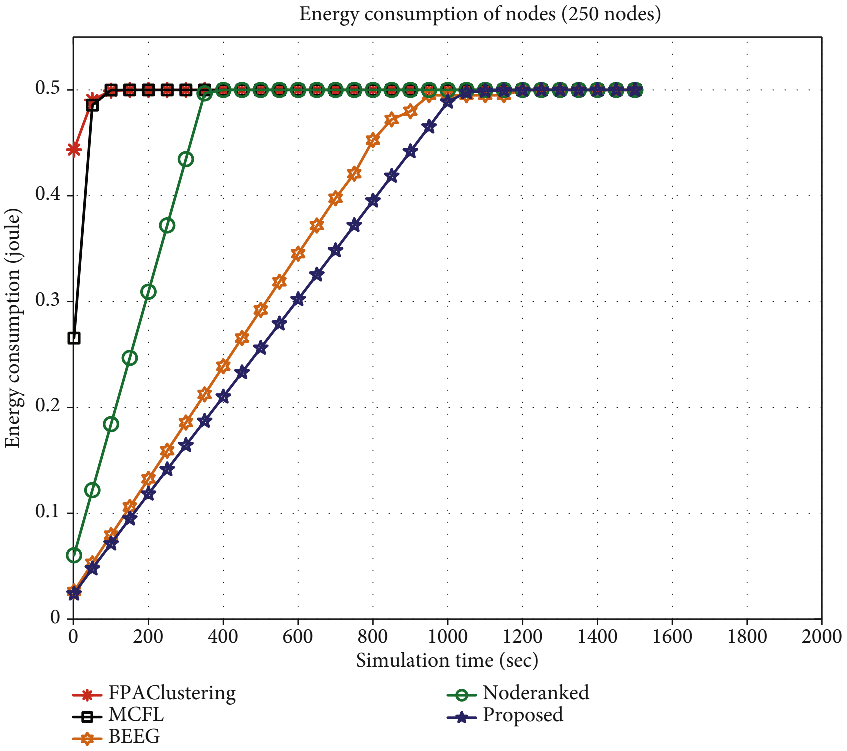
(a)

(b)

Figure 2: Continued.

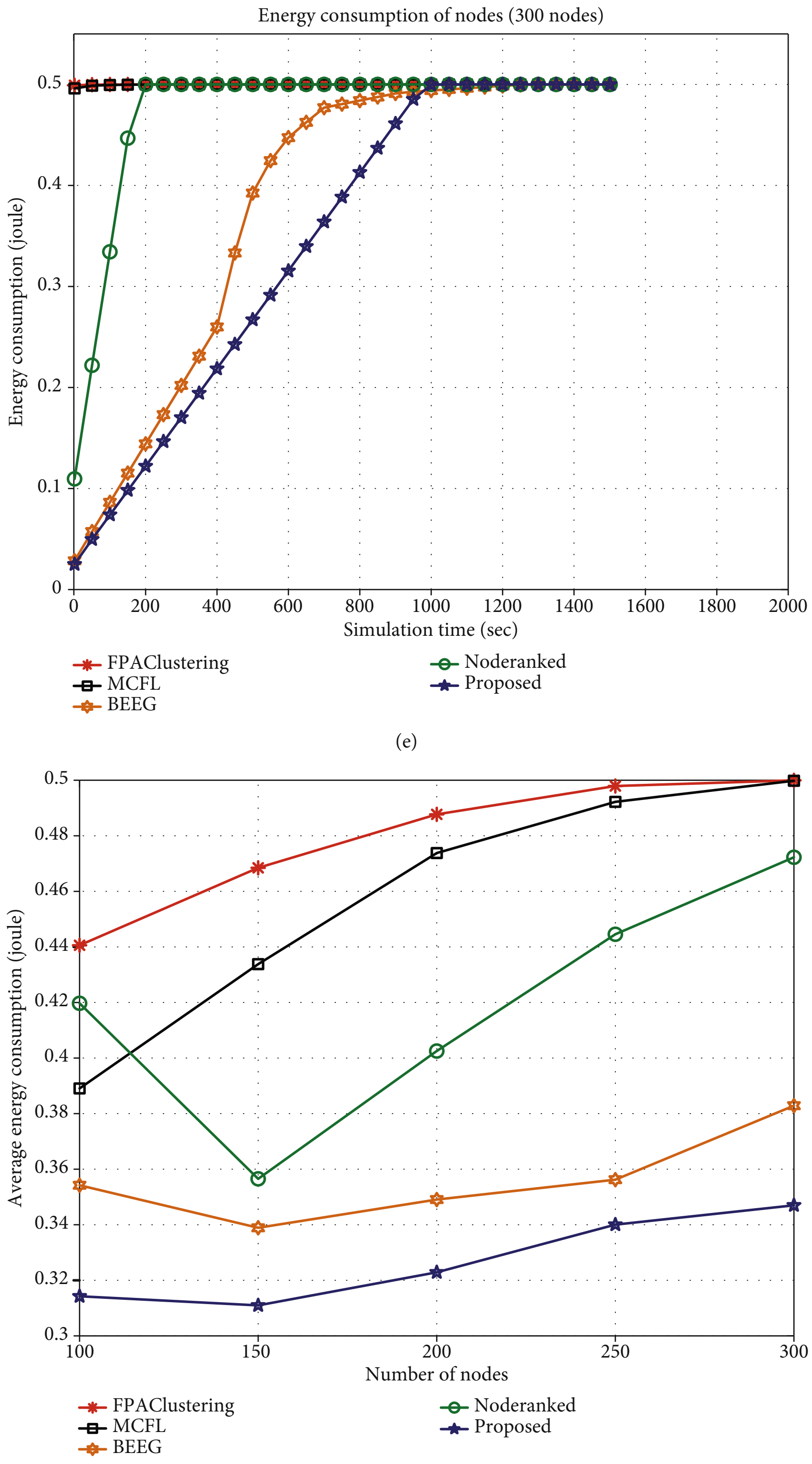


(c)



(d)

Figure 2: Continued.



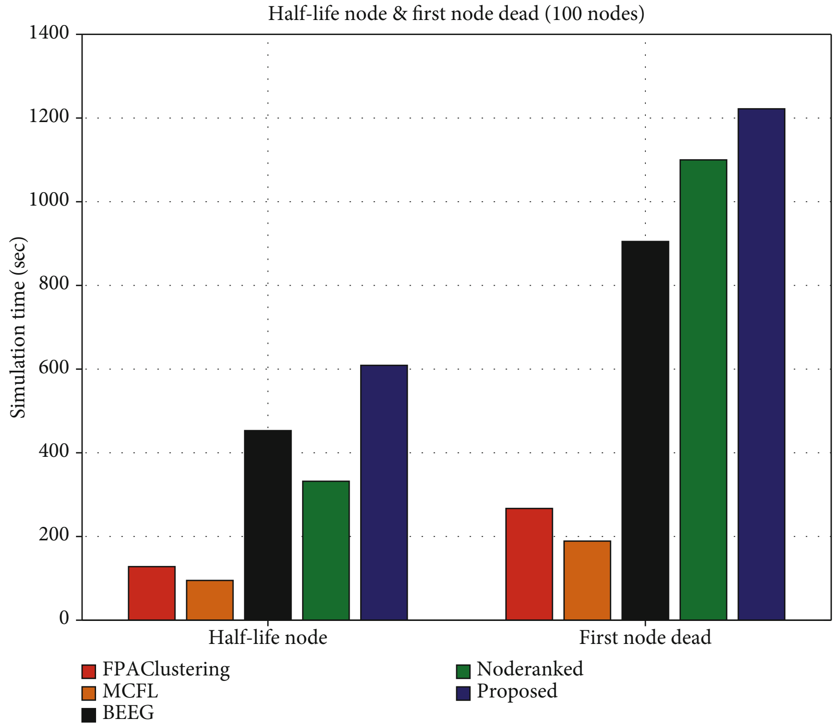
(e)

(f)

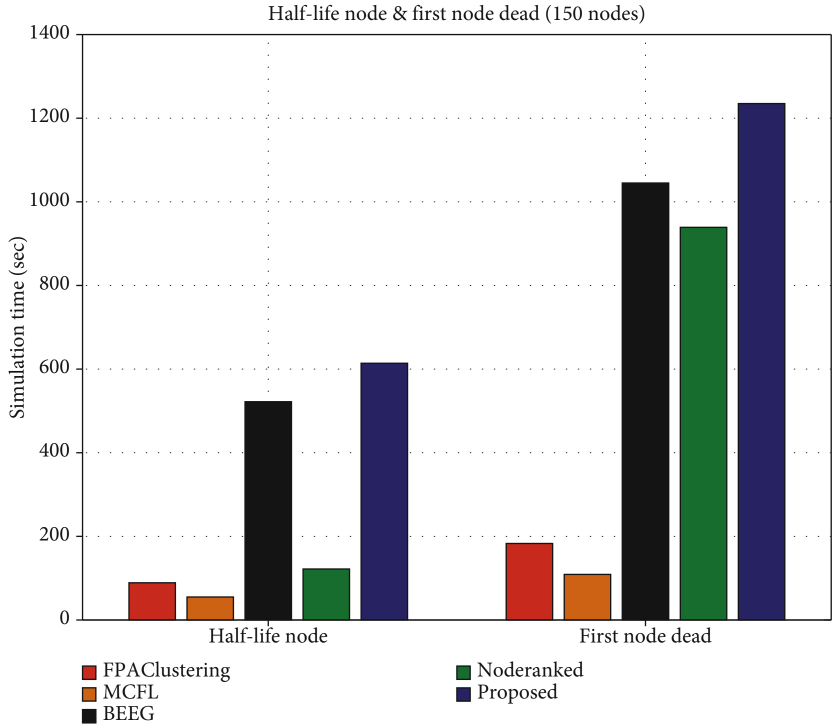
Figure 2: Average energy consumption.

routing packets have been sent. The routing overhead is seen in various situations in Figure 6.

The routing overhead in the proposed method is lower than the rest of the approaches. It is because of the increase in steady-state duration and local clustering. More routing packets are sent due to centralized execution in MCFL, Node-Ranked, and FPA clustering. But, the reclustering process is distributed locally in the BEEG.

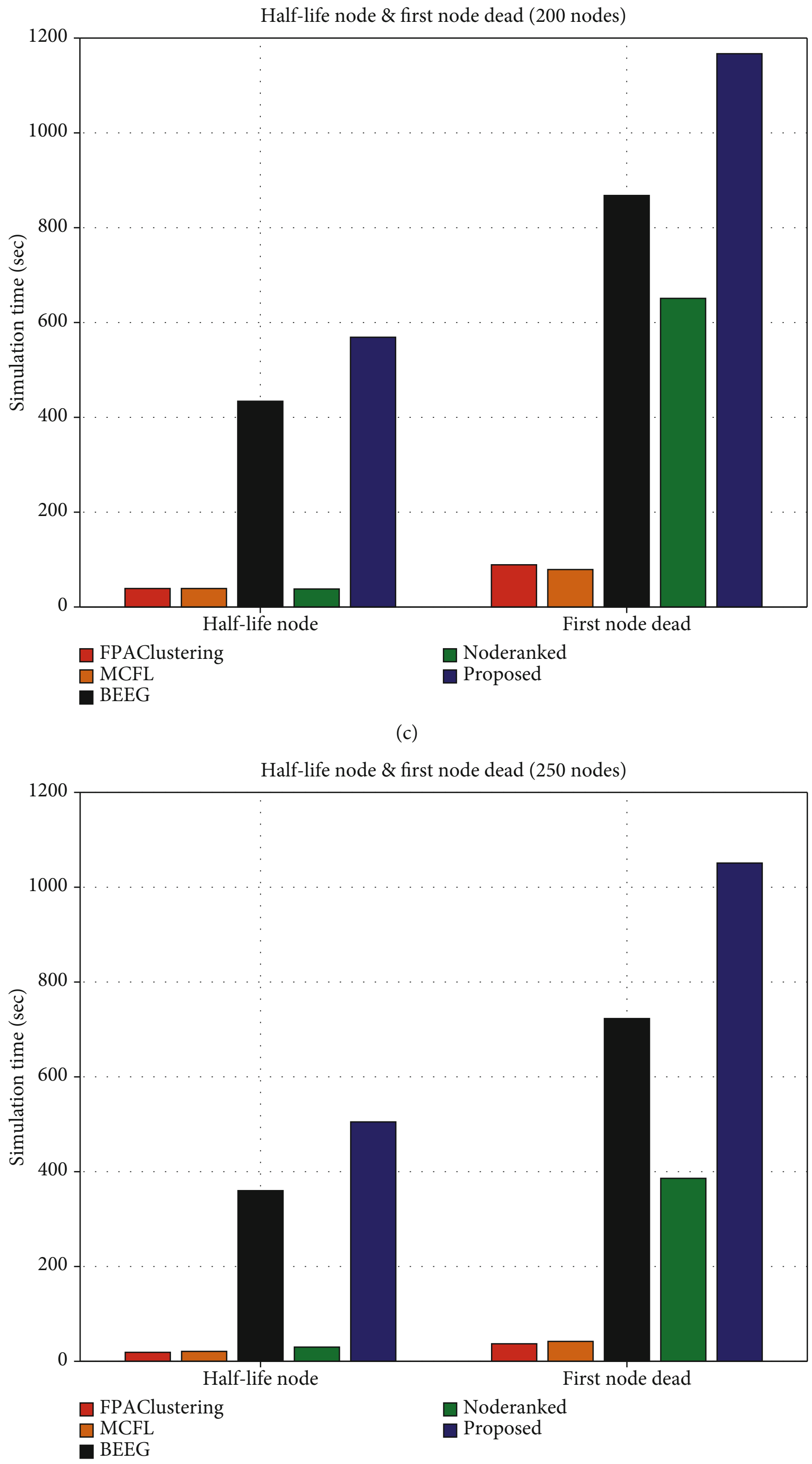


(a)



(b)

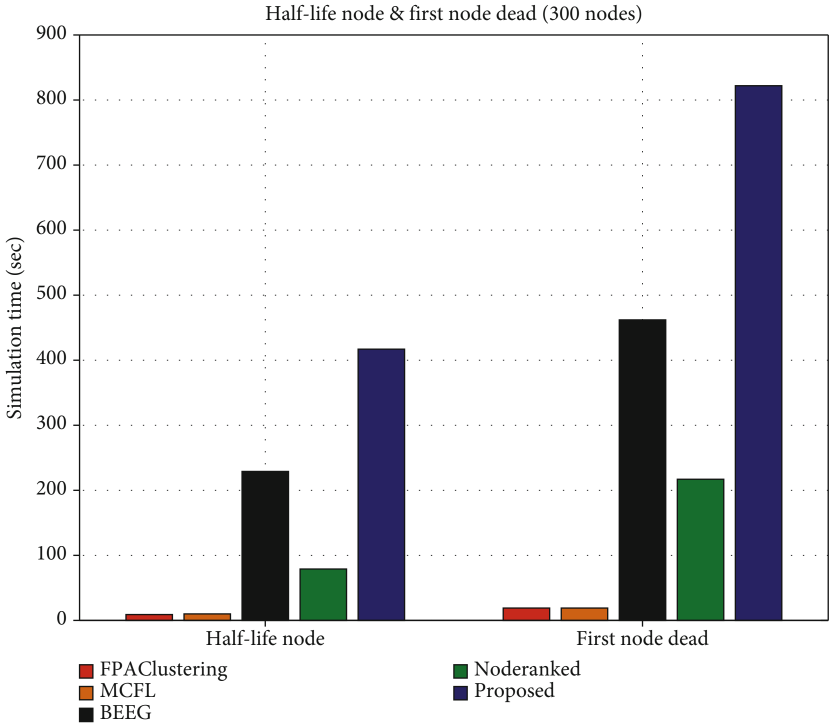
Figure 3: Continued.



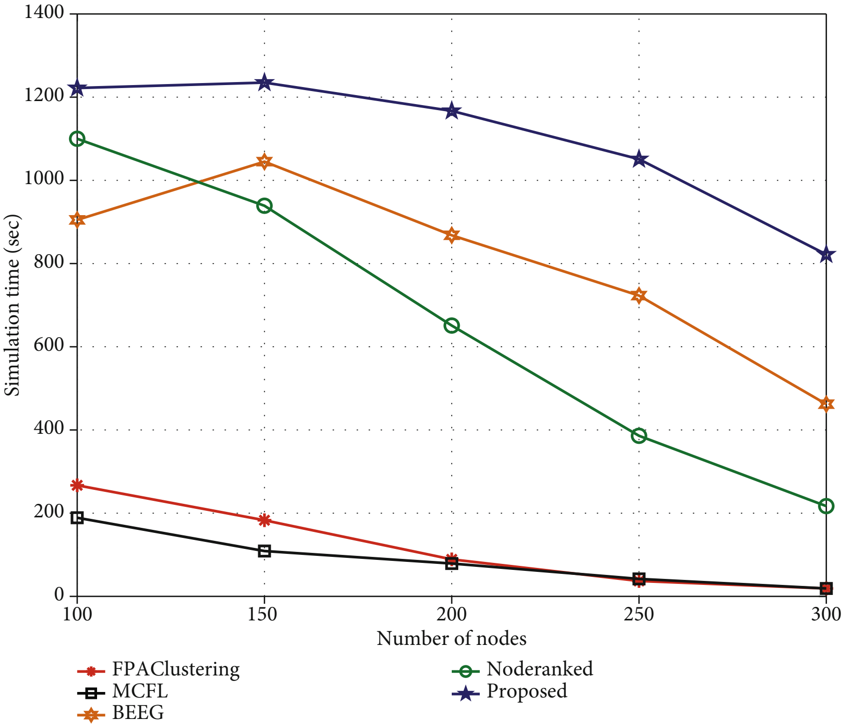
(c)

(d)

Figure 3: Continued.



(e)



(f)

Figure 3: Continued.

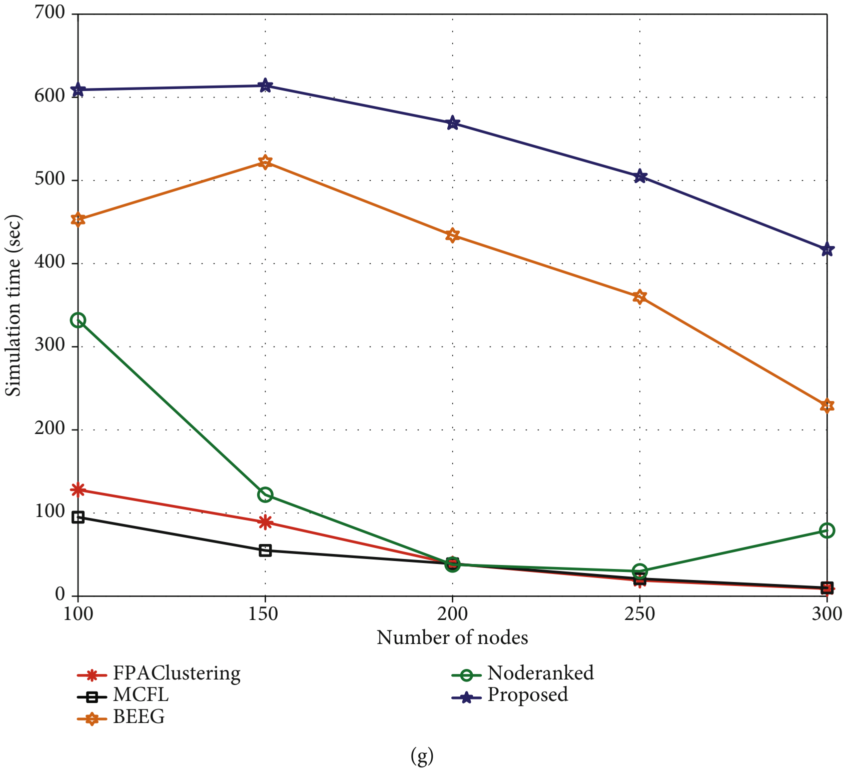


Figure 3: Comparison of the half-life nodes and the first dead nodes.

5.7. Computational Complexity. The FPA in each round requires determining and selecting the best CHs using the evolutionary algorithm. The time complexity of FPA equals to O(N_(iter)* N_pop* N_nodes)$O(N_{iter} \times N_{pop} \times N_{nodes})$. Where N_{iter} is the number of evolutionary algorithm iterations, N_{pop} is the original population, and N_{nodes} is the number of nodes. It needs to collect all network information in the sink for clustering. Therefore, this method is not scalable.

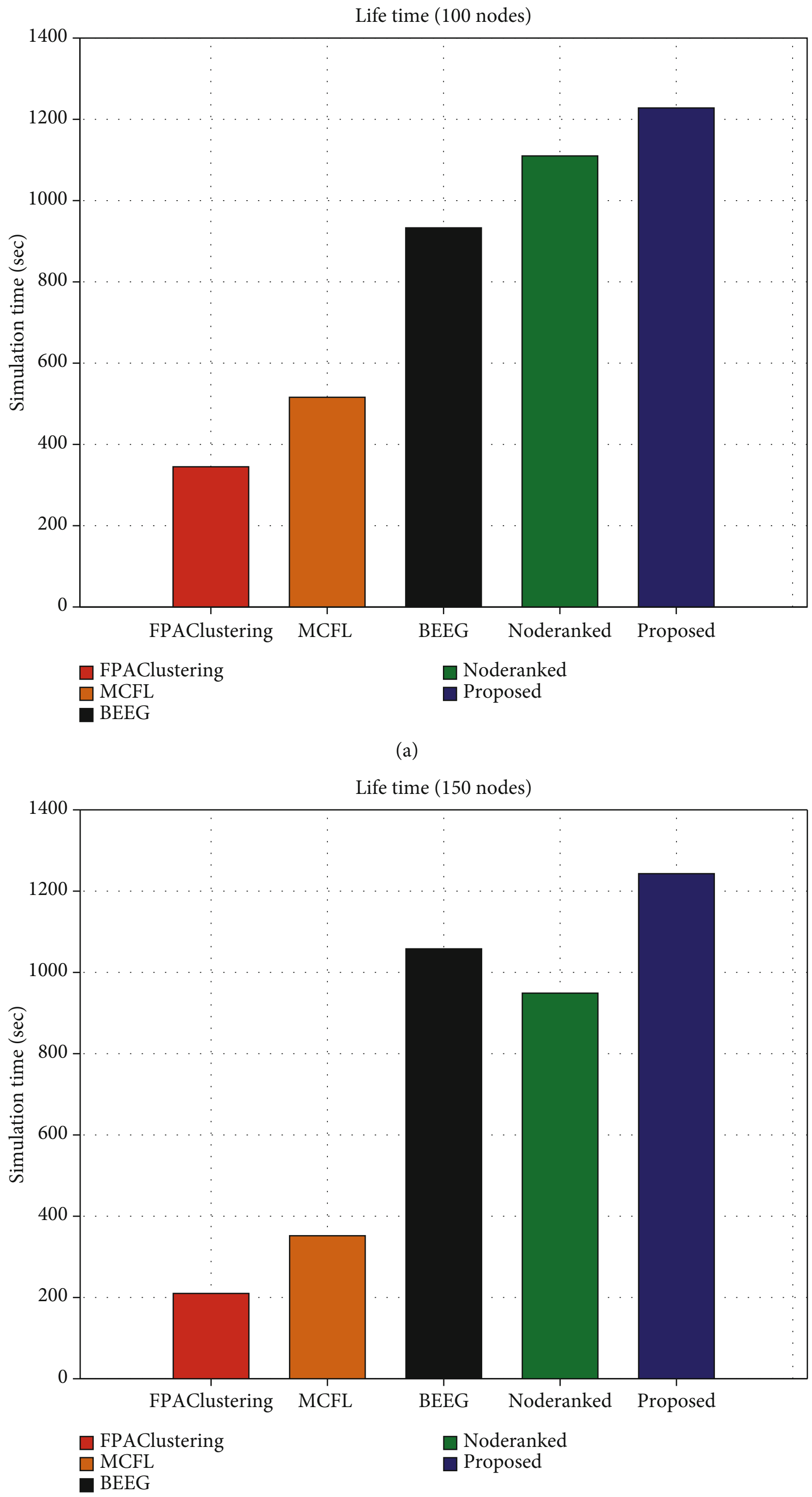
In the MCFL algorithm, clustering is performed in the sink. It attempts to improve the steady phase, and the CHs will remind in the next round if have the appropriate condition. MCFL uses a fuzzy scheme. The time complexity of a fuzzy system ranges from $O(NI \times NIF + (MOD + NI) \times L)$ to $O(NI \times NIF \times NID + (MOD + NI) \times L)$ base on the form of membership functions and the process of defuzzification (Kim, 2000). Where NI is the number of inputs, NIF is the number of fuzzy sets of inputs, NID is the number of discretization of the discourse input universe, and L is the number of rules.

The BEEG algorithm uses a virtual grid such that sensors information does not have to be collected in the sink. The clustering is performed locally depending on the maximum remaining resources. A node is chosen as the CHs in each cell. This approach guarantees that CHs are balanced and distributed and avoid any broadcasting packets. The complexity of time for this process is $O(k \times m)$, while k is the number of virtual network cells and m is the average number of members per cell.

The Node-Ranked algorithm has to rank the nodes, and this ranking is performed in the sink, so the broadcast problem influences it. The time complexity for choosing the right CHs is $O(nlogn)$. Because we have n nodes and must rank each node's value before selecting the k nodes with the highest rank value. Ranking with complexity $O(n)$ is completed; then, sorting with complexity $O(nlogn)$ is used to choose the top k nodes. First, n nodes are ranked, and then, k nodes with the highest rank are selected from them. The ranking complexity is $O\,(n)$, and sorting with $O\,(nlogn)$ complexity is used to select k nodes.

The proposed approach uses local clustering such that we do not need to broadcast packets. But to take advantage of SDN, information is collected periodically in the sink. The localization of clustering is also decreased with the improvements in the steady phase. The time complexity for finding the right CH in this method is $O(n)$. However, an evolutionary algorithm and virtual grid have been used to ensure clusters' distribution and equilibrium at the beginning of the network operations. Since this step is done rarely and periodically, it can be overlooked.

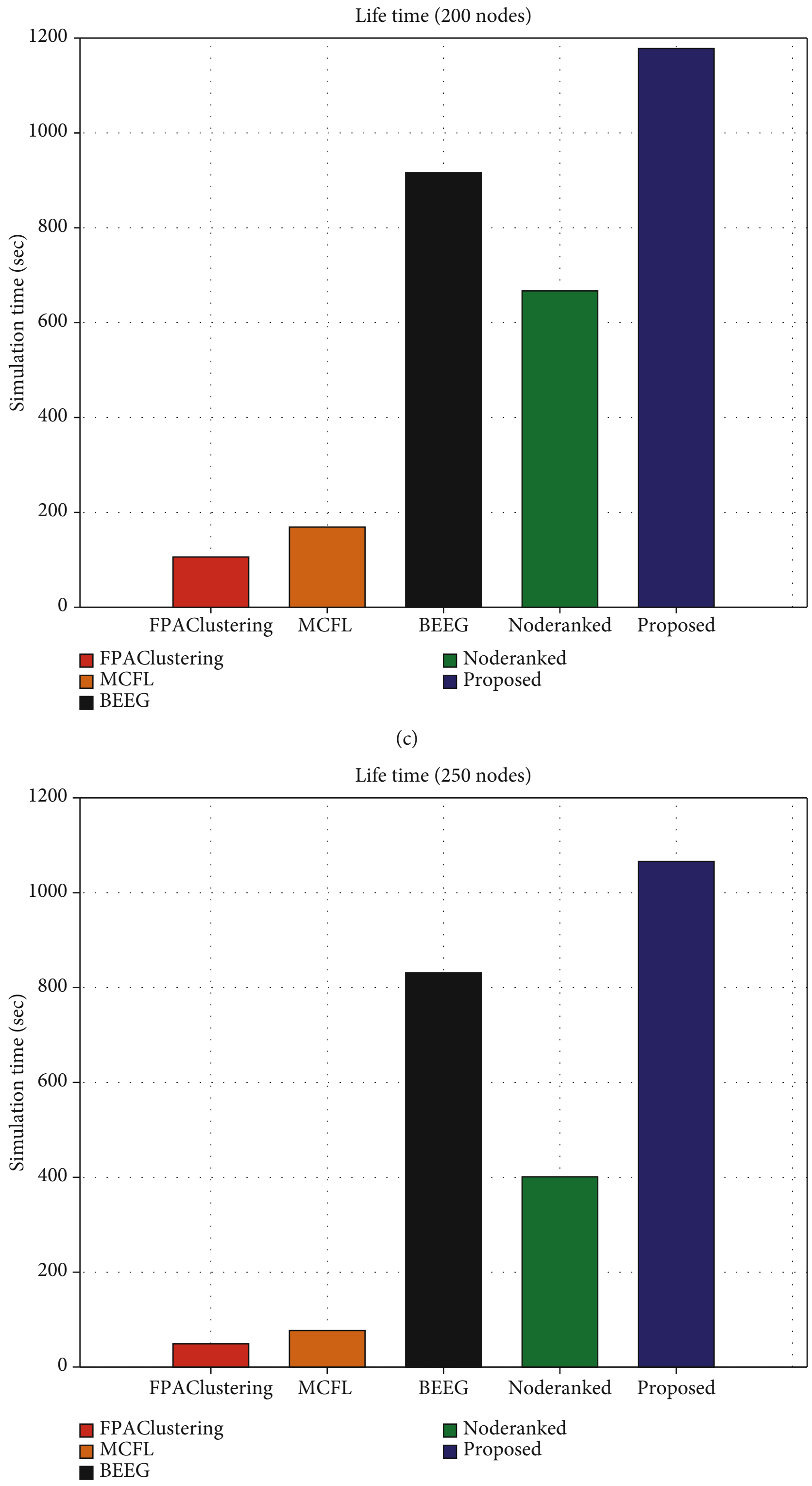
It is worth mentioning that since our proposed architecture takes the use of SDN, it considerably outperforms other solutions in terms of various performance metrics. As aforementioned in the previous sections, the number of clusters in the network significantly affects the network's energy consumption and routing overhead. For this reason, leveraging SDN architecture to compute the number of required clusters with respect to the current status of the network leads to conservation in energy as well as decreasing the overhead of routing. In a nutshell, the centralized nature of SDN helps to make optimize solutions in network management and



(a)

(b)

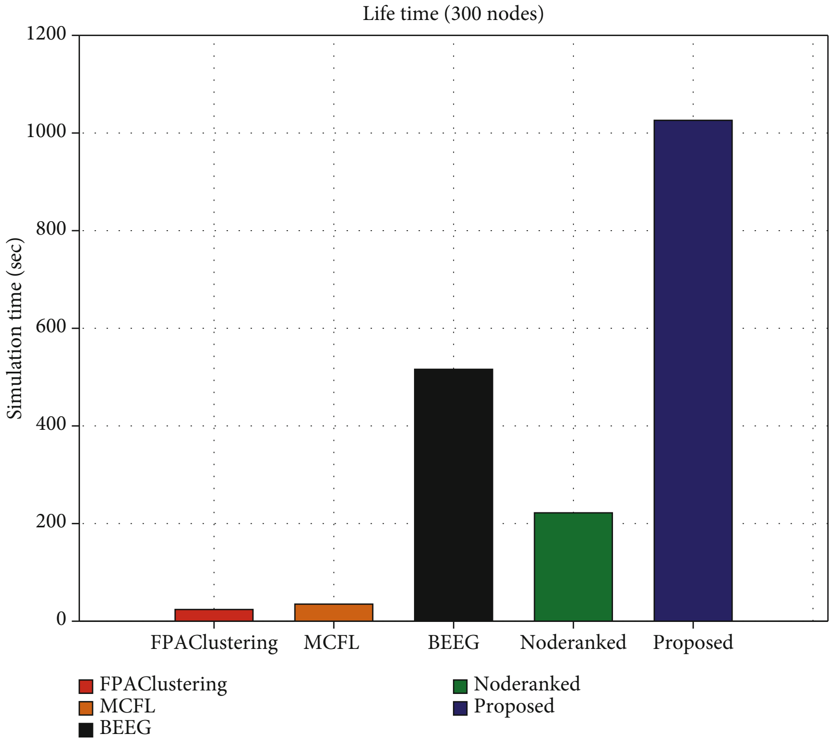
Figure 4: Continued.



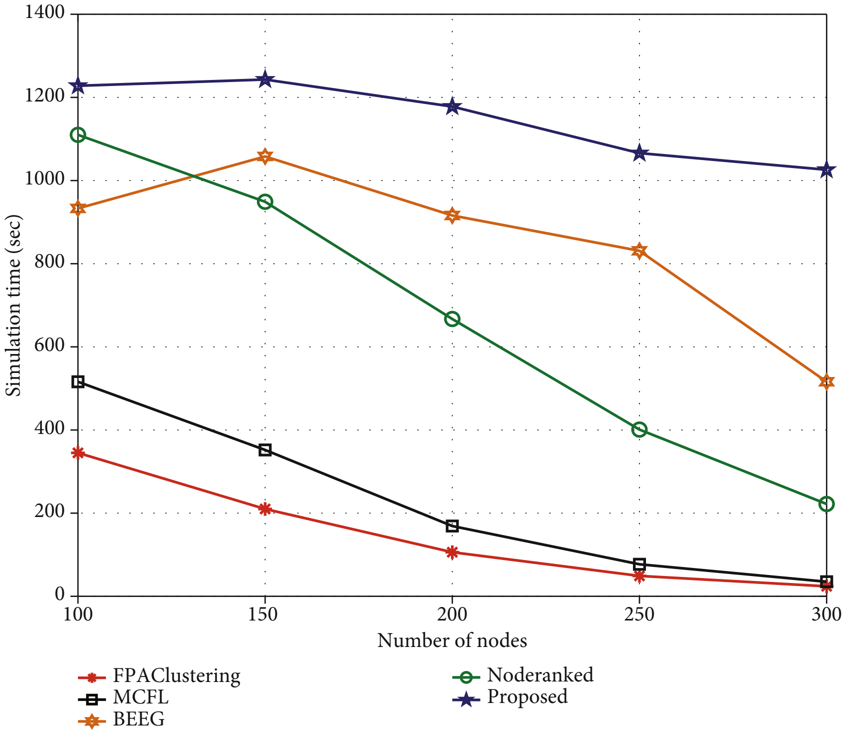
(c)

(d)

Figure 4: Continued.

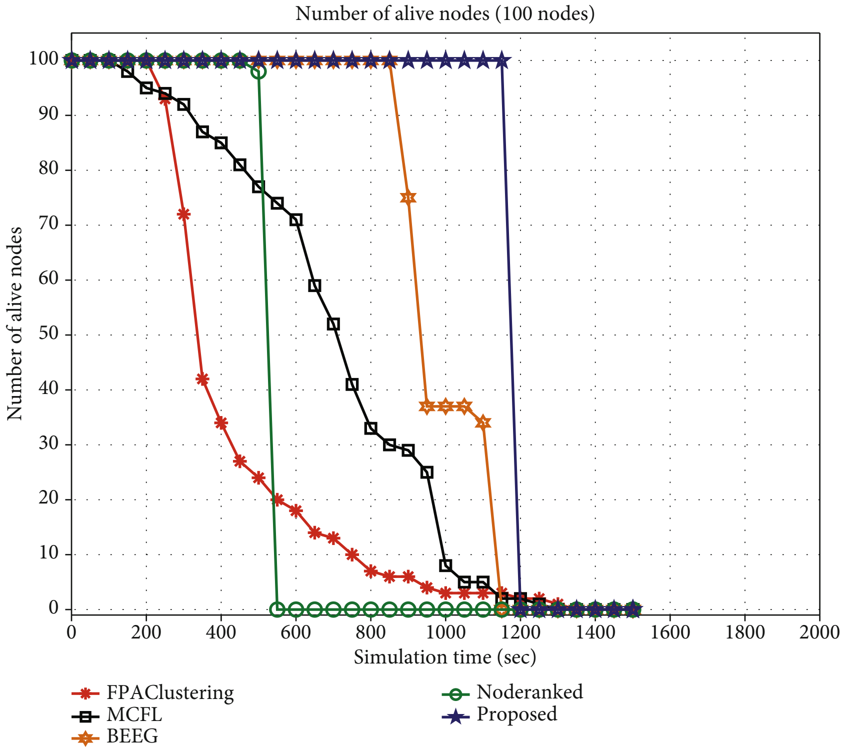


(e)

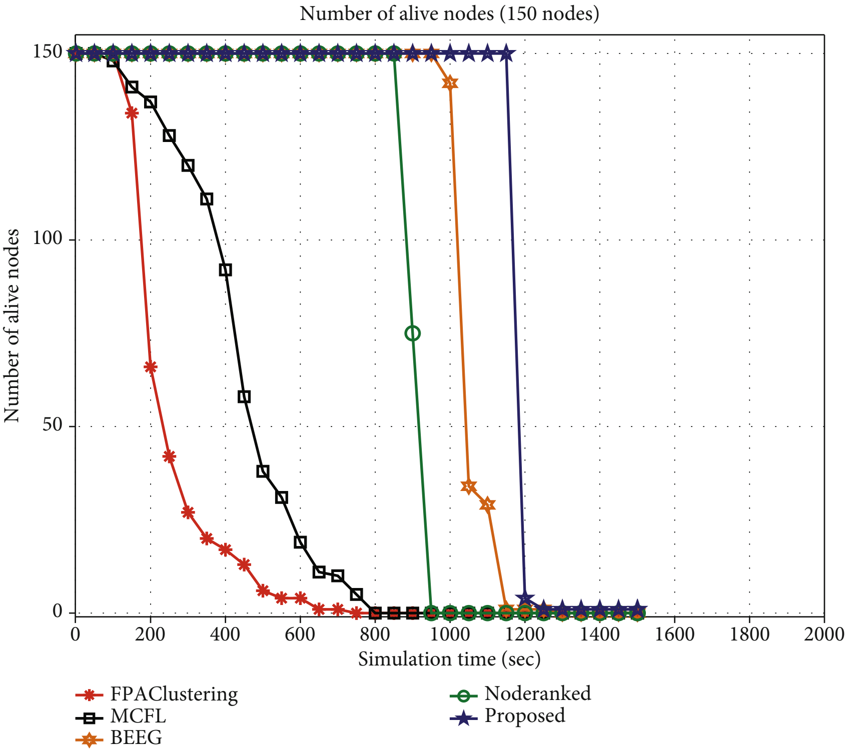


(f)

Figure 4: Comparison of the network lifetime.

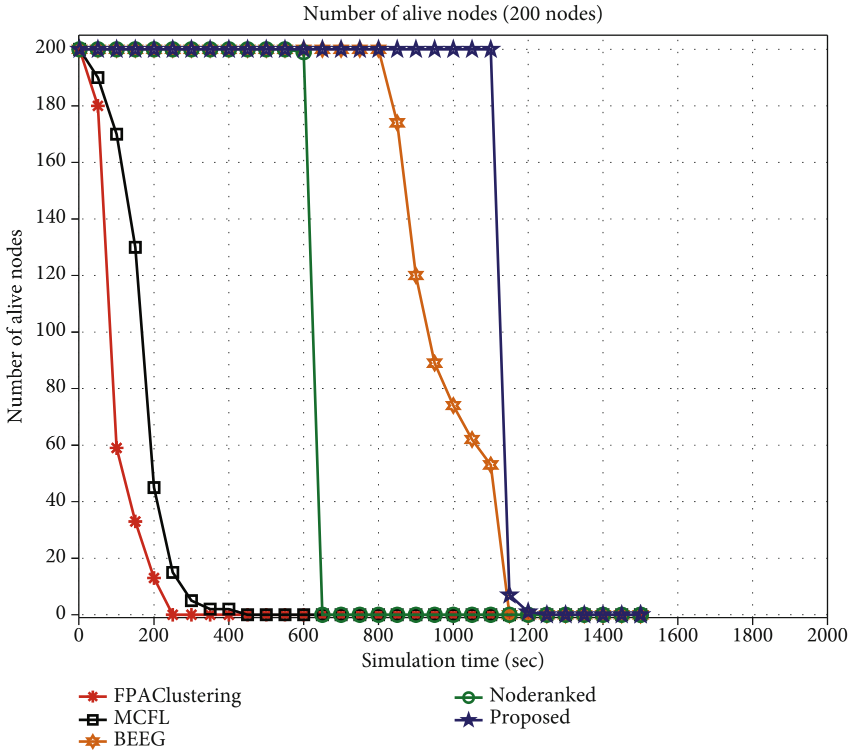


(a)

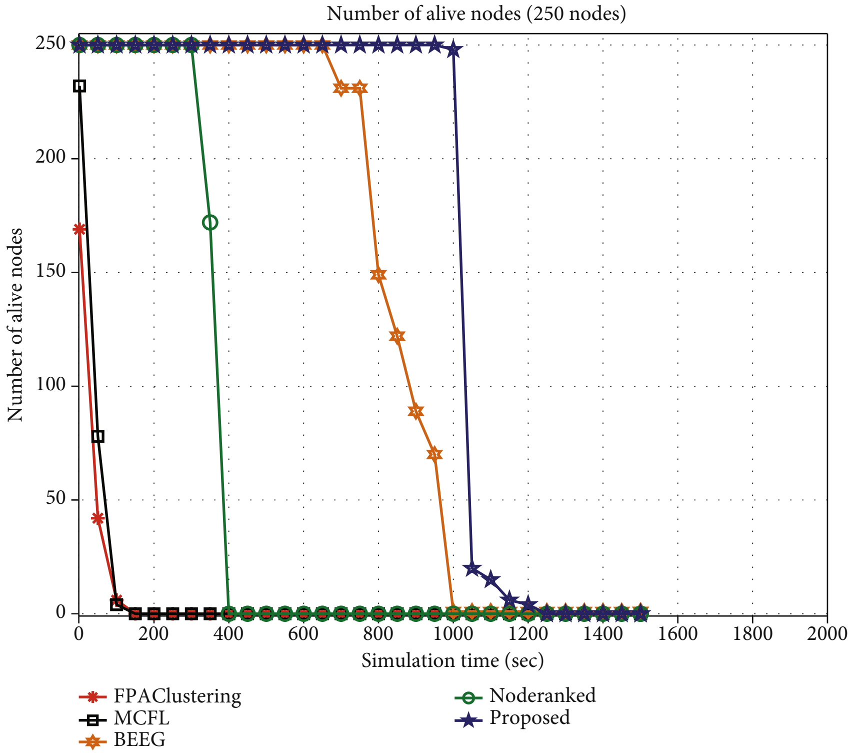


(b)

Figure 5: Continued.

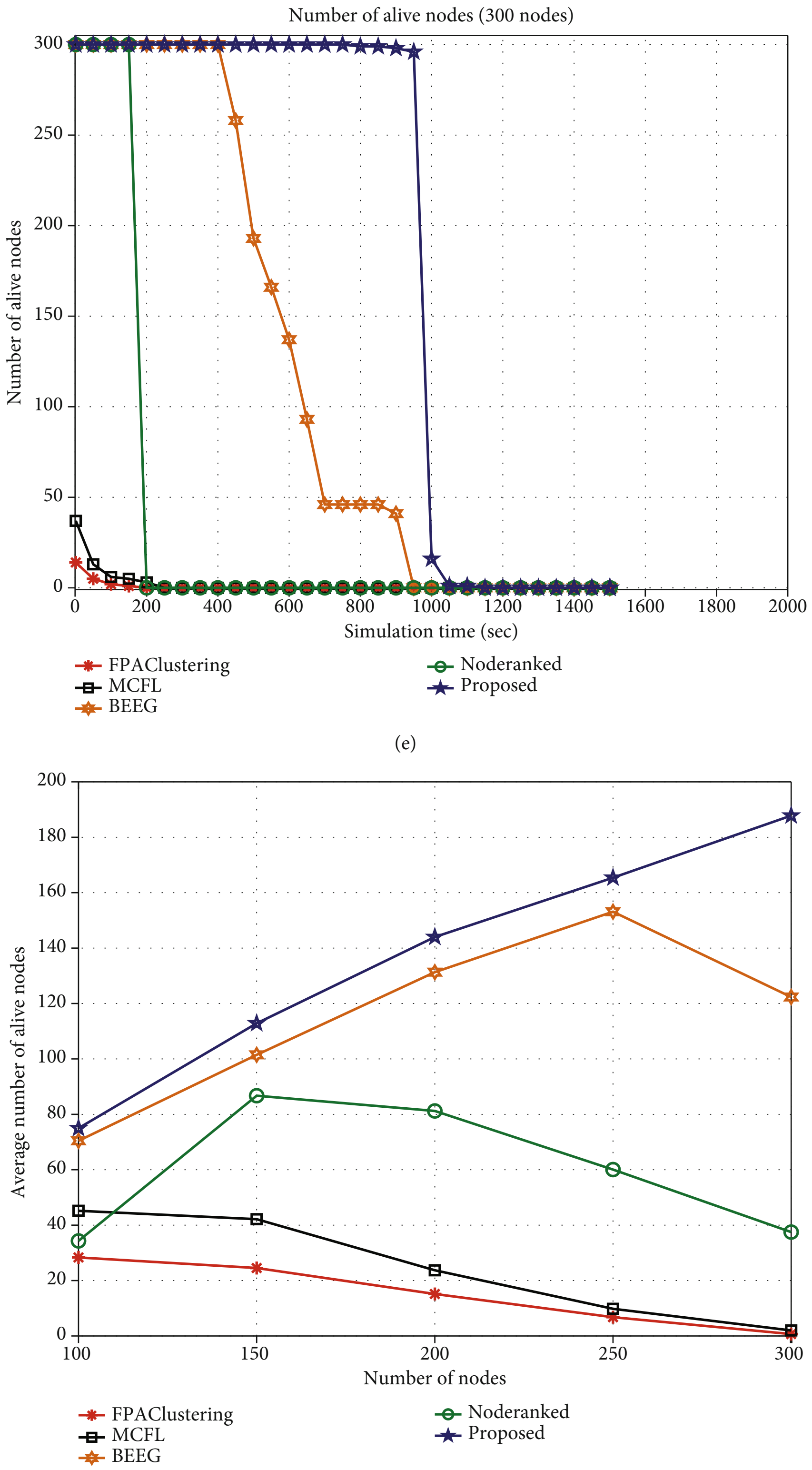


(c)



(d)

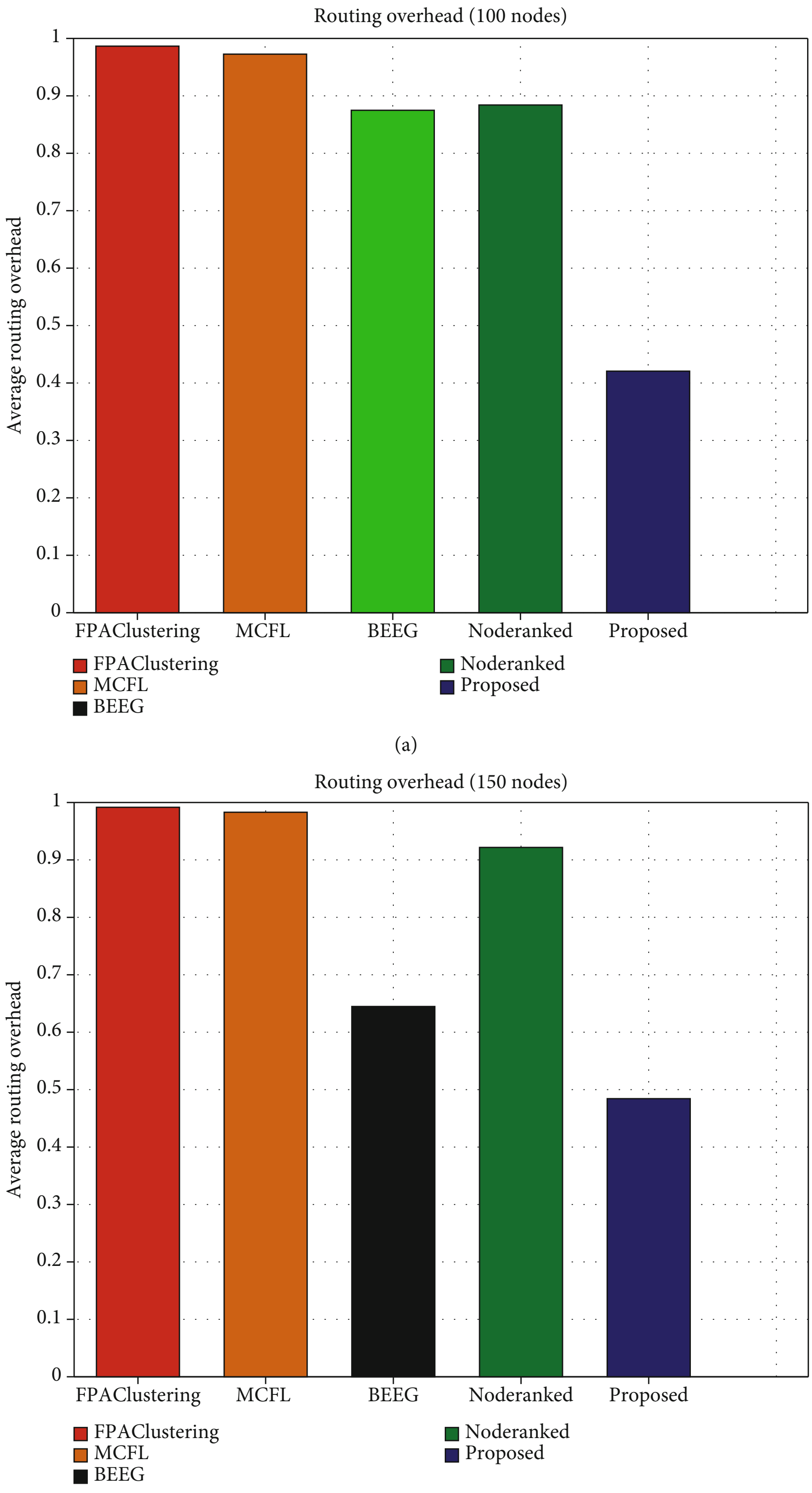
Figure 5: Continued.



(e)

(f)

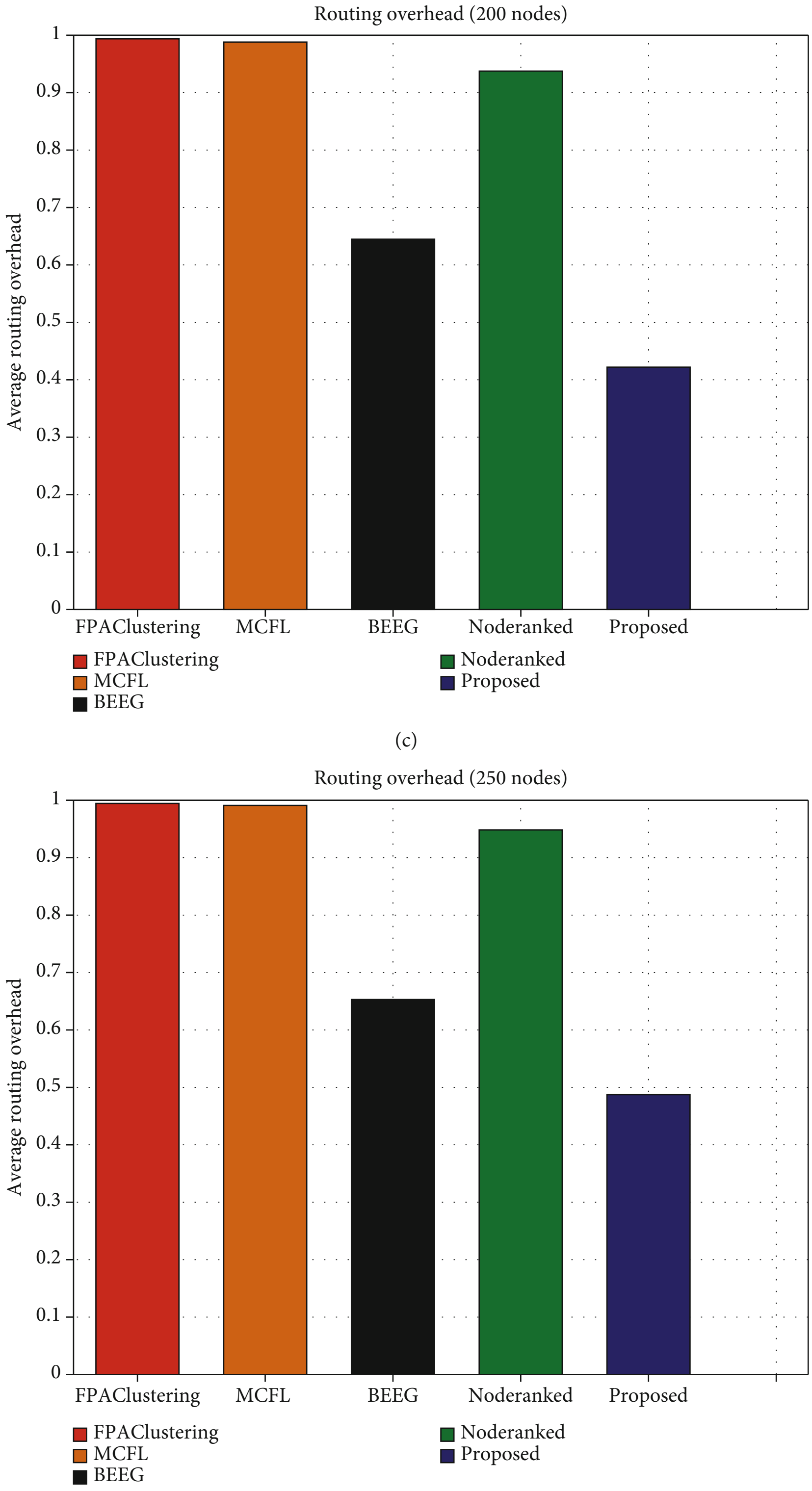
Figure 5: Comparison of the number of alive nodes per round.



(a)

(b)

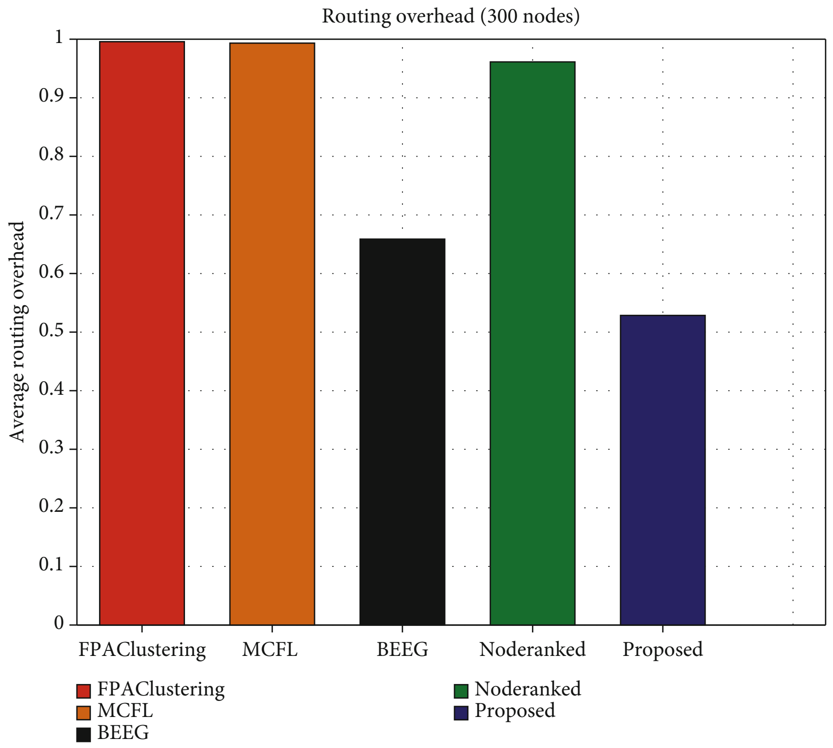
Figure 6: Continued.



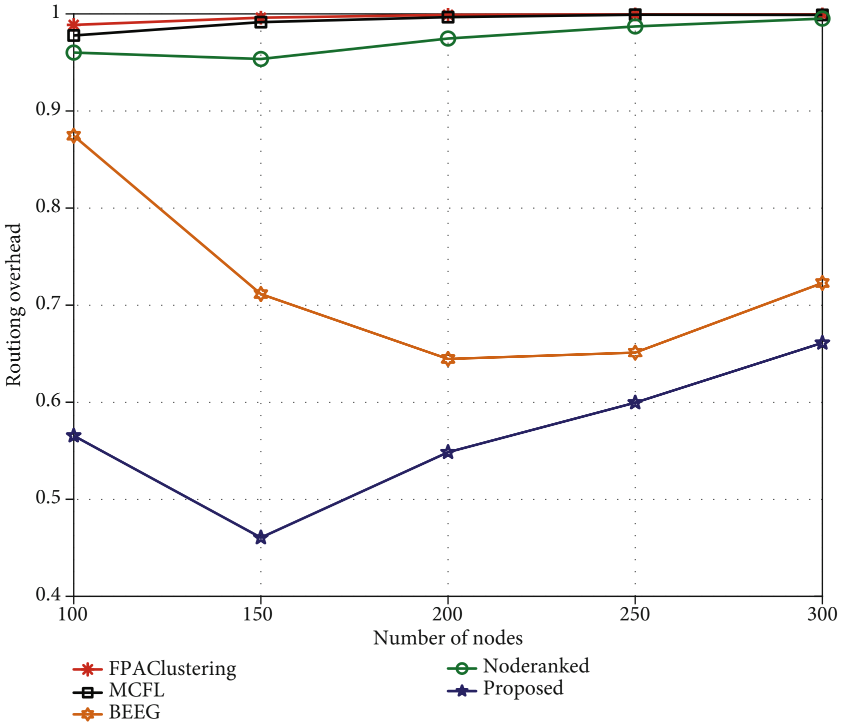
(c)

(d)

Figure 6: Continued.



(e)



(f)

Figure 6: Comparison of the routing overhead for all scenarios.

increase its efficiency together with end-users satisfaction. Our simulation results confirm that using SDN architecture is feasible for smart homes and superior to other non-SDN solutions.

6. Conclusion

This paper introduces a new method for optimizing energy efficiency on the IoT using an effective SDN-based energy clustering protocol. The proposed method involves four stages of set-up, the steady state, the routing phase, and the reclustering phase. Clusters are optimally generated in the set-up process using the SDN platform and using a multiobjective optimization algorithm. The goal of this phase is to accurately determine the number of clusters required and cluster them in a balanced manner. For this purpose, a virtual grid distributes clusters in the environment, and a genetic algorithm is used to create balanced clusters. In the clustering, the centrality of the cluster has also been considered in addition to the balance of clusters.

In the steady state, data are collected and sent to the sink. As long as the CH energy is not less than the threshold that is set dynamically and intelligently, the CH will remain in the next round. Otherwise, it enters the third phase. Increasing the duration of the stability phase reduces both traffic and energy consumption. In the third phase, the current CH identifies the next CH based on the remaining energy. Each CH selects the appropriate node to be the next cluster head locally. The fitness function, which uses the two criteria of distance and energy, is used to choose the next cluster from among the members of the cluster. In the fourth phase, packets are routed through a greedy algorithm. In routing, the node with the shortest distance to the sink and the highest energy among all members of the cluster is selected as the next hop.

The implementation results indicate that the proposed method due to suitable performance can be used in smart homes. It effectively reduces energy consumption and increases network lifetime. Also, it shows a significant improvement in other criteria. However, the calculation overhead of the proposed method is a reasonable amount. An SDN-based routing method could be considered in future work.

Smart homes are the cornerstone of smart cities and can play an important role in urban management. They are used in various fields such as health care, security, and energy saving. On the other hand, the IoT is expected to cover all aspects of life in the coming years. SDN-based applications based on the IoT-Fog-Cloud architecture seem to be a good solution for smart homes and smart cities.

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 that they have no conflicts of interest.

References

[1] L. Khelladi, D. Djenouri, M. Rossi, and N. Badache, "Efficient on-demand multi-node charging techniques for wireless sensor networks," *Computer Communications*, vol. 101, pp. 44–56, 2017.

[2] T. Wang, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, and J. Cao, "Big data reduction for a smart city's critical infrastructural health monitoring," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 128–133, 2018.

[3] A. Yessembayev, D. Sarkar, and F. Sikder, "Detection of good and bad sensor nodes in the presence of malicious attacks and its application to data aggregation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 549–563, 2018.

[4] B. Salma, B. Youssef, and H. Abderrahim, "Software defined networking based for improved wireless sensor network," in *Paper presented at the International Conference on Artificial Intelligence and Symbolic Computation*, Cham, 2019Springer.

[5] E. Stattner, N. Vidot, P. Hunel, and M. Collard, "Wireless sensor network for habitat monitoring: a counting heuristic," in *37th Annual IEEE Conference on Local Computer Networks-Workshops*, Clearwater, FL, USA, 2012.

[6] F. T. Jaigirdar and M. M. Islam, "A new cost-effective approach for battlefield surveillance in wireless sensor networks," in *2016 International Conference on Networking Systems and Security (NSysS)*, Dhaka, Bangladesh, 2016.

[7] S. Razdan and S. Sharma, "Internet of Medical Things (IoMT): overview, emerging technologies, and case studies," in *IETE Technical Review*, pp. 1–14, Taylor & Francis, 2021.

[8] V. Jelicic, M. Magno, D. Brunelli, G. Paci, and L. Benini, "Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring," *IEEE Sensors Journal*, vol. 13, no. 1, pp. 328–338, 2013.

[9] B. Gupta, "Analysis of IoT concept applications: smart home perspective," *Future Access Enablers for Ubiquitous and Intelligent Infrastructures: 5th EAI International Conference, FABULOUS 2021, Virtual Event, May 6–7, 2021, Proceedings*, vol. 382, 2021.

[10] A. R. Javed, L. G. Fahad, A. A. Farhan et al., "Automated cognitive health assessment in smart homes using machine learning," *Sustainable Cities and Society*, vol. 65, article 102572, 2021.

[11] R. Mohammadi and R. Javidan, "On the feasibility of telesurgery over software defined networks," *International Journal of Intelligent Robotics and Applications*, vol. 2, no. 3, pp. 339–350, 2018.

[12] Q. Xu and J. Zhao, "A WSN architecture based on SDN," in *Paper presented at the 4th International Conference on Information Systems and Computing Technology*, Shanghai, China, 2016.

[13] R. Kadel, K. Paudel, D. B. Guruge, and S. J. Halder, "Opportunities and challenges for error control schemes for wireless sensor networks: a review," *Electronics*, vol. 9, no. 3, p. 504, 2020.

[14] M. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *4th international workshop on mobile and wireless communications network*, Stockholm, Sweden, 2002.

[15] A. Al-Baz and A. El-Sayed, "A new algorithm for cluster head selection in LEACH protocol for wireless sensor networks,"

International Journal of Communication Systems, vol. 31, no. 1, article e3407, 2018.

[16] S. El Khediri, R. U. Khan, N. Nasri, and A. Kachouri, "MW-LEACH: low energy adaptive clustering hierarchy approach for WSN," *IET Wireless Sensor Systems*, vol. 10, no. 3, pp. 126–129, 2020.

[17] S. Karimullah, D. Vishnuvardhan, K. Riyazuddin, K. Prathyusha, and K. Sonia, "Low power enhanced Leach protocol to extend WSN lifespan," in *ICCCE 2020*, pp. 527–535, Springer, Singapore, 2021.

[18] M. T. Rahama, M. Hossen, and M. M. Rahman, "A routing protocol for improving energy efficiency in wireless sensor networks," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2016.

[19] S. K. Singh, M. Singh, and D. K. Singh, "Routing protocols in wireless sensor networks–a survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, pp. 63–83, 2010.

[20] S. Wang, J. Yu, M. Atiquzzaman, H. Chen, and L. Ni, "CRPD: a novel clustering routing protocol for dynamic wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 545–559, 2018.

[21] X. X. Ding, T. T. Wang, H. Chu, X. Liu, and Y. H. Feng, "An enhanced cluster head selection of LEACH based on power consumption and density of sensor nodes in wireless sensor networks," *Wireless Personal Communications*, vol. 109, no. 4, pp. 2277–2287, 2019.

[22] J. Huang, Y. Hong, Z. Zhao, and Y. Yuan, "An energy-efficient multi-hop routing protocol based on grid clustering for wireless sensor networks," *Cluster Computing*, vol. 20, no. 4, pp. 3071–3083, 2017.

[23] R. Yarinezhad and A. Sarabi, "Reducing delay and energy consumption in wireless sensor networks by making virtual grid infrastructure and using mobile sink," *AEU-International Journal of Electronics and Communications*, vol. 84, pp. 144–152, 2018.

[24] A. Amer, A. E. Fawzy, M. Shokair, W. Saad, S. El-Halafawy, and A. Elkorany, "Balanced energy efficient grid based clustering protocol for wireless sensor Networks," *International Journal of Computing and Digital Systems*, vol. 6, no. 1, pp. 1–12, 2017.

[25] R. Yarinezhad and S. N. Hashemi, "Solving the load balanced clustering and routing problems in WSNs with an fpt-approximation algorithm and a grid structure," *Pervasive and Mobile Computing*, vol. 58, article 101033, 2019.

[26] A. S. Raghuvanshi, S. Tiwari, R. Tripathi, and N. Kishor, "Optimal number of clusters in wireless sensor networks: a FCM approach," *International Journal of Sensor Networks*, vol. 12, no. 1, pp. 16–24, 2012.

[27] G. Pau and V. M. Salerno, "Wireless sensor networks for smart homes: a fuzzy-based solution for an energy-effective duty cycle," *Electronics*, vol. 8, no. 2, p. 131, 2019.

[28] M. Mirzaie and S. M. Mazinani, "Adaptive MCFL: an adaptive multi-clustering algorithm using fuzzy logic in wireless sensor network," *Computer Communications*, vol. 111, pp. 56–67, 2017.

[29] B. Baranidharan and B. Santhi, "DUCF: distributed load balancing unequal clustering in wireless sensor networks using fuzzy approach," *Applied Soft Computing*, vol. 40, pp. 495–506, 2016.

[30] Z. Sun, X. Xing, T. Wang, Z. Lv, and B. Yan, "An optimized clustering communication protocol based on intelligent computing in information-centric Internet of Things," *IEEE access*, vol. 7, pp. 28238–28249, 2019.

[31] J. Kaur, S. Randhawa, and S. Jain, "A novel energy efficient cluster head selection method for wireless sensor networks," *International Journal of Microwave and Wireless Technologies*, vol. 8, no. 2, pp. 37–51, 2018.

[32] D. Ruan and J. Huang, "A PSO-based uneven dynamic clustering multi-hop routing protocol for wireless sensor networks," *Sensors*, vol. 19, no. 8, p. 1835, 2019.

[33] M. Alazab, K. Lakshmanna, T. Reddy, Q.-V. Pham, and P. K. R. Maddikunta, "Multi-objective cluster head selection using fitness averaged rider optimization algorithm for IoT networks in smart cities," *Sustainable Energy Technologies and Assessments*, vol. 43, article 100973, 2021.

[34] E. Demirors, J. Shi, A. Duong et al., "The seanet project: toward a programmable internet of underwater things," in *2018 Fourth Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, 2018.

[35] H. Luo, K. Wu, R. Ruby, Y. Liang, Z. Guo, and L. M. Ni, "Software-defined architectures and technologies for underwater wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2855–2888, 2018.

[36] Z. ABBOOD, M. SHUKR, Ç. AYDIN, and D. Ç. ATİLLA, "Extending wireless sensor networks' lifetimes using deep reinforcement learning in a software-defined network architecture," *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, vol. 9, no. 1, pp. 39–46, 2021.

[37] A. Ouhab, T. Abreu, H. Slimani, and A. Mellouk, "Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020.

[38] R. Mohammadi, A. Nazari, M. Nassiri, and M. Conti, "An SDN-based framework for QoS routing in internet of underwater things," *Telecommunication Systems*, vol. 78, no. 2, pp. 253–266, 2021.

[39] R. K. Naha, S. Garg, D. Georgakopoulos et al., "Fog computing: survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[40] S. Javanmardi, M. Shojafar, R. Mohammadi, A. Nazari, V. Persico, and A. Pescapè, "FUPE: a security driven task scheduling approach for SDN-based IoT–Fog networks," *Journal of Information Security and Applications*, vol. 60, article 102853, 2021.

[41] P. Azad and V. Sharma, "Cluster head selection in wireless sensor networks under fuzzy environment," *International Scholarly Research Notices*, vol. 2013, Article ID 909086, 8 pages, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5415562, 10 pages
https://doi.org/10.1155/2022/5415562

Research Article

Joint Optimization in Intelligent Reflecting Surface-Aided UAV Communication for Multiaccess Edge Computing

Chao He[1] **and Jia Xiao**[2]

[1] *State Key Laboratory of Network and Switching Technology Institute, Beijing University of Posts and Telecommunications, Beijing, China*
[2] *Huaxia Jingwei Information Technology Co., Ltd., Beijing, China*

Correspondence should be addressed to Chao He; chaohe@bupt.edu.cn

Received 25 January 2022; Revised 1 February 2022; Accepted 15 February 2022; Published 31 March 2022

Academic Editor: Mohammad R Khosravi

Intelligent reflecting surface (IRS) is a key enabling technology for b5G and 6G networks, which can provide a reconfigurable electromagnetic environment while reducing energy consumption. In this article, the communication link between user equipment (UE) and the base station (BS) is severely blocked, so we deployed IRS on the Unmanned Aerial Vehicle (UAV) to assist UE for offloading the computing task to the multiaccess edge computing (MEC) server on the base station, which provides mobile users with low-latency edge computing services. By jointly optimizing active beamforming of UE transmitter, passive beamforming of the IRS, UAV hovering position, and computing task scheduling, the response time of user tasks is minimized. In order to solve this complex nonconvex problem, we propose an alternating optimization (AO) algorithm combined with the genetic algorithm to decouple the problem, alternate optimization, until the convergence condition is met, to find the approximate optimal solution of the problem. Numerical results show that with the assistance of IRS, MIMO channels can significantly improve the performance of edge computing and meet the needs of users for high speed and low latency.

1. Introduction

The explosive growth of network traffic and computing demands has prompted the continuous integration of communication and computing technologies, thereby promoting the continuous innovation and evolution of 5G and 6G technologies [1–5]. Meanwhile, the continued evolution of user requirements and the emergence of new applications have resulted in higher demands on network infrastructure, such as delays, reliability, safety, and energy effectiveness. On the one hand, mobile cellular networks need to cope with the diverse and dynamic demands of massive UE. On the other hand, network operators are constantly confronted with high hardware costs and new demands, which require new technologies to reduce costs while improving quality of service (QoS) and energy efficiency.

MEC [6–8] is a key technology for the mobile communication system to enhance the capabilities of service applications. MEC [9] pushes services and functions of cloud computing to the edge of wireless access network, provides computing and caching services for local mobile users, and deeply integrates communication and computing technologies to meet the needs of mobile users in different scenarios, thereby improving the user experience and promoting network intelligence. MEC [10] uses network function virtualization (NFV) and software-defined network (SDN) to reduce equipment costs and improve equipment utilization; that is, MEC can provide users with low-latency network services at lower hardware costs by dynamically allocating communication and computing resources in real time. Computation offloading [11–14] is an important user-oriented use case in MEC, which is aimed at offloading computation tasks to MEC from resource-constrained UE to meet the real-time requirements of computation-intensive applications. In [6], the authors introduced the important framework of edge computing and categories of computation offloading and then illustrated the offloading model in terms of communication, computation, and energy harvesting. However, when we offload computation tasks, the channel strength between the mobile users and the edge of the network changes dynamically with time and frequency, especially when there are buildings, trees, hills, and other obstacles between the channels, which will cause signal attenuation. And this leads to higher energy consumption, deployment, backhaul, and maintenance costs and more serious and complex network interference problems.

In order to increase the channel gain, the cellular network can use higher frequency bands and package more antennas, i.e., the use of ultramassive multi-input multi-output (MIMO) and terahertz communication. However, this will increase hardware cost, energy consumption, and signal processing complexity. In the context of the above issues, the IRS [15] is considered a disruptive and innovative technology that can intelligently reconstruct the wireless propagation environment. The IRS [16] is an elaborately designed two-dimensional artificial surface. The amplitude and phase of the incident signal of each element IRS are regulated through the control circuit, thereby significantly improving the channel fading and interference problems, i.e. improving the spectral efficiency. Compared with traditional reflective antenna arrays and active surfaces, the IRS is composed of a large number of passive components; therefore, it has the advantages of cost and energy efficiency. Reference [17] proposed an alternating optimization approach to jointly optimize the MIMO input covariance matrix and the IRS reflection coefficients for maximizing the capacity of the IRS-enhanced MIMO system with narrowband and broadband transmission, respectively, and the numerical results showed that the algorithm for getting a suboptimal solution can significantly improve the capacity of the network.

In terrestrial wireless networks, IRS [18] can be deployed on the exterior walls of buildings, ceilings, and billboard. By controlling reflections to avoid obstacles, establish a virtual line-of-sight (LOS) link between UE and BS, thereby significantly improving communication throughput. The performance of a wireless system still depends on its channel, that is, reflection, refraction, diffraction, and path loss in the channel before reaching the receiver. In [19], the authors presented an AO algorithm to jointly optimize resource scheduling, IRS reflection coefficients, and UAV trajectory for maximizing the sum rate in the IRS-assisted UAV system. The results showed that the IRS and UAV increase the degrees of freedom for communication system design and bring promising performance gains such as energy efficiency, passive beamforming, and channel. Reference [20] proposed a successive convex approximation (SCA) algorithm to iteratively optimize active beamforming, the trajectory of UAV, and passive beamforming for maximizing the received signal power, which can reduce the complexity of the solution. Compared with ground IRS, deploying IRS on a rotary-wing UAV to dynamically adjust its hovering position can establish a sustainable LOS link between UE and IRS and between IRS and BS. The aerial intelligent reflection surface (AIRS) [21] communication framework is presented for maximizing the worst-case signal-to-noise ratio (SNR) by jointly optimizing transmit power, AIRS location, and phase shifts of the IRS with the suboptimal solution.

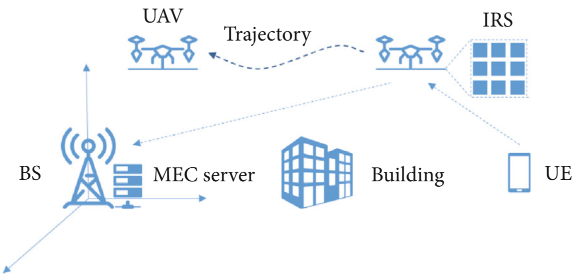


Figure 1: The system of UAV-IRS-enhanced MEC.

The wireless relaying system aerial intelligent reflecting surface (AIRS) [21] can extend the coverage area of cellular network and improve the network performance. The authors proposed a suboptimal solution to tackle the maximizing worst-case signal-to-noise ratio (SNR) problem by jointly optimizing transmit power, AIRS position, and reflection coefficients. To improve the QoS of wireless network, the IRS-assisted single-input single-output (SISO) MEC system [22] was presented to offload the computation task to the edge node of the access point (AP). The presented system was aiming at maximizing the total computational bits by jointly optimizing the CPU frequency, the offloading time assignment, and the transmit power allocation, as well as the IRS phase shifts for promoting the performance of applications. The IRS-assisted MIMO system [17] can achieve increased capacity by a convex relaxation-based alternating optimization method. The authors optimized the transmission covariance matrix and the IRS reflection factors to get a suboptimal solution of achievable rate. Inspired by above views, we propose a UAV-IRS (UIRS) enhanced MIMO MEC system to assign computing and communication resources for offloading the computation tasks to the MEC server on the BS, which is shown in Figure 1. Obstacles in urban and suburban environments can block LOS links, which cause signal loss and attenuation. Therefore, the UAV can provide a higher LOS probability than the ground link. We assume that the radio signal from the transmitter to the receiver is severely blocked by obstacles (i.e., buildings) and the signal is interrupted. In this scenario, we place the IRS on a highly maneuverable rotary-wing UAV to expand the coverage of the IRS which makes it easier to establish a virtual LOS link between the UE and the BS, which can assist mobile users in computation offloading. Our objective is to minimize the computational offloading time by jointly optimizing transmit beamforming, IRS passive beamforming, UAV hovering location, and computing task allocation. To solve the above-mentioned nonconvex optimization problem, we use the AO algorithm combined with the genetic algorithm to decompose the complex problem into four subproblems for iterative optimization to reduce the computational complexity, so that the algorithm can at least accelerate the convergence to a local optimal solution. Moreover, the main notations presented in this article are summarized in Table 1.

Table 1: Main notations.

Notation	Definition
$\mathbf{m}$	UE antennae vector
$\mathbf{s}$	IRS position vector
$\mathbf{b}$	BS position vector
N	Number of IRS element
d_m^s	Distance between UE and IRS
d_s^b	Distance between IRS and BS
$\mathscr{K}$	Rician factor
H	UE-IRS channel matrix
G	IRS-BS channel matrix
los	Channel matrix LOS exponent
nlos	Channel matrix NLOS exponent
$d(n, k)$	Distance between UE kth antenna and IRS nth element
$d(l, n)$	Distance between BS lth antenna and IRS nth element
$\mathscr{G}$	UE-BS virtual link channel matrix
ρ	Free-space path loss
d_m	Distance between UE antenna and IRS plane
d_b	Distance between BS antenna and IRS plane
Θ	Reflection coefficient matrix
β_n	Reflection amplitude of the nth element
θ_n	Phase shift of the nth IRS element
B	Bandwidth
σ^2	Noise power
Q	Input covariance matrix
R	Transmission rate of UE
A	Data size of computation task
A_l	Data size of local execution
A_b	Data size of edge server execution
W	Workload of computation task
T_l	Delay of the local execution
T_b	Delay of the computation offloading

The rest of this article is organized as follows. Section 2 introduces the UAV-IRS MEC system model and problem formulation and formulates the problem to minimize the computation task delay. Section 3 proposes an alternate optimization algorithm, which is used to solve each subproblem decomposed in different closed forms and obtain the optimal solution. Section 4 presents the numerical results and analysis. Section 5 summarizes this paper.

2. System Model

As shown in Figure 2, we propose an edge computing system that deploys IRS on UAV to assist UE's computation tasks for offloading. The UE's signal is severely blocked by the building; then, the user's intensive computing tasks cannot be directly offloaded to the base station. At this time, the IRS on the UAV can provide UE with intelligent reflection in the air to assist in offloading the user's computing tasks. The UAV plays a role as a mobile communication base station but only for acquiring channel state information (CSI) for enhancing IRS's signal reflection. In this paper, the channel is assumed to be quasistatic flat fading; i.e., the channel state, the reflection coefficient, and UAV's and UE's location are unchanged and independent over each transmission block.

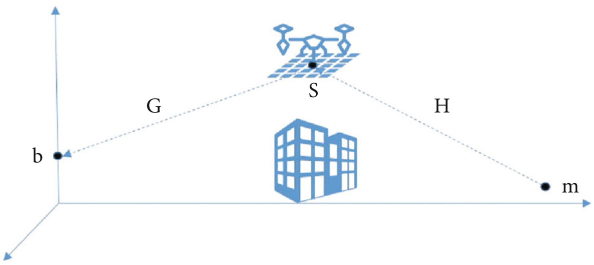


Figure 2: The illustration of UAV-IRS-enhanced MEC.

2.1. Channel Model. We assume that the UE and BS are equipped with K antennas and L antennas, respectively, which are placed in uniform linear array (ULA) and vertical to the ground, i.e., the XOY plane. In this three-dimensional (3D) coordinate system, we take the midpoint of the transmit ULA, receive ULA, and the center point of IRS as the reference point to represent their location. We set the coordinate of K antennae reference point as $\mathbf{m} = [X_m, Y_m, Z_m]^T$, the coordinate of IRS's reference point as $\mathbf{s} = [X_s, Y_s, Z_s]^T$, and the coordinate of L antenna reference point as $\mathbf{b} = [X_b, Y_b, Z_b]^T$. The IRS is equipped with $N = N_r \times N_c$ elements, i.e., placed in a uniform planar array (UPA) with N_r as the number of IRS elements along the x-axis and N_c as the number of elements along the y-axis, where the nth element is $n \in \mathscr{N} = 1, 2, \cdots, N$ and parallel to the XOY plane. The intervals between adjacent elements along the x-axis and y-axis are both $\lambda/2$, where λ is the carrier wavelength. In view of the air traffic control (ATC), we suppose that UAV can only fly at a fixed area and altitude; that is, $X_{\min} \le X_s \le X_{\max}$, $Y_{\min} \le Y_s \le Y_{\max}$, and Z_s is a constant. The vertical distance from the reference point of the UE antennae to the plane where the IRS is located and that from the BS antennae reference point to the IRS plane can be written as $d_m = |Z_s - Z_m|$ and $d_b = |Z_s - Z_b|$, respectively. Meanwhile the distance between the reference point of UE's antenna and the reference point of IRS can be given by $d_m^s = |\mathbf{m} - \mathbf{s}|$ and also, the distance from IRS's reference point to BS antenna's reference point can be denoted by $d_s^b = |\mathbf{s} - \mathbf{b}|$.

As the direct link between UE and BS is blocked, IRS is deployed on UAV for assisting offloading computation tasks. We adopt the Rician fading channel model, and thus, the channel matrix $\mathrm{H} \in \mathbb{C}^{N\times K}$ between the UE and IRS is expressed by

$$H = \frac{\left(\sqrt{\mathscr{K}} H_{\mathrm{los}} + H_{\mathrm{nlos}}\right)}{\sqrt{\mathscr{K} + 1}}, \qquad (1)$$

where $\mathscr{K}$ is the Rician factor, $H_{\text{los}} = e^{-j(2\pi/\lambda)d(n,k)}$, $d(n, k)$ is the distance matrix between IRS's nth element and UE's k th antenna, $k \in \{1, 2, \cdots K\}$, H_{nlos} is independent and identically distributed (i.i.d.), and $H_{\text{nlos}} \sim \mathscr{C}\mathscr{N}(0, 1)$. The channel matrix $G \in \mathbb{C}^{L\times N}$ between IRS's nth element and BS's lth antenna is given by

$$\text{G} = \frac{\left(\sqrt{\mathscr{K}}G_{\text{los}} + G_{\text{nlos}}\right)}{\sqrt{\square+1}}, \tag{2}$$

where $G_{\text{los}} = e^{-j(2\pi/\lambda)d(l,n)}$, $d(l, n)$ is the distance matrix from BS's lth antenna to IRS's nth element, and $l \in \{1, 2, \cdots L\}$; G_{nlos} is i.i.d. according to $\mathscr{C}\mathscr{N}(0, 1)$.

As a result of blocked LOS link, IRS-assisted virtual LOS link channel matrix between UE and BS is denoted by

$$\mathscr{G} = \sqrt{\rho^{-1}}G\Theta H, \tag{3}$$

where ρ is the free-space path loss and, according to [23–25], can be expressed by

$$\rho = \frac{256\pi^2\left(d_m^s\right)^2\left(d_s^b\right)^2}{\lambda^4\left(\left(d_m/d_m^s\right) + \left(d_b/d_s^b\right)\right)^2}, \tag{4}$$

and $\Theta \in C^{N\times N}$ is the reflection coefficient matrix which is expressed by

$$\Theta = \text{diag}\left(\beta_1 e^{j\theta_1}, \beta_2 e^{j\theta_2}, \cdots, \beta_N e^{j\theta_N}\right), \tag{5}$$

where $\beta_n \in [0, 1]$ is the reflection amplitude of the nth passive element and $\theta_n \in [0, 2\pi]$ is the phase shift of the nth passive IRS element. Therefore, the computation offloading transmission rate of UE according to [10] is denoted by

$$R = B \log_2 \det\left(I + \frac{\mathscr{G}Q\mathscr{G}^H}{\sigma^2}\right), \tag{6}$$

where B is the bandwidth, σ^2 is the noise power, and $Q \in \mathbb{C}^{K\times K}$ is the input covariance matrix.

2.2. Computing Model. Considering the limited computation resources of UE, we focus on partial offloading: computation-intensive part of the task is offloading to the edge server for remote executing, and the remaining part is computed locally. The UE's computation tasks are characterized by the tuple <A, W > , where $A = A_l + A_b$ (in bits) is the total amount of the computing task and W (in cycles/bit) is the required workload of the task. Therefore, the delay of the user computation task is split into two parts in parallel, and the delay of the local execution part is

$$T_l = \frac{A_l W}{f_l}, \tag{7}$$

where A_l is the data size of local execution and f_l is the local computation capacity. Generally, the results of computation task are very small compared to the input data; thus, the delay of returning the results can be ignored and the delay of the computation offloading is

$$T_b = \frac{A_b}{R} + \frac{A_b W}{f_b}, \tag{8}$$

where A_b is the data size of edge server execution and f_b is the computing capacity of MEC server. The total delay of computation task is

$$T = \max\left\{T_l, T_b\right\}. \tag{9}$$

2.3. Problem Formulated. In this paper, we aim to minimize the delay of computation tasks for single UE by jointly optimizing the transmission covariance matrix Q, the IRS reflection coefficients Θ, and the computation task allocation A_l, A_b, and the UAV hovering location is the **s**. Then, the problem of offloading time is formulated as

$$(\text{P1}) \underset{\{\mathbf{s},\text{A}_l,\text{A}_b,\text{Q},\Theta\}}{\text{minimize}}\ T \tag{10}$$

$$\text{s.t.} \quad \text{Tr}(Q) \le P_t, \quad Q \succeq 0, \tag{11}$$

$$\begin{aligned} A &= A_l + A_b, \\ A_l &\ge 0, \\ A_b &\ge 0 \end{aligned} \tag{12}$$

$$\begin{aligned} X_{\min} &\le X_s \le X_{\max}, \\ Y_{\min} &\le Y_s \le Y_{\max}, \end{aligned} \tag{13}$$

$$\begin{aligned} 0 \le \theta_n &\le 2\pi, \\ \beta_n \in [0, 1], &\quad \forall n \in \{1, 2, \cdots, N\}, \end{aligned} \tag{14}$$

where (11) is the transmission power constraint, i.e., the active beamforming constraint, (12) denotes the task assignment constraint, (13) restricts the area of UAV hovering, and (14) is the passive beamforming constraint of the IRS, i.e., the phase shifts θ_n and the amplitude β_n, $\forall n \in \{1, 2, \cdots, N\}$.

It is worth noting that the controller is embedded in the UIRS. UIRS can calculate the phase shifts and the amplitude according to the CSI and send the instructions to the controller to adjust the reflection coefficient for assisting UE in computation offloading. Moreover, the P1 problem is a nonconvex optimization problem and there are four optimization variables coupling into the min-max formulation, which makes it more difficult to tackle.

3. Proposed Method

The P1 problem in the previous section is a high-complexity nonconvex problem. In order to solve this problem, we use an alternating optimization method to decouple multiple variables of the problem, that is, iteratively optimize the active beamforming, the UAV hovering location, the

```
Input: initial variable
Output: suboptimal solution
(1) Randomly generate an initial population with a certain number of individuals
(2) Use the fitness function to evaluate the population to determine whether the stopping condition is met; if so, stop and output the
    optimal solution; otherwise, continue to operate
(3) Individuals that can be updated are selected according to their fitness. Individuals with high fitness have a high probability of
    being selected, and individuals with low fitness may be eliminated
(4) Generate new individuals according to a certain crossover probability and method
(5) Generate new individuals according to a certain mutation probability and method
(6) Generate a new generation of population by crossover and mutation; return to step 2
```

Algorithm 1: GA process.

computation task scheduling, and passive beamforming variables while the other variables are fixed. In each iteration, we use a heuristic genetic algorithm to get a feasible solution to the optimization problem to be solved in acceptable time and space cost. Specifically, the P1 problem can be transformed into more solvable formulation by changing the optimization variables in task assignment and passive beamforming. We set a task allocation factor φ to replace the A_l and A_b, i.e., $A_l = \varphi A$, $A_b = (1-\varphi)A$, and $\varphi \in [0, 1]$. The reflection factors are continuously adjusted and can be defining $\Theta_n = \beta_n e^{j\theta_n}$ as the coefficient per element, that is, $|\Theta_n| \leq 1$ for simplicity. Therefore, the P1 problem can be transformed as follows:

$$(\text{P2}) \ \underset{\{\mathbf{s},\varphi,\mathrm{Q},\Theta\}}{\text{minimize}} \quad T \tag{15}$$

$$\text{s.t.} \quad \operatorname{Tr}(Q) \leq P_t, \quad Q \succeq 0, \tag{16}$$

$$|\Theta_n| \leq 1, \quad \forall n \in \{1, 2, \cdots, N\} \tag{17}$$

$$\begin{aligned} A_l &= \varphi A, \\ A_b &= (1-\varphi)A, \quad \varphi \in [0, 1], \end{aligned} \tag{18}$$

$$\begin{aligned} X_{\min} &\leq X_s \leq X_{\max}, \\ Y_{\min} &\leq Y_s \leq Y_{\max}. \end{aligned} \tag{19}$$

To solve the P2 problem, we use the alternating optimization [26] to decompose the nonconvex problem into four optimization variables, which can be denoted by

$$\begin{cases} s^{i+1} = \arg \min\limits_{s} \ T\left(s, \varphi^i, \mathrm{Q}^i, \Theta^i\right), \\ \varphi^{i+1} = \arg \min\limits_{\varphi} T\left(s^{i+1}, \varphi, \mathrm{Q}^i, \Theta^i\right), \\ \mathrm{Q}^{i+1} = \arg \min\limits_{\mathrm{Q}} T\left(s^{i+1}, \varphi^{i+1}, \mathrm{Q}, \Theta^i\right), \\ \Theta^{i+1} = \arg \min\limits_{\Theta} T\left(s^{i+1}, \varphi^{i+1}, \mathrm{Q}^{i+1}, \Theta\right), \end{cases} \tag{20}$$

where s^i is the UIRS hovering location, φ^i is the task assignment factor, Q^i is the transmit covariance matrix, and Θ^i is the reflection factors in the ith iteration. Then, in the iteration, we first update the location s by genetic algorithm (GA) [27, 28] when all the other variables are fixed, then put the updated s into the fitness function to update the allocation factor φ while the matrix Q and Θ are unchanged; next, we put the new factor φ to the objective function to update the matrix Q. Finally, we update the active beamforming Θ with hovering location s, task allocation coefficient φ, and passive beamforming Θ fixed.

Table 2: Detailed parameters of the UIRS-enhanced MEC system.

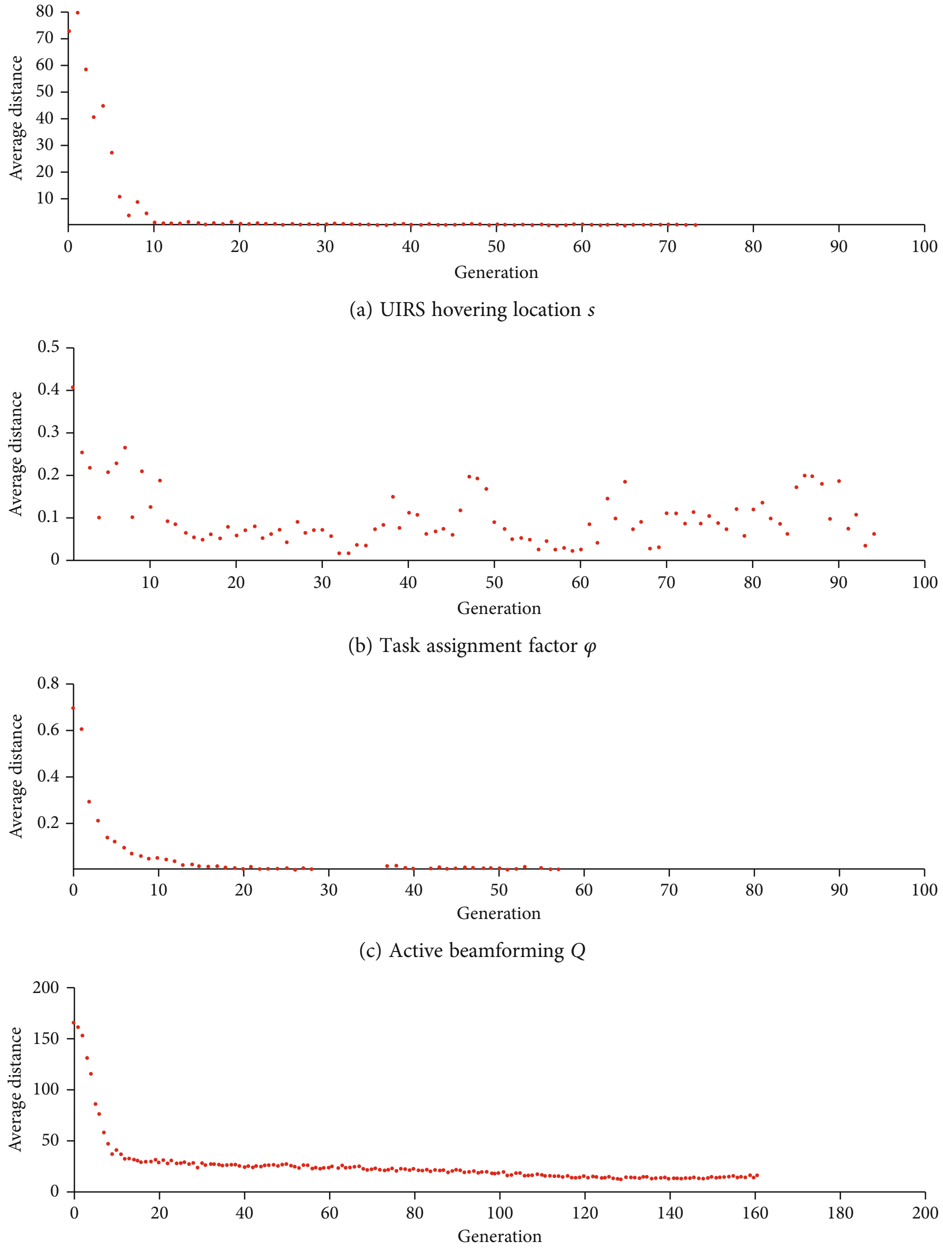
Parameter	Value
Number of IRS element N	20×20
Number of transmitters Nt	10
Number of receivers Nr	6
Transmit power: Pt	1 watt
Noise power: σ^2	-120 dB
Data size of task: A	$1e6$ bits
Task requirement: W	500 cycles/bit
Local CPU frequency: f_l	$1e9$ Hz
Local CPU frequency: f_b	$5e9$ Hz
Wavelength: λ	0.15 m
Rician factor: $\mathcal{K}$	1
UE position: $\mathbf{m}$	[100,0,1]
BS position: $\mathbf{b}$	[0,0,60]
UIRS x-axis: X_s	0~100
UIRS y-axis: Y_s	-100~100
UIRS altitudes: Z_s	100

The detailed GA process is given in Algorithm 1.

GA can use very complex fitness functions (i.e., objective functions) and place limits on the range of variables. GA is not always the best optimization strategy, but it can find good solutions, even in very complex feasible set. For any specific optimization problem, adjusting the parameters of the GA can make the problem converge quickly. That is to say, the GA can jump out of the local optimum and find the global optimum. Therefore, our proposed hybrid optimization method of GA and AO algorithm can find a suboptimal solution to the P2 problem.

4. Simulation Results

In this section, we provide extensive simulation results to corroborate the performance of our presented UIRS-



(a) UIRS hovering location s

(b) Task assignment factor φ

(c) Active beamforming Q

(d) Passive beamforming Θ

Figure 3: Average distance between individuals.

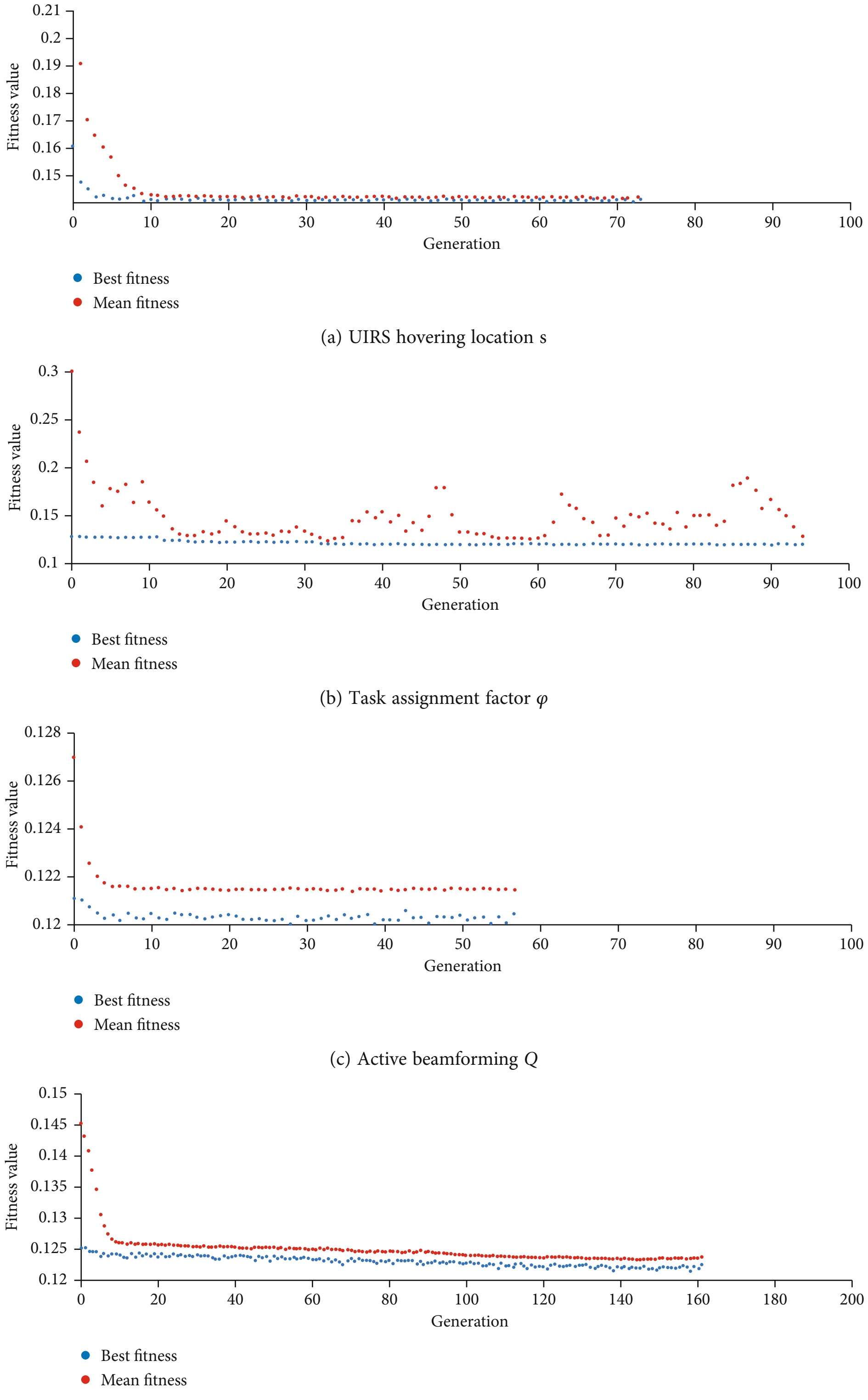
enhanced MEC system. The simulation is for narrowband flat-fading channels under MIMO as well as single-user setups. The channel between UE and BS is interrupted because of severe blockage, and the mobile user offloads computation task to the MEC server by the UE-UIRS link and UIRS-BS link. The execution of computing tasks is divided into the following three ways:

(i) Local execution: the computation resources of the UE are sufficient to perform computation task locally; thus, the task allocation factor $\varphi = 1$

(ii) Full offloading: the UE's computation resources are limited, and the whole computation task can be offloaded to the MEC server, i.e., the task assignment factor $\varphi = 0$

(iii) Partial offloading: the UE offloads partial computation task to the edge server, and the rest of the task is executed locally, that is, the task scheduling factor $\varphi \in (0, 1)$

In the simulations, our proposed system is a 3D coordinate system, the BS is located at the origin coordinate; the UE is located on the x-axis. Their antennas are both equipped with the ULA perpendicular to the XOY plane, and the spacing of antenna is $\lambda/2$. The IRS is deployed on the UAV and is equipped with the UPA parallel to



(a) UIRS hovering location s

(b) Task assignment factor φ

(c) Active beamforming Q

(d) Passive beamforming Θ

Figure 4: Decay trend of offloading time optimization.

the XOY plane. The system parameters are summarized in Table 2.

Figure 3 shows the average distance between individuals in each generation, which is a good measure of population diversity. From Figures 3(a) to 3(d), we can find out that as the population evolves, the average distance between individuals approaches 0 as the number of mutations decreases. Figure 4 shows the decay trend of fitness function (i.e., the computation offloading time), and the optimal value can be obtained after generations iteratively. We use the

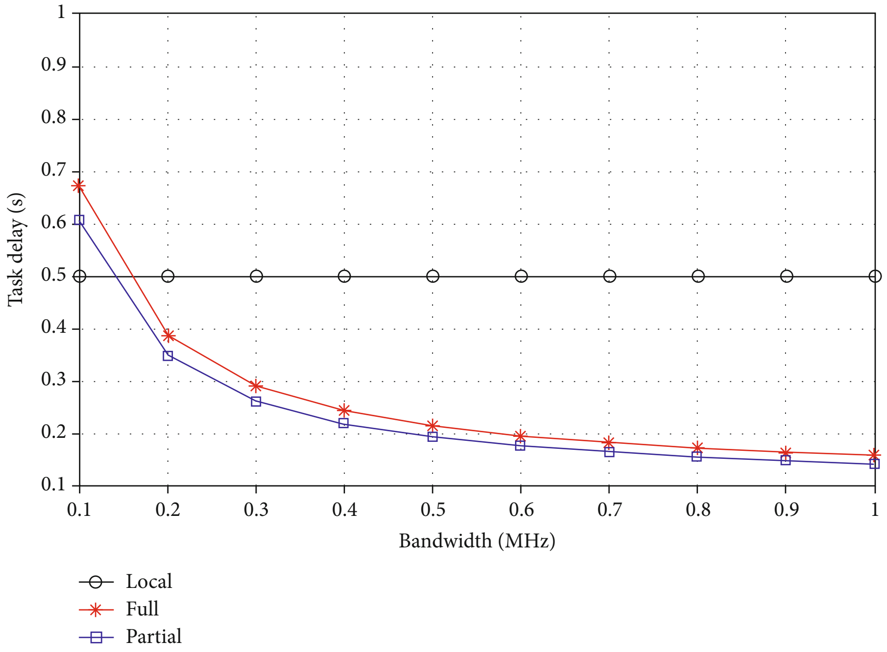


Figure 5: Computation task delay versus system bandwidth.

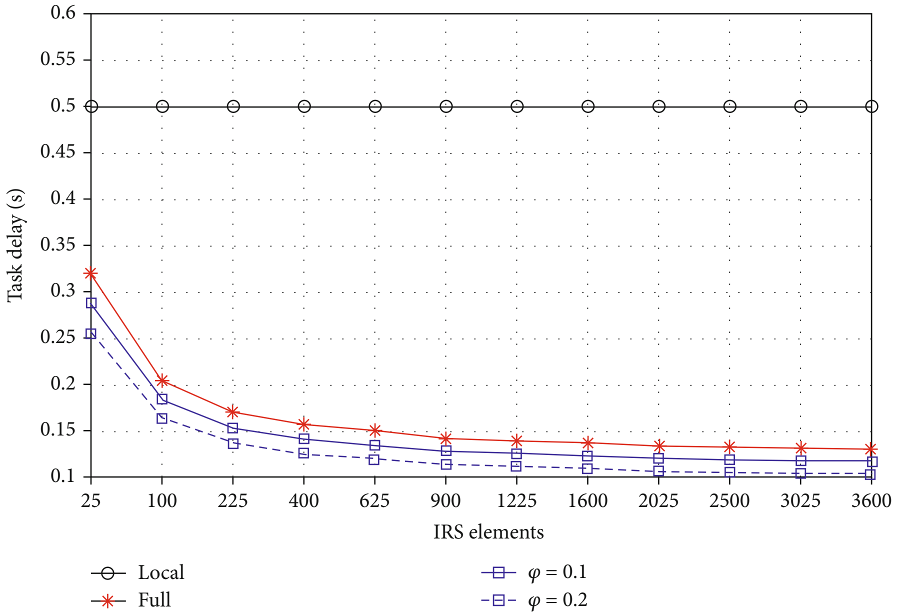


Figure 6: Computation task delay versus IRS elements.

parameters in Table 1 as the initial parameters to search optimal solution with GA. The optimized UIRS hovering location, task assignment factor, active beamforming, and passive beamforming can be found while all the other variables are fixed.

In addition, the bandwidth and the number of IRS reflecting elements also affect the response time of computation offloading. Under the premise of the optimal solutions obtained from previous experiments, Figure 5 plots the computation task delay with system bandwidth under three task execution ways. The local execution is independent of bandwidth, and all the variables are optimized. As a result, it is a straight line in the figure. The full offloading requires offloading all the computation task to the MEC server, and the task delay is affected by the IRS parameter and bandwidth. Partial offloading divides computing tasks into local execution and edge execution, and the two are executed in parallel, so the task latency is less than the full offloading latency. We observe that the full offloading and partial offloading decrease with increasing bandwidth, which is due to the fact that the transmission rate is affected by system bandwidth. As the bandwidth increases, performance of UIRS-aided MEC begins to outperform the local execution, and the gap becomes more pronounced, which further illustrates the important role of IRS in the proposed system.

Figure 6 shows that the computation task delay obtained by local execution, full offloading, and partial offloading under the different numbers of IRS elements, where the task assignment factor $\varphi = 0.1$ and $\varphi = 0.2$. It can be observed that as the number of IRS elements increases, the task response time of UIRS-assisted system decreases, while the exception of local execution latency is unchanged due to the irrelevant with the IRS. Another important finding is that the number of IRS elements cannot grow indefinitely. On the one hand, it is due to the constraints of hardware cost and control complexity. On the other hand, it can be seen from Figure 6 that when the number of IRS elements increases to a certain extent, the task delay will not be significantly improved. We can also observe that our presented UIRS system can significantly improve the MEC performance of the UE; that is, the IRS can play an important role in wireless network. Moreover, the partial offloading outperforms the full offloading, and the gap between them depends on the task allocation factor. We can get this conclusion from the variety of the polyline when$\varphi = 0.1$and$\varphi = 0.2$in Figure 6.

5. Conclusions

In this paper, we present an alternating algorithm based on GA for computation task delay optimization in the UIRS-enhanced MEC system. Due to the severe blockage, UIRS can provide virtual LOS link between the UE and BS. The BS provides MEC service for the UE so as to obtain more efficient power management, fewer storage requirements, and higher application performance. We aim to minimize the task latency and jointly optimize the UIRS hovering location, task allocation factor, active beamforming, and passive beamforming to find the suboptimal solution in limited time. The IRS and UAV provide new degrees of freedom to further improve the performance of wireless links, paving the way for the convergence of computing and communications. Simulation results validate the effectiveness of the proposed solution and that MEC can significantly decrease the delay of computation task and improve the performance of UE. In general, our proposed method is an efficient, parallel, global search method suitable for running on hardware with limited memory or computational power. Meanwhile, our proposed method can also be used as a benchmark for future work. In addition, there are still some challenging problems that can be completed in the future. For example, discrete reflection coefficient and multiuser computation offloading problem should be considered in our future work.

Data Availability

The data used to support the findings of this study are included in the article. Some or all data used during the study are available from the corresponding author by request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 615–637, 2021.

[2] W.-C. Chien, S.-Y. Huang, C.-F. Lai, and H.-C. Chao, "Resource management in 5G mobile networks: survey and challenges," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 896–914, 2020.

[3] Y. Zhao, J. Zhao, W. Zhai, S. Sun, D. Niyato, and K.-Y. Lam, "A survey of 6G wireless communications: emerging technologies," in *Future of Information and Communication Conference*, pp. 150–170, Springer, 2021.

[4] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and Y. Shui, "Compatibility-aware web APIs recommendation for mashup creation via textual description mining," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 20, pp. 1–19, 2021.

[5] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, pp. 135–151, 2022.

[6] Q.-V. Pham, F. Fang, V. N. Ha et al., "A survey of multi-access edge computing in 5G and beyond: fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[7] Q. He, G. Cui, X. Zhang et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, 2020.

[8] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Cost-effective app data distribution in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 31–44, 2021.

[9] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. Fitzek, "Device-enhanced MEC: multi-access edge computing (MEC) aided by end device computation and caching: a survey," *IEEE Access*, vol. 7, pp. 166079–166108, 2019.

[10] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: a survey," *IEEE Access*, vol. 8, pp. 197017–197046, 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 102781, 2020.

[12] A. Shakarami, A. Shahidinejad, and M. Ghobaei-Arani, "A review on the computation offloading approaches in mobile edge computing: a g ame-theoretic perspective," *Software: Practice and Experience*, vol. 50, no. 9, pp. 1719–1759, 2020.

[13] Y. Liao, L. Shou, Q. Yu, Q. Ai, and Q. Liu, "Joint offloading decision and resource allocation for mobile edge computing enabled networks," *Computer Communications*, vol. 154, pp. 361–369, 2020.

[14] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent task offloading in vehicular edge computing networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 126–132, 2020.

[15] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, “Intelligent reflecting surface aided wireless communications: a tutorial,” *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[16] J. Y. Dai, W. Tang, M. Z. Chen et al., “Wireless communication based on information metasurfaces,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 3, pp. 1493–1510, 2021.

[17] S. Zhang and R. Zhang, “Capacity characterization for intelligent reflecting surface aided MIMO communication,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1823–1838, 2020.

[18] Q. Wu, J. Xu, Y. Zeng et al., “A comprehensive overview on 5G-and-beyond networks with UAVs: from communications to sensing and intelligence,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 2912–2945, 2021.

[19] Z. Wei, Y. Cai, Z. Sun et al., “Sum-rate maximization for IRS-assisted UAV OFDMA communication systems,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2530–2550, 2021.

[20] L. Ge, P. Dong, H. Zhang, J.-B. Wang, and X. You, “Joint beamforming and trajectory optimization for intelligent reflecting surfaces-assisted UAV communications,” *IEEE Access*, vol. 8, pp. 78 702–78 712, 2020.

[21] H. Lu, Y. Zeng, S. Jin, and R. Zhang, “Enabling panoramic full-angle reflection via aerial intelligent reflecting surface,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Dublin, Ireland, 2020.

[22] Z. Chu, P. Xiao, M. Shojafar, D. Mi, J. Mao, and W. Hao, “Intelligent reflecting surface assisted mobile edge computing for internet of things,” *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 619–623, 2021.

[23] F. H. Danufane, M. Di Renzo, J. de Rosny, and S. Tretyakov, “On the path-loss of reconfigurable intelligent surfaces: an approach based on Green’s theorem applied to vector fields,” 2020, https://arxiv.org/abs/2007.13158.

[24] S. W. Ellingson, “Path loss in reconfigurable intelligent surface enabled channels,” 2019, https://arxiv.org/abs/1912.06759.

[25] S. J. Orfanidis, *Electromagnetic Waves Antennas*, Rutgers Univ, New Brunswick, NJ, USA, 2002.

[26] M. Cui, W. Han, D. Xu, P. Zhao, and W. Zou, “Hybrid precoding design based on alternating optimization in mmWave massive MIMO systems aided by intelligent reflecting surface,” *Computer Communications*, vol. 180, pp. 188–196, 2021.

[27] M. Yong-Jie and Y. Wen-Xia, “Research progress of genetic algorithm,” *Application Research of Computers*, vol. 29, no. 4, pp. 1201–1206, 2012.

[28] S. Mirjalili, “Genetic algorithm,” in *Evolutionary Algorithms and Neural Networks*, pp. 43–55, Springer, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1860946, 19 pages
https://doi.org/10.1155/2022/1860946

Research Article

Intelligent Roller Bearing Fault Diagnosis in Industrial Internet of Things

Ji Xu, Hong Zhou, and Yanjun Fang

Department of Automation, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Hong Zhou; hzhouwuhee@whu.edu.cn

Received 26 November 2021; Accepted 13 January 2022; Published 30 March 2022

Academic Editor: Mohammad R Khosravi

Advanced research studies on industrial Internet of things require effective feature extraction and accurate machinery health state evaluation. For roller bearing, a well-known mechanical component most extensively used in the industry, its running status directly affects the operation of the entire machinery and equipment. For intelligent fault diagnosis of roller bearing, the selection of the intrinsic mode function (IMF) modes in approaches of ensemble empirical mode decomposition (EEMD)/variational mode decomposition (VMD) becomes a tricky problem. To solve this problem, this study proposed an efficient scheme on roller bearing fault diagnosis that combines the refined composite multivariate multiscale sample entropy (RCMMSE) with different classifiers. Firstly, the synthetic noise signals are introduced to compare the effectiveness of the multiscale sample entropy (MSE) and the RCMMSE models. Secondly, the random noise signals are used to compare the performance of EEMD and VMD methods, where the envelope spectrum characteristics of fault signals are well described. Moreover, EEMD/VMD methods are utilized to decompose the roller bearing vibration signals into various modes to get the entropy values. Finally, the obtained RCMMSE is adopted as a feature vector and subsequently employed as an input of support vector machine, random forest, and probabilistic neural network models to conduct roller bearing fault identification. The extensive experimental results prove that this proposed scheme performs well and the classification accuracy of VMD-RCMMSE is higher than EEMD-RCMMSE.

1. Introduction

With the rapid development of Internet of things (IoT) [1, 2] and Industry 4.0 [3], there are increasingly massive real-time data from various types of mechanical equipment [4]. The availability of these data that contain abundant information about machine health has attracted more and more enterprises' attention. It has been proved that large volume, high velocity, and diversity mechanical big data are the major properties of mechanical big data [5, 6]. Effective feature extraction from these data and accurate machinery health state evaluation with ever-accelerated updating of schemes have become hot research issues in the prognostic and health management systems in the era of industrial IoT [7].

For roller bearing, a well-known mechanical component most extensively used in the industry, its operating status directly affects the operation of the entire machinery and equipment. Roller bearing failure is an important factor leading to mechanical equipment failure, so timely detection and fault diagnosis are of great significance, and analysing the vibration signal collected by the sensors to determine its failure is a commonly adopted scheme [8]. As the roller bearings are usually working in the vibration source environment, with the destruction of complex forms, the vibration signal is typically nonlinear and nonstationary. How to excavate the fault feature based on the vibration signals has been a research hotspot [9, 10].

The widely used signal decomposing schemes include wavelet analysis [11, 12] and EMD [13]. However, the wavelet transformation is essentially nonadaptive due to the configuration of wavelet basis and decomposing layers. In EMD, the complicated signal can be self-adaptively decomposed into some IMFs with a residual component. Nevertheless, mode mixing is a stumbling block in EMD. To deal with this challenge, EEMD is proposed [14], in which a complex signal can be disintegrated into IMFs in terms of the local time-scale characteristic of the signal. In recent, EEMD has been extensively employed in fault detection [15].

Using the EMD or EEMD schemes for fault feature extraction has been widely concerned. However, these two schemes have some disadvantages. The recursive mode decomposition in EMD/EEMD will propagate the envelope estimation error continuously. The signal contains no noise or intermittent signal, which leads to the decomposition of the mode mixing. Although the white noise scheme was used to suppress the mode mixing, the scheme needs to be carried out several hundreds of times of EMD/EEMD operation and will break out the signal composed of more than a real component. Moreover, EMD and EEMD cannot be separated correctly from the close frequencies, and it is quite challenging to select the suitable number of IMFs when applying EMD and EEMD.

To address the abovementioned problems, a neoteric scheme of signal decomposition estimation called VMD was proposed [16–18]. The whole framework of the alternating direction multiplier scheme was resorted to consecutively refresh the modes and their centre frequencies and gently demodulate the modes into the corresponding base frequency bands. Ultimately, each mode and corresponding centre frequencies were abstracted together. In comparison with the recursive filter pattern of EMD and EEMD, VMD converts the signal into variational and non-recursive decomposition patterns, and it consists of adaptive Wiener filter groups in nature. VMD can isolate two pure harmonic signals with similar frequencies [16–18]. The IMFs obtained by EEMD denote the natural oscillatory pattern inserted into the signals, where the entropy values of every IMF were often abstracted as a feature to discover the properties of the vibration signals [19, 20].

Entropy was one of the schemes to evaluate the time-series complexity. S. M. Pincus proposed the approximate entropy (AE) [21]. However, the length of the time series makes a critical influence on the performance of AE scheme. Hence, the value of AE is conformably lower than the intended one and fails relative coherence particularly when the data length is short. Based on AE, SE was proposed to deal with this challenge [22]. It has the advantages of short data, stable and low noise and interference capacity, and good consistency in the region of large parameter range. It can be noticed that the irregularity of time series can be only reflected by AE and SE individually. When the roller bearings fail, not only the frequencies but also the corresponding complexity of vibration signals has tremendous deviations. Accordingly, ME can be viewed as a property index for fault diagnosis [23]. Considering the MSE, the feature of the vibration signals can be abstracted under all kinds of conditions, and the eigenvectors were counted as the input of adaptive neuro-fuzzy inference system (ANFIS) for roller bearing fault recognition [22]. Moreover, the SE, to some extent, was undefined as no standard vectors, which were in accordance with one another. Undefined or imprecise SE leads to the degradation of authenticity of MSE algorithm. To overcome this drawback, the RCMMSE was revolved by Zhang et al. [24] to overcome these challenges. It is demonstrated that RCMMSE can not only ascend the precision of entropy assessment but also descend the probability of inducing undefined entropy.

However, after EEMD and VMD, a necessary step is how to select the number of modes. For example, the number of patterns was usually determined by the correlation coefficient and mutual information scheme between each component and the original signals [18, 25]. In addition, MSE and RCMMSE can only get the single-channel signal. Therefore, it cannot precisely display the overall signal information. The multivariate multiscale sample entropy (MMSE) was a scheme that gets different time series and conducts various embedding aspects, delay time, and amplitude ranges of data channels in a strict way. Hence, it can directly analyse multichannel data [26]. Therefore, a scheme that relied on VMD and RCMMSE was proposed in this study to get the vibration feature of roller bearings.

With the advent of computer techniques, a lot of fault recognition algorithms, such as SVM, RF, and PNN [27, 28], have been broadly applied to fault diagnosis. Therefore, the SVM, RF, and PNN were resorted to fulfill the fault recognition.

As mentioned above, a scheme based on RCMMSEVMD and VMD is designed. At first, the vibration signals were disintegrated into series IMF/BL-IMF modes by the EEMD/VMD. Secondly, the RCMMSE was utilized to figure out the entropy values. Finally, the SVM/RF/PNN models were utilized to achieve the fault recognition.

The rest of the study is organized as follows. It is shown that the review of VMD and RCMMSE schemes, respectively, is presented in Section 2. The comparison of MSE/RCMMSE and EEMD/VMD is presented in Section 3. Section 4 gives the procedures of the proposed scheme, experimental data sources, and parameter selection. Section 5 provides the experimental validation. Conclusions are given in Section 6.

2. Theoretical Framework of VMD and RCMMSE Models

2.1. Basic Principle of Variational Mode Decomposition. The VMD process is divided into establishment and solution of variational constraint problems. Assuming there is a limited bandwidth in each mode, the variational problem is formulated as k mode functions $u_k(t)$, which could minimize the estimated bandwidth. The summation of each mode is constrained to be corresponded to the input signal f. In particular, the structure of this problem could be divided into the following steps.

2.1.1. Variational Problem Formulation

① Hilbert transform: the analytic signal of every mode function is obtained, whose purpose is to get its spectrum:

$$\left[\delta(t) + \frac{j}{\pi t}\right] * u_k(t). \tag{1}$$

② The centre frequency $e^{-j\omega_k t}$ of each modal signal is estimated. The spectrum of each mode is modified to be adaptive to its own baseband.

$$\left[\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)\right]e^{-j\omega_k t}. \tag{2}$$

③ The L^2 is derived, and it denotes the signal gradient, and the signal's bandwidth is obtained. Therefore, constrained variational questions can be obtained as follows:

$$\begin{cases} \min\limits_{\{u_k\},\{\omega_k\}}\left\{\sum\limits_k \left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)\mu_k(t)\right]^{-j\omega_k t}\right\|^2\right\} \\ s.t. \sum\limits_k \mu_k = f \end{cases}, \tag{3}$$

where $\{\mu\}$ is the decomposition of the k mode components, $\{\mu\}=\{\mu_1,\mu_2,\ldots,\mu_k\}$, $\{\omega\}$ is the centre frequencies of each mode, and $\{\omega\}=\{\omega_1,\omega_2,\ldots,\omega_k\}$.

2.1.2. The Solution of the Problem

① The above constraint problem could be modified as a nonbinding problem by adding it to the penalty factor ∂ and the Lagrangian multiplication operator $\lambda(t)$, and here, the second penalty factor is used to guarantee the accuracy of the signal with the Gaussian noise, and the Lagrangian operator is adopted to ensure the constraint condition satisfied strictly, and the extended Lagrangian description is as follows:

$$L(\{u_k\},\{\omega_k\},\lambda) := \partial\sum_k\left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)\right]e^{-j\omega_k t}\right\|^2 + \left\|f(t)-\sum_k u_k(t)\right\|_2^2 + \left\langle \lambda(t), f(t)-\sum_k u_k(t)\right\rangle. \tag{4}$$

② VMD used multiplication operator alternating direction method of multipliers (ADMMs) to solve the above variable problem. By alternately updating A, B, and C to seek the "saddle" of the Lagrangian expression:

$$u_k^{n+1} = \underset{u_k\in k}{\operatorname{argmin}}\left\{a\left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)\right]e^{-jw_k t}\right\|_2^2 + \left\|f(t)-\sum_i u_i(t)+\frac{\lambda(t)}{2}\right\|_2^2\right\}, \tag{5}$$

where ω_k is equal to ω_k^{n+1}, and $\sum_t u_i(t)$ is equal to $\sum_{i\neq k}u_i(t)^{n+1}$.

Equation (5) is transformed to another domain by frequency by the Parseval/Plancherel Fourier equidistant transformation:

$$\widehat{u}_k^{n+1} = \operatorname{argmin}\left\{a\left\|j\omega\left[(1+\operatorname{sgn}(\omega+\omega_k))\cdot\widehat{u}_k(\omega+\omega_k)\right]\right\|_2^2 + \left\|\widehat{f}(\omega)-\sum_i\widehat{u}_i(\omega)+\frac{\widehat{\lambda}(\omega)}{2}\right\|_2^2\right\}. \tag{6}$$

The first item of ω with $\omega-\omega_k$ instead is as follows:

$$\widehat{u}_k^{n+1} = \arg\min\left\{a\left\|j(\omega-\omega_k)\left[(1+\operatorname{sgn}(\omega))\widehat{u}_k(\omega)\right]\right\|_2^2 + \left\|\widehat{f}(\omega)-\sum_i\widehat{u}_i(\omega)+\frac{\widehat{\lambda}(\omega)}{2}\right\|_2^2\right\}. \tag{7}$$

Equation (7) is converted into a nonnegative integral form with frequency interval:

$$\widehat{u}_k^{n+1} = \underset{\widehat{u}_k,u_k\in X}{\arg\min}\left\{\int_0^\infty 4a(\omega-\omega_k)^2\left|\widehat{u}_k(\omega)\right|^2 + 2\left|\widehat{f}(\omega)-\sum_i\widehat{u}_i(\omega)+\frac{\widehat{\lambda}(\omega)}{2}\right|^2 d\omega\right\}. \tag{8}$$

In this case, the solution of the quadratic optimization problem is as follows:

$$\widehat{u}_k^{n+1}(\omega) = \frac{\widehat{f}(\omega)-\sum_{i\neq k}\widehat{u}_i(\omega)+\widehat{\lambda}(\omega)/2}{1+2a(\omega-\ \omega_k)^2}. \tag{9}$$

Based on the above process, the centre frequency value is converted to the frequency domain:

$$\omega_k^{n+1} = \arg\min\left\{\int_0^\infty (\omega-\omega_k)^2\left|\widehat{u}_k(\omega)\right|^2 d\omega\right\}. \tag{10}$$

The centre frequency updated scheme is as follows:

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega\left|\widehat{u}_k(\omega)\right|^2 d\omega}{\int_0^\infty \left|\widehat{u}_k(\omega)\right|^2 d\omega}, \tag{11}$$

where $\widehat{u}_k^{n+1}(\omega)$ is equivalent to the current remaining amount of $\widehat{f}(\omega) - \sum_{i\neq k}\widehat{u}_i(\omega)$ Wiener filtering. A is the central gravity of the spectrum of the current mode function. For inverse Fourier transmission of $\widehat{u}_k(\omega)$, then the real part is $\{u_k(t)\}$.

The procedure of VMD algorithm is as follows:

(1) Initialize the $\{\widehat{u}_k^{(1)}\}$, $\{\omega_k^{(1)}\}$, $\{\widehat{\lambda}^{(1)}\}$, and n.

(2) Update u_k and ω_k followed by (9) and (11).

(3) Update λ, and

$$\widehat{\lambda}^{n+1}(\omega) \leftarrow \widehat{\lambda}^{n}(\omega) + \tau\left[\widehat{f}(\omega) - \sum_k \widehat{u}_k^{n+1}(\omega)\right]. \tag{12}$$

(4) For a given accuracy $e > 0$, if $\sum_k \|\widehat{u}_k^{n+1} - \widehat{u}_k^{n}\|_2^2 / \|\widehat{u}_k^{n}\|_2^2 < e$, stop iteration; otherwise, return to Step 2.

2.2. Basic Principle of Refined Composite Multivariate Multiscale Sample Entropy

2.2.1. Sample Entropy. The SE uses the measurement of the exponential function for two sequences with a tolerance r from m points to remain r of each other at the next point. For a time series with N sample points, the procedure of SE calculation is shown as follows:

(1) The m length vectors $X_m(i)$ are formed:

$$\begin{aligned}&\{X_m(i) = x(i), x(i+1), \ldots, x(i+m-1)\} - u(i),\\ &1 \leq i \leq N - m + 1,\end{aligned} \tag{13}$$

where m represents embedding dimension and $X_m(i)$ has m consecutive values. Commencing with the ith point and generalized by removing a baseline:

$$u(i) = \frac{1}{m}\sum_{j=0}^{m-1} x(i+j). \tag{14}$$

(2) For each $X_m(i)$, the similarity degree between the $X_m(i)$ and its neighboring vector $X_m(j)$ is calculated by $D_{i,j}^m$:

$$D_{i,j}^m = \mathrm{dist}\left(d_{i,j}^m, r\right), \tag{15}$$

where $d_{\mathrm{i,j}}^m$ denotes the maximum absolute difference in the corresponding scalar components of $X_m(i)$ and $X_m(j)$.

(3) For each $X_m(i)$ and the fixed tolerance r, let A_i be the number of vectors that satisfy, and then, $B_i^m(r)$ is denoted as follows:

$$B_i^m(r) = \frac{A_i}{N - m + 1}, \quad 1 \leq i \leq N - m. \tag{16}$$

(4) The average of the $B_i^m(r)$ is designated as follows:

$$B^m(r) = \frac{1}{N-m}\sum_{i=1}^{N-m} B_i^m(r), \tag{17}$$

where r represents the boundary width of the exponential function.

(5) Increasing dimension m to $m+1$, steps (1)-(4) are repeated to calculate the corresponding of SE values to find $B^{m+1}(r)$, and SE is defined as follows:

$$SampleEn(m, r, N) = \sum_{N\longrightarrow\infty} -\ln\frac{B^{m+1}(r)}{B^m(r)}. \tag{18}$$

When N is finite, the SE can be estimated as follows:

$$\mathrm{SampleEn}(m, r, N) = \left[-\ln\frac{B^{m+1}(r)}{B^m(r)}\right] = \ln B^m(r) - \ln B^{m+1}(r). \tag{19}$$

SE determines the time sequence irregularity on the single scale. The smaller the value of FE is, the higher time sequence self-similarity can be achieved. Conversely, the greater the FE value is, the more complicated time sequences without rules can be achieved.

2.2.2. Multiscale Entropy and Multiscale Sample Entropy. ME could be represented as a series of times with different scales, which is obtained through a coarse-grained process. The irregularity of time series is then presented by the ME, where the self-similarity of various scales is reflected. Considering the scale entropy, when there is a sequence with a higher entropy than another sequence, the corresponding complexity will also be higher than the other. That is, if there is a case that the increasing scale of a time series conversely results in the decreased entropy value, there are more chances that the time series holds a relatively simple sequence structure.

The MSE is obtained as follows:

(1) Consider a discrete-time series $\{X_m(i)\colon 1 \leq i \leq N\}$. Firstly, the value of embedding dimension m and similar tolerance r of the SE are set. Another auxiliary time sequence is then constructed in a vector form, which is presented as $y_k^\tau = \left\{y_{k,1}^\tau, y_{k,2}^\tau, \ldots y_{k,p}^\tau\right\}$ and named coarse-grained vector.

$$y_k^\tau = \frac{1}{\tau}\sum_{i=(j-1)\tau+k}^{j\tau+k-1} X_m(i), 1 \leq j \leq \frac{N}{\tau}, 1 \leq k \leq \tau, \tag{20}$$

where τ is the scale factor. Actually, we can find that when $\tau = 1$, the coarse-grained time series is equal to the previous one $\{X_m(i)\}$, which means that by dividing the length of time series, we can decompose the original time series into a coarse-grained vector series y_k^τ. It is worth mentioning that the number of coarse-grained time series y_k^τ is τ and the length is N/τ.

(2) Based on the different scales τ of the time sequence, the SE values are obtained. Generally, the r in SE takes the standard deviation of the primary time

series when calculating the value of SE. This procedure is called MFE analysis.

$$\mathrm{MSE}(X, \tau, m, r) = \mathrm{SampleEn}\left(m, r, y_1^{\tau}\right). \tag{21}$$

2.2.3. Refined Composite Multiscale Sample Entropy. In the process of applying coarse graining, the length of the original time series is declined by introducing a factor of τ, as shown in equation (13). For example, the authors developed the RCMMSE algorithm with the aim to boost the accuracy of the MSE. In particular, the SE of all the time series is obtained at a certain scale of factor τ. The RCMMSE value is defined as follows:

$$\mathrm{RCMMSE}(X, \tau, m, r) = -\ln \frac{\overline{B}^{m+1}(r)}{\overline{B}^{m}(r)}, \tag{22}$$

where $\overline{B}^{m+1}(r) = 1/\tau \sum_{k=1}^{\tau} B^{m+1}(r)$ and $\overline{B}^{m}(r) = 1/\tau \sum_{k=1}^{\tau} B^{m}(r)$. Thus, the RCMMSE value can be obtained as follows:

$$\mathrm{RCMMSE}(X, \tau, m, r) = -\ln \frac{\overline{B}^{m+1}(r)}{\overline{B}^{m}(r)} = -\ln\left(\frac{1/\tau \sum_{k=1}^{\tau} B^{m+1}(r)}{1/\tau \sum_{k=1}^{\tau} B^{m}(r)}\right) = -\ln\left(\frac{\sum_{k=1}^{\tau} B^{m+1}(r)}{\sum_{k=1}^{\tau} B^{m}(r)}\right). \tag{23}$$

2.2.4. Refined Composite Multivariate Multiscale Sample Entropy. Given time series $X_m^p(i)$ with p variables, the RCMMSE is calculated according to the following procedures:

(1) Each coarse-grained time series $y_k^{\tau} = \{y_{k,1}^{\tau}, y_{k,2}^{\tau}, \ldots, y_{k,p}^{\tau}\}$ requires to compute the RCMMSE, the multiple embedding vectors M, and therefore, the original $X_m^p(i)$ is reconstructed using the multiple embedding vectors M.

$$X_m^p(i) = [x_1(i), \ldots x_1(i + (m_{1-1})\lambda_1), x_2(i), \ldots x_2(i + (m_{2-1})\lambda_2), \ldots x_1(i), \ldots x_1(i + (m_{1-1})\lambda_1)], \tag{24}$$

here $\lambda = [\lambda_1, \lambda_2, \ldots \lambda_p]$ is the time delay vector.

(2) The maximum distance of two vectors $X_m^p(i)$ and $X_m^p(i)$ is defined as follows:

$$D[X_m^p(i), X_m^p(j)] = \max_{l=1,2,\ldots,m} \{|x(i + l - 1) - x(j + l - 1)|\}. \tag{25}$$

(3) For a composite time delay $X_m^p(i)$ and threshold r, the parameter P_i is the number of the vector pairs, and therefore, $D[X_m^p(i), X_m^p(j)] \leq r, \quad j \neq i$, and then, the probability $B_i^m(r)_{\mathrm{RCMMSE}}$ is computed as follows:

$$B_i^m(r)_{\mathrm{RCMMSE}} = \frac{P_i}{N - m + 1}, \quad 1 \leq i \leq N - m, \tag{26}$$

here $m = \max\{M\} * \max\{\lambda\}$.

(4) The mean of the $B_i^m(r)_{\mathrm{RCMMSE}}$ is obtained as follows:

$$\overline{B}_i^m(r)_{\mathrm{RCMMSE}} = \frac{1}{N - m} \sum_{i=1}^{N-m} B_i^m(r). \tag{27}$$

Increasing dimension m to $m + 1$, steps (1)-(4) are repeated to get the corresponding SE values to find $B_i^{m+1}(r)_{\mathrm{RCMMSE}}$ and $\overline{B}_i^{m+1}(r)_{\mathrm{RCMMSE}}$.

$$\overline{B}_i^{m+1}(r)_{\mathrm{RCMMSE}} = \frac{1}{p(N - m - 1)} \sum_{i=1}^{p(N-m-1)} B_i^{m+1}(r). \tag{28}$$

The calculation of the RCMMSE is defined as follows:

$$\mathrm{RCMMSE}(X, \tau, m, r) = -\ln\left(\frac{\overline{B}_i^{m+1}(r)_{\mathrm{RCMMSE}}}{\overline{B}_i^{m}(r)_{\mathrm{RCMMSE}}}\right). \tag{29}$$

3. Scheme Comparisons

3.1. Comparison of EEMD/VMD. To compare the EEMD and VMD models, a simulation signal is used to prove that the VMD model is better than EEMD. The simulation signal $X(t)$ is superimposed by the multifrequency signal and the random noise signal with the standard deviation of 1, and the following formula is the signal definition.

$$\begin{aligned} X(t) = {} & [1 + \cos(2\pi \times 30t)] \times \cos(2\pi \times 125t) \\ & + [1 + \cos(2\pi \times 30t)] \times \cos[2\pi \times 155t + \cos(2\pi \times 5t)] \\ & + [1 + \cos(2\pi \times 30t)] \times \cos(2\pi \times 185t). \end{aligned} \tag{30}$$

The time-domain waveforms and its envelope spectrum of the original signal are given in Figure 1. As shown in Figure 1(b), the main frequency of the original signal focuses on 0–200 Hz, especially 30.27 Hz. Then, EEMD and VMD were used to decompose the aforementioned signal.

To achieve the VMD, firstly, the number of modes k should be determined. It should be noted that if the value of k in VMD is chosen too tiny, the original signal with time frequency could not be fully captured by the decomposition of the mode. Larger k values produce the similar frequency between the BL-IMF components, and they may be over-decomposition. Therefore, we use the observing centre frequency of the signal scheme to select the applicable k in

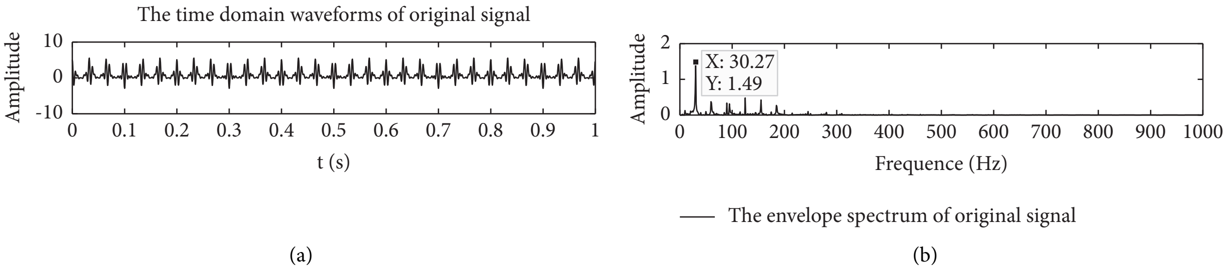


Figure 1: Time-domain waveforms and the envelope spectrum of the original signal. (a) Time domain and (b) envelope spectrum.

this study [19]. The results of the centre frequency corresponding to different k values are given in Table 1.

As shown in Table 1, there is a centre frequency difference of less than 0.008 kHz from BL-IMF3 to BL-IMF5 modes (0.0636 kHz in BL-IMF3, 0.0582 kHz in BL-IMF4, and 0.051 kHz in BL-IMF5) when $k = 5$. These three modes exist similar frequency. Therefore, the number of k values is selected as 4 in this study to decompose the above simulation signal.

The EEMD/VMD models were used to decompose the original signal, and the results of IMFs (EEMD) and BL-IMFs (VMD) are given in Figures 1(a) and 2(a). Figures 1(b) and 2(b) are the corresponding spectrum analysis results. As shown in Figure 2, each BL-IMF component is mainly distributed around a single frequency (30.27 Hz), compared with VMD in Figure 1(b), and some IMF components in Figure 2(b) have a series frequency from IMF4 mode.

The mentioned result shows that the decomposition effect of EEMD algorithm is not ideal for the multicomponent synthesis simulation signal, and the mode mixing is serious. Because some slight signals drown in the signal, which is to be decomposed, EEMD in the selection process of the three-spline envelope fitting leads to decomposition bias. The weak signal is embedded in the vast majority of the strong signal where the EEMD can be filtered and extracted, but when the weak signal appears only in the maximum slope range of the strong signal, the weak signal will be in the form of wave frequency modulation and does not produce additional local extreme points. Therefore, EEMD is difficult to extract the useful components and easy to produce some components with mode mixing.

It can be seen from Figure 1(b) that VMD can not only effectively remove the dummy components, but also each BL-IMF exhibits a mode in the range of a certain scale, and there is no mode mixing problem between them. Therefore, the scale characterization and decomposition effect of VMD and better than EEMD, and this indicates that VMD has good robustness Figure 3.

3.2. *Comparison of Validity of MSE/RCMMSE.*

In this section, the $1/f$ noise signal was employed to demonstrate the superior performance of the RCMMSE compared with MSE. Resorting to MSE and RCMMSE, the probabilities of inducing undefined entropy were presented to address the white noise and $1/f$ noises with 200 samples. For each sample, the length is chosen as 1000. The results of the corresponding probabilities are shown in Table 2. From Table 2, we can see that the probability of undefined entropy showed the same tendency as the time scale increased, while there was a decrease in the length of the time series. As given in Table 1, the probability of undefined entropy is about zero when they were used to analyse white noise in the MSE model. On the contrary, the probability of inducing undefined entropy is 0.01 when $\tau = 4$ for all 200 $1/f$ noise samples. In the MSE algorithm, the entropy is undefined when $B^m(r)$ or $B^{m+1}(r)$ is zero. However, the RCMMSE can successfully get the values of entropy from 1 to 20 when the white and $1/f$ noises are considered. Hence, the RCMMSE model is superior to MSE.

3.3. *Comparison of Accuracy in MSE/RCMMSE.*

In this section, the efficiency of the MSE/RCMMSE models was verified through case studies. In each simulation study, we employ noise samples with 1000 data points containing white and $1/f$ noises. The results of MSE/RCMMSEs are presented in Figure 4. Meanwhile, Figures 5 and 6 present the results of means and standard deviations of entropies.

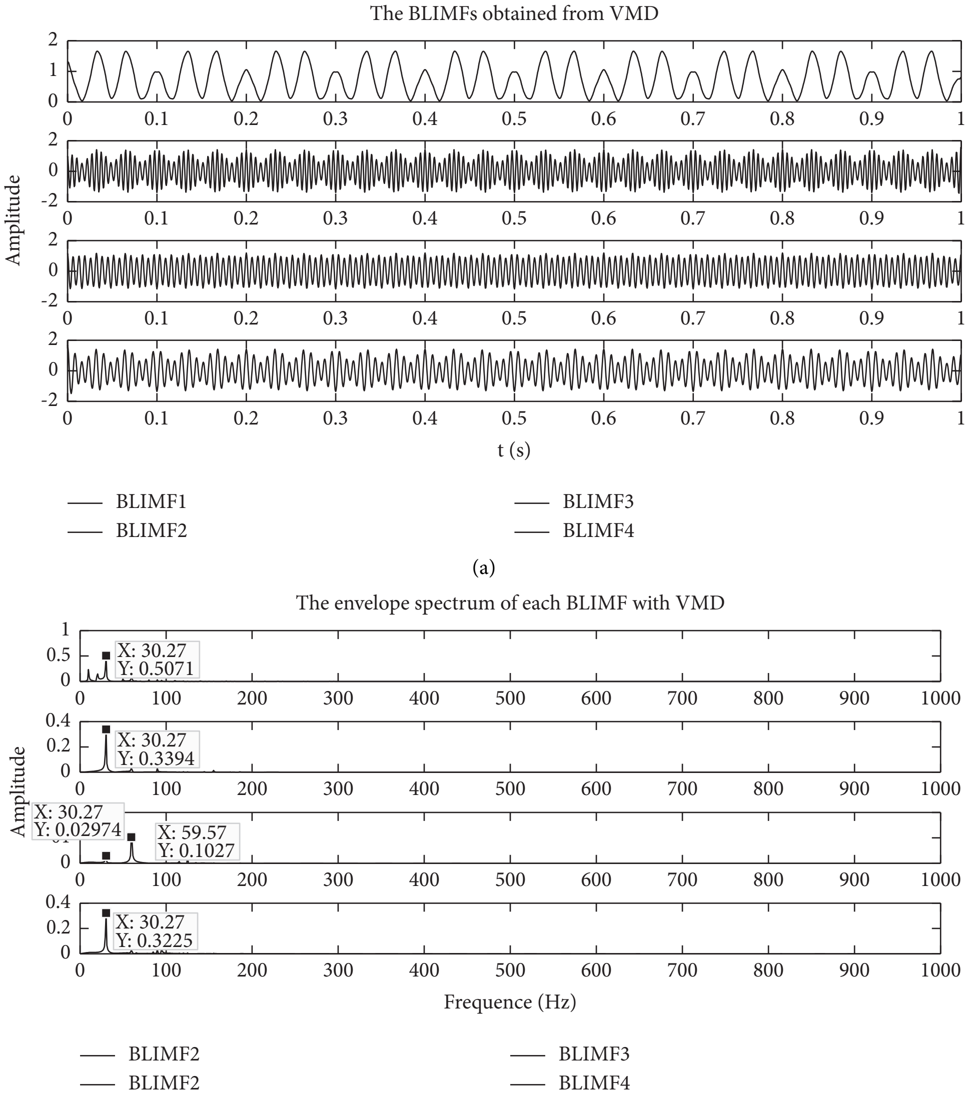
As seen in Figure 4(a), the overall recently RCMMSE model shows higher performance compared with the MSE model in the condition of white noise, resulting from the tolerance r in SE, which is utilized to evaluate the similarity between any two time series. Note that r was often chosen as $0.15 \times \mathrm{SD}$ of the original time series.

It can be seen from Figures 5(a) and 5(b) that the means of the entropy values obtained using the MSE/RCMMSE are nearly equal in white noise. Nevertheless, the means of the entropy values of the MSE are higher than that of the RCMMSE (see Figure 5(b)). Figure 6 shows that the standard deviation of RCMMSE is all lower than that of MSE, and this result indicates that the entropies obtained using the RCMMSE algorithm were more consistent than those obtained using the MSE algorithm.

With the purpose of studying the relationship between the data length and the effectiveness of the MSE/RCMMSE models, several different data lengths ($N = 600$, 1200, 1800, and 2400) are used to get the entropies. In Table 3, the means and standard deviation of the entropies are presented. It can be found that the results of the means of entropies by MSE and RCMMSE are nearly equivalent with the long primary time series ($N = 2400$). However, with short original time series, the results of means show a significant difference. On the other hand, the results of the RCMMSE algorithm hold the lowest error when the discrimination between the

Table 1: Centre frequency under different k values in VMD.

K	Centre frequency (kHz)					
	BL-IMF1	BL-IMF2	BL-IMF3	BL-IMF4	BL-IMF5	BL-IMF6
2	0.0027	0.0820	—	—	—	—
3	0.0022	0.0883	0.0621	—	—	—
4	0.0022	0.0888	0.0669	0.0574	—	—
5	0.0020	0.0869	0.0636	0.0582	0.0510	—
6	0.0020	0.0798	0.0943	0.0603	0.0510	0.0513



(a)

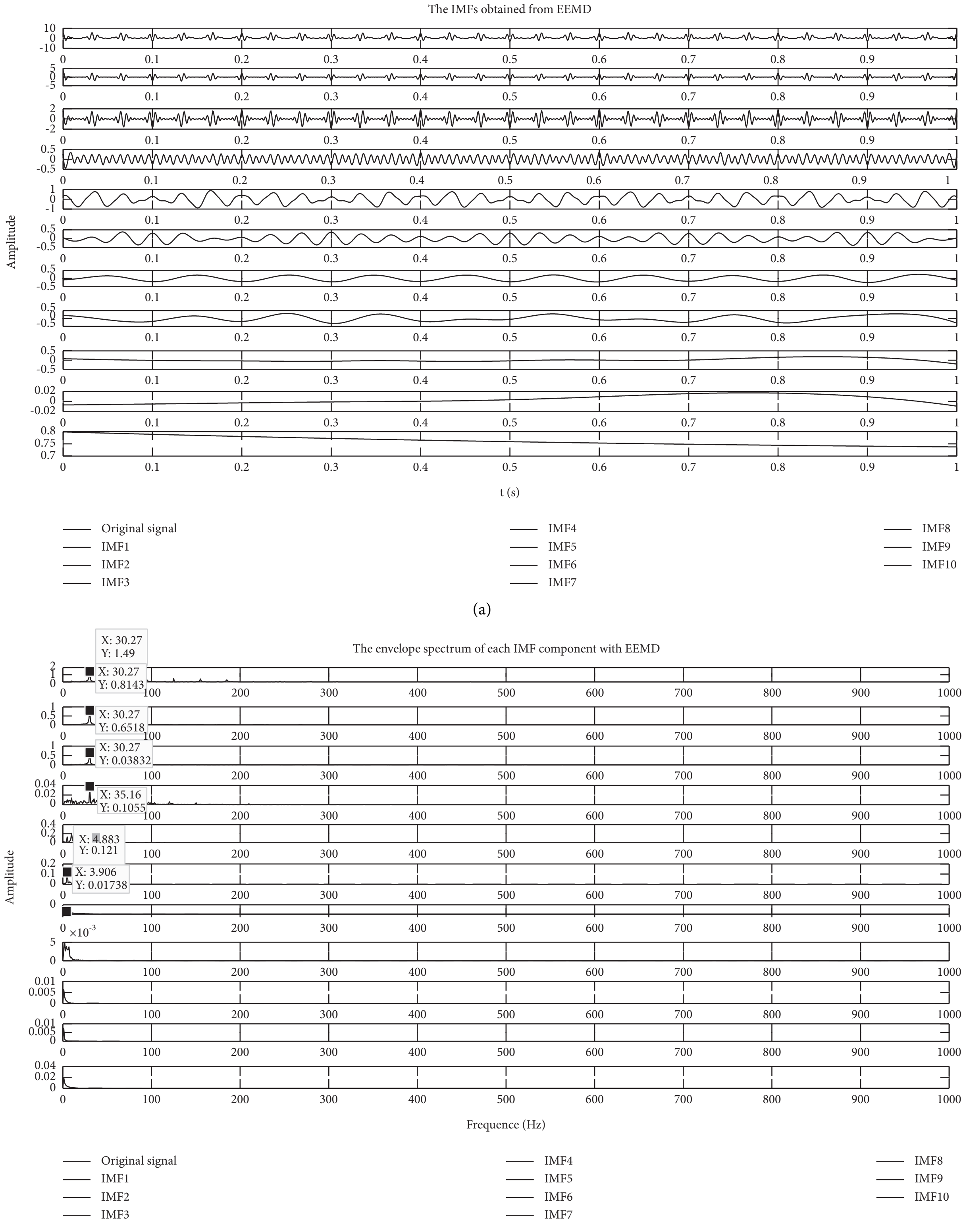
(b)

Figure 2: VMD signal decomposition. (a) The BL-IMF components. (b) The spectrum of each BL-IMF.

entropy of $1/f$ noise with 500 data points and the entropy of $1/f$ noise with 2000 data points is considered. That is to say, the RCMMSE has higher performance compared with other algorithms at any length of the time series. In conclusion, the RCMMSE is more reliable than MSE according to the above discussions.

4. Experimental Setup

4.1. The Dataset Source. The performance of the proposed scheme is verified through experiments in this part. Case Western Reserve University Bearing Data are used, which was obtained from an induction motor. It should be noted



(a)

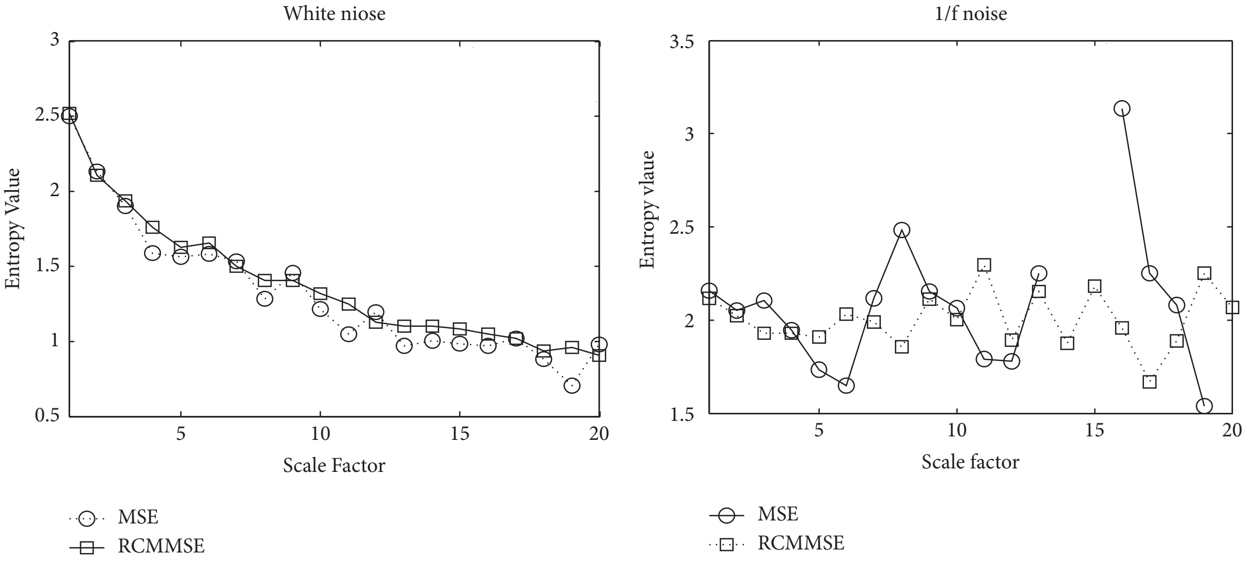
(b)

Figure 3: Signal decomposition (EEMD). (a) The IMF components. (b) The spectrum of each IMF mode.

that the data are generated by accelerometers at the drive end and fan end. Besides, EDM is employed to model the possible faults in the motor bearings. In particular, the fault tolerance is set as 0.1778 mm. After the fault detection, the data located in the outer race were aggregated in a state, which holds the time configuration at 6:00clock. To ensure

Table 2: Probabilities of undefined entropy.

Scale factor	MSE		RCMMSE	
	White noise	$1/f$ noise	White noise	$1/f$ noise
1–3	0	0	0	0
4	0	0.01	0	0
5	0	0.015	0	0
6	0	0.045	0	0
7	0	0.11	0	0
8	0	0.155	0	0
9	0	0.275	0	0
10	0	0.35	0	0
11	0	0.36	0	0
12	0	0.4824	0	0
13	0	0.4949	0	0
14	0	0.5606	0	0
15	0	0.5377	0	0
16	0	0.5714	0	0
17	0	0.7059	0	0
18	0	0.7211	0	0
19	0	0.6402	0	0
20	0	0.6667	0	0



(a)

(b)

Figure 4: MER/RCMMSE values of white and $1/f$ noises. (a) White noise. (b) $1/f$ noise.

the data acquisition system suitable for the vibration signals, the amplifier is particularly designed with a high bandwidth. On the other hand, to improve the accuracy, a sampling frequency of 12000 Hz of each channel was set for the data recorder.

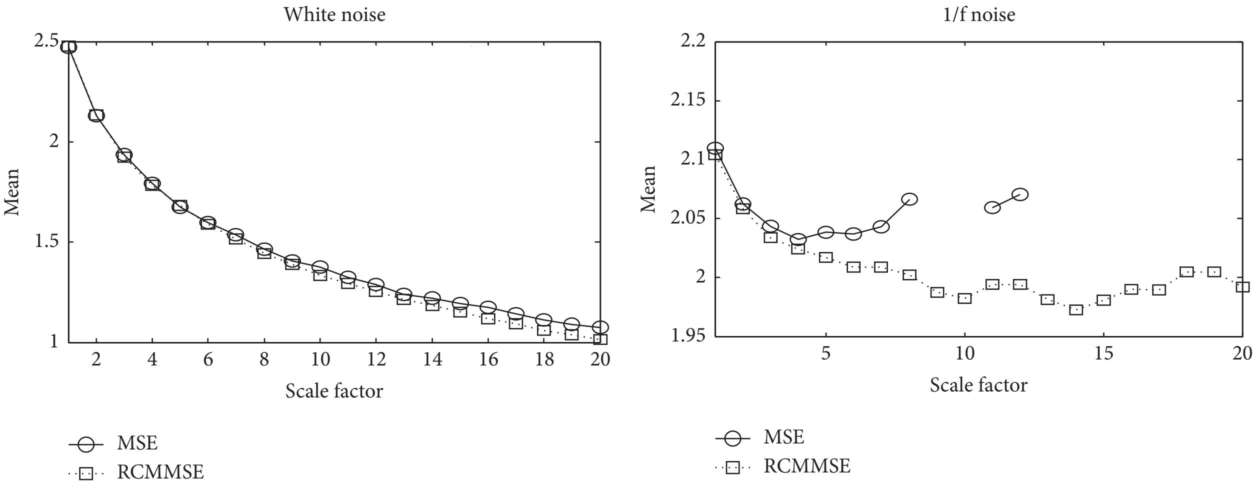
Table 4 demonstrates different working conditions, which are considered in this experiment. Note that 0.1778 mm is the fault tolerance. “Normal” and “slight” denote the fault severity. In addition, considering the drive end of the motor 1785 rpm is chosen as the motor revolving speed. In total, 200 data samples are adopted with each fault condition consisting of 51 samples. In particular, in each data sample, there are 2048 data points.

4.2. Procedure of Our Proposed Scheme. The procedure of the proposed scheme can be shown as follows:

(1) Resorting to EEMD and VMD models, the vibration signals under different cases were disintegrated into a sequence of IMF and BL-IMF modes.

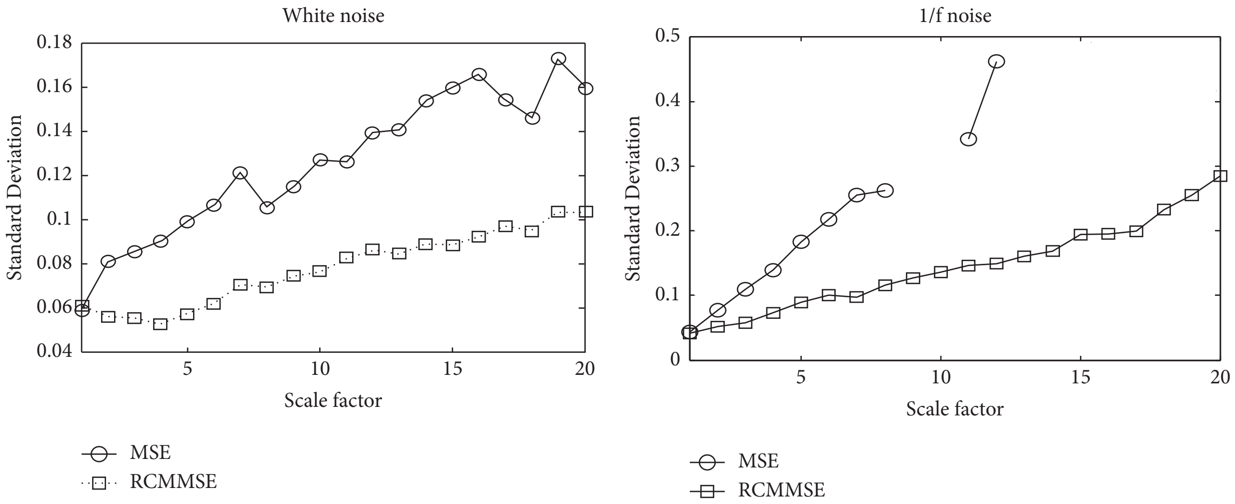
(2) Using the RCMMSE model to get the entropy values, the entropy values were regarded as the input of SVM/RF/PNN models for training and testing.

(3) Testing samples were regarded as the input of the trained SVM/RF/PNN classifier, while the operating conditions can be recognized by the output of the SVM/RF/PNN classifiers. The procedure of our



(a) (b)

Figure 5: Mean deviation of 200 analysis results of white and $1/f$ noises using MSE and RCMMSE. (a) White noise. (b) $1/f$ noise.



(a) (b)

Figure 6: Standard deviation of 200 analysis results of white noise and $1/f$ noises using MSE and RCMMSE. (a) White noise. (b) $1/f$ noise.

Table 3: Means and standard deviations of the MSE and RCMMSE of $1/f$ noise at a scale factor of 20.

Data length	MSE		RCMMSE	
	Means	Standard deviations	Means	Standard deviations
500	—	—	2.1323	0.5234
1000	—	—	2.0185	0.2667
1500	—	—	2.0185	0.1761
2000	—	—	1.9373	0.1230

Table 4: Experimental data of the roller bearings.

Fault category	Fault tolerance (mm)	Motor speed (rpm)	Number of samples	The fault severity
NR	0	1785	51	Normal
IRF	0.1778	1785	51	Slight
BF	0.1778	1785	51	Slight
ORF	0.1778	1785	51	Slight

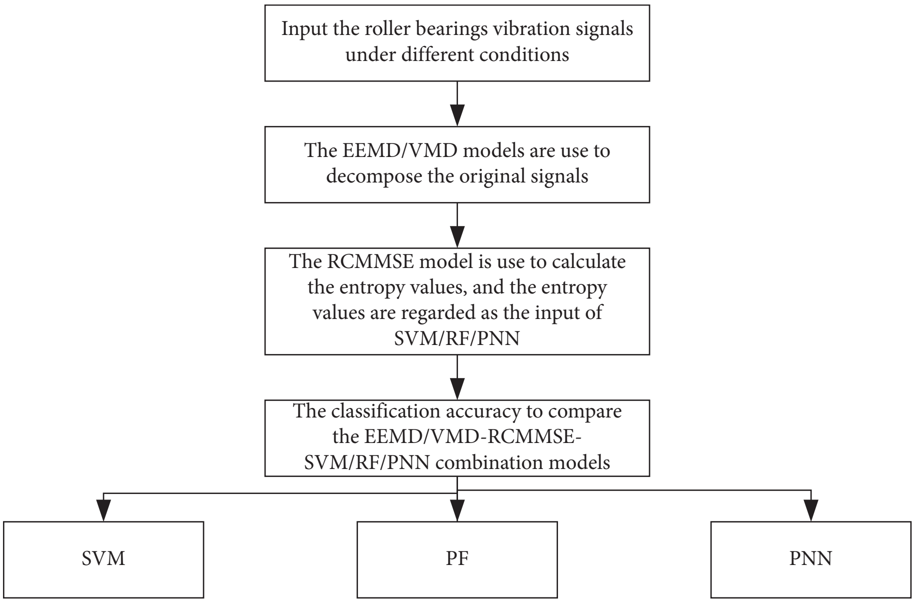

Input the roller bearings vibration signals under different conditions
The EEMD/VMD models are use to decompose the original signals
The RCMMSE model is use to calculate the entropy values, and the entropy values are regarded as the input of SVM/RF/PNN
The classification accuracy to compare the EEMD/VMD-RCMMSE-SVM/RF/PNN combination models
SVM
PF
PNN

Figure 7: Procedure of our proposed scheme.

proposed fault diagnosis scheme is shown in Figure 7.

4.3. Parameter Selection. There are some parameters in different models that should be present before its calculation:

(1) EEMD: considering EEMD, there are two parameters that need to be determined, namely the ensemble number m and $n_i(t)$ related to the white noise. In particular, $n_i(t)$ denotes the amplitude. It should be mentioned that there is a one-to-one correspondence between the result and the ensemble number. Note that the ensemble number should be chosen as a few hundred. In addition, consider the situation that the input signal holds the standard deviation, and a fraction of one percent of error will be caused by the remaining noise with the added noise. In particular, m is set as 100.

(2) VMD: the number of BL-IMF modes k is according to the centre frequency. The second penalty factor ∂ in equation (4) is often set as 2000.

(3) SE/RCMMSE:

① SE: before the process of SE, there are three parameters that must be chosen suitably. Firstly, considering the different lengths of sequences in SE, embedding dimension m should be designed. In particular, the dynamic process will be reconstructed in a more detailed manner with a larger m. However, it should also be mentioned that if m is chosen too large, the need of $N = 10^m - 30^m$ would be greatly limited, which will cause the loss of important information, and the general condition will be hard to satisfy. In the literature, m is often set as 2. Another two parameters, namely, similarity tolerance n and nr, are dependent on the gradient of the exponential function and its bound, respectively. Experimentally, $r = (0.1 - 0.25)\mathrm{SD}$, and r is chosen as 0.15SD in SE.

② RCMMSE: the multiple embedding vector in RCMMSE scheme is set as $M = [2_1, 2_2, 2_3, \ldots, 2_k]$, and here, k is the number of the BL-IMF components, and therefore, the time delay is set as 1.

(4) SVM: the kernel function in SVM is selected as the radial basis function (RBF).

(5) PNN: the distribution density of PNN is set as 1.5.

(6) RF: there are two parameters that need to be confirmed before using RF model, such as the number of input variables $mtry$ is selected randomly based on the M input variables. Regarding the scale factor τ as 20 and EEMD/VMD-RCMMSE as feature, the number of input variables $M = \tau = 20$, and the parameters often meet the condition $mtry \le \sqrt{M}$ [15]. Therefore, we let $mtry = 4$ and the number of the DT is set as 4 and 1000 in this simulation.

5. Experimental Results and Analysis

5.1. EEMD/VMD. The data are selected from the experiments in which SKF bearings are used. The number of each sample is set as NR:1–50, IRF: 51–100, BF:101–150, ad ORF:151–200. As limited space, here with a sample of each state for an example, the time-domain waveforms of vibration signals under different working conditions are shown in Figure 8.

The vertical axis is the acceleration vibration amplitude. Because of the influence of noise, it is difficult to find significant differences in different states. As shown in Figure 8, it is hard to distinguish the four signals; in particular, there is no obvious regularity in two states of NR and BF signals.

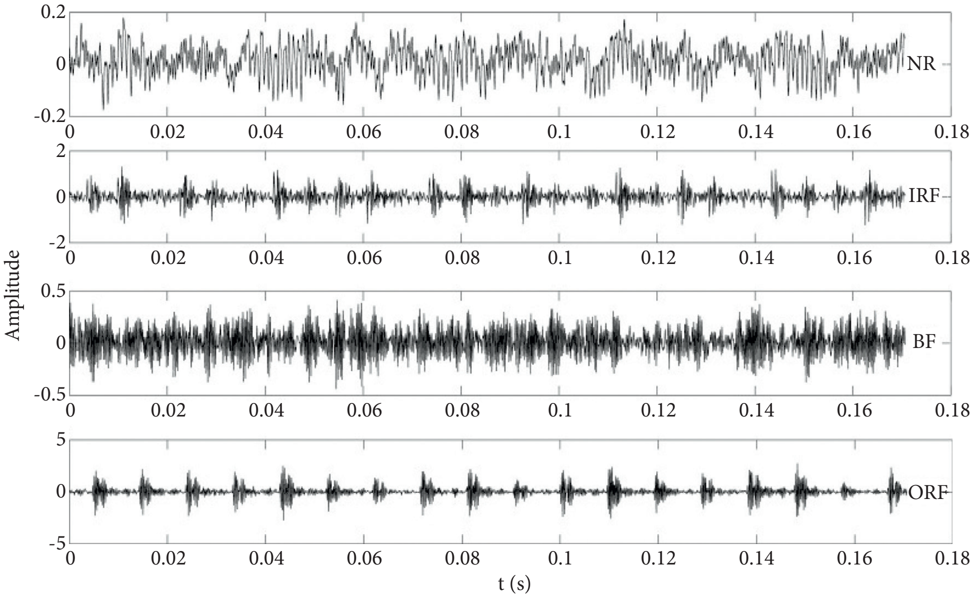


Figure 8: Time-domain waveforms of each working condition.

The EEMD and VMD were used to disintegrate the vibration signals into series modes (IMF and BL-IMF). Here, we use the IRF vibration signals to observe the frequency change situation, and the waveforms of one IRF vibration signal and its spectrum envelope are shown in Figure 9. As shown in Figure 5(b), the IRF signal fault frequency is 164.1 Hz. The results of the centre frequency under different k values with VMD are given in Table 5. There are some modes, such as BL-IMF4 (0.2787 kHz) and BL-IMF5 (0.3071 kHz), and with similar centre frequencies, when $k =$ 5 in Table 5, it is considered that there is over-decomposition. Therefore, the number of k values is selected as 4 to decompose the vibration signals.

Employing EEMD/VMD, the roller bearing primary signals in Table 4 are decomposed into IMF1-10 and BL-IMF1-4. Figures 10 and 11 show the VMD and the corresponding spectrum of each mode for an IRF signal.

It can be shown from Figure 6(b) that the VMD can not only effectively remove the dummy components, but also each IMF in the range of a certain scale, and there is no mode mixing problem between them. Therefore, the scale characterization and decomposition effect of VMD are better than EEMD, and this result reveals that the VMD owns a good robustness.

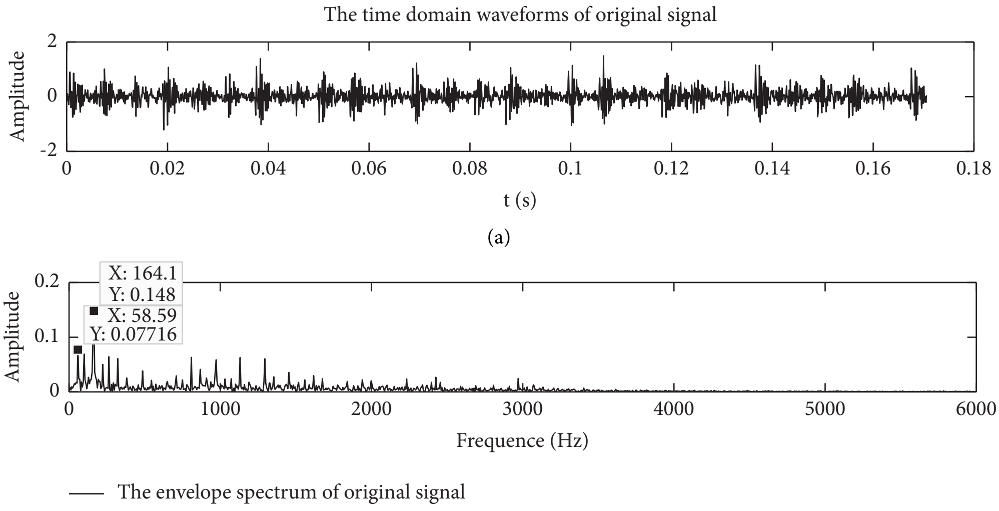
The correlation coefficient scheme is utilized to prove the degree of correlation between each mode and the primary signal to further prove the effectiveness of VMD scheme. In Tables 6 and 7, the overall average correlation values of BL-IMF component and original IRF signals are higher than IMF. Therefore, the decomposition effect of VMD is superior to EEMD.

After the vibration signal feature extraction using VMD and EEMD, the RCMMSE was employed to get the entropy value. The value of VMD-RCMMSE values under different k values and scale factors is shown in Figure 12.

Figure 12 shows that the RCMMSE values of the VMD effective mode decrease with the increasement in the scale factor and the number of modes k, and the overall RCMMSE values are close to the steady state when the value of k is more than 3, because the VMD algorithm decomposes the original signals into several BL-IMFs with information of different frequency bands. This can also be supported from Table 5. The correlation coefficient of most EEMD modes and the original signal is much smaller than that of VMD mode. This indicates that the VMD algorithm can overcome the mode mixing shortcoming in EEMD. The entropy values of EEMD-RCMMSE and VMD-RCMMSE ($k = 4$) are shown in Figure 13.

As shown in Figure 13(a), it is difficult to distinguish the four types of roller bearing signals with EEMD-RCMMSE, especially the NR and IRF, BF, and ORF. On the one hand, the complexity of signals under normal states and the conditions with ball fault is similar. In other words, the feature of the two kinds of signals is alike. On the other hand, much noise hidden in the signals may affect the identification of them. However, the RCMMSE curves of BL-IMF components can recognize the four states clearly in Figure 13(b).

5.2. Fault Identification. Regarding the extracted RCMMSE vectors as the inputs of the chosen fault classifier, such as SVM/RF/PNN, there are four kinds of data with 200 samples in total. 10, 20, and 30 samples are used under different statuses as the training samples, and the rest samples (20, 30, 40) as the testing samples are used to compare the

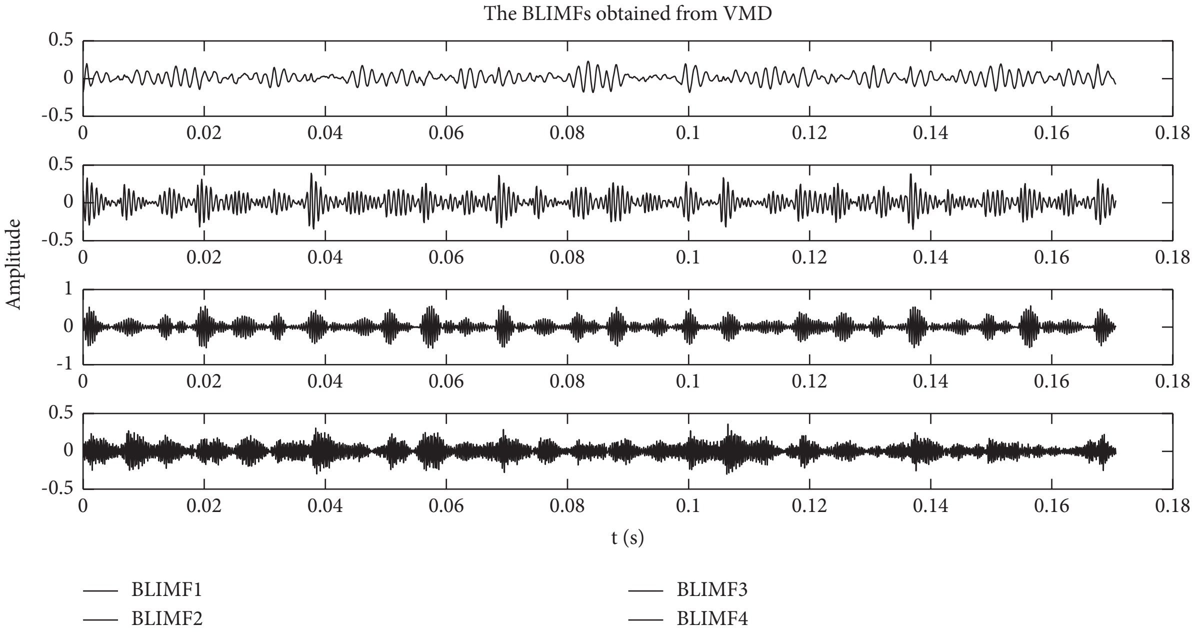


(b)

Figure 9: Waveforms and the envelope spectrum of the original signal in time domain. (a) Time domain. (b) Envelope spectrum.

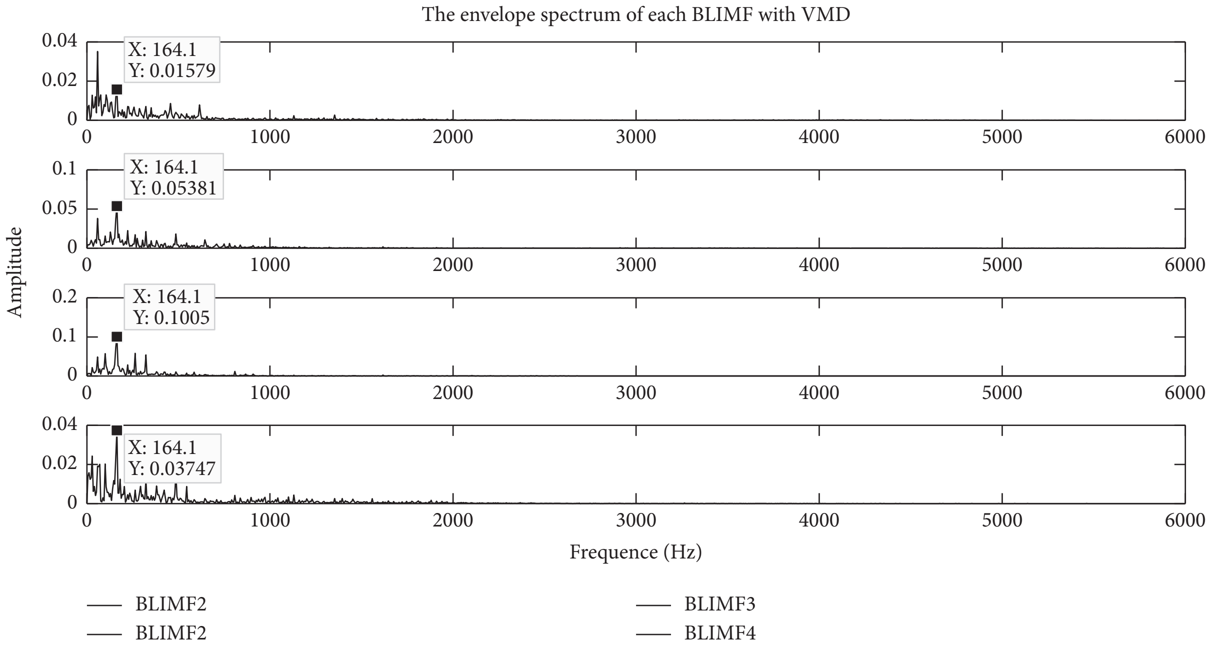
Table 5: Centre frequency under different k in VMD.

K	Central frequency (KHz)					
	BL-IMF1	BL-IMF2	BL-IMF3	BL-IMF4	BL-IMF5	BL-IMF6
2	0.0922	0.2631	—	—	—	—
3	0.0895	0.2246	0.2873	—	—	—
4	0.0494	0.1140	0.2268	0.2878	—	—
5	0.0490	0.1137	0.2239	0.2787	0.3071	—
6	0.0489	0.1136	0.2228	0.2730	0.2890	0.3144



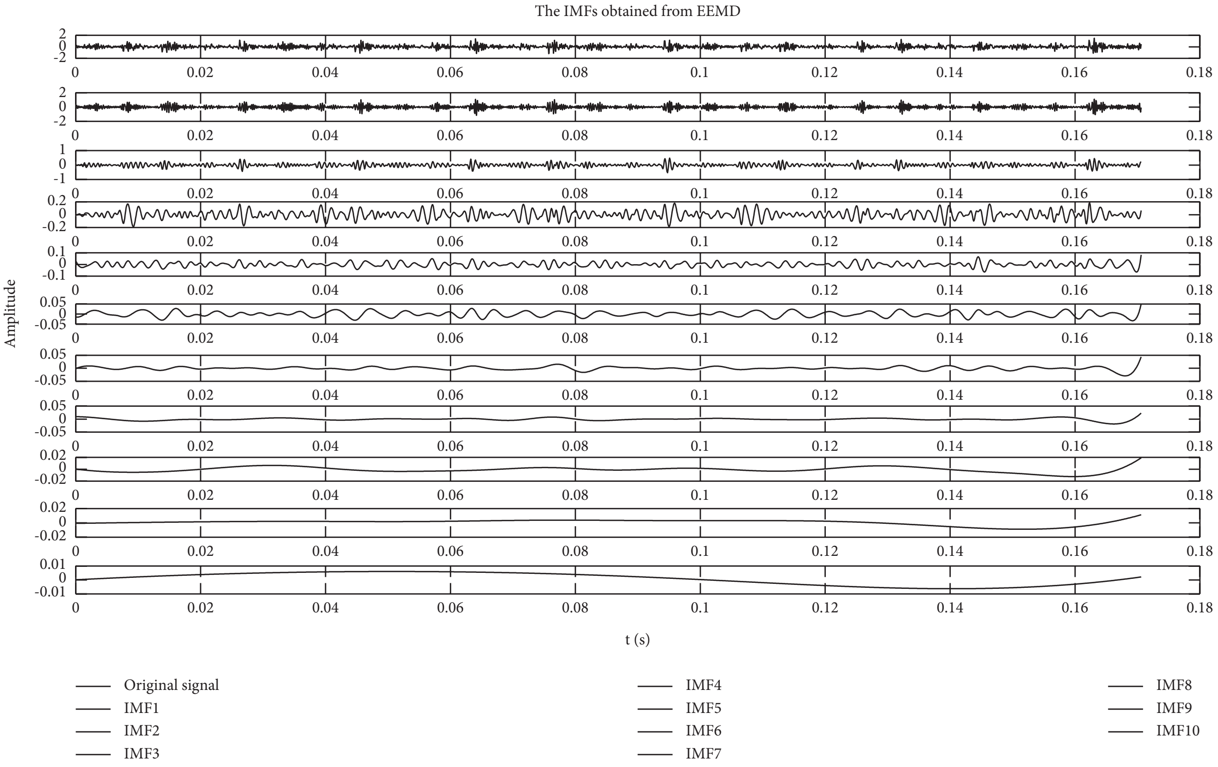
(a)

Figure 10: Continued.



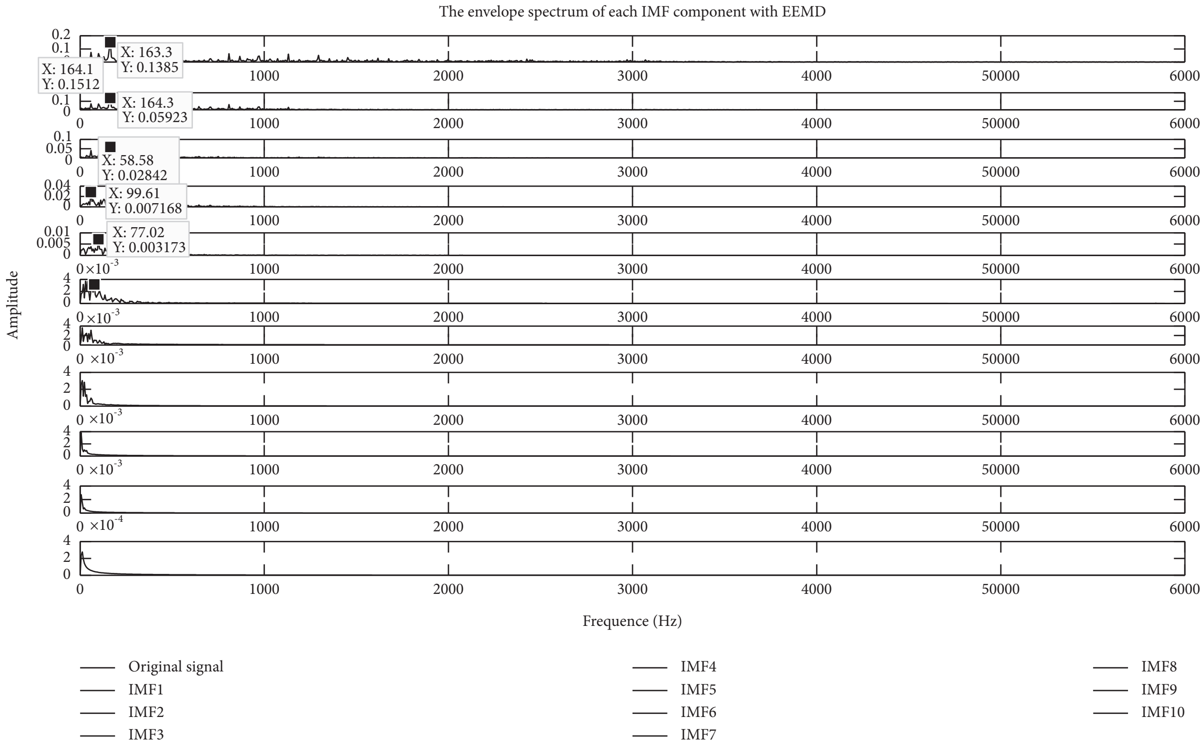
(b)

Figure 10: Signal decomposition based on VMD. (a) The BL-IMF components. (b) The spectrum of every BL-IMF.



(a)

Figure 11: Continued.



(b)

Figure 11: NR signal based on EEMD. (a) The IMF components. (b) The spectrum of each IMF.

Table 6: Total average correlation values of each IMF with EEMD.

	Mode	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10
EEMD	NR	0.54	0.62	0.46	0.45	0.45	0.27	0.12	0.022	0.0047	0.0089
	IRF	0.91	0.38	0.178	0.101	0.052	0.0201	0.00149	0.00086	0.00025	$2.46\,e{-}05$
	BF	0.84	0.406	0.328	0.266	0.085	0.021	0.004	0.003	0.0013	0.0013
	ORF	0.86	0.308	0.366	0.189	0.125	0.053	0.0041	0.00268	0.0013	0.0001

Table 7: Total average correlation values of each BL-IMF with VMD.

Mode	Mode	BL-IMF1	BL-IMF2	BL-IMF3	BL-IMF4
VMD ($k = 4$)	NR	0.6152	0.4463	0.4940	0.5384
	IRF	0.2275	0.3718	0.5773	0.7702
	BF	0.3849	0.4141	0.4943	0.7376
	ORF	0.4201	0.5187	0.7421	0.2708

classification accuracy. Part of the experiment classification results is shown in Figure 14 and Table 8.

As given in Table 8, the highest classification precision is up to 100% when $k = 4$ in VMD-RCMMSE model. On the contrary, the lowest accuracy is 63.75% in EEMD-RCMMSE model. The overall classification accuracy in VMD-RCMMSE model is higher than EEMD-RCMMSE model. The overall classification accuracy in SVM model is higher than RF/PNN models. As mentioned above, the experimental results show that VMD-RCMMSE combination model can recognize different working conditions of the roller bearings effectively.

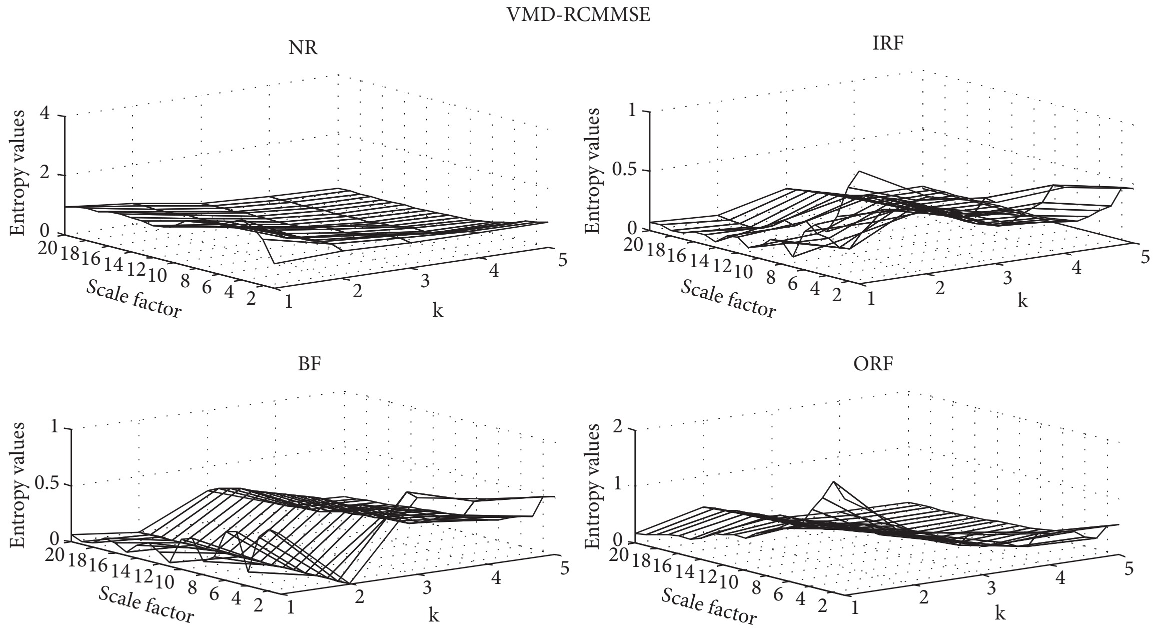
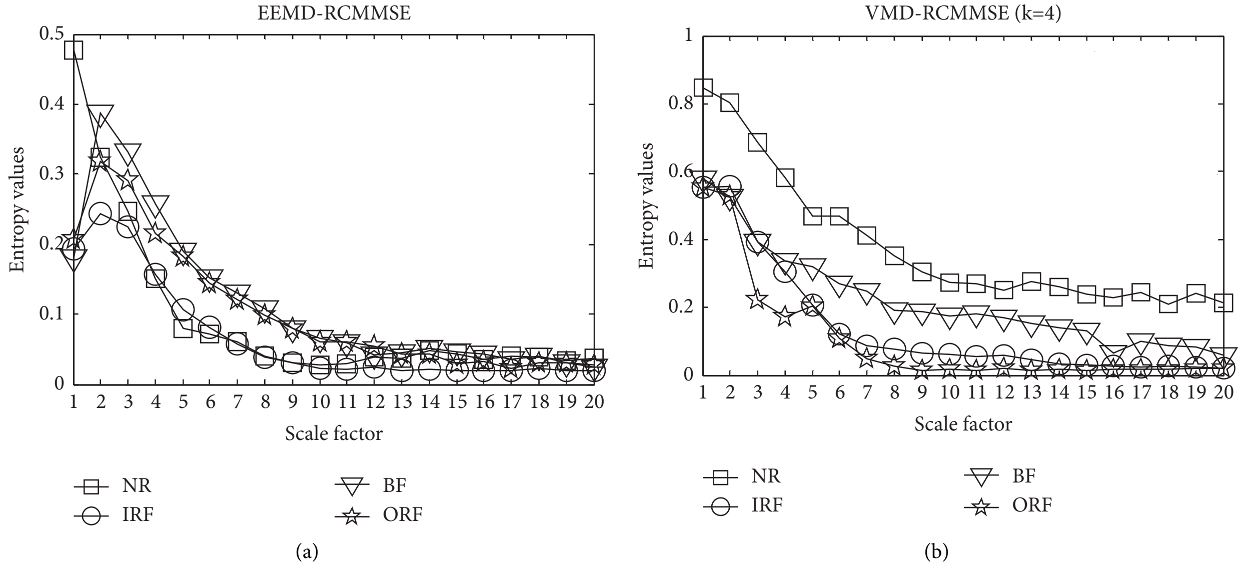


Figure 12: RCMMSE values under different k values and scale factors.



(a)

(b)

Figure 13: RCMMSE values with EEMD and VMD ($k = 4$) modes.

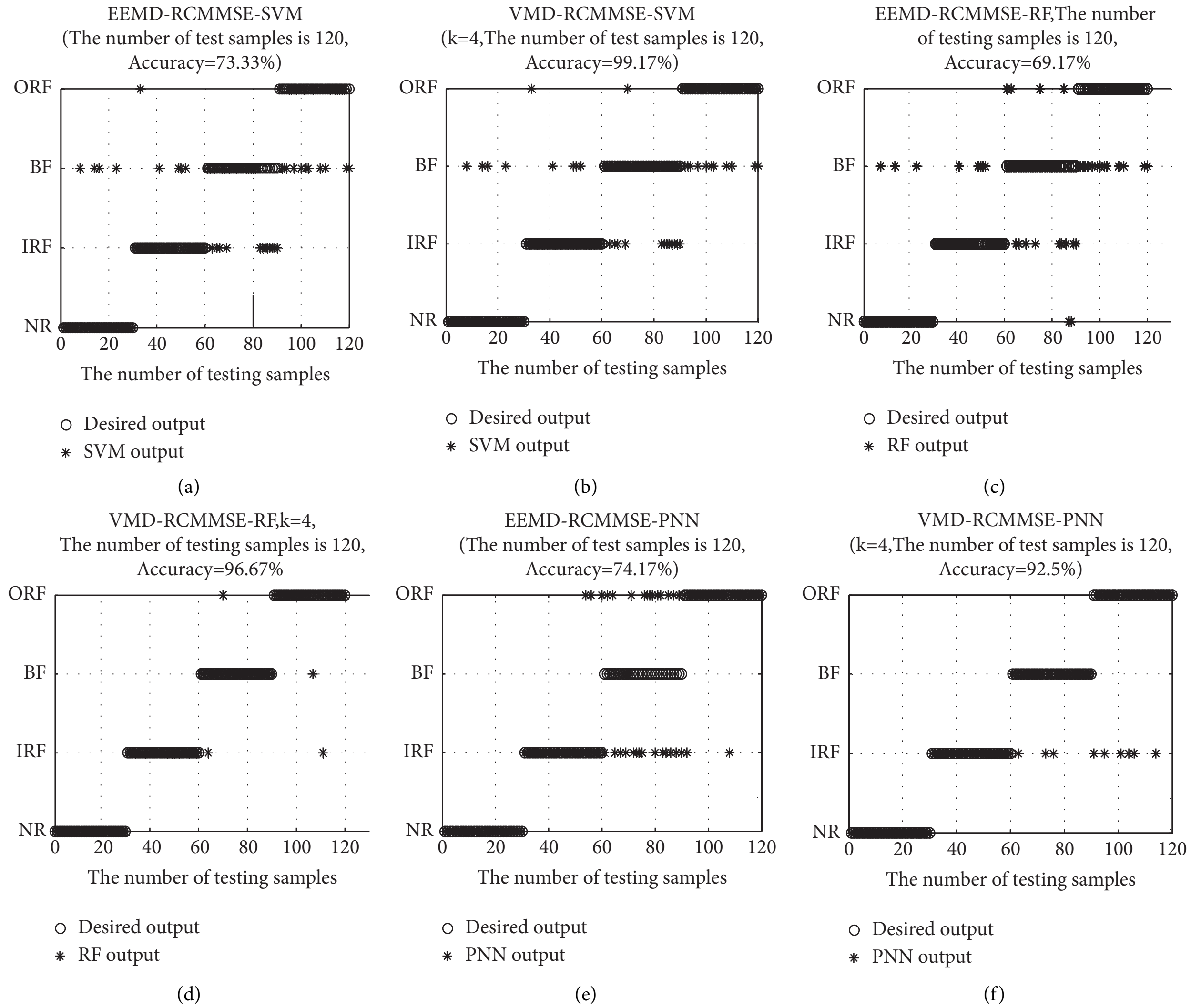


Figure 14: Waveforms in the time domain.

Table 8: Mean and standard deviation of the MSE/MFE/RCMSE and RCMMSE of 1/f noise at a scale factor of 20.

Mode	Precision (%)		
	Total number of testing samples		
	160	120	80
EEMD-RCMMSE-SVM	69.375	73.33	86.25
EEMD-RCMMSE-RF	63.75	69.17	80
EEMD-RCMMSE-PNN	71.25	74.16	71.25
VMD-RCMMSE-SVM ($k = 3$)	85	88.3	91.25
VMD-RCMMSE-RF ($k = 3$)	81.25	88.3	91.25
VMD-RCMMSE-PNN ($k = 3$)	83.125	85	86.25
VMD-RCMMSE-SVM ($k = 4$)	98.125	99.1667	100
VMD-RCMMSE-RF ($k = 4$)	96.25	96.67	97.5
VMD-RCMMSE-PNN ($k = 4$)	93.75	90.83	91.25
VMD-RCMMSE-SVM ($k = 5$)	98.75	97.5	98.75
VMD-RCMMSE-RF ($k = 5$)	97.5	95	96.25
VMD-RCMMSE-PNN ($k = 5$)	86.875	90	91.25

6. Conclusion

The VMD analysis model and RCMMSE are proposed in this study. The VMD model is utilized to analyse the complex vibration signals. Therefore, BL-IMF can be obtained from the decomposition of the roller bearing vibration signals, the RCMMSE can deal with multichannel data, and the EEMD/VMD-RCMMMSE values are regarded as eigenvectors. Because the VMD can achieve adaptive subdivision of each component in the frequency domain of signals, the roller bearing vibration signals are distinguished. Finally, the VMD-RCMMSE-SVM/RF/PNN combination models can distinguish roller bearing fault types effectively.

With the rapid development of Internet of things (IoT) [1, 2] and Industry 4.0 [3], there are increasingly massive real-time data from various types of mechanical equipment [4]. The availability of these data that contain abundant information about machine health has attracted more and more enterprises' attention. It has been proved that large volume, high velocity, and diversity mechanical big data are the major properties of mechanical big data [5, 6]. Effective feature extraction from these data and accurate machinery health state evaluation with ever-accelerated updating of schemes have become hot research issues in the prognostic and health management systems in the era of industrial IoT [7].

7. Future Ideas

Most related roller bearing signal fault diagnoses depend on manual aids and expert knowledge. To diagnose more smartly, convolutional neural network (\textsc{CNN}), as well as long short-term memory (LSTM), will be considered for an end-to-end scenario. For ensuring high recognition accuracy, the fault classifier based on CNN may be used to extract the effective signal from the severely distorted signal; the LSTM could cope with time-varying bearing failure. Additionally, the aforementioned SVM fault classifier performs well when the data samples are insufficient. In the future, we may consider the generative adversarial networks (GANs) and transfer learning for lack of data. By simulating the distribution of fault samples, GAN could obtain more data. Besides, the transfer learning-related diagnosis model can reuse the previous information in the new task. Consequently, small-sized samples could achieve accurate fault identification.

Data Availability

The prior studies and data are cited at relevant places within the text as references [20].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Zhao, J. Li, A. Al-Dubai, A. Y. Zomaya, G. Min, and A. Hawbani, "Routing schemes in software-defined vehicular networks: design, open issues and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 4, pp. 217–226, 2021.

[2] L. Zhao, W. Zhao, A. Hawbani et al., "Novel online sequential learning-based adaptive routing for edge software-defined vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2991–3004, 2021.

[3] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in Internet of vehicles based on short-term traffic prediction," *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 1–18, 2021.

[4] J. Lu, H. Liu, Z. Zhang, J. Wang, S. K. Goudos, and S. Wan, "Towards fairness-aware time-sensitive asynchronous federated learning for critical energy infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3462–3472, 2021.

[5] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, "A context-aware user-item representation learning for item recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–29, 2019.

[6] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: an efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1616–1629, 2021.

[7] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1984–1992, 2020.

[8] M. P. Paidoussis and E. B. Deksnis, "Articulated models of cantilevers conveying fluid: the study of a paradox," *Journal of Mechanical Engineering Science*, vol. 12, no. 4, pp. 288–300, 1970.

[9] J. Liu, L. Dai, and L. J. Zhao, "Modeling and simulation of flexible multi-body dynamics of concrete pump truck arm," *Chinese Journal of Mechanical Engineering*, vol. 43, no. 11, pp. 131–134, 2007.

[10] G. Cazzulani, C. Ghielmetti, H. Giberti, F. Resta, and F. Ripamonti, "A test rig and numerical model for investigating truck mounted concrete Pumps," *Automation in Construction*, vol. 20, no. 8, pp. 1133–1142, 2011.

[11] J. Rafiee, M. A. Rafiee, and P. W. Tse, "Application of mother wavelet functions for automatic gear and bearing fault diagnosis," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4568–4579, 2010.

[12] X. Lou and K. A. Loparo, "Bearing fault diagnosis based on wavelet transform and fuzzy inference," *Mechanical Systems and Signal Processing*, vol. 18, no. 5, pp. 1077–1095, 2004.

[13] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[14] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 1, no. 1, pp. 1–41, 2009.

[15] X. Zhang and J. Zhou, "Multi-fault diagnosis for rolling element bearings based on ensemble empirical mode decomposition and optimized support vector machines," *Mechanical Systems and Signal Processing*, vol. 41, no. 1-2, pp. 127–140, 2013.

[16] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[17] N. Huang, H. Chen, G. Cai, L. Fang, and Y. Wang, "Mechanical fault diagnosis of high voltage circuit breakers based on variational mode decomposition and multi-layer classifier," *Sensors (Basel, Switzerland)*, vol. 16, no. 11, pp. 1–19, 2016.

[18] X. An, L. Pan, and P. Luo, "Bearing fault diagnosis of a wind turbine based on variational mode decomposition and permutation entropy," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 231, no. 2, pp. 200–206, 2017.

[19] K. Zhu, X. Song, and D. X. Xue, "Fault diagnosis of rolling bearings based on IMF envelope sample entropy and support vector machine," *Journal of Information and Computational Science*, vol. 10, no. 16, pp. 5189–5198, 2013.

[20] F. Xu, Y. J. Fang, and R. Zhang, "A fault diagnosis method combined with ensemble empirical mode decomposition, base-scale entropy and clustering by fast search algorithm for roller bearings," *Journal of Vibro engineering*, vol. 18, no. 7, pp. 04472–04490, 2016.

[21] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Physical Review Letters*, vol. 89, no. 6, pp. 068102–068118, 2002.

[22] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of biological signals," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 71, no. 5, pp. 021906–021918, 2005.

[23] J. D. Zheng, J. S. Cheng, and Y. Yang, "A rolling bearing fault diagnosis approach based on multiscale entropy," *Journal of Hunan University (Natural Sciences)*, vol. 39, no. 5, pp. 38–41, 2012.

[24] L. Zhang, G. Xiong, H. Liu, H. Zou, and W. Guo, "Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy inference," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6077–6085, 2010.

[25] S.-D. Wu, C.-W. Wu, S.-G. Lin, K.-Y. Lee, and C.-K. Peng, "Analysis of complex time series using refined composite multiscale entropy," *Physics Letters A*, vol. 378, no. 20, pp. 1369–1374, 2014.

[26] C. L. Liu, Y. G. Wu, and C. G. Zhen, "Rolling bearing fault diagnosis based on variational mode decomposition and fuzzy C means clustering," *Proceedings of the CSEE*, vol. 35, no. 13, pp. 3358–3365, 2015.

[27] M. U. Ahmed and D. P. Mandic, "Multivariate multiscale entropy analysis," *IEEE Signal Processing Letters*, vol. 19, no. 2, pp. 91–94, 2012.

[28] B. Gu, V. S. Sheng, and K. Y. Tay, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403–1416, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3229294, 12 pages
https://doi.org/10.1155/2022/3229294

Research Article

Wireless Communication Technologies for IoT in 5G: Vision, Applications, and Challenges

Quy Vu Khanh,[1] Nam Vi Hoai,[1] Linh Dao Manh,[1] Anh Ngoc Le,[2] and Gwanggil Jeon[3]

[1] *Hung Yen University of Technology and Education, Hung Yen, Vietnam*
[2] *Swinburne Vietnam, FPT University, Hanoi, Vietnam*
[3] *Incheon National University, Incheon, Republic of Korea*

Correspondence should be addressed to Anh Ngoc Le; anhngoc@epu.edu.vn and Gwanggil Jeon; gjeon@inu.ac.kr

Received 19 October 2021; Revised 4 November 2021; Accepted 4 January 2022; Published 7 February 2022

Academic Editor: Mohammad R Khosravi

Communication technologies are developing very rapidly and achieving many breakthrough results. The advent of 5th generation mobile communication networks, the so-called 5G, has become one of the most exciting and challenging topics in the wireless study area. The power of 5G enables it to connect to hundreds of billions of devices with extreme-high throughput and extreme-low latency. The 5G realizing a true digital society where everything can be connected via the Internet, well known as the Internet of Things (IoT). IoT is a technology of technologies where humans, devices, software, solutions, and platforms can connect based on the Internet. The formation of IoT technology leads to the birth of a series of applications and solutions serving humanity, such as smart cities, smart agriculture, smart retail, intelligent transportation systems, and IoT ecosystems. Although IoT is considered a revolution in the evolution of the Internet, it still faces a series of challenges such as saving energy, security, performance, and QoS support. In this study, we provide a vision of the Internet of Things that will be the main force driving the comprehensive digital revolution in the future. The communication technologies in the IoT system are discussed comprehensively and in detail. Furthermore, we also indicated indepth challenges of existing common communication technologies in IoT systems and future research directions of IoT. We hope the results of this work can provide a vital guide for future studies on communication technologies for IoT in 5G.

1. Introduction

The development history of mobile communication systems demonstrated that aim to meet the requirements of humanity, the data rate of mobile communication is constantly being improved and achieved breakthrough results. Mobile generations have evolved through 5 periods, starting from 1G to the current 5G [1]. Network generations from 1G to 3G have shown the continuous evolution of services and speeds. The 4G was proposed in the early 2000s. 4G was the first network generation entirely based on the IP packet switching method [2]. After about ten years of implementation, the former advantages of 4G have converted into disadvantages. Nowadays, 4G has access speed has become too low with high latency [3]. Humanity needs a solution to connect with data rates up to Gbps. The advent of the next-generation network called 5G in the early 2020s marks a comprehensive digital society. In particular, in 5G, a new concept is considered the Internet of Things (IoT) [4, 5]. IoT is an integrated system of advanced technologies and solutions that allows devices, people, platforms, software, and solutions to be connected through the Internet [6, 7].

According to Cisco, more than 500 billion devices will be connected to the Internet by 2030. These devices will be endogenously equipped with IoT modules that allow device-to-device (D2D) communications to each other, forming IoT networks [8]. IoT applications will be deployed in almost all humanity domains, including smart cities [9, 10], smart transportation [11, 12], smart agriculture [13, 14], and smart homes [15]. In [16], we presented a detailed survey of IoT applications for humanity. We illustrate several typical IoT applications as in Figure 1.

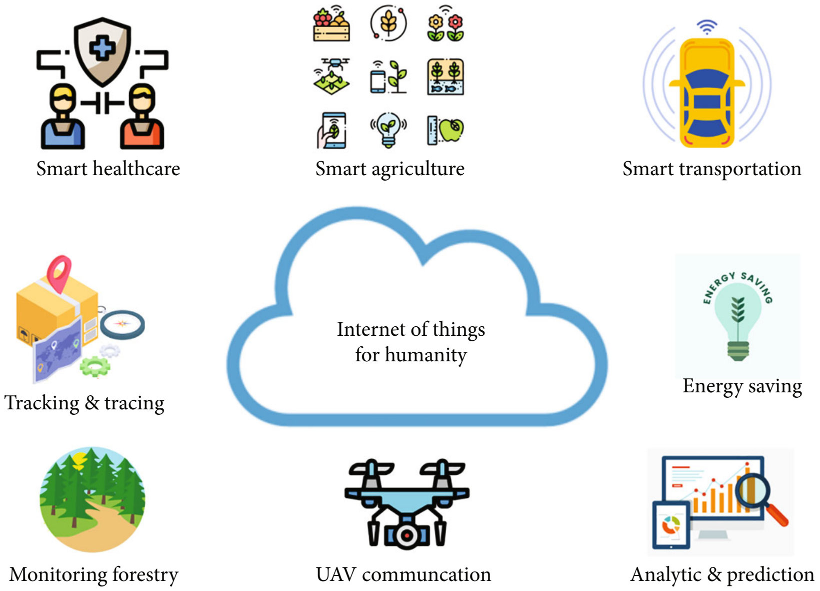


FIGURE 1: An illustration of IoT applications for humanity.

However, the survey results also showed that IoT networks in 5G have a series of challenges such as performance improvement, support QoS, saving energy, privacy, and security [17, 18]. Communication solutions including architecture, routing algorithm, protocol, and spectrum have been proposed to solve these problems. In this study, we conduct a comprehensive survey of communication technologies for IoT in 5G. The main contributions of this survey are listed as follows:

(i) The vision of the Internet of Things in 5G: architecture and research timeline

(ii) A comprehensive survey of the recent communication technologies for IoT in 5G

(iii) The breakout technologies and solutions for IoT in 5G

(iv) Challenges and attractive research topics in the future of communication for IoT

(v) The vision of the Internet of Things in 5G

The evolutionary history of network generations has proven that each generation is born to correct the weakness of previous generations and do some things that the previous generation could not do [19]. In the early 2020s, the Internet of Things concept was born simultaneously with the emergence of 5G [20]. Therefore, to define the vision of IoT in 5G, we need to clarify the advent context of IoT in 5G.

For the convenience of following the article, we have compiled the acronyms in Table 1.

1.1. Forming of IoT in 5G. The development history of mobile communication systems began in the early 1980s. During its development, mobile radio communication systems always tend to integrate all systems. End-user devices are smarter more and more, lighter, save energy, support all types of data such as voice, video, and real-time multimedia applications. The data rate and bandwidth increase with costs decrease. The 1G–3G network generations are standardized and deployed widely worldwide, in references [1, 21]; so, we will not consider these issues to focus on presents the 4G network generation.

The 4th mobile network generation (4G) is formed after 3G and before 5G. Besides the provided services of 3G, it also provides added services such as broadband Internet access, IP phone (VoIP), video conferencing, online games, high-definition Internet TV, 3D TV, and cloud computing. The two technologies were standardized for 4G as Wimax and LTE [21]. One difference with previous generations, 4G unsupports the traditional circuit switching mechanism but relies entirely on IP protocol with the packet switching mechanism. Aim to speed up data transmission, spectrum modulation technologies of previous generations are replaced by OFDMA technology, combined with MIMO multipoint transceiver mechanism and smart antenna. [22]. As a result, the bit rate in 4G is significantly higher than in 3G.

With many advantages mentioned above, 4G has become a pioneering technology and commercialized in many countries. In Vietnam, 4G was deployed in 2016 [23]. However, after many years of deployment, 4G has revealed the limitations of this network generation. According to Cisco, over 500 billion devices will be connected to the Internet in the future. This is beyond the provided capacity of 4G [24]. Moreover, the delay of 4G is too large for the real-time applications, approximately 10 ms, and the data rate of 4G is relatively low, approx. 3 (Mbps). With the number of devices increasing hundreds of times today, 4G will consume a huge amount of energy.

The limitations of 4G were indicated that the advent of 5G is an inevitable trend. Humans need a new network

Table 1: Acronyms used in the survey and definations.

Acronym	Definition
3GPP	3rd generation partnership project
5G	5th generation mobile networks
AAC	Adaptive admission control
AI	Artificial intelligence
ANN	Artificial neural network
AR	Augmented reality
D2D	Device-to-device
D2D	Device to device
eNB	Evolved node B
GPRS	General Packet Radio Service
GSMA	Global System for Mobile Communications
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet protocol
LoRa	Long range
LoRaWAN	Long Range Wide-Area Network
LPWANs	Low-Power Wide-Area Technologies
LTE	Long-term evolution
MIMO	Multiple in, multiple out
NB-IoT	Narrowband IoT
NFC	Near field communication
OFDMA	Orthogonal frequency-division multiple access
QoS	Quality of service
RFID	Radio frequency identification
SC-FDMA	Single-carrier FDMA
SDN	Software-defined networking
UAV	Unmanned aerial vehicle
VoIP	Voice over internet protocol

generation that the data rate increases hundreds of times faster, but energy consumption reduces many times compared to 4G. Some countries such as China, Korea, the United Kingdom, and the United States are currently pioneering in the studies and deployment of 5G. Although still not yet official standardized, GSMA and some organizations and suppliers such as Ericsson and Huawei have proposed the standard of 5G network generation [25] as follows:

Aim to achieve these goals, in Table 2, many breakout technologies and solutions need to be implemented synchronously. However, like previous generations, the improvement of the radio access layer has always been a significant challenge to meet the goals of 5G. In this study, we approach 5G from an Internet of Things perspective. The concept of IoT was first mentioned in 5G. IoT is an advanced technology that allows things, machines, devices, solutions, and people to connect through the Internet. IoT is expected to become popular in all areas serving people, such as smart agriculture, smart transportation, smart cities, health, rescue and disaster recovery, retail, management house, and green energy. A very diverse survey of IoT applications is presented in [26].

Table 2: Main characteristics of 5G network generation.

Characteristics	Goal
Mobile access speed	1 Gbps
Fixed access speed	1-10 Gbps
Data transmission delay	1 ms
Reliability	99.999%
Energy consumption	Reduce many times compared to 4G

1.2. The Architecture of IoT in 5G. IoT in the 5G framework consists of main four-layer architecture, as shown in Figure 2, and is related to data collection, processing, analysis, and sharing of information between equipment and communication networks.

(i) Thing layer: This layer includes physical systems such as actuators, devices, sensors, and communicates with the network layer

(ii) Network layer: The network layer consists of two sublayers: (1) low power wide area technologies (LPWANs) such as SigFox, LoRa, ZigBee, NB-IoT, and (2) backhaul-based connections of 5G. In this study, in order to focus on detailing communication solutions in IoT, communication technologies in the backhaul layer are not within the scope of this research

(iii) Middleware layer: this layer is considered the heart of the network. The IoT framework focuses on advanced technologies and solutions as fog computing, edge computing, cloud computing, AI vision, and big data analytics are deployed

(iv) Application layer: this layer presents IoT applications that are deployed in a series of domains as management factories and buildings, agriculture, traffic system, and IoT ecosystems. This layer integrates all solutions, technologies, and applications to interact with humans through the Internet connection

A specific illustration of this architecture is presented in Figure 3. The sensor devices of IoT applications interact with the IoT gateway based on low-power communication networks such as SigFox, LoRa, or NB-IoT. These IoT gateways collect information from IoT devices and then transmit it to the Cloud through the 5G backhaul communications. In the middleware layer, the collected data is processed and stored, combining autonomous decision-making systems or human controls to make under layer tasks.

1.3. Research Timeline IoT in 5G. Nowadays, study activities on different aspects of the Internet of Things in 5G are exciting in both academic research and industry. Some of the top mobile telecommunication corporations and excellent research labs perform studies and experiments to provide applications and solutions of IoT in 5G.

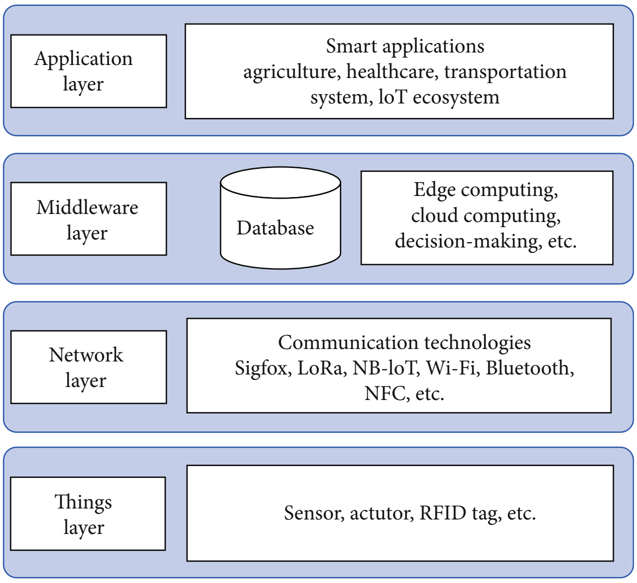

Application layer
Smart applications
agriculture, healthcare, transportation system, loT ecosystem
Middleware layer
Database
Edge computing, cloud computing, decision-making, etc.
Network layer
Communication technologies
Sigfox, LoRa, NB-loT, Wi-Fi, Bluetooth, NFC, etc.
Things layer
Sensor, actutor, RFID tag, etc.

FIGURE 2: An illustration of the IoT in 5G architecture.

1.3.1. Intel. This corporation has pioneered in the IoT field. The company predicts that IoT devices will generate over 55% of global data by 2025. In order to accelerate the application of IoT in various areas serving humanity, Intel is developing an IoT ecosystem at all layers of the IoT architecture with the key technologies and solutions [27] as follows:

(i) In thing layer, Intel providers unique performance scalability with four processor families for IoT applications. Besides, processors of Intel run a variety of operating systems such as Linux, Microsoft, and Google

(ii) In network layer, Intel supports many networking interfaces and protocols to provide the necessary connectivity. Besides, Intel also provides Gateway solutions for the IoT

(iii) In middleware layer, Intel server technology is extensively used in the network and cloud infrastructure. Moreover, Intel is focusing on three IoT computing projects, including edge computing, cloud computing, and AI and computer vision

(iv) In application layer, Intel provides foundations to support the IoT application in various other domains, such as Figure 4

1.3.2. Samsung. According to Samsung, the total number of IoT devices is expected to increase to 21.5 billion by 2025. The number of devices also increases to 34.2 billion if it includes smartphones, laptops, and tablets. Furthermore, Samsung also forecasts that the global IoT market will archive around $1.600 billion. Relying on the expectation that all devices will be connected to the Internet, Samsung has built IoT ecosystems in 5G to realize aspirations such as smart homes, smart cities, smart factories, healthcare, smart agriculture, and logistics [28]. Some of the recent developments in the field of IoT are as follows:

(i) In application layer, Samsung is providing IoT solutions that allow users to control home appliances. The Samsung electronic devices such as TVs, washing machines, and refrigerators can be controlled by remote based on a Samsung smartphone

(ii) In middleware layer, Samsung is implementing research projects related to optimal computing solutions, specifically edge computing, cud computing, and AI vision

(iii) In thing layer, products and devices designed for Samsung IoT platforms, including phones, tablets and wearables, digital signage, and automation solutions. In particular, Samsung designed the unique IoT modules, called Samsung ARTIK modules, which can be customized based on the size, ability, and capabilities of the Samsung products. Moreover, the Samsung ARTIK Smart IoT platform combines open-source modules and cloud services with an ecosystem of tools and partners that is motivation to drive the development of the IoT in 5G. Figure 5 is an illustration of the Samsung Artik 530 development kit

1.3.3. Ericsson. According to Ericsson, the expected IoT numbers of connections would increase over 3.5 times from about 1.7 billion in 2020 to approx 6 billion by 2026. Erricson also forecasts there will have over 24 billion interconnected IoT devices Internet by 2050. Consequently, almost everything is around us as home appliances, vehicles, traffic lights, personal devices, learning devices, and health monitoring would be connected to the Internet. This will be a very exciting area both in academic and industrial research in the coming years. With the ambition to connect anything, anywhere, Ericsson is driving the growth of the IoT through its major contributions in the domain of real-time network performance and cloud computing solutions [29]. Some researches dedicated by Ericsson for IoT in 5G are as follows:

(i) In application layer, besides developing IoT solutions and applications for a wide range of fields such as healthcare and smart agriculture, Ericsson developed an IoT Accelerator Developer Portal to support the development of IoT solutions for the community of application developers worldwide, as presented in Figure 6

(ii) In network layer, Ericsson has focused on researching spectrum sharing solutions, exploiting mmWave, THz frequency bands, and intelligent communication solutions between devices

(iii) In middleware layer, Ericsson promotes research into architectures and solutions of cloud computing and edge computing

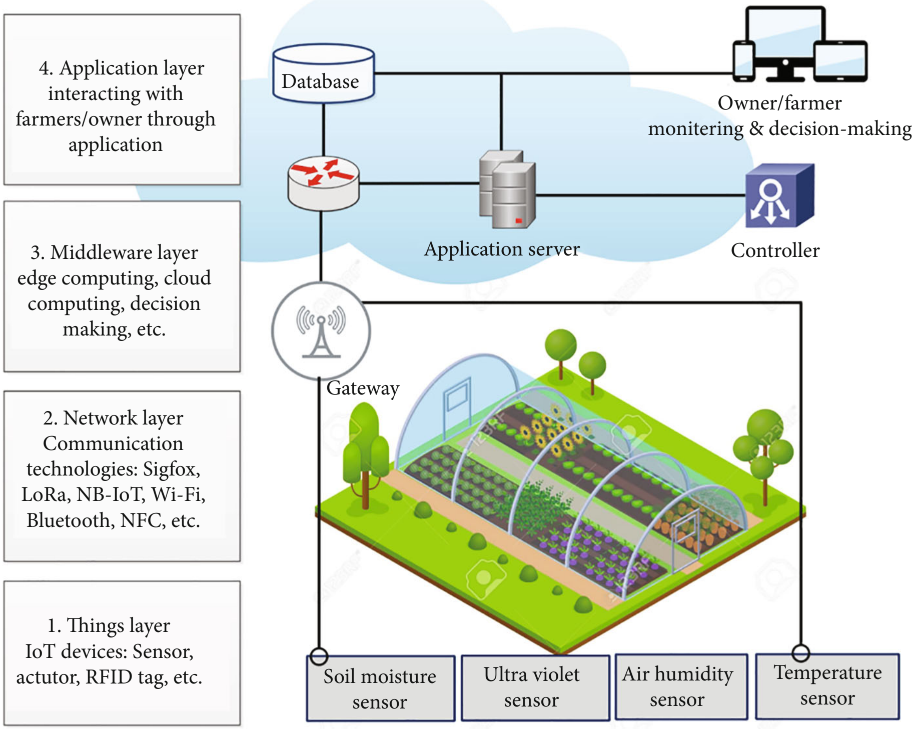

4. Application layer interacting with farmers/owner through application
3. Middleware layer edge computing, cloud computing, decision making, etc.
2. Network layer Communication technologies: Sigfox, LoRa, NB-IoT, Wi-Fi, Bluetooth, NFC, etc.
1. Things layer IoT devices: Sensor, actutor, RFID tag, etc.
Database
Owner/farmer monitering & decision-making
Application server
Controller
Gateway
Soil moisture sensor
Ultra violet sensor
Air humidity sensor
Temperature sensor

Figure 3: An illustration of practise IoT structure for smart agriculture area.

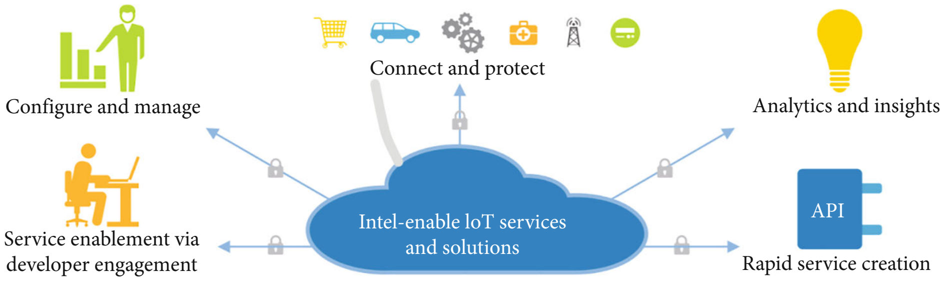


Figure 4: The foundation for connected IoT.

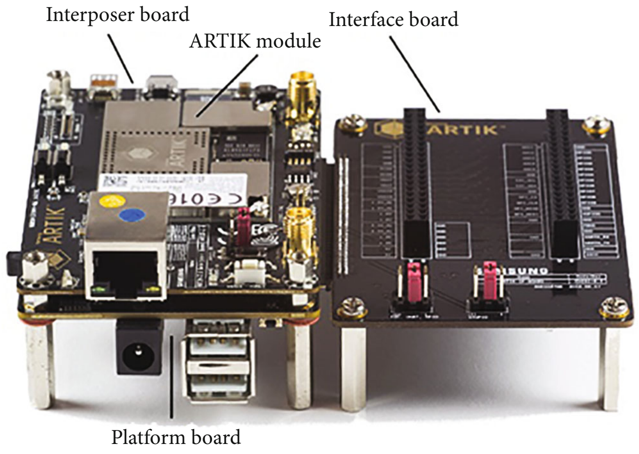


Figure 5: An illustration of the Samsung Artik 530 development kit.

1.3.4. Huawei. This corporation is a pioneering provider of communication solutions for IoT in 5G with a very diverse IoT ecosystem. Huawei has developed a Huawei IoT Connection Management Platform that aims to provide a full connection between people and things and fast integration for the vertical industry applications.

According to Huawei, promoting the development of IoT is based on five factors: (1) flexible deployment, (2) multiple connections, (3) intelligent management, (4) data security, and (5) open ecosystems. Reply to these factors, Huawei is driving the development of IoT through a series of significant contributions in all layers of IoT architecture, from the things layer to the application layer [30], as presented in Figure 7. Along with the achieved breakthrough study results by top telecommunication corporations, a series of research labs around the world are also driving the research process to find promising solutions for IoT in 5G aim enhance data rate, exploit spectrum more efficiently, extend communication distances, optimize energy consumption,

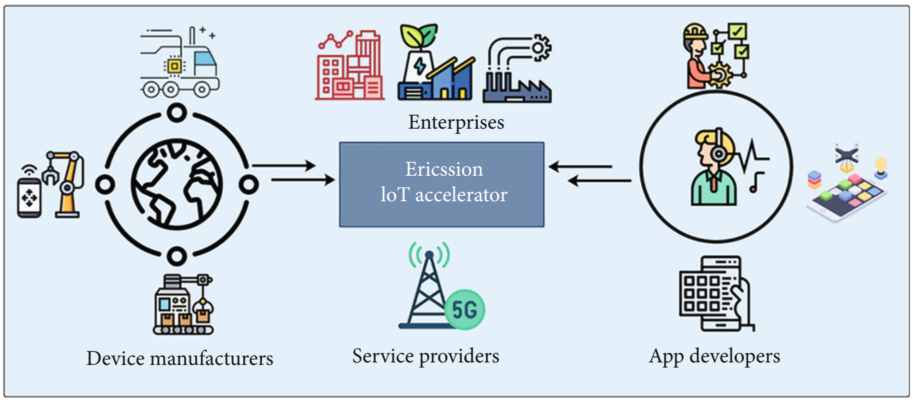


Figure 6: An illustration of IoT Accelerator Developer Portal.

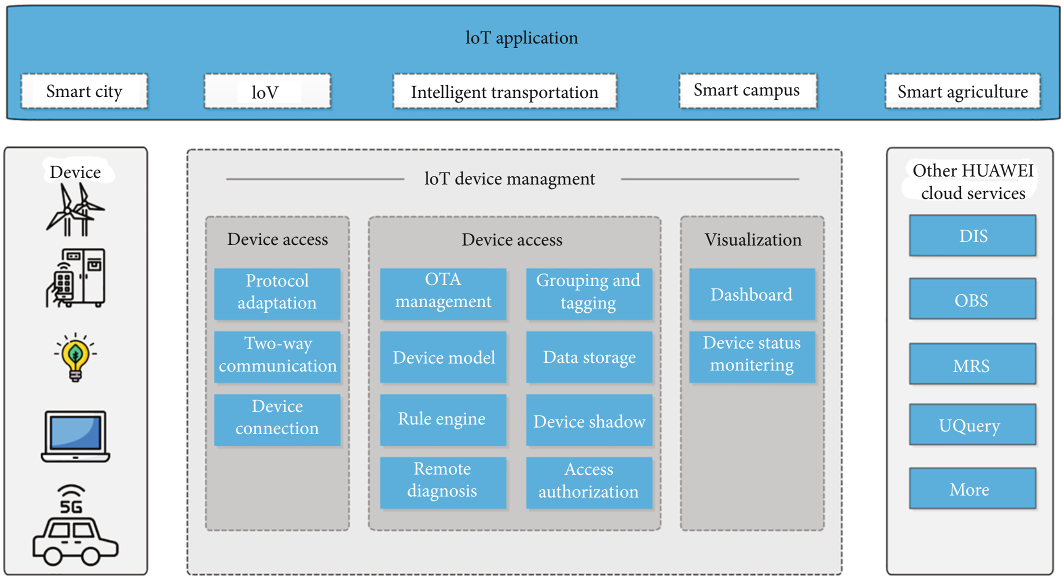


Figure 7: An illustration of solutions and products' architecture in the IoT domain of Huawei.

and extend scale networks up to hundreds of billions of Things. IoT in 5G could be the most revolutionary technology in the communication and information technology area. It could be applied in a series of different domains from popular applications in life such as payment utilities, smart retail, manage home appliances to expert apps such as self-driving vehicles, monitoring traffic status, collision warning between vehicles and monitoring, and controlling green energy systems, smart cities management. In the agriculture area, IoT can also be applied in applications such as forestry management, farm management, monitoring forest fire, tracing, and tracking products. In the industrial area, actuators and robots with the support of AI technology can perform tasks day and night replace humans with extremely high productivity and accuracy. It realizes the dream of smart and green factories.

2. Survey of Recent Communication Solutions

Advances in the semiconductor, electronics, and automation industries are driving the development of communication solutions for IoT in 5G. These solutions are smarter, more reliable, robust, high data rate, and energy saving. As a result, various low-power communication technologies have been proposed for IoT in 5G, such as SigFox and LoRa. Survey results have demonstrated that low power technologies are suitable for IoT 5G networks due to their unique characteristics such as wide coverage, low power, high energy efficiency, and suitable data rate. In this section, we present the recently proposed communication solutions for IoT in 5G. We divide these proposals into four categories based on technology. The detailed survey results are presented in the following subsections and are summarized in Table 3.

2.1. SigFox. SigFox technology was introduced in the 2010s to connect low-power devices such as electricity meters and smartwatches, which need to be continuously operated on and have extremely low data rates. SigFox uses the industrial, scientific, and medical radio band, which uses 868 MHz in Europe and 902 MHz in the US with a channel bandwidth of 100 MHz. SigFox uses a wide-reaching signal that passes freely through solid objects, called ultra narrowband and

Table 3: Some typical LPWAN communication technologies for IoT in 5G.

Type	Transmission distance	Type of network	Frequency	Data rate
802.11a/b/g/n/ac	100 m	WLAN	2.4-5 GHz	2-700 Mbps
802.11ah	1000 m	WLAN	Sub-GHz	78 Mbps
802.11p	1 km	WLAN	5.9 GHz	3-27 Mbps
802.11af	1 km	WLAN	54-790	25-550 Mbps
SigFox	Rural: 50 km Urban:10 km	LPWA	Zwave	100-600 bps
LoRaWAN	20 km	LPWA	Sub-GHz	0.3-100 Kbps
NB-IoT	35 km	LPWA	Zwave	250 Kbps
ZigBee	1 km	LPWA	2.4 GHz	250 Kbps

requires low energy; so, it also is an LPWAN technology. It uses a one-hop star topology. SigFox is used to cover large areas and to reach underground objects. SigFox cells have a coverage range of about 30-50 km in rural areas and reduced to under 10 km in crowded areas. Overall, SigFox enables to provide a wide area network with low-power consumption. Nowadays, the SigFox IoT system has covered around 72 countries with over 1.3 billion of the world population. Several recent IoT applications are based on SigFox communication, as follows:

In [31], Joris et al. (2019) designed an autonomous SigFox sensor node capable collected data from an area of sensors, then transmitting data to the Cloud for smart agriculture applications. Aim to enhance the ability of this system, the sensor nodes are designed to use solar energy. Experimental data show that the system can transmit data every 5 minutes in cloudy conditions. In [32], Lavric et al. (2019) analyzed the responsiveness of SigFox under different scale and density conditions of sensors for IoT networks. The figures indicated that the maximum number of sensors that can transmit data at the same time is approximately 100. The results indicated that, as the number of sensors increases above 100, the network performance could be decreased. Moreover, this study also proposes solutions to improve performance, large-scale, and high-density of sensors in SigFox IoT networks.

In [33], Mikhaylov et al. (2019) evaluated the performance of SigFox communication technology in the real world. Specifically, they deployed a SigFox-based communications network at 311 different locations in Brno city, Czech Republic. Then, they conduct tests to evaluate the performance and characteristics of the radio channel. The experimental results show that the packet delivery ratio achieved over 94% in the urban environment in the real world.

2.2. LoRa. The LoRa is emerging as one of the most promising low-power wide-area (LPWA) communication technologies. It enables the energy-constraint devices distributed over wide-scale areas to establish connectivity at an affordable cost. The LoRa uses a low-power wide-area network modulation technique and unlicensed frequency bands like 433 MHz, 868 MHz (Europe), 915 MHz (Australia and North America), and 923 MHz (Asia). LoRa enables long range transmissions with low power consumption. The LoRa technology covers the physical layer, while other technologies and protocols such as LoRaWAN (Long Range Wide-Area Network) cover the network layer. Depending upon the spreading factor, it can achieve data rates between 0.3 kbps and 27 kbps. However, how to implement a flexible LoRa network with an effective cost is still an open challenge.

In [34], Zhou et al. (2019) designed and introduced an open LoRa system for IoT networks. Contributions of this work include (1) design and hardware implementation of a LoRa gateway, (2) use LoRa open-source codes on GitHub, and (3) improve server LoRa through the uses of the messages system for the interaction between modules to guarantee scalability and flexibility. The experimental results have shown that the proposed system has improved the performance of the LoRa network compared to the traditional LoRa network.

In [35], Lee et al. (2018) designed and evaluated the performance of a LoRa mesh network to examine the applicability of LoRa networks for urban scenarios. This work installed 19 mesh LoRa devices in range [800 × 600] m on a university campus and installed a gateway that collected data at 1-min intervals. The experimental results showed that the proposed LoRa system has an average packet delivery ratio of 88.49%, whereas the star LoRa topology only achieved 58.7% under the same conditions.

The LoRa is one of the most successful technologies of the LPWAN (Low-Power Wide-Area Network) family. It enables robust long-distance low power communications and is proven to be effective in the Internet of Things (IoT) applications. The LoRa is also promising for Industrial IoT scenarios. However, a limitation of LoRa does not offer support to real-time data flows. To solve this problem, In [36], Leonardi et al. (2019) proposed a new medium access strategy for LoRa, called RT-LoRa, which aim to support real-time LoRa-based IoT applications. The simulation results demonstrated that RT-LoRa could support real-time flows for IoT applications.

2.3. Wi-Fi. Wi-Fi is a known-well family of wireless communication technologies based on the IEEE 802.11 family of standards. It is commonly used for local area networks of devices and Internet access within 100 (m). It operates in the 2.4-5 GHz frequency band. Wi-Fi is suitable for short-range communication; so, it is a feasible communication solution for IoT networks.

In [37], Pokhrel et al. (2020) proposed a queue management solution for the home IoT access points based on Wi-Fi. The focus of this work proposal adaptive admission control mechanism at the Wi-Fi access point aims to reduce the response time of the access point. The experiment results demonstrated that the proposed system is more stable than traditional home IoT systems based on Wi-Fi. In [38], Sheth et al. (2019) proposed a saving energy communication solution based on WIOTAP for IoT systems based on Wi-Fi. The focus of this work uses an intelligent Wi-Fi access point. Then, it presents a downlink packet scheduling mechanism to reduce downlink channel access contention and queuing delay of regular stations in IoT systems. The results demonstrated that the proposed system improved over 38% of energy consumption and over 41% of the delay compared to traditional solutions.

Real-time locating and tracking are the most critical problems of IoT applications. GPS-based positioning applications are well known for outdoor environments. However, it is not feasible for indoor scenarios. In [39], Ruo et al. (2019) proposed an IoT solution to tracking and locating indoor based on Wi-Fi signals for indoor environments. The focus of this work uses a message type that is built-in the 802.11-REVmc2 Wi-Fi standard. Then, through measurements of the round-trip time and signal strength to improve the accuracy and ability of the positioning system. Experiment results demonstrated that the proposed system enhanced the performance and achieved an average positioning accuracy of 1.435 m with an update time of every 0.19 s for indoor scenarios.

2.4. ZigBee. ZigBee is a communication technology that uses the IEEE.802.15.4 standard and operates in the industrial, scientific, and medical radio frequency bands. It is a low-power wide-area communication solution for IoT networks. ZigBee technology in IoT networks has advantages compared to other communication technologies because of its simplicity, flexibility, and low cost. The transmission distance of ZigBee is about 100 m, with a data rate that is about 250 kbps, depending on power output and environmental features. ZigBee is typically used in extreme-low data rate networks, short-range, and long-lasting battery life such as home automation, medical device data collection, and industrial equipment control.

In [40], Pirayesh et al. (2021) proposed a ZigBee receiver based on MIMO against jamming attacks for IoT networks. This work designed a prototype of the ZigBee receiver based on MIMO technology and a learning mechanism to mitigate the unknown interference. The experiment results demonstrated that the proposed system could provide an average of over 26.7 dB jamming mitigation capability compared to the traditional ZigBee receiver.

In [41], Farha et al. (2021) introduced a new security schema based on a timestamp against replay attacks for ZigBee networks. This solution improves energy consumption significantly. Besides, to enhance feasibility, this solution uses powered devices to provide energy for power-constrained devices with the current timestamp. The proposal is designed to be suitable for all ZigBee networks. The experiment results indicated that the proposed solution improves significantly against ability reply attacks in the ZigBee-based IoT networks.

In [42], Ali et al. (2019) designed the smart sensors that combined two communication modules include ZigBee and LoRa, to measure temperature and humidity factors for IoT applications. The collected data from sensors are sent to the central receiver unit by using the ZigBee or LoRa transceiver modules. The choice of the communication module can be controlled remote or based on the Cloud. The practical design and experiment figures indicated the benefits of the low-power, long range communication solutions for IoT applications.

2.5. Narrowband Internet of Things. Narrowband Internet of Things (NB-IoT) is a new LPWAN radio technology developed by 3GPP to support massive connections, wide-area coverage, ultra-low power consumption, and low cost for IoT in 5G. NB-IoT is a promising emerging communication technology for IoT in 5G. NB-IoT focuses specifically on indoor coverage, low cost, long battery life, and high connection density. It uses the bandwidth to narrow-band 200 kHz and OFDM modulation for downlink communication and SC-FDMA for uplink communications.

In [43], Chen et al. (2020) designed a prototype NB-IoT network based on open source for IoT in 5G applications. The open-source NB-IoT results from cooperation between three providers, including EURECOM, B-COM, and NTUST, based on the open-source eNB of LTE technology. This work presents a method to use the existing commercial NB-IoT module to transmit the collected data from sensors to the Internet via the open-source NB-IoT network.

In [44], Chen et al. (2019) evaluated the performance and improved NB-IoT protocol for IoT networks in 5G. The focus of this work includes the following: (1) use the stochastic network to analyze the delay metric in the NB-IoT system and (2) improve NB-IoT protocol through the improvement of the k -means algorithm to cluster NB-IoT devices and perform a scheduling strategy based on the priority. The experiment results indicated that the proposed uplink traffic scheduling schema enhanced performance compared to existing uplink traffic scheduling schemas.

In [45], Kanj et al. (2020) introduced a method to design the physical layer of the NB-IoT device. The focus of this work presents the characteristics and the scheduling of downlink and uplink physical channels at the NB-IoT base station and end-user device to help readers without having to read all the 3GPP specifications.

3. Discussions, Challenges, and Open Issues

In this study, we have highlighted the revolutionary contributions of IoT in 5G in a wide range of fields to serve humanity. Low power communication technologies will play an essential role in supporting and driving IoT applications more public. The survey results have indicated that many applications have been presented in Table 4. The proposals are applied in a variety of domains such as environment, city, home, building, factory, and agriculture.

Communication technologies such as ZigBee, SigFox, LoRa, and NB-IoT have advantages such as low energy

Table 4: Statistics of recently proposed IoT applications based on communication technologies in 5G.

Ref. no	Technology	Application domain	Case study: Key focus
[31]	SigFox	Agriculture	This proposal aims to design an autonomous IoT sensor to collect data in smart agriculture based on the solar energy system.
[32]	SigFox	City, industry, building, home, traffic	This research evaluates the performance of the SigFox communication protocol under different scale and density conditions.
[33]	SigFox	City, environment	This research deployed a network system in the real world at 311 locations of Brno city to measure the real performance of SigFox.
[34]	LoRa	City, environment, healthcare, agriculture	This research designs and deployment a LoRa network for performance improvement, flexible, and reduced cost.
[35]	LoRa	Building	This research real deployed a LoRa mesh system on a university campus to consider the real system performance.
[36]	LoRa	Industry	This research introduced a medium access strategy for LoRA, called RT-LoRa, to reduce service response time for real-time IIoT applications based on LoRa communication protocol.
[37]	Wi-Fi	Home	This research introduced the AAC mechanism at the Wi-Fi access point to reduce service response time for home Wi-Fi IoT applications.
[38]	Wi-Fi	Home	This research introduced the Wi-Fi IoT access point (Wiotap) to address saving energy and reducing delay for home IoT applications.
[39]	Wi-Fi	Home	This research introduced a tracking and location IoT solution based on Wi-Fi to improve accuracy and reduce delay indoor environments.
[40]	ZigBee	Security IoT networks	This research designs a ZigBee receiver against jamming attacks for IoT networks based on MIMO technology.
[41]	ZigBee	Security IoT networks	This research proposes a new security schema based on a timestamp against ability reply attacks in the ZigBee-based IoT networks.
[42]	ZigBee	Environment	This research designs an IoT sensor that combines two communication modules, ZigBee and LoRa, to improve energy consumption and performance.
[43]	NB-IoT	IoT ecosystems	This research designs a prototype NB-IoT network based on open source for IoT in 5G applications.
[44]	NB-IoT	IoT ecosystems	This research introduced a scheduling schema to improve the NB-IoT protocol to enhance performance.
[45]	NB-IoT	IoT ecosystems	This research presents how to design the physical layer of the NB-IoT device according to 3GPP.

consumption, large-coverage, use of unlicensed frequency bands, and suitable for the characteristics of IoT networks and increasingly popular applied in IoT applications. Communication solutions of IoT in 5G aim to provide connectivity for IoT applications. With hundreds of billions of IoT devices connected to the network, these technologies face several significant challenges. In our opinion, the two crucial issues are the security-aware and energy efficiency. Then, we present the challenges of communication technologies for IoT applications in 5G and indicate possible research directions.

3.1. Privacy and Security. The Internet of Things development forms a truly open world, where everything is connected to the Internet. Consequently, objects are easily vulnerable to attacks from the Internet. Therefore, according to Roukounaki et al. (2019) [46], privacy and security are the most critical factors to promote the development of IoT applications to become popular. In IoT applications, attacks can be performed in multiple layers, specifically as

(i) *Security for IoT devices*: the IoT devices with low computing capability and massive numbers are unsuitable for setting up robust security algorithms. Consequently, attacks focus on exploiting the vulnerabilities of IoT devices

(ii) *Security for gateway devices*: the gateway devices play an important role in communication between things layer devices and upper layers. As a result, it is the heart of IoT applications. Denial of service attacks or data spoofing always focuses on the gateway of IoT applications

(iii) *Security for devices at the edge*: recently proposed solutions use edge computing technology to reduce service response times for real-time IoT applications. Consequently, the security of edge computing servers is one of the major challenges

(iv) *Security for cloud servers*: with the huge amount of data is provided by IoT devices, cloud infrastructure will be a possible solution in storing and processing big data. Consequently, the security of cloud servers will be one of the significant challenges

In [47], Zhou et al. (2021) presented a survey comprehensive of security logic bugs in IoT devices, platforms, and systems in all layers. In [48], Lins et al. (2021) presented a complete picture of potential threats as well as solutions aimed to mitigate attacks on IoT gateways. In [49], Wang et al. (2018) presented potential risks at the application layer, including data collection, storage, and data processing in the cloud of cloud-based IoT systems. Attacks into cloud servers to gain control or execute tasks to affect autonomous devices in smart factories and farms. In [50], Hassija et al. (2019) presented a diverse survey of attacks and security threats and proposed several architectural solutions to mitigate attacks on IoT systems.

In our opinion, security is one of the most critical problems of communication solutions in the IoT 5G network. This issue will continue to be a research topic timely and attract both academic and industry researchers in the future.

3.2. Energy Efficiency. Assuming that when IoT applications in 5G become popular, tens of billions of IoT devices will operate and transmit data continuously day and night. As a result, it will consume a huge amount of energy while energy resource is exhausted day by day. This is not feasible. Therefore, energy-efficient communication solutions are a real challenge.

In [51], Popli et al. (2019) presented a comprehensive survey of energy-saving solutions for IoT systems based on NB-IoT technology. The survey concluded that NB-IoT technology would be an essential technology to realize green IoT networks in the future. In [52], Ding et al. (2019) presented an optimal scheduling solution based on the multiobjective fuzzy algorithm to save energy for IoT networks. In [53], Al-Kadhim et al. (2019) presented a reliable and saving energy data transmission solution for cloud-based IoT systems. The figures demonstrate that the proposed solution reduced energy consumption by 57% and improved reliability by 60% compared to the traditional solution.

In our opinion, energy efficiency can be considered based on some of the solutions as follows:

(i) *Communication technology-based*: integration of smart, flexibly, and low-power communication technologies such as NB-IoT and ZigBee. In [17], the authors presented a survey of the energy harvesting communication technology for autonomously power IoT devices. This technology promises green energy in the future

(ii) *Trade-off based*: In reality, performance and energy-saving have an antagonistic relationship. Therefore, a smart, flexible trade-off solution should be considered. In [54], Couso et al. (2018) proposed a trade-off solution for inverters to balance energy saving and performance for IoT-based smart grid applications

(iii) *Cloud-based IoT networks*: cloud will continue to be the backhaul infrastructure for IoT applications due to its robust storage, computing, and processing ability. However, cloud services have a high response time due to the edge computing solutions that are proposed. Consequently, an intelligent offload schema to optimal resource allocation between cloud and edge servers should be considered. In [55], Aljanabi et al. (2021) proposed a hybrid fog-cloud offloading schema to optimal performance and energy for IoT applications

4. Conclusion

In this study, we introduced the vision, architecture, wireless communication technologies, and research timelines of IoT in 5G. Based on the analysis of the core components for IoT in 5G, we conducted a short survey of low power communication technologies for IoT in 5G. The survey results showed that the Internet of Things would be the future of humanities, where all things such as software, systems, and people are connected through the Internet. The advent of IoT in 5G led to the formation of a series of applications serving humanity, such as smart homes, smart cities, smart agriculture, smart factories, green energy, and IoT systems. Besides, we have provided a full picture of promising communication technologies for IoT in 5G such as SigFox, LoRa, Wi-Fi, and LoRaWAN. These solutions are suitable for the operating characteristics of IoT networks such as large coverage areas, high energy efficiency, and low energy consumption level, which support a large number of IoT devices.

Moreover, the survey results also point out some challenges of communication technologies for IoT in 5G, including (1) privacy and security and (2) saving energy. In our opinion, the security and saving energy problems of communication technologies will continue to be exciting research topics in the future and receive attention from both academic research and industry. We hope that this study will play an important role as a guide for future research on communication technologies for IoT applications in 5G.

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] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, “Survey of cellular mobile radio localization methods: from 1G to 5G,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2018.

[2] M. Agiwal, A. Roy, and N. Saxena, “Next generation 5G wireless networks: a comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[3] M. Agiwal, H. Kwon, S. Park, and H. Jin, “A survey on 4G-5G dual connectivity: road to 5G implementation,” *IEEE Access*, vol. 9, pp. 16193–16210, 2021.

[4] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, “A survey on security and privacy of 5G technologies: potential solutions, recent advancements, and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2020.

[5] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[6] S. Sinche, D. Raposo, N. Armando et al., "A survey of IoT management protocols and frameworks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1168–1190, 2020.

[7] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[8] https://www.cisco.com/c/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.html.

[9] J. An, F. le Gall, J. Kim et al., "Toward global IoT-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5753–5765, 2019.

[10] F. Cirillo, D. Gómez, L. Diez, I. Elicegui Maestro, T. B. J. Gilbert, and R. Akhavan, "Smart city IoT services creation through large-scale collaboration," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5267–5275, 2020.

[11] A. J. V. Neto, Z. Zhao, J. J. P. C. Rodrigues, H. B. Camboim, and T. Braun, "Fog-based crime-assistance in smart IoT transportation system," *IEEE Access*, vol. 6, pp. 11101–11111, 2018.

[12] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F. -Y. Wang, "Parallel transportation systems: toward IoT-enabled smart urban traffic control and management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4063–4071, 2020.

[13] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of Things for the future of smart agriculture: a comprehensive survey of emerging technologies," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 718–752, 2021.

[14] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of IoT in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156237–156271, 2019.

[15] H. Uddin, M. Gibson, G. A. Safdar et al., "IoT for 5G/B5G applications in smart homes, smart cities, wearables and connected cars," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–5, Limassol, Cyprus, 2019.

[16] A. Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, "Future trends and current state of smart city concepts: a survey," *IEEE Access*, vol. 8, pp. 86448–86467, 2020.

[17] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting IoTs: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2020.

[18] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[19] M. H. Alsharif and R. Nordin, "Evolution towards fifth generation (5G) wireless networks: current trends and challenges in the deployment of millimetre wave, massive MIMO, and small cells," *Telecommunication Systems*, vol. 64, no. 4, pp. 617–637, 2017.

[20] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020.

[21] https://www.ericsson.com/en/blog/2020/4/impact-of-mobile-network-generations-on-business/.

[22] T. Mumtaz, S. Muhammad, M. I. Aslam, and N. Mohammad, "Dual connectivity-based mobility management and data Split mechanism in 4G/5G cellular networks," *IEEE Access*, vol. 8, pp. 86495–86509, 2020.

[23] S. Won and S. W. Choi, "Three decades of 3GPP target cell search through 3G, 4G, and 5G," *IEEE Access*, vol. 8, pp. 116914–116960, 2020.

[24] H. Beyranvand, M. Lévesque, M. Maier, J. A. Salehi, C. Verikoukis, and D. Tipper, "Toward 5G: FiWi enhanced LTE-A HetNets with reliable low-latency fiber backhaul sharing and Wi-Fi offloading," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 690–707, 2017.

[25] K. Samdanis and T. Taleb, "The road beyond 5G: a vision and insight of the key technologies," *IEEE Network*, vol. 34, no. 2, pp. 135–141, 2020.

[26] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "IoT connectivity technologies and applications: a survey," *IEEE Access*, vol. 8, pp. 67646–67673, 2020.

[27] "Intel," https://www.intel.com/.

[28] "Samsung," https://www.samsung.com/ph/.

[29] "IoT Platform," https://www.ericsson.com/en.

[30] "Huawei," https://www.huaweicloud.com/intl/en-us/.

[31] L. Joris, F. Dupont, P. Laurent, P. Bellier, S. Stoukatch, and J. M. Redoute, "An autonomous Sigfox wireless sensor node for environmental monitoring," *IEEE Sensors Letters*, vol. 3, no. 7, pp. 1–4, 2019.

[32] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.

[33] K. Mikhaylov, M. Stusek, P. Masek et al., "Communication performance of a real-life wide-area low-power network based on Sigfox technology," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.

[34] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, "Design and implementation of open LoRa for IoT," *IEEE Access*, vol. 7, pp. 100649–100657, 2019.

[35] H. Lee and K. Ke, "Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: design and evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177–2187, 2018.

[36] L. Leonardi, F. Battaglia, and L. Lo Bello, "RT-LoRa: a medium access strategy to support real-time flows over LoRa-based networks for industrial IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10812–10823, 2019.

[37] S. R. Pokhrel, H. L. Vu, and A. L. Cricenti, "Adaptive admission control for IoT applications in home Wi-Fi networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2731–2742, 2020.

[38] J. Sheth and B. Dezfouli, "Enhancing the energy-efficiency and timeliness of IoT communication in Wi-Fi networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9085–9097, 2019.

[39] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, "Indoor smartphone localization: a hybrid Wi-Fi RTT-RSS ranging approach," *IEEE Access*, vol. 7, pp. 176767–176781, 2019.

[40] H. Pirayesh, P. Kheirkhah Sangdeh, and H. Zeng, “Securing ZigBee communications against constant jamming attack using neural network,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4957–4968, 2021.

[41] F. Farha, H. Ning, S. Yang, J. Xu, W. Zhang, and K. K. R. Choo, “Timestamp scheme to mitigate replay attacks in secure ZigBee networks,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 342–351, 2022.

[42] S. Z. Ali, S. K. Partal, and H. P. Partal, “ZigBee and LoRa based wireless sensors for smart environment and IoT applications,” in *2019 1st global power, Energy and Communication Conference (GPECOM)*, pp. 19–23, Nevsehir, Turkey, 2019.

[43] C. Chen, R.-G. Cheng, C.-Y. Ho, M. Kanj, B. Mongazon-Cazavet, and N. Nikaein, “Prototyping of open source NB-IoT network,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–5, Taipei, Taiwan, 2020.

[44] X. Chen, Z. Li, Y. Chen, and X. Wang, “Performance analysis and uplink scheduling for QoS-aware NB-IoT networks in mobile computing,” *IEEE Access*, vol. 7, pp. 44404–44415, 2019.

[45] M. Kanj, V. Savaux, and M. Le Guen, “A tutorial on NB-IoT physical layer design,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[46] A. Roukounaki, S. Efremidis, J. Soldatos, J. Neises, T. Walloschke, and N. Kefalakis, “Scalable and configurable end-to-end collection and analysis of IoT security data: towards end-to-end security in IoT systems,” in *2019 Global IoT Summit (GIoTS)*, pp. 1–6, Aarhus, Denmark, 2019.

[47] W. Zhou, C. Cao, D. Huo et al., “Reviewing IoT security via logic bugs in IoT platforms and systems,” *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11621–11639, 2021.

[48] F. A. A. Lins and M. Vieira, “Security requirements and solutions for IoT gateways: a comprehensive study,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8667–8679, 2021.

[49] W. Wang, P. Xu, and L. T. Yang, “Secure data collection, storage and access in cloud-assisted IoT,” *IEEE Cloud Computing*, vol. 5, no. 4, pp. 77–88, 2018.

[50] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, “A survey on IoT security: application areas, security threats, and solution architectures,” *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[51] S. Popli, R. K. Jha, and S. Jain, “A survey on energy efficient narrowband internet of things (NBIoT): architecture, application and challenges,” *IEEE Access*, vol. 7, pp. 16739–16776, 2019.

[52] X. Ding and J. Wu, “Study on energy consumption optimization scheduling for internet of things,” *IEEE Access*, vol. 7, pp. 70574–70583, 2019.

[53] H. M. Al-Kadhim and H. S. Al-Raweshidy, “Eergy efficient and reliable transport of data in cloud-based IoT,” *IEEE Access*, vol. 7, pp. 64641–64650, 2019.

[54] C. Couso, J. Martin-Martinez, M. Porti, and M. Nafría, “Performance and power consumption trade-off in UTBB FDSOI inverters operated at NTV for IoT applications,” *IEEE Journal of the Electron Devices Society*, vol. 6, pp. 55–62, 2018.

[55] S. Aljanabi and A. Chalechale, “Improving IoT services using a hybrid fog-cloud offloading,” *IEEE Access*, vol. 9, pp. 13775–13788, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4058729, 9 pages
https://doi.org/10.1155/2022/4058729

Research Article

Personalized Travel Recommendation Based on the Fusion of TGI and POI Algorithms

Lu Fan[1] and Wenliang Zhang[2]

[1] *School of Henan College of Transportation, Zhengzhou, Henan 450000, China*
[2] *School of Zhengzhou University, Zhengzhou, Henan 450001, China*

Correspondence should be addressed to Wenliang Zhang; zwl@zzu.edu.cn

Received 26 November 2021; Revised 20 December 2021; Accepted 27 December 2021; Published 24 January 2022

Academic Editor: Mohammad R Khosravi

On the way to travel, the public expect to get a tourism experience with low cost, convenient travel, and high comfort. At the same time, they also have different tourism needs such as history and culture, natural landscape, and food shopping. To address the problem that traditional travel route recommendation algorithms have limited accuracy and only analyze text or pictures alone, we propose a personalized travel route recommendation algorithm that integrates text and photo information from travelogues and obtain the historical tourism footprint of tourists by analyzing travel notes. According to the frequency and cooccurrence of scenic spots in the travel notes and the number of photos taken by each scenic spot, the popularity of scenic spots and the interest preferences of various types of tourists are analyzed. Under the given starting and ending points or passing points, the optimal tourism route generation method is designed. Experiments on the real data set of Ctrip Travel website show that the recommendation accuracy of this algorithm is significantly improved compared with the traditional algorithm which only uses travel notes text or photos. Compared with the algorithm that only considers the popularity of interest points or tourists' interest preferences, the accuracy of the route recommended by the algorithm is improved. Compared with the algorithm that only considers the cooccurrence of scenic spots or only considers the influence of photos, this algorithm can obtain a better popularity score of scenic spots. This method integrates the two kinds of information including picture and text, fully considering the interest of users with high practicability.

1. Introduction

When traveling to an unfamiliar city, users usually select points of interest (POI) first and then make a travel itinerary based on their interests and the time [1]. For example, when a person comes to a new place, he must be interested in visiting the most POIs in the shortest time. Therefore, the recommendation of tourist attraction is conducive to promoting the rapid development of tourism.

With the development of information technology, the Internet is becoming an important source for people to plan their travels [2]. In the field of tourism, various forms of tourism temporal and spatial trajectory data have been formed, such as GPS trajectory, BeiDou navigation information, and check-in records. These data and a large number of travel experience, travel photos, and other data shared by users jointly form tourism big data [3].

Scientific travel route planning can not only help travelers formulate their travel routes according to their time and budget but also improve their travel experience [4]. Based on the problems encountered by current users in travel planning, tourism route planning came into being. To get a high-quality solution in travel planning, we need to consider many factors and establish corresponding evaluation models according to different standards [5, 6]. For example, Rahimi and Xin further extended the existing work by studying the periodicity of space and time in user check-in data and proposed two new recommendation algorithms [7]. Zhang et al. studied some representative topic model extraction methods based on the spatial and temporal features of short text [8]. Bin et al. proposed a neural multicontext modeling framework (NMMF) combined with rich heterogeneous tourism information [9].

When using a single type of user generated content for tourism route planning, there is often great uncertainty, so it is difficult to ensure the accuracy of the user trajectory [10]. Therefore, the comprehensive use of various content to more accurately mine the user's historical trajectory has become the focus of current research [11]. Feng and Qian studied a new method to help users digest a large number of available opinions in an easy way [12]. Marrese-Taylor et al. used a multisource social media integration method to integrate fragmented tourism information from many aspects to recommend routes to users [13]. Hu et al. proposed a new framework called scenic planner for travel route recommendation, including scenic road network modeling and scenic spot route planning [14].

More and more travelers are posting their travelogues, including experiences, suggestions, and experiences, to online social media platforms [15]. Travelogues contain a large number of unique experiences and reliable travel advice from different travelers, providing an excellent reference for travel planners to select their routes [16]. As a result, some studies today make more accurate and personalized travel route recommendations by uncovering graphic information on travelogues (TGI) and finding out travelers' preferences, habits, and the popularity of attractions [17].

Lim et al. used the positioning information and shooting time in the picture to calculate the preference of tourists and the popularity of scenic spots [18]. Peng et al. recommended scenic spot areas according to the clustered areas by using social media pictures [19]. However, they only analyzed the picture information and ignored the text. Instead, the massive travel notes released by social media are full of pictures and texts. Coincidentally, Murphy and Banerjee paid more attention to the text information in travel notes [20]. Tai et al. and Lu et al. did not combine user behavior habits, interest preferences, and route popularity, and the personalization of route planning results was not high [21, 22]. In addition, based on privacy protection and other considerations, photos in travelogues often deliberately hide some attribute data [23]. This paper uses the number of pictures taken by tourists to calculate the tourist preference and scenic spot popularity, supplemented by the text description of the travel notes to make up for the lack of some attribute data. Therefore, the comprehensive use of picture and text information can often obtain more accurate recommendation results [24]. For example, Arain et al. extracted semantic information of tourist attractions and user preferences by using photos with geographical labels and user context information [25]. Huang developed a heuristic algorithm calculating context similarity, which can be used in photo data and GPS track [26].

Personalized tourism recommendation is more difficult to analyze and mine useful information from numerous tourism data. At present, many methods have many deficiencies in recommendation quality and speed. The POI + TGI model mainly considers the generated content of users and has significant advantages in preference extraction and fast route generation. This paper proposes a personal trip recommendation based on interest and popularity (PTRIP) algorithm. The algorithm comprehensively considers the text description and photo data in online travel notes, uses the cooccurrence information of scenic spots and the number of photos taken by tourists for a certain type of POI to calculate the popularity of scenic spots and tourist preference for various scenic spots, respectively, and, combined with the time cost of tourists conversion between POIs, generates a tourism route transfer map for tourism route recommendation. Compared with the original method, this algorithm uses the text and picture information in the travel notes at the same time, comprehensively considers the two factors of tourist preference and scenic spot popularity, and effectively improves the accuracy of recommendation.

2. Basic Definition

This paper uses the orientation problem to return an optimal travel route for the user, considering the interest preference of users and the popularity of POI. The personalized recommendation method proposed in this paper incorporates two social factors: preferences of user in travelogue graphic information and interpersonal interest similarity [27]. Therefore, we first introduce the user interest factors. Then, we derive the objective function of the proposed personalized recommendation model. Finally, the training method of the model is given. In the following, we present our definitions and methods in detail. We take Wuhan, China, as the empirical research object of this article. This city has many types of scenic spots, such as natural scenery, urban prosperity, and historical heritage, which is very attractive.

Let a directed weighted POI transfer graph be $G = <V, E>$, where V is the set of nodes and E is the set of edges. A node $p \in V$ represents a POI, and each p has the category attribute Cat_p (e.g., beach and castle), longitude, and latitude. The value on node p represents the score of POI_p, while C denotes the set of all POIs. (c_i, pop_i) is the attribute of node p_i, where c_i denotes the category and pop_i denotes the popularity. Each directed edge (p_i, p_j) represents a feasible route between two POIs, and the weights on the edges represent the travel time (in h) spent to visit the two POIs consecutively. An example of a POI transfer map is shown in Figure 1. Each p represents a scenic spot in Wuhan. The specific digital source and path analysis will be described in the next chapter.

Definition 1. Given a visitor t, its POI preferences of category c_i can be expressed as Equation (1).

$$\mathrm{Int}(t) = <\mathrm{Int}(t, c_1), \mathrm{Int}(t, c_2), \cdots, \mathrm{Int}\left(t, c_{|C|}\right)>. \tag{1}$$

Definition 2. The time from p_i to p_j is defined as Equation (2).

$$T^{\mathrm{travel}}\left(p_i, p_j\right) = \frac{\mathrm{Dist}\left(p_i, p_j\right)}{\mathrm{speed}}. \tag{2}$$

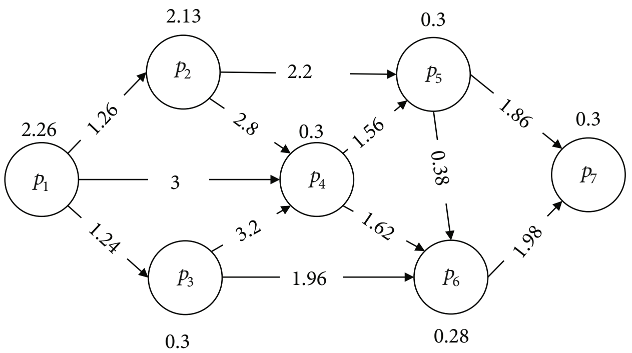


Figure 1: An example of POI transfer graph.

$\text{Dist}(p_i, p_j)$ denotes the distance between p_i to p_j. Its value is the actual distance recorded by Gaode Map. Generally, the bus and self-driving tours are greatly affected by traffic conditions, while the walking time of tourists is within the predictable range. Therefore, this paper uses the walking time between POIs as the travel time and takes speed = 5 km/h.

Definition 3. Given a user u and the set of POIs he/she has visited, S_u defines his/her historical travel footprint in chronological order, as in Equation (3).

$$S_u = \left(\left(p_1, t_{p1}^a, t_{p1}^d\right), \left(p_2, t_{p2}^a, t_{p2}^d\right), \cdots, \left(p_n, t_{pn}^a, t_{pn}^d\right)\right). \quad (3)$$

Each triplet $(p_x, t_{px}^a, t_{px}^d)$ consists of three elements: the arrival time t_{px}^a at p_x and the departure time t_{px}^d from p_x. The first photo taken by the user at each POI is the arrival time, and the last photo is the departure time. Thus, the visit time of u at p_x (i.e., the stay time) can be defined by the difference between t_{px}^a and t_{px}^d. Similarly, for the travel sequence S_u, t_{px}^a and t_{px}^d represent the start and end time, respectively. Simplicity, we put S_u as $S_u = (p_1, \cdots, p_n)$.

Definition 4. The POI scores in this paper are obtained by combining attraction popularity and visitor preferences by weighting. The score of POI_{pi} for c_i is defined in Equation (4).

$$\text{score}(p_i) = \alpha \bullet \text{Int}(\,t, c_i\,) + (1 - \alpha)\text{pop}_i, \quad (4)$$

where α is a weight adjustment parameter to adjust the weight of preference and the popularity of POI in the tour route?

Definition 5. Given a tourist t, the total number of route attractions n and the set of POI scores, the visitor gain is defined as Equation (5).

$$\text{profit}(t) = \frac{\sum_{i=1}^{n} \text{score}(p_i)}{\sum_{i=1}^{n-1} \sum_{j=2}^{n} T^{\text{travel}}\left(p_i, p_j\right)}. \quad (5)$$

3. Recommendation Method

3.1. Method Framework. The travel route recommendation algorithm proposed in this paper is divided into data preprocessing, POI mode, association graph construction, and tourist interest preference learning and route recommendation. The POI transfer graph is constructed offline and learns interest preferences of tourists. The POI and the interest preferences of tourists are obtained by analyzing the cooccurrence information of attractions and photo data in the travelogue. The route recommendation is conducted online. Based on the personal information entered by tourists, the number of expected attractions and the designated tour points, the PTRIP algorithm is used to recommend the routes with the highest benefits to tourists, considering the POI popularity and tourists' preferences. The detailed framework is shown in Figure 2. The basis for tourist itinerary recommendations mainly comes from visitor information, design tour sites, budget number of attractions, and profit of route. The first three bases mainly refer to the subjective will of the referee; the last basis is the problem to be solved by the algorithm proposed in this paper. Route profit is calculated by popularity of POI and visitor preference, and the two indicators have their weights. Moreover, the POI mode is the most important part in the travel itinerary recommendation model.

3.2. Construct the POI Transfer Graph. The POI transfer graph is constructed offline. Treating all POIs as nodes on the way, the travel routes can be generated by visiting the directed edges in the graph consecutively.

(1) Map the photo

The web travelogues shared by tourists contain textual description information such as travel routes, travel feelings, and their photos taken at each attraction. The travelogue number and tourist number can be extracted from them. The structure of the photo data shared by the user conclude Photo ID, User ID, Time, Longitude, Latitude, and Category. Based on the longitude and latitude of each photo, the distance of each POI can be calculated by using Formula Haversine. If the result is less than 200 meters, it is assumed that the photo is taken at this POI. And the list of POI is $S_t = (p_1, p_2, \cdots, p_n)$. Meanwhile, the time cost of inter-POI transition can be calculated in walking mode.

(2) The popularity of POI

The popularity of POI is calculated by combining the number of photos in the historical travelogues shared by visitors and the cooccurrence information of attractions by weighting, as in Equation (6).

$$\text{pop}(p) = \beta \bullet \frac{N(p)}{N_{\max}} + (1 - \beta) \bullet \frac{F(p)}{F_{\max}}. \quad (6)$$

In Equation (6), $N(p)$ is the number of photos taken by visitors to POI_p; $N_{\max}$ is the maximum number of photos taken by visitors to POI_p; $F(p)$ is the number of times

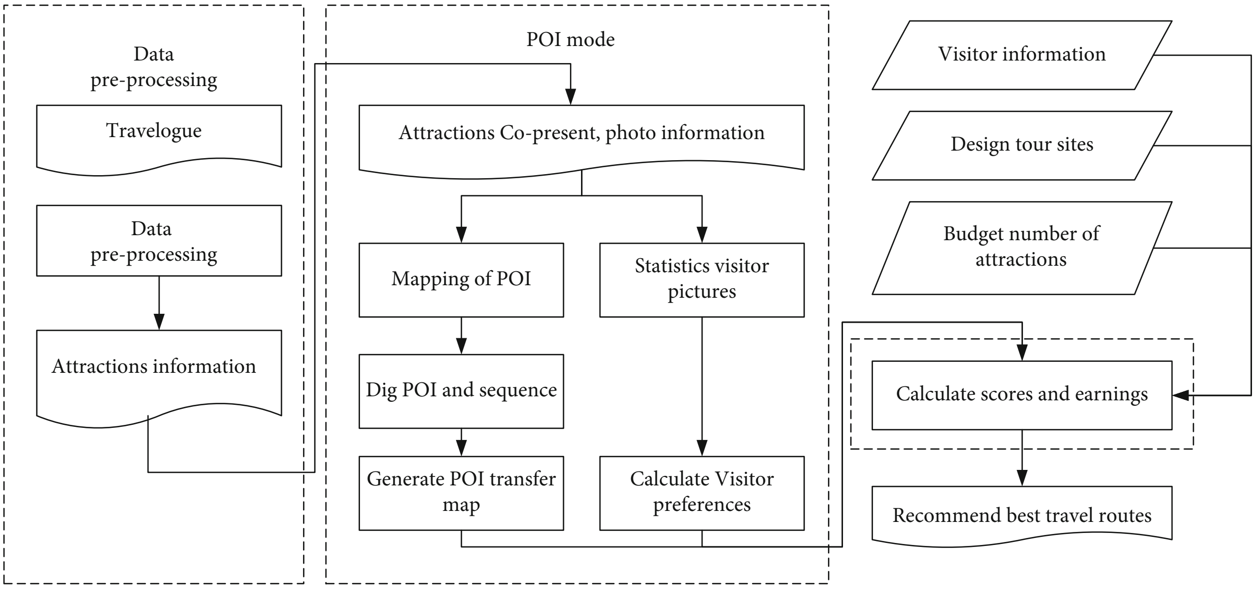

Data
pre-processing
Travelogue
Data
pre-processing
Attractions information
POI mode
Attractions Co-present, photo information
Mapping of POI
Statistics visitor pictures
Dig POI and sequence
Generate POI transfer map
Calculate Visitor preferences
Visitor information
Design tour sites
Budget number of attractions
Calculate scores and earnings
Recommend best travel routes

Figure 2: Framework for travel itinerary recommendations.

POI_p was mentioned in the travelogue; $F_{\max}$ is the maximum number of times POI_p was mentioned in the travelogue.

3.3. Interest Preferences of Tourists. We propose a time-based user interest preference from the historical travel footprint of users. When one visits a POI, he stays there for a certain amount of time. From the historical travel footprints of all users, the visit time (i.e., stay time) of each user at each POI is calculated according to Definition 4, so that the average time required for any user to visit POI can be calculated. In this paper, $\bar{V}(p)$ is the average visit time at POI_p for any user, as Equation (7).

$$\bar{V}(p) = \frac{1}{n}\sum_{u\in U}\sum_{p_x\in S_x}\left(t^d_{p_x} - t^a_{p_x}\right)\sigma(p_x = p);\forall p \in P. \tag{7}$$

In Equation (7), U is all users, n is the number of users accessing p, and $\sigma(p_x = p) = \begin{cases} 1, & p_x = p \\ 0, & \text{else} \end{cases}$.

The average access time of a user at each POI does not truly reflect his interest preference for the POI. Therefore, we propose a time-based interest preference. The preference of user for the category attribute of POI is given by Equation (8):

$$\mathrm{Int}(t) = \sum_{u\in U}\frac{t^d_{p_x} - t^a_{p_x}}{\bar{V}(p_x)}\sigma\left(Cat_{p_x} = c\right);\forall c \in C. \tag{8}$$

In Equation (6), Cat_p is the category attribute, and $\sigma($

$$\mathrm{Cat}_{p_x} = c) = \begin{cases} 1, & \mathrm{cat}_{p_x} = c \\ 0, & \text{esle} \end{cases}.$$

Equation (8) determines the interest of user in category attribute of a particular POI. Relative to the average access time of all users at the same POI, it is calculated based on the time cost by the user at each POI with category attribute. In other words, a user may spend more time visiting the POI that he is interested in, which in turn determines the interest of users in such POIs.

3.4. PTRIP Algorithm. Orienteering problem (OP) has already widely used in travel route recommendation. In a directed band power diagram $G(V, E)$, V is the set of all points on the graph, and E is the set of all edges on the graph. Each point has a corresponding score which can be expressed as a gain. And each edge has a corresponding weight which represents the travel time between two points. The start and end points are specified, and some points are selected from diagram G, and a path is planned through these points and the specified start and end points, while maximizing the score under the condition that the total weight of the path does not exceed a certain time budget.

In this paper, we propose the PTIR route recommendation algorithm based on the TGI and POI. PTIR can provide a route with the highest score and satisfied time budget, i.e., $R = \{p_1, p_2, \cdots, p_N\}$. Time budget is calculated by function $\mathrm{Cost}(p_x, p_y) = T(p_x, p_y)\,\bar{V}(p_y)$. From this, it follows that the travel route recommendation model in this paper can be expressed integer programming problem satisfying multiple constraints:

$$\mathrm{Max}\sum_{i=2}^{N-1}\sum_{j=2}^{N} x_{i,j}\,\mathrm{profit}(p_i). \tag{9}$$

In Equation (9), $x_{i,j} = 1$ indicates the route from i to j, i.e., (p_i, p_j), or $x_{i,j} = 0$.

$$\sum_{j=1}^{N} x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1, \tag{10}$$

$$\sum_{j=2}^{N} x_{k,j} = \sum_{i=1}^{N-1} x_{i,k} \leq 1; \forall k = 2, 3, \cdots, N-1, \tag{11}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \text{Cost}(i, j) x_{i,j} \leq B, \tag{12}$$

$$2 \leq u_i \leq N, \tag{13}$$

$$u_i - u_j + 1 \leq (N-1)\left(1 - x_{i,j}\right); \forall i, j = 2, 3, \cdots, N. \tag{14}$$

Equation (9) is the objective function that maximizes the POI popularity and user interest preferences in the recommended route. Equations (10)–(14) are constraints. Equation (10) ensures that the starts at p_1 and ends at p_n; Equation (11) ensures that the itinerary is coherent and that each POI in the itinerary is visited only once; Equation (12) ensures that the time spent on the trip is within budget; Equation (13) and Equation (14) ensure that there are no subcircuit routes in this integer programming problem. The lpsolve (BERKELAAR M, 2004) linear programming package is used to solve the proposed integer programming problem.

To explain the algorithm, we take Figure 1 for example and give set $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, as in Table 1. Given a tourist t_1 and the set of POIs, he has visited $S_{t1} = \{p_1, p_2\}$. The POIs visited by tourist $t_1$and the number of photos at each POI are shown in Table 2. The average number of photos at each POI calculated from Equation (5) based on historical tourist data is shown in Table 3, and the popularity of each POI calculated from Equation (4) is shown in Table 4. The time cost required for a visitor t_1 to transfer between POIs and the rating value of each POI are represented by the values of the directed edges and the values of the nodes in Figure 1.

We assume that p_4 is a mandatory site for tourists to visit and plan to go to 4 attractions. The amount of interest preference can be calculated by Equation (8). The result is Int(t_1) =<2.8,2.67,0,0,0>, and from this, we can get 13 routes based on PTRIP algorithm, as shown in Table 5.

And the benefits of seven routes are calculated by Equation (5). The results are the following: $\text{pro}(R_1) = 0.89$, pro$(R_2) = 0.86$, $\text{pro}(R_3) = 0.53$, $\text{pro}(R_4) = 0.52$, $\text{pro}(R_5) = 0.64$, $\text{pro}(R_6) = 0.49$, $\text{pro}(R_7) = 0.48$, $\text{pro}(R_8) = 0.64$, $\text{pro}(R_9) = 0.49$, $\text{pro}(R_{10}) = 0.70$, $\text{pro}(R_{11}) = 0.19$, $\text{pro}(R_{12}) = 0.18$ $\text{pro}(R_{13}) = 0.17$. Finally, the R_1 route that has the highest profit may be recommended to visitor t_1, i.e., Yellow Crane Tower, Riverbank Park, Wuhan University, and Tumultuan Lin. Combined with the reality of tourism websites, this route is adopted more frequently, which preliminarily proves the effectiveness of the algorithm.

Table 1: The properties of POI.

POI	Category properties
Yellow Crane Tower (p_1)	Ancient ruins and buildings (c_1)
Riverbank Park (p_2)	Natural scenery (c_2)
Guiyuan Buddhist Temple (p_3)	Folk religion (c_3)
Wuhan University (p_4)	Humanities, academia (c_4)
Tumultuan Lin (p_5)	Bussiness, street(c_5)
Hubu Lane (p_6)	Ancient ruins and buildings (c_1)
East Lake (p_7)	Natural scenery (c_2)

Table 2: The visited POI of $t_{1.}$

POI	Number of photos
P1	35
P2	20

Table 3: Average photos of POI.

POI	Average photos
p_1	12.5
p_2	7.5
p_3	4.5
p_4	13.8
p_5	6.3
p_6	11.7
p_7	8.2

Table 4: The popularity of POI.

POI	pop(p)
p_1	1.00
p_2	0.87
p_3	1.00
p_4	1.00
p_5	1.00
p_6	0.93
p_7	1.00

4. Experimental Results and Analysis

4.1. Experimental Data. In this paper, we use 2,638 travelogues obtained from the Ctrip Travel website with "Wuhan" as the keyword as the experimental dataset. After data preprocessing, the dataset contains two main aspects: 116,396 photos of Wuhan and its surrounding areas, including the number of the travelogue to which the photos belong and the location where they were taken, and the cooccurrence statistics of 168 POIs in the travelogue, as well as 5,238 historical single-day travel routes and the actual distance information between the connected POIs in the routes. This experiment uses the leave-one-out cross-validation method commonly used in recommendation system validation to experimentally validate the algorithm, which loops the records in the specified dataset as the test set or training set, respectively, and calculates the predicted conclusions of each loop in a comprehensive manner to derive the measurement index.

Table 5: Initial recommended route of Wuhan.

Route	POI passed
R_1	$\{p_1, p_2, p_4, p_5\}$
R_2	$\{p_1, p_2, p_4, p_6\}$
R_3	$\{p_1, p_3, p_4, p_5\}$
R_4	$\{p_1, p_3, p_4, p_6\}$
R_5	$\{p_1, p_4, p_5, p_6\}$
R_6	$\{p_1, p_4, p_5, p_7\}$
R_7	$\{p_1, p_4, p_6, p_7\}$
R_8	$\{p_2, p_4, p_5, p_6\}$
R_9	$\{p_2, p_4, p_5, p_7\}$
R_{10}	$\{p_2, p_4, p_6, p_7\}$
R_{11}	$\{p_3, p_4, p_5, p_6\}$
R_{12}	$\{p_3, p_4, p_5, p_7\}$
R_{13}	$\{p_3, p_4, p_6, p_7\}$

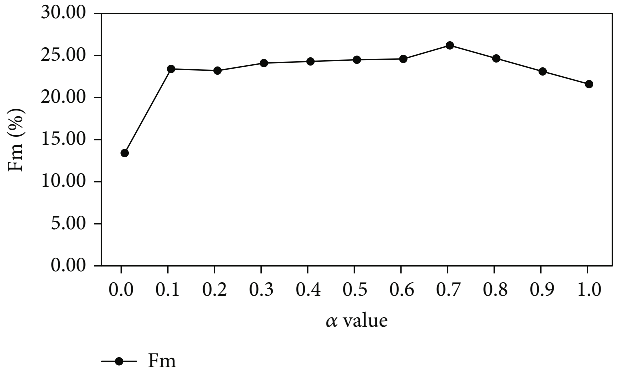


Figure 3: The effect of α on the recommendation accuracy.

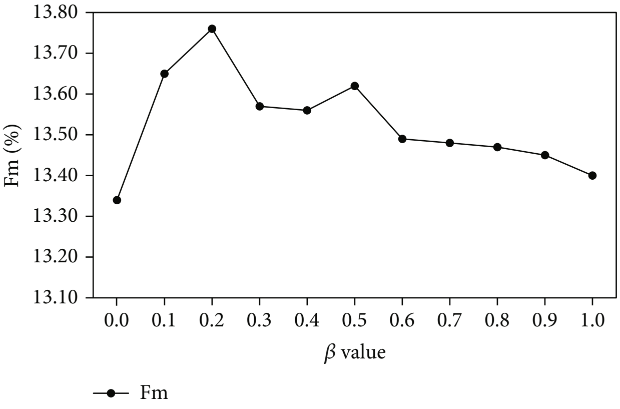


Figure 4: The effect of β on the recommendation accuracy.

4.2. Algorithm Accuracy Analysis. The accuracy of recommendation is the most basic metric for evaluating algorithms. In this paper, the precision and recall are used as the criteria to measure the algorithm performance. The calculation formulas are as Equations (15) and (16).

$$\text{precision} = \frac{|P_r \cap P_v|}{|P_r|}, \tag{15}$$

$$\text{recall} = \frac{|P_r \cap P_v|}{|P_v|}. \tag{16}$$

Precision represents the probability that a user is interested in the recommended route, and the recall represents the probability that one preferred POI is recommended, and the higher the precision and recall, the better the recommendation. P_r represents the set of POIs in the recommended route, and P_v represents the set of POIs in the real travel sequence that the user has visited. To better verify the recommendation quality of the algorithm in this paper, the metrics F_m are introduced, as Equation (17).

$$F_m = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{17}$$

The value of α is used to determine the weight assignment of visitor interest preferences and POI popularity when calculating route profit. For a given number of tour points and a specified number of tour points, the effect of α on the recommendation accuracy is shown in Figure 3.

When $\alpha = 0$, the weight of $\text{pop}(p)$ is 1, and the recommendation of visitor route is just based on the popularity of POIs. When $\alpha = 1$, the weight of $\text{Int}(t, c_i)$ is 1, and the recommendation of visitor route is just based on the interest preferences of tourists. As shown in Figure 4, the accuracy of route recommendation tends to increase and then decrease with the increase of α. It shows that the recommendation effect of considering popularity of POI and visitor preference is better than that of considering only one of them, and the best recommendation result is achieved when $\alpha = 0.7$.

The value of β is used to determine the weight distribution between the number of photos and the cooccurrence information of attractions when calculating POI popularity. In the case of route recommendation using POI popularity only, the effect of β value on the accuracy of route recommendation is shown in Figure 4, given the number of attractions visited and the specified tour points.

When $\beta = 0$, the weight of $\text{Int}(t)$ is 1, calculation of POI popularity based only on the cooccurrence data of attractions in travelogues. When $\alpha = 1$, the weight of $\text{Int}(t, c_i)$ is 1, calculation of POI popularity based only on the number of photos in the historical travelogues shared. As shown in Figure 4, the accuracy of route recommendation fluctuates with the variation of β. The lowest values at both ends of the curve, which means the popularity of POI calculated by considering the cooccurrence of attractions and the number of photos in the travelogue is more accurate. And the best recommendation result is achieved when $\beta = 0.2$.

Comparing the analysis of Figures 3 and 4, it is found that β has a small effect on the accuracy, fluctuating in the range of 1%, while the change of α makes the accuracy fluctuate in the range of 10%. From this, we can find that the calculation of POI popularity is related to the number of photos and photo cooccurrence information, but the weight

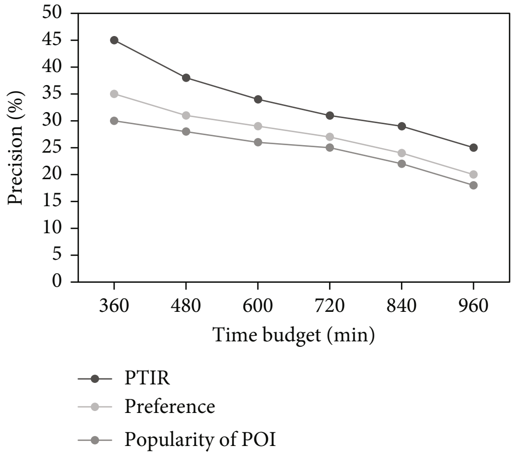


Figure 5: The experimental result of precision.

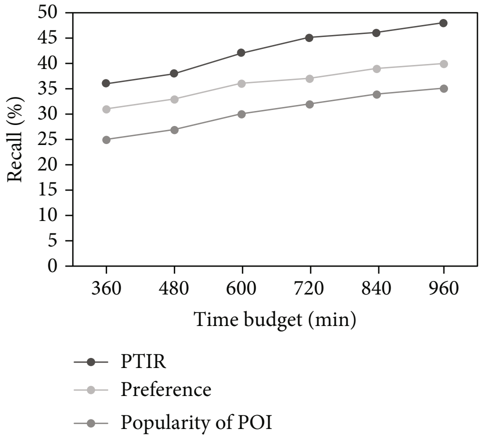


Figure 6: The experimental result of precision.

between them is not very important. The focus of the recommended route is on the subjective will of the tourists, so personalized customization is the future development direction of customized travel routes.

To verify the effectiveness of the PTIR algorithm, this paper uses the traditional travel route recommendation algorithm as a control, in which the recommendation algorithm considering only POI popularity and the recommendation algorithm considering only user interest preference are used as the metric, respectively. Under different time budgets B, the traditional algorithm is compared with PTIR, a travel route recommendation algorithm based on POI popularity and user interest preferences, and the experimental results are shown in Figures 5 and 6.

Figure 5 shows the difference in precision between the PTIR algorithm and the traditional algorithm. The precision of PTIR algorithm is much higher than algorithms that only consider user interest or only consider POI popularity. Figure 6 shows the difference in recall between the PTIR algorithm and the traditional algorithm. The recall of PTIR algorithm has the same situation with recall rate indicator. Among them, the accuracy and recall accuracy of the algorithm considering user interest are higher than the POI popularity only. One of the influencing factors is that both the algorithm PTIR and the algorithm that considers only the user's interest consider the user's interest because users prefer to visit places that interest them. The high accuracy and high recall of algorithm PTIR indicate that the algorithm proposed in this paper can recommend routes that reflect real travel sequences of users more accurately. It shows that when recommending travel itineraries to tourists, they should be guided by interests of users.

Overall, the precision increases with the increase of time budget, while the recall rate is the opposite. There are large uncertainties in the process of personalized tourism recommendation. In addition to the popularity of POI and preference of visitor, it is also necessary to consider time budget of them. By controlling the time budget and comprehensively considering the accuracy and recall of the algorithm, the time budget point corresponding to the best experimental result can be found. Then, while planning the tourist route for tourists, it is suggested to travel time.

5. Conclusion

To improve the accuracy of travel route recommendation and make comprehensive use of the graphic information in travel notes, this paper proposes a personalized route recommendation algorithm PTRIP.

Firstly, the algorithm uses the scenic spot cooccurrence information and photo data shared in the online travel notes to calculate the POI popularity and tourists' interest preferences and then comprehensively uses the above information to construct a personalized travel route recommendation framework to recommend the optimal travel route to tourists.

Finally, the experimental verification is carried out by using the real data set shared on the Ctrip Travel website. It is proved that the recommendation accuracy of the PTRIP algorithm proposed in this paper is significantly higher than that of the traditional recommendation algorithm which only uses the cooccurrence information of text scenic spots and also higher than that of the original algorithm which only uses the photo information of tourists.

The accuracy of PTRIP algorithm is much higher than the traditional algorithm considering only the popularity of POI. It is also better than the traditional algorithm that only considers tourists' preferences. Moreover, the comprehensive use of graphic information in travel notes can maximize the use of the information recorded in travel notes on the one hand and make up for the incomplete basic attributes of tourist photos caused by privacy and other reasons [28]. The POI popularity score quality calculated by PTRIP algorithm is also higher than the traditional algorithm considering text or picture alone. Experiments show that PTRIP algorithm can effectively make comprehensive use of the graphic information of Travel Notes published in social media to make more accurate personalized travel route recommendation.

The proposal of global tourism, smart tourism, and other strategies and the proliferation of user shared content have not only brought opportunities but also greater challenges to tourism route planning. The planning method based on user generated content is not perfect. In real life, tourists may have multiple tourism needs to be optimized at the same time. How to efficiently solve the multiobjective optimization problem of personalized tourism route recommendation will be the next research direction.

Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no competing interests.

References

[1] R. A. Abbaspour and F. Samadzadegan, "Time-dependent personal tour planning and scheduling in metropolises," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12439–12452, 2011.

[2] T. Majeed, A. Stämpfli, A. Liebrich, and R. Meier, "What is of interest for tourists in an alpine destination: personalized recommendations for daily activities based on view data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4545–4556, 2020.

[3] Z. Ning, X. Hu, Z. Chen et al., "A cooperative quality-aware service access system for social Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2506–2517, 2018.

[4] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: a survey," *Geo Informatica*, vol. 19, no. 3, pp. 525–565, 2015.

[5] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2013.

[6] D. Gavalas, V. Kasapakis, C. Konstantopoulos, G. Pantziou, N. Vathis, and C. Zaroliagis, "A personalized multimodal tourist tour planner," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pp. 73–80, Melbourne, Australia, 2014.

[7] S. M. Rahimi and W. Xin, *Location Recommendation Based on Periodicity of Human Activities and Location Categories*, Springer, Berlin Heidelberg, 2013.

[8] J. Zhang, X. Hu, Z. Ning et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2018.

[9] C. Bin, T. Gu, Z. Jia, G. Zhu, and C. Xiao, "A neural multicontext modeling framework for personalized attraction recommendation," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 14951–14979, 2020.

[10] S. Banerjee and A. Y. K. Chua, "In search of patterns among travellers' hotel ratings in TripAdvisor," *Tourism Management*, vol. 53, pp. 125–131, 2016.

[11] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user Interest and social circle," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763–1777, 2014.

[12] H. Feng and X. Qian, "Mining user-contributed photos for personalized product recommendation," *Neurocomputing*, vol. 129, pp. 409–420, 2014.

[13] E. Marrese-Taylor J. D. Velásquez et al., "A novel deterministic approach for aspect-based opinion mining in tourism products reviews," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7764–7775, 2014.

[14] B. Hu, G.-P. Gao, L. L. He, X.-D. Cong, and J.-N. Zhao, "Bending and on-arm effects on a wearable antenna for 2.45 GHz body area network," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 378–381, 2016.

[15] C. Chen, X. Chen, Z. Wang, Y. Wang, and D. Zhang, "Scenic-Planner: planning scenic travel routes leveraging heterogeneous user-generated digital footprints," *Frontiers of Computer Science*, vol. 11, no. 1, pp. 61–74, 2017.

[16] A. Majid, L. Chen, G. Chen, H. T. Mirza, I. Hussain, and J. Woodward, "A context-aware personalized travel recommendation system based on geotagged social media data mining," *International Journal of Geographical Information Science*, vol. 27, no. 4, pp. 662–684, 2013.

[17] A. S. Tewari and A. G. Barman, "Sequencing of items in personalized recommendations using multiple recommendation techniques," *Expert Systems with Application*, vol. 97, pp. 70–82, 2018.

[18] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized tour recommendation based on user interests and points of interest visit durations," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 1778–1784, California, 2015.

[19] H. Peng, B. Hu, Q. Shi et al., "Removal of ocular artifacts in EEG—an improved approach combining DWT and ANC for portable applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 600–607, 2013.

[20] H. C. Murphy, M. M. Chen, and M. Cossutta, "An investigation of multiple devices and information sources used in the hotel booking process," *Tourism Management*, vol. 52, pp. 44–51, 2016.

[21] C. H. Tai, D. N. Yang, L. T. Lin, and M.-S. Chen, "Recommending Personalized Scenic Itinerary with Geo-Tagged Photos," in *Proceedings of 2008 IEEE International Conference on Multimedia and Expo*, pp. 1209–1212, Hannover, Germany, 2008.

[22] X. Lu, C. Wang, J. M. Yang, Y. Pang, and L. Zhang, "Photo 2Trip: generating travel routes from geo-tagged photos for trip planning," in *Proceedings of the 18th International Conference on Multimedia*, pp. 25–29, Firenze, Italy, 2010.

[23] G. Cui, J. Luo, and X. Wang, "Personalized travel route recommendation using collaborative filtering based on GPS trajectories," *International Journal of Digital Earth*, vol. 11, no. 3, pp. 284–307, 2018.

[24] X. Hu, J. Cheng, M. Zhou et al., "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.

[25] Q. A. Arain, H. Memon, I. Memon et al., "Intelligent travel information platform based on location base services to predict user travel behavior from user-generated GPS traces," *International Journal of Computers and Applications*, vol. 39, no. 3, pp. 155–168, 2017.

[26] H. Huang, "Context-aware location recommendation using geotagged photos in social media," *International Journal of Geo-Information*, vol. 5, no. 11, p. 195, 2016.

[27] B. Zheng, H. Su, K. Zheng, and K. Zhou, "Landmark-based route recommendation with crowd intelligence," *Data Science and Engineering*, vol. 1, no. 2, pp. 86–100, 2016.

[28] D. Gavalas, C. Konstantopoulos, K. Mastakas et al., "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, no. 3, pp. 291–328, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3364921, 13 pages
https://doi.org/10.1155/2022/3364921

Research Article

An End-to-End Deep Learning Approach for Plate Recognition in Intelligent Transportation Systems

Jamshid Pirgazi, Mohammad Mehdi Pourhashem Kallehbasti, and Ali Ghanbari Sorkhi

Department of Computer Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran

Correspondence should be addressed to Jamshid Pirgazi; j.pirgazi@mazust.ac.ir

Received 24 November 2021; Revised 22 December 2021; Accepted 28 December 2021; Published 24 January 2022

Academic Editor: Mohammad R Khosravi

Accurate and fast recognition of license plates is one of the most important challenges in the field of license plate recognition systems. Due to the high frame rate of surveillance cameras, old license plate recognition systems cannot be used in real-time applications. On the other hand, the presence of natural and artificial noise and different light and weather conditions make the detection and recognition process of these systems challenging. In this paper, an end-to-end method for efficiently detecting and recognizing plates is presented. In the proposed method, vehicles are first detected using a single-shot detector- (SSD-) based deep learning model in the video frames and the input images. This will increase the speed and accuracy in identifying the location of the plate in the given images. Then, the location of the plate is identified using the proposed architecture based on convolutional networks. Finally, using a deep convolutional network and long short-term memory (LSTM), the characters related to the plate are recognized. An advantage of our method is that the proposed deep network is trained using different images with different qualities that leads to high performance in detecting and recognizing plates. Also, considering that in the proposed method the vehicles are first detected and then the plate is detected in the vehicle image, there is no limit in the number of identified plates. Moreover, plate detection in the vehicle rectangle, instead of the whole frame, speeds up our method. The proposed method is evaluated using several databases. The first part of the evaluation focuses on robustness and recognition speed. The proposed method has the accuracy of 100% for vehicle detection, 100% for plate detection, and 99.37% for character recognition. In the second part of evaluation, the proposed method is evaluated in terms of overall speed. The experimental results witness that the proposed method is capable of processing 30 frames per second without losing any data and also outperforms several methods proposed in recent years, in terms of time and accuracy.

1. Introduction

Due to the increasing number of vehicles, manually controlling and monitoring traffic is time-consuming, costly, inaccurate, and sometimes impossible. This makes automatic vehicle plate recognition a recurrent research topic. Since the plate is the unique ID of vehicles, several prominent applications are found for automatic plate recognition, including traffic control, driving offence detection, vehicle speed estimation, self-driving vehicles, and surveillance [1]. To this end, many cameras are installed in cities, roads, highways, borders, parking lots, and protected areas for better and more accurate control of vehicles. These cameras are constantly monitoring the images of passing vehicles. Vehicles and their plates cannot be detected and recognized without processing and analyzing these images.

There is a need for a system based on image processing and machine learning to detect vehicles and extract other information such as plate number [2]. In vehicle plate detection and recognition systems, the quality of the input images has a direct impact on the result accuracy and processing speed. Many parameters are influential in the quality of the captured images including environmental and weather conditions such as light projection angle, light intensity, rainfall, fog, dust, humidity, dark, glare, occluded, rainy, snowy, tilted, or blurred scenarios.

There are other concerns in automatic plate recognition systems, including different plate alignments on vehicles, size, plate aspect ratio, various shapes of plates, various angles of camera placement, too low or too high lighting, presence of several plates in the image, various plate background colors, excessive plate dirtiness, and various arrangements of letters and numbers in plates [3, 4]. As shown in Figure 1, a vehicle plate detection and recognition system consists of the three main steps: (i) plate detection, (ii) segmentation of characters in the plate, and (iii) character recognition [5].

The most important and challenging step is plate detection. If plates are not detected properly, the subsequent steps will not work as expected and the final result will be entirely wrong.

There have been many attempts to efficiently detect plates. In order to make this step more efficient, a phase of preprocessing is performed to denoise and improve image quality. The location of the plates is then identified in the input image. According to the previous studies on plate detection, existing approaches fall in five categories: edge-based methods, color-based methods, texture analysis methods, methods based on image global features, and hybrid methods. Different methods of image processing and machine vision have been used in each of these categories [6–11]. In the next step, plate segmentation is performed. In this step, the detected plate image is first converted to a binary image. Then, based on methods such as morphology [12], connected component analysis algorithm [13], and histogram-based methods, the parts related to characters are separated. The problem with this step is that most binarization methods have acceptable results only for clean plates. Using these methods on dirty plates leads to huge data loss. More precisely, some character parts of the plate may not be recognized as characters or some noncharacter parts may be recognized as characters. As can be seen in Figure 2, some noncharacter parts are considered as characters.

The next step in plate recognition systems is character recognition. This step is divided into two phases. In the first phase, different features are extracted from the segmented parts of the previous step. Then, the training of machine learning models is done based on the extracted features. In the feature extraction phase, various methods are proposed including active regions [14], HOG [15], horizontal and vertical mapping [16], and multiclass AdaBoost methods [17]. Some algorithms use key points locating methods such as SIFT [18] and SURF [19]. After creating the feature vector, classification is done for each section. In various works, artificial neural networks, support vector machine, Bayesian classifier, K nearest neighbor, etc. have been used for character recognition [20, 21]. Extracted features can be numerous and not all of them are useful for classification purposes. In other words, some features are redundant and irrelevant. The existence of large number of features makes the machine learning models prone to overfitting. To address this problem, feature selection and dimension reduction are performed in some works.

Nowadays, due to the massive production of labeled data in different phases and increasing computing power, the use of deep learning methods has received considerable attention. Some works propose end-to-end methods for plate detection and recognition. In these works, only one or two of these steps are performed using deep learning methods due to lack of required data to train the deep learning model.

Gou et al. [22] proposed a method based on character-specific extremal regions and hybrid discriminative restricted Boltzmann machines (HDRBMs). Vertical edge detection, morphological operators, and different evaluations have been used for top-hat transfer plate detection. Specific character regions are identified as candidate regions for plate characters. Then, a trained model of HDRBM is used for character recognition. Their proposed method is resistant to changes in light intensity and weather conditions.

Wang et al. [23] proposed a multifunctional convolutional neural network (CNN) to detect and recognize Chinese vehicle plates with better accuracy and lower computational cost. In their proposed method, the detection network consists of three layers (P-Net, R-Net, and O-Net) and also a fully connected layer. The output of this part is the input of the detection network. In this part, the features are normalized and then the plate characters are recognized using a CNN. Wen-bin et al. [24] used an improved convolutional recurrent neural network to recognize Chinese plates. In this method, deep CNNs, recurrent neural networks (RNNs), spatial transformer networks, and connectionist temporal classification model are combined. In this method, there is no need for plate segmentation that causes erroneous plate detection.

Li et al. [25] proposed a plate recognition method that is a combination based on deep neural networks. In this method, recurrent neural networks with LSTM were used for plate feature extraction. The extracted sequence features in this part are given to a 37-class CNN for character detection. The advantage of using this method is that there is no need for plate segmentation.

The differences between the character features in terms of width, height, distance, and presence of noise such as heavy shadows, uneven light, different optical geometries, and poor contrast are challenging in plate recognition systems. Bulan et al. [26] used a two-stage classification method plate detection. For this purpose, the plate candidate points are first classified using a weak classifier, and then, the plate main points are detected using a strong classifier based on deep convolutional networks. After that, using the ALexnet architecture, features are extracted from the regions extracted from the previous step, and using support vector machine classification, the character detection is done based on the features obtained from the previous step.

In [27], an end-to-end method is proposed to detect and recognize vehicle plates. In this method, features are first extracted from the input image using CNNs. Plate regions are then identified based on the deep recurrent network.

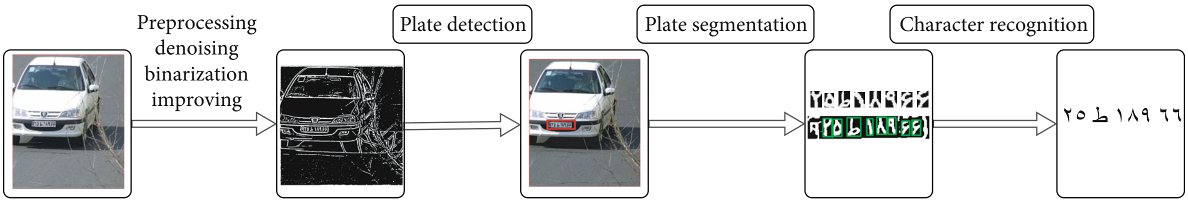


Figure 1: General steps of the vehicle plate recognition and detection system.

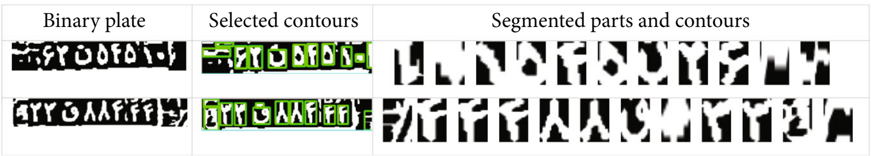


Figure 2: General steps of the vehicle plate recognition and detection system.

After that, the plate characters are recognized using convolutional and recurrent networks. Performing these steps makes the model to perform with acceptable speed.

In [28], fully convolutional networks are combined with a broad learning system. A fully convolutional network, which has been designed as a two-stage classification method at the pixel level, is used to detect objects by combining multiscale and hierarchical features. The AdaBoost cascade classifier was used to classify characters, and an extensive learning system was used to recognize characters.

Chen et al. [29] proposed a method based on CNNs for plate detection. In this method, an end-to-end network is proposed that simultaneously detects vehicles and plates. Different convolutional layers have been used in this method. Gao et al. [30] proposed an end-to-end network for plate detection and recognition based on encoder and decoder. The efficiency of traditional plate detection methods is affected by several factors including light intensity, shadow, and complex background. Using deep learning methods, plate recognition algorithms can extract deep features and improve the plate detection and recognition rate [31].

An efficient shared adversarial training network has been proposed for plate detection in [32]. This model can learn environment independent semantic features without perspective from real plates using prior knowledge of standard plates.

Authors in [33–35] considered using YOLO3 architecture for plate detection and recognition. In [33], YOLO3 is used for Brazilian plate detection and a three-layer convolutional network is used for character recognition. Authors in [34] used a two-stage convolutional layer in YOLO3 architecture in order to extract temporal features from plates and to reduce false positive detection rate. Authors in [35] used seven convolutional layers for plate detection and character recognition.

In the current paper, due to the massive production of labeled data in different parts, an end-to-end method for plate detection and recognition is proposed. Vehicles are first detected using the SSD-based deep learning model in the video frames and the input images. This increases the speed and accuracy in identifying the location of the plate in the given image. Vehicle detection in the proposed method makes it suitable for smart transportation systems. In this mode, traffic volume measurement can be done based on vehicle detection. Moreover, vehicle movement direction and speed and also stopped vehicles in highways can be detected. In the next step, the location of the plate is identified using the proposed architecture based on convolutional networks. Finally, the characters related to the plate are detected and recognized using the deep convolutional network and LSTM.

An advantage of our approach is that the proposed deep network is trained with different images with different qualities that leads to high performance in detecting and recognizing plates. Apart from that, the vehicles are first detected and then the plates are detected in the vehicle images, instead of the whole image that speeds up our method. Moreover, there is no limit in the number of identified plates.

In real-time systems, analysis is performed on video frames captured by monitoring cameras. These cameras capture between 30 and 90 frames per second. Conventional systems can process 1 to 2 frames per second, while the proposed method is capable of processing 30 frames per second without losing any data. The rest of this paper is organized as follows. Section 2 presents the proposed method and describes its architecture in detail. Section 3 describes the experimental evaluation and compares the proposed method to other methods. Finally, Section 4 concludes the paper.

2. Proposed Method

Most of plate recognition methods fail to detect all the plates when there are many of them in the frame being processed. This makes these methods inapplicable for real-time situations when there can be many plates in every frame and the response time is limited. To address this issue, an SSD architecture is used in the proposed approach to make the processing speed fast enough for real-time plate recognition. In the proposed method, first vehicles are detected. Then, based on the deep learning architecture, in two phases, plates

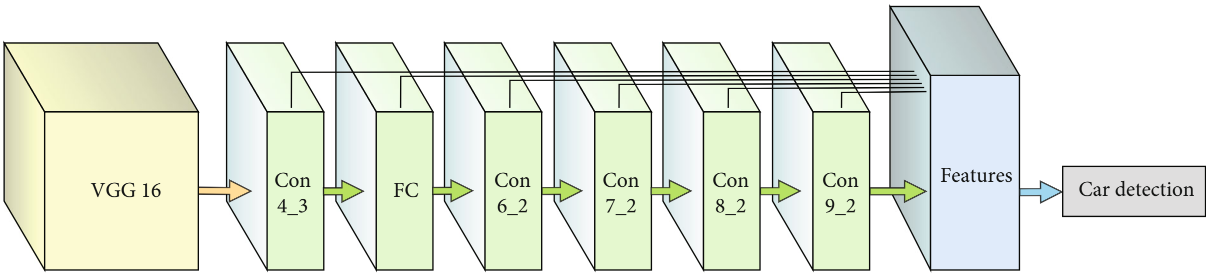


Figure 3: The deep architecture used in the proposed method for vehicle detection.

are first detected and then the text of the plates are extracted using a deep network with convolutional layers and recurrent layers. The proposed method is an end-to-end method that detects vehicles and plates simultaneously and recognizes the plates. The following sections explain different steps of this method.

2.1. Vehicle Detection. In order to achieve a better detection rate in real-time applications, it is better to start the process with vehicle detection. Conventional systems based on deep learning look for plate in the whole frame. Since plates compose a very small part of a frame, this process makes the existing approach slow. In the proposed method, vehicles are first detected in order to avoid processing of the parts of the frame that are unrelated to vehicles. This step eliminates a great deal of unnecessary processing effort and therefore speeds up the overall process. Vehicle detection is done using a deep network composed of several convolutional layers.

Vehicle detection is done based on the high-level features that are extracted by the designed convolutional network. In fact, the proposed method combines low-level featured in primary convolutional layer and high-level features extracted by convolutional and recurrent network to detect vehicles. Moreover, several layers are used for various feature extractions from input images. Since the kernels in various layers vary in size, features are extracted with different details. This helps to have images with different details and to extract relevant and discriminant features. Figure 3 depicts the proposed architecture for vehicle detection.

In [36], we proposed a practical method that produces data for various phases of Iranian plate recognition system. Using deep learning methods requires sufficient data for training models. Due to the lack of data for Iranian plate recognition, the current paper uses the proposed approach in [36].

In the proposed architecture, a model based on VGG16 is used for vehicle detection, that is followed by convolutional layers. These layers decrease in size as we move forward such that we are able to detect vehicles with different sizes. Each layer extracts a different high-level feature from vehicle for vehicle detection.

In the proposed model, training objective is derived from the multibox objective [37, 38] but is extended to handle multiple object categories. Let $x_{i,j}^{p} = \{0, 1\}$ be an indicator for matching the i-th default box to the j-th ground truth box of category p. In the matching strategy above, we can have $\sum_i \boxtimes x_{i,j}^{p} > 1$. Equation (1) presents the overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss, where N is the number of matched default boxes.

$$L(x, c, l, g) = \frac{1}{N}\left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)\right). \tag{1}$$

If $N = 0$, the loss is set to 0. The localization loss is a smooth L_1 loss [33] between the predicted box (l) and the ground truth box (g) parameters. The weight term α is set to 1 by cross validation.

2.2. Plate Detection. Most of plate images are noisy due to environmental conditions. Using Gaussian filter is a common way to address this issue. Gaussian filter is used as a preprocessing step in image processing and is a linear filter that smooths and removes noise. The first activity in this step is to improve quality of the vehicle image using Gaussian filter. Then, a deep convolutional network-based method is proposed in order to detect the plate location. Figure 4 shows the proposed architecture for plate detection. The input resolutions in deep network are 320 × 320, 416 × 416, and other multiples of 32. Therefore, the input vehicle image is resized to a 1 : 1 square, in the proposed method. Deep network preserves the aspect ratio of the images, and if the input image is not in a square shape, it automatically adds black bars to conform to a square shape.

In the proposed method, 416 × 416 resolution is used for plate detection. Plate detection works the best using this value, which is obtained from several experiments on the size of Iranian vehicles. Deep architecture of proposed method performs segmentation of size $M \times M$, where M is the size of window that can have different values. Two parameters should be given in prior for plate detection process: N that is the number of bounding boxes and S that is the confidence score for each bounding box and determines whether there is an object inside the box or the probability of its existence. We should calculate evaluation metric such as the Intersection over Union (IoU) to compare predicted bounding boxes and the dimensions and location of ground truth, to measure the S parameter. The IoU is computed as shown in Equation (2), where X and Y are the two bounding boxes. Higher values for IoU indicate that the two bounding boxes have

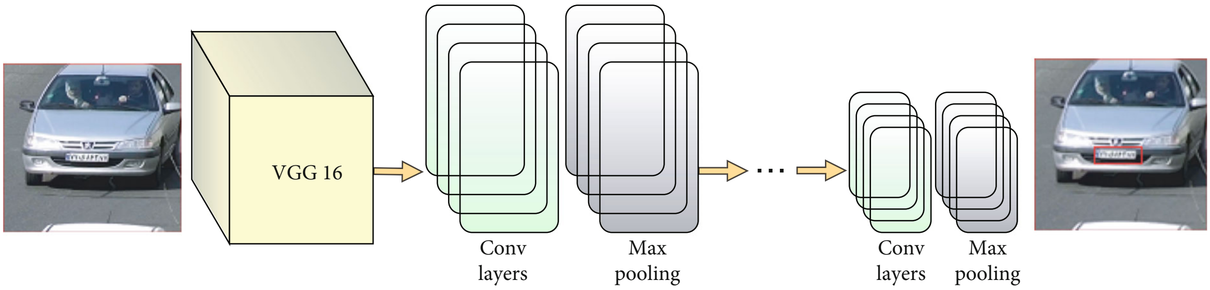


FIGURE 4: The deep architecture used in the proposed method for plate detection.

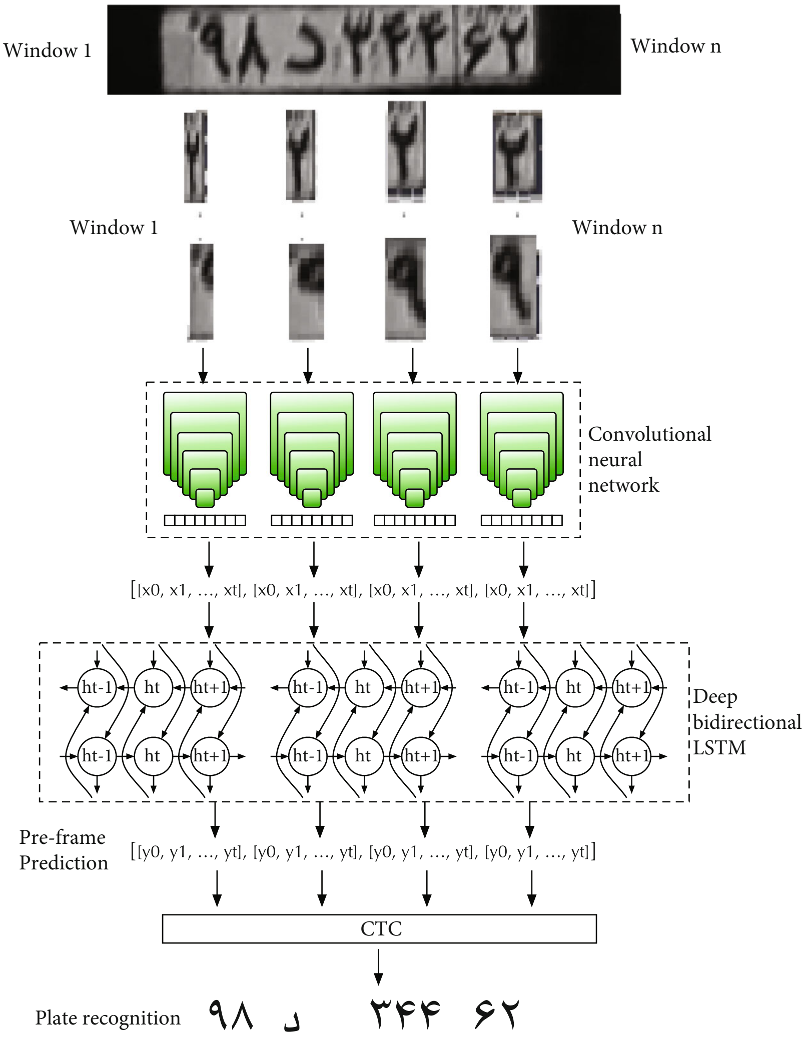

Window 1
Window n
Window 1
Window n
Convolutional neural network
[[x0, x1, ..., xt], [x0, x1, ..., xt], [x0, x1, ..., xt], [x0, x1, ..., xt]]
ht-1
ht
ht+1
Deep bidirectional LSTM
Pre-frame Prediction
[[y0, y1, ..., yt], [y0, y1, ..., yt], [y0, y1, ..., yt], [y0, y1, ..., yt]]
CTC
Plate recognition
۹۸ د ۳۴۴ ۶۲

FIGURE 5: The deep architecture used in the proposed method for character recognition.

a similar location and dimension, to a large extent.

$$\mathrm{IoU}(X, Y) = \frac{\text{Area of overlap } (X, Y)}{\text{Area of union } (X, Y)}. \quad (2)$$

For the sake of better performance, other parameters such as confidence threshold and anchors are constant based on the default setting. There are special features that are extracted by the deep architecture of the proposed method. These features include maximum and minimum length and height, maximum and minimum number of pixels, and area. Spall plates with unusual resolution are ignored in the proposed architecture. Classification is done in the late stage of process. Those objects that have plate features are classified as plate and are ignored otherwise.

2.3. Character Recognition. Character detection and recognition is one of the most important steps in plate recognition. In this paper, a method based on convolutional and recurrent networks is used for plate detection. LSTM architecture is very efficient here due to the fact that characters in Iranian plates follow a specific pattern. This method is fast enough for real-time applications since plate images enter to the network, and in the end, characters are recognized.

Unlike classic methods, this method does not require separate stages of segmentation, feature extraction, and classification. Figure 5 depicts the proposed deep architecture for plate detection that contains convolutional and recurrent layers. The network configuration used in our experiments is summarized in Table 1. The main advantage of the proposed architecture is that it only requires plate images and their label sequences and there is no need for plate segmentation. Moreover, various features must be extracted from plate for classifying characters to 37 possible classes. Convolutional networks help to extract these discriminant features. Features extracted from plates using CNN enter LSTM networks. Using LTSM units in RNN is very useful since characters and digits have a special order in Iranian plates. In this method, features are extracted from plate image based on shape and sequence using convolutional layers. These features are used as the input of recurrent layers in the end part of network. These layers consider all the sequence record. LSTM units are used that contain memory cells and gates instead of RNN units, in order to avoid gradient vanishing. In this stage, characters are classified in 37 classes by the network. These classes include 10 digits (0 to 9) and 24 characters that are shown in Table 2. There is also a special character for disabled drivers. In this kind of plates, an image of a wheelchair replaces a Persian character. Moreover, there are two English characters, *S* and *D*, that are used for special purposes. It is worthy to mention that scarcity of plates with special characters prevents training networks with these characters. To address this issue, we used deep networks to produce plates with these special characters. Samples of these plates are shown in Figure 6. This paper also proposes a method based on generative adversarial networks (GAN) to produce plate images with different qualities. Existing data is used to train the network in the proposed method; then, this trained network is used to produce various plate images.

Table 1: Network configuration summary: "*k*", "*s*", and "*p*" stand for kernel size, stride, and padding size, respectively.

Type	Configuration
Input	$W \times 32$
Conv	#kernels: 64, k: 3×3, s: 1, p: 1
Max pooling	Windows: 2×2, s: 2
Conv	#kernels: 128, k: 2×2, s: 1, p: 1
Max pooling	Windows: 2×2, s: 2
Conv	#kernels: 256, k: 3×3, s: 1, p: 1
Conv	#kernels: 256, k: 3×3, s: 1, p: 1
Max pooling	Windows: 1×2, s: 2
Conv	#kernels: 512, k: 3×3, s: 1, p: 1
Batch normalization	
Conv	#kernels: 512, k: 3×3, s: 1, p: 1
Batch normalization	
Max pooling	Windows: 1×2, s: 2
Conv	#kernels: 512, k :2×2, s: 1, p: 0
Map to sequence	
Bidirectional LSTM	#hidden unit: 256
Bidirectional LSTM	#hidden unit: 256
Bidirectional LSTM	#hidden unit: 256
Transcription	

3. Experimental Results

Since the proposed method is based on deep learning, there is a need for a large volume of data that are labeled in various phases. To this end, data are gathered from many cameras in streets and highways during several days and nights under various illumination and weather conditions. There are over four million frames, and each of which contains several, one, or no vehicle.

Classic methods are used for labeling data in vehicle detection, plate detection, and plate recognition process. The proposed model is trained using these frames that are labeled for vehicle detection. Three million frames are used for plate detection and recognition. Images have various illumination, shadow, reflection, weather conditions, and qualities in order to better train the model. It is worthy to mention that all hyperparameters are chosen based on numerous random search tests. First, a set of hyperparameters was chosen and the model was trained using the training data. The set of hyperparameter was then evaluated based on the testing data. This process was repeated, and the hyperparameters with highest accuracy were chosen. All hyperparameters keep their values in different datasets.

In this step, the neural network presented in Section 2.1 is used in order to address the problem of scarcity of plates with special character; some of which are shown in Figure 6. Plates containing these characters are rare and belong to few organizations.

Figure 7 shows several plate samples in our database. It should be noted that background color of Iranian plates can be white, red, blue, yellow, or green. The advantages of deep learning-based methods over other methods such as color-based methods are that they are not dependent on color, have lower fault rate, and can be used under different illumination situation such as high, low, and uneven illumination. In the proposed method, vehicles are first detected using deep networks to expedite the plate detection process.

The output of the proposed method in the vehicle detection phase with unfixed shooting angles and different qualities in real scenes is shown in Figure 8. As it can be seen, the proposed method is capable of detecting vehicles under various illumination conditions and from front, back, and different angles. The proposed method is able to detect vehicles with high accuracy owing to extraction of high-level features for vehicle detection process. The advantages of this method is that there is no limitation in the number

Table 2: Different letters and digits in Iranian plates.

Type	Characters
1	ط ل م ژ ز و 0123456789
2	SD ♿ ی ه ن گ ک ق ع د ج
3	الف ف ص س ش ت ث پ ب

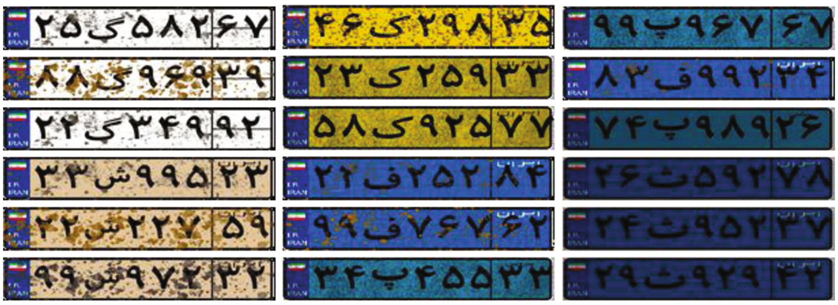


Figure 6: Samples of plates with special characters.

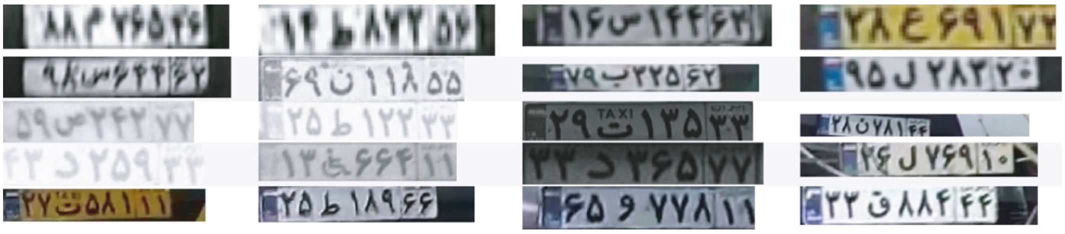


Figure 7: Samples of plates with special characters.

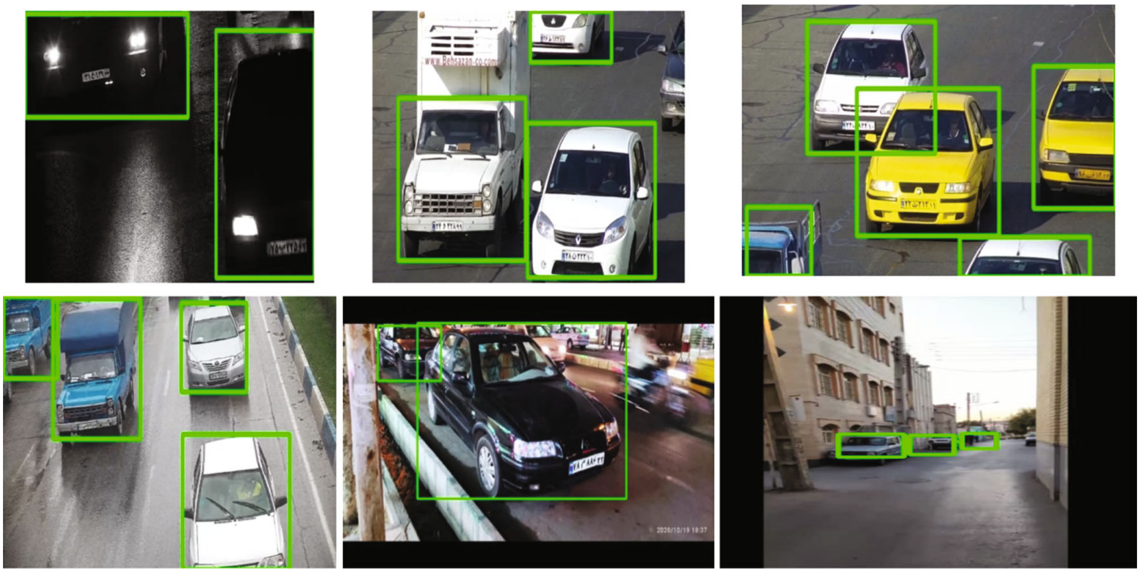


Figure 8: Output of the proposed method in the vehicle detection phase.

of vehicles and the vehicles can even occlude in the image. The proposed model is trained using images with different qualities, shooting angles, and illumination conditions in order to improve detection rate. Training images also include images taken during day and night and different weather conditions. This phase of the proposed method is

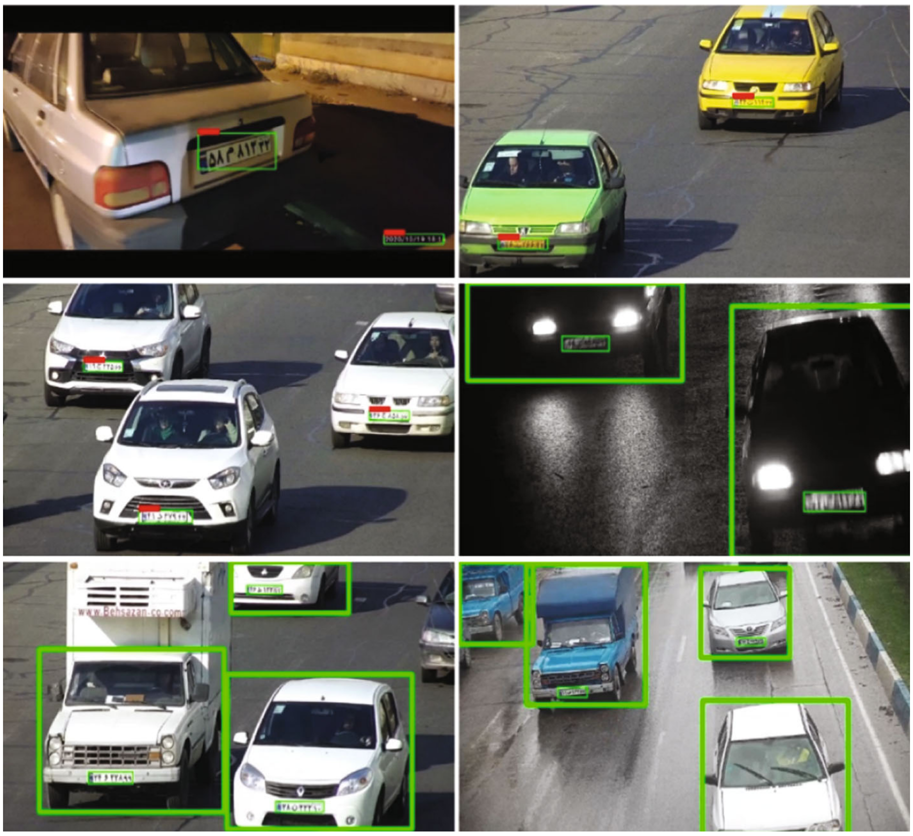


Figure 9: Output of the proposed method in the vehicle detection phase.

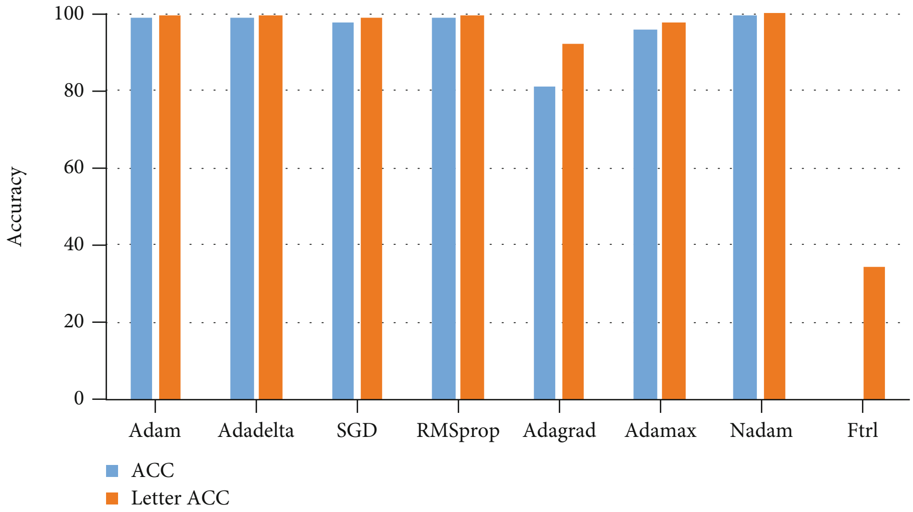


Figure 10: Detection rate of the proposed method with different optimizer functions.

based on labeled data, and the detection rate is 100%. This detection rate guarantees that no vehicle is missed by the proposed method.

The next step is plate detection where the detected vehicle image from the previous step is searched instead of the whole image. This considerably speeds up the plate detection process. Training the proposed deep network is done using three million images with labeled plates. Image processing methods are used for labeling images and detecting plates. There are images with different qualities and shooting angles in the training set in order to improve the detection rate in the proposed method. Images with different illumination condition are also present in the training set. In the proposed method, different plate features including maximum and minimum length and width and plate area are extracted based on the training dataset. Figure 9 depicts output of the proposed method for six images.

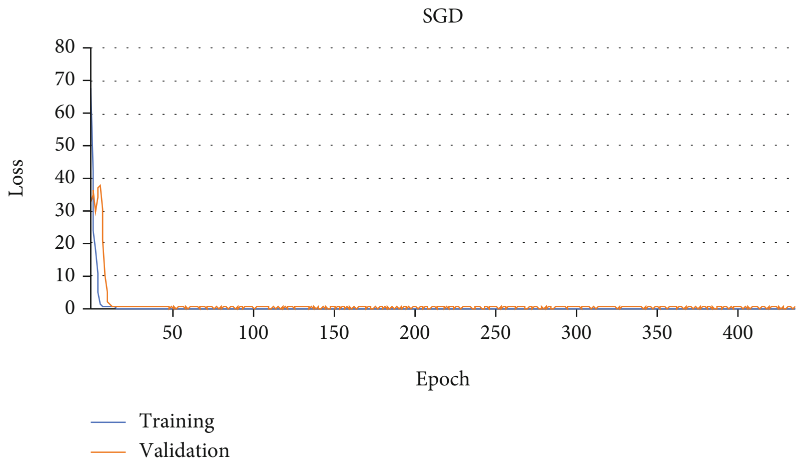


Figure 11: Loss diagram of the proposed method with SGD optimizer function.

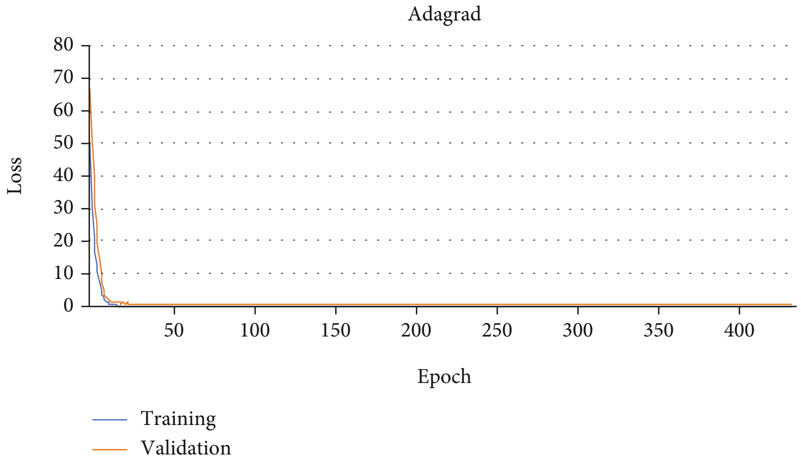


Figure 12: Loss diagram of the proposed method with Adagrad optimizer function.

Table 3: Comparison between the proposed method and other methods on different datasets.

Dataset	Plate detection Acc		Plate recognition Acc		Overall Acc	
	Reported Acc	Our method Acc	Reported Acc	Our method Acc	Reported Acc	Our method Acc
[10]	99%	100%	97%	100%	96%	100%
[17]	96%	100%	94%	100%	90%	100%
[34]	98%	100%	98%	100%	96%	100%

As it can be seen, the proposed method is able to detect plates in images with different shooting angles (including front and back), distances, and illumination conditions. Detection rate in this step is 100%. In the next step, the characters of the detected plates should be recognized. To this end, the convolutional and recurrent network mentioned in Section 2.3 needs to be trained using different plates. Then, the proposed model is evaluated based on the test data.

Since the proposed model is trained using different datasets, different features from plates are extracted in various layers. These features include edges, corners, and structures and are extracted based on CNN layers. Moreover, in LSTM layers, sequence features are extracted. This architecture helps to successfully extract discriminant features for character recognition.

The experiments on the test data show that the detection rate of all characters in the proposed method is 99.37%. Moreover, character detection rate is 99.92% that is calculated based on each character and not the whole plate. Choosing a proper optimizer function is a determining factor. The proposed model is tested with several optimizers.

The validation results using different optimizers are shown in Figure 10. As it can be seen, Adma-based optimizer functions outperform other functions and have a recognition rate of more than 99%. Ftrl optimizer does not

Table 4: Comparison between the proposed methods and the other methods.

Methods	Reported Acc			Overall Acc	Plate characters	Number of plates	Image size	Processing time	Method D: detection method R: recognition method
	Plate detection	Character segmentation	Plate recognition						
[5]	98.7%	100%	97.6%	96.33%	Persian	10000	Variable	180	D: CCA and RANSAC R: probabilistic SVM
[10]	99.33%	NR	96.6%	96%	Persian	150	640 × 480	NR	D: color features R: ANN
[14]	97.3%	NR	94.5%	91.94%	Persian	320	640 × 480	NR	D: edge features (Sobel) R: MLP
[17]	96.93%	98.75%	94.5%	90.45%	Persian	1185	1024 × 768	NR	D: morphological operations and Adaboost R: SAMME
[23]	97.7%	—	98.8%	NR	Chinese	250 K	96 × 32	NR	D: MTCNN R: MTLPR
[29]	100%	—	96.78%	NR	Chinese	—	300 × 300	—	D: CNN R: CNN
[35]	NR	99%	NR	NR	Korean	120	640 × 480	NR	D: sliding concentric windows R: ANN
[30]	100%	NR	98.4%	NR	Chinese	2507	1600 × 1200	42	YOLOv3 (352 × 288)
[30]	98.27%	NR	98.1%	NR	Chinese	2507	1600 × 1200	161	Faster_RCNN_ ResNet101 (800 × 600)
[27]	98.04%	NR	94.12%	NR	Chinese	2049	48 × 640	400	Unified deep neural network
[39]	97.16%	98.34%	97.88%	93.54%	English Japanese	9026	NR	288	D: improved Bernsen algorithm R: SVM
[39]	96.5%	NR	89.1%	86%	English	1334	NR	276	D: sliding concentric windows and CCA R: probabilistic NN
[40]	95.9%	NR	92.3%	90%	Chinese English	5026	720 × 576	125	D: edge features (Sobel) R: feed forward NN
[41]	97.3%	NR	95.7%	93.1%	English	1176	640 × 480	223	D: salient features R: self-defined classifier
[42]	97.1%	NR	96.4%	93.6%	English	332	867 × 623	594	D: color features and Hough transform R: feed forward NN
[43]	94.43%	—	99.37%		Arabic	600	380 × 540	NR	D: DSSN R: CNN
[44]	97.76%	—	95.05%	NR	Persian	5719	Variable	54.18	D: YOLO-v3 R: YOLO-v3
[45]	99.37%	—	99.53%	98.9%	Brazilian	—	1024 × 768	—	D: YOLO-v3 R: CNN
[46]	99.72%	—	87%	—	Jordanian	187200	1920 × 1080	—	D: YOLO-v3 R: CNN
[47]	98.22%	—	87%	—	English	2049	Variable	—	D: YOLO-v3 R: YOLO-v3
The proposed method	100%	—	99.37%	99.6%	Persian	3 million	Variable	46	D: CNN R: CNN+LSTM

properly fit in the proposed method and shows plate recognition rate of 0% and character recognition rate of 37%. Therefore, Adadelta optimizer function that is based on Adam is used in the proposed method. In order to examine the convergence of the proposed method, loss diagram of the proposed method is shown with two optimizer functions of SGD and Adagrad in Figures 11 and 12, respectively. As it can be seen, loss rates gradually converge to zero on the

Table 5: Average processing times (in millisecond) of the proposed method in different scenarios for different steps.

Total time	Recognition plate	Plate detection	Vehicle detection	Number of vehicle
8	0	3	5	0
16	3	4	9	1
25	6	5	14	2
34	9	7	17	3
41	12	8	21	4
48	15	9	24	5
56	18	10	27	6
61	21	11	29	7

training and testing dataset. It shows that the model parameters are well learned based on the input data and lead to loss rate reduction during repetitive process of learning. In Figures 11 and 12, the horizontal axis presents epoch in the evaluation steps in the training and validation sets, and the vertical axis presents amount of loss in these sets.

3.1. Comparison with Other Methods. Several datasets are used in order to compare the proposed method with other methods in terms of efficiency. Table 3 shows this comparison using several datasets containing images of Iranian vehicles in terms of efficiency in different steps. According to the results, the proposed method performs well for dirty and low-quality images. In this part, used datasets contain images of already cropped vehicles in order to evaluate the proposed method in the plate detection step. Plate detection rate and character recognition rate are 100% in the proposed method. The images in the used dataset are captured by handheld cameras and have good quality.

Table 4 shows the result of experiments with four criteria to evaluate the proposed method and compare it with other methods. These criteria are plate characters, number of plates, image size, and processing time. Plate detection and recognition process are considered in this set of experiments. The advantage of the proposed method over other methods, that do not use deep learning, is that it is segmentation-free. This improves recognition rate and also speeds up the plate detection and recognition process.

As it can be seen in Table 4, the proposed method shows a great performance in vehicle and plate detection, and its accuracy is 100%. It should be noted that the overall accuracy (Overall Acc) is the amount of accuracy when plate detection and character recognition are done simultaneously.

3.2. Real-Time Evaluation. This set of experiments focuses on applicability of the proposed method in real-time situations. Here, the input is a video with 30 frames per second. Plate detection is a time-consuming step in plate recognition process and it gets more complicated as the number of plates increases in the input image. In the proposed method, vehicles are first detected to speed up the plate detection process. In this way, exploration is done only in the detected vehicle image instead of the whole image. The character recognition process becomes slower as the number of vehicles in the image increases; therefore, inputs with different number of vehicles are used in order to demonstrate applicability of the proposed method. Table 5 shows average processing times for a 10-time execution of the proposed method in each step for different scenarios. The total processing time increases as the number of vehicles increases. As it can be seen, the proposed method is able to be used in real-time situation even with many vehicles in a single image.

4. Conclusion

Monitoring systems are necessary in highways and roads, especially due to the increasing number of vehicles and increasing traffic volume. Plates are used to recognize vehicles in videos and images captured by monitoring cameras, since plates are the unique identification of registered vehicles. In this paper, vehicle plate recognition contains three steps: vehicle detection, plate detection, and character recognition. The proposed method in this paper detects incoming vehicles in each frame that are inputs for the next step (plate detection and extraction). CNN is used for vehicles and plate detection, and deep model with CNN and RNN layer is used for character recognition.

The proposed method is evaluated using the frames of an input video. The experimental evaluations show that the proposed method is capable of performing all three steps of vehicle detection, plate detection, and character recognition efficiently with negligible error rates. Moreover, the proposed method is shown to be viable for real-time applications based on its response time. The proposed method in this paper can substantially improve plate recognition accuracy based on the obtained results.

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 that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Shan, 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, 2012.

[2] C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, “Vehicle detection and classification based on deep neural network for intelligent transportation applications,” in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1605–1608, Honolulu, HI, USA, 2018.

[3] 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)*, pp. 580–596, Munich, Germany, 2018.

[4] Y. Yuan, W. Zou, Y. Zhao, X. Wang, H. Xuefeng, 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.

[5] R. Panahi and I. Gholampour, “Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.

[6] M. Y. Arafat, A. S. M. Khairuddin, and R. Paramesran, “Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework,” *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 712–723, 2020.

[7] S. Yu, B. Li, Q. Zhang, C. Liu, and M. Q.-H. Meng, “A novel license plate location method based on wavelet transform and EMD analysis,” *Pattern Recognition*, vol. 48, no. 1, pp. 114–125, 2015.

[8] M. S. Al-Shemarry, Y. Li, and S. Abdulla, “Ensemble of Adaboost cascades of 3L-LBPs classifiers for license plates detection with low quality images,” *Expert Systems with Applications*, vol. 92, pp. 216–235, 2018.

[9] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, “Vertical-edge-based car-license-plate detection method,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 26–38, 2013.

[10] R. Azad, F. Davami, and B. Azad, “A novel and robust method for automatic license plate recognition system based on pattern recognition,” *Advances in Computer Science: an International Journal*, vol. 2, no. 3, pp. 64–70, 2013.

[11] A. Mousa, “Canny edge-detection based vehicle plate recognition,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 3, pp. 1–8, 2012.

[12] P. Dollár, T. Zhuowen, P. Perona, and S. Belongie, “Integral channel features,” in *Proceedings of the British Machine Vision Conference*, BMVC Press, London, 2009.

[13] B. Yang, J. Yan, Z. Lei, and S. Z. Li, “Aggregate channel features for multi-view face detection,” in *IEEE International Joint Conference on Biometrics*, pp. 1–8, FL, USA, 2014.

[14] S. Ghofrani and M. Rasooli, “Farsi license plate detection and recognition based on characters features,” *Majlesi Journal of Electrical Engineering*, vol. 5, no. 17, pp. 44–51, 2011.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[16] M. M. Dehshibi and R. Allahverdi, “Persian vehicle license plate recognition using multiclass Adaboost,” *International Journal of Computer and Electrical Engineering*, vol. 4, no. 3, pp. 355–358, 2012.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: speeded up robust features,” in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.

[19] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, “Automatic license plate recognition with convolutional neural networks trained on synthetic data,” in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, Luton, UK, 2017.

[20] V. H. Pham, P. Q. Dinh, and V. H. Nguyen, “CNN-based character recognition for license plate recognition system,” in *Asian Conference on Intelligent Information and Database Systems*, pp. 594–603, Springer, 2018.

[21] C. Henry, S. Y. Ahn, and S.-W. Lee, “Multinational license plate recognition using generalized character sequence detection,” *Access*, vol. 8, pp. 35185–35199, 2020.

[22] C. Gou, K. Wang, Y. Yao, and Z. Li, “Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.

[23] W. Wang, J. Yang, M. Chen, and P. Wang, “A light CNN for end-to-end car license plates detection and recognition,” *IEEE Access*, vol. 7, pp. 173875–173883, 2019.

[24] G. O. N. G. Wen-bin, S. H. I. Zhang-song, and J. I. Qiang, “Non-segmented Chinese license plate recognition algorithm based on deep neural networks,” in *2020 Chinese Control and Decision Conference (CCDC)*, pp. 66–71, Hefei, China, 2020.

[25] H. Li, P. Wang, M. You, and C. Shen, “Reading car license plates using deep neural networks,” *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.

[26] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, “Segmentation- and annotation-free license plate recognition with deep localization and failure identification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.

[27] H. Li, P. Wang, and C. Shen, “Toward end-to-end car license plate detection and recognition with deep neural networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.

[28] C. L. P. Chen and B. Wang, “Random-positioned license plate recognition using hybrid broad learning system and convolutional networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 444–456, 2020.

[29] S.-L. Chen, C. Yang, J.-W. Ma, F. Chen, and X.-C. Yin, “Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3686–3695, 2020.

[30] F. Gao, Y. Cai, Y. Ge, and S. Lu, “EDF-LPR: a new encoder–decoder framework for license plate recognition,” *IET Intelligent Transport Systems*, vol. 14, no. 8, pp. 959–969, 2020.

[31] W. Weihong and T. Jiaoyang, “Research on license plate recognition algorithms based on deep learning in complex environment,” *IEEE Access*, vol. 8, pp. 91661–91675, 2020.

[32] S. Zhang, G. Tang, Y. Liu, and H. Mao, “Robust license plate recognition with shared adversarial training network,” *IEEE Access*, vol. 8, pp. 697–705, 2020.

[33] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, 2015.

[34] M. Nejati, A. Majidi, and M. Jalalat, “License plate recognition based on edge histogram analysis and classifier ensemble,” in *2015 Signal Processing and Intelligent Systems Conference (SPIS)*, pp. 48–52, Tehran, Iran, 2015.

[35] K. Deb, V. V. Gubarev, and K.-H. Jo, “Vehicle license plate detection algorithm based on color space and geometrical properties,” in *International Conference on Intelligent Computing*, pp. 555–564, Springer, 2009.

[36] J. Pirgazi, A. G. Sorkhi, and M. M. P. Kallehbasti, “An efficient robust method for accurate and real-time vehicle plate recognition,” *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1759–1772, 2021.

[37] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2154, Columbus, OH, USA, 2014.

[38] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, “Scalable, high-quality object detection,” 2014, http://arxiv.org/abs/1412.1441.

[39] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, “An algorithm for license plate recognition applied to intelligent transportation system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.

[40] J. Jiao, Q. Ye, and Q. Huang, “A configurable method for multi-style license plate recognition,” *Pattern Recognition*, vol. 42, no. 3, pp. 358–369, 2009.

[41] Zhen-Xue Chen, Cheng-Yun Liu, Fa-Liang Chang, and Guo-You Wang, “Automatic license-plate location and recognition based on feature salience,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3781–3785, 2009.

[42] Jing-Ming Guo and Yun-Fu Liu, “License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1417–1424, 2008.

[43] N. Omar, A. Sengur, and S. G. S. Al-Ali, “Cascaded deep learning-based efficient approach for license plate detection and recognition,” *Expert Systems with Applications*, vol. 149, article 113280, 2020.

[44] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. Y. Suen, “A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems,” *IEEE Access*, vol. 8, pp. 201317–201330, 2020.

[45] D. M. Izidio, A. Ferreira, H. R. Medeiros, and E. N. Barros, “An embedded automatic license plate recognition system using deep learning,” *Design Automation for Embedded Systems*, vol. 24, pp. 23–43, 2020.

[46] S. Alghyaline, “Real-time Jordanian license plate recognition using deep learning,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, 2020.

[47] Hendry and R.-C. Chen, “Automatic license plate recognition via sliding-window darknet-YOLO deep learning,” *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1519198, 20 pages
https://doi.org/10.1155/2022/1519198

Research Article

Computational Intelligence and Metaheuristic Techniques for Brain Tumor Detection through IoMT-Enabled MRI Devices

Damandeep Kaur,[1] Surender Singh,[1] Wathiq Mansoor ,[2] Yogesh Kumar,[3] Sahil Verma ,[1] Sonali Dash,[4] and Apeksha Koul[5]

[1] *Department of Computer Science & Engineering, Chandigarh University, Gharuan, Mohali, India*
[2] *University of Dubai, Dubai, UAE*
[3] *Indus Institute of Technology & Engineering, Indus University, Rancharda, Via Thaltej, Ahmedabad 382115, India*
[4] *Department of Electronics and Communication Engineering, Raghu Institute of Technology (A), Visakhapatnam, 531162 Andhra Pradesh, India*
[5] *Department of Computer Science and Engineering, Punjabi University, Patiala, Punjab 147002, India*

Correspondence should be addressed to Sahil Verma; sahilverma@ieee.org

Received 21 October 2021; Accepted 16 December 2021; Published 18 January 2022

Academic Editor: Mohammad R Khosravi

The brain tumor is the 22nd most common cancer worldwide, with 1.8% of new cancers. It is likely the most severe ailment that necessitates early discovery and treatment, and it requires the competence of neurosubject-matter experts and radiologists. Because of their enormous increases in data search and extraction speed and accuracy, as well as individualized treatment suggestions, machine and deep learning techniques are being increasingly commonly applied throughout healthcare industries. The current study depicts the methodologies and procedures used to detect a tumor inside the brain utilizing machine and deep learning techniques. Initially, data were preprocessed using contrast limited adaptive histogram equalization. Then, features were extracted using principal component analysis and independent component analysis (ICA). Next, the image was smoothed using multiple optimization techniques such as firefly and cuckoo search, lion, and bat optimization. Finally, Naïve Bayes and recurrent neural networks were utilized to classify the improved results. According to the findings, the ICA with cuckoo search and Naïve Bayes has the best mean square error rate of 1.02. With 64.81% peak signal-to-noise and 98.61% accuracy, ICA with hybrid optimization and a recurrent neural network (RNN) proved to better than the other algorithms. Furthermore, a Smartphone application is designed to perform quick and decisive actions. It helps neurologists and patients identify the tumor from a brain image in the early stages.

1. Introduction

A tumor is a mass of tissue that forms as a result of an aggregation of irregular cells. Normally, our body's cells die and are replaced by new ones as we age. However, brain cancer and other cancers inevitably break this pattern. In reality, tumor cells expand even though our bodies do not need them, and they do not die like normal cells in the body. As a result, the tumor will continue to grow as cells are added to the mass. Tumors are divided into two types: cancerous and noncancerous [1]. A cancerous tumor can start in any part of the body, and it is formed when cancer cells form a lump or growth. It is grown into nearby tissues and spreads to the lymph nodes and different parts of the body via hemoglobin or the lymphatic system.

In contrast, noncancerous tumors do not spread to other parts of the body [2]. Noncancerous tumors do not reappear once they are eliminated and tend to have a regular and smooth shape, with a delimiting border called a capsule [3]. Benign, premalignant, and malignant are the three types of tumor. The benign and premalignant tumors are not cancerous, but premalignant tumors can become malignant. Such tumors are cancerous, and their cells can multiply and migrate to other areas of the body unless a doctor removes them. Likewise, when a brain tumor grows, it becomes malignant and harmful (dangerous) or remains noncancerous. The tumor causes the pressure inside the skull to expand, causing harm to the brain, which is hazardous [3, 4]. Although the mechanisms leading to the development of brain tumors are not always precise, some risk factors are being exposed to the Epstein–Barr virus, also called Human gammaherpesvirus 4, one of the most common viruses in humans. This virus can cause infectious mononucleosis and other illnesses. In addition, such a virus is being exposed to ionizing radiation. It has genetic syndromes such as neurofibromatosis, tuberous sclerosis, and von Hippel–Lindau disease. This rare, inherited disorder causes tumors and cysts to grow in certain parts of the body, brain, spinal cord, eyes, inner ear, etc.

Brain tumors are classified into two types: primary and secondary. A primary brain tumor has its origins in the brain [5, 6]. These tumors are not malignant, but their size and position might cause significant problems and death. They are often referred to as benign brain tumors. Several types of primary tumors include the following: chordomas, craniopharyngiomas, gangliocytomas, glomus jugulare tumors, meningiomas, pineocytomas, pituitary adenomas, and schwannomas [7]. As cancer cells migrate from another organ, such as the lung or breast, to the brain, they form a secondary brain tumor, which is generally referred to as a metastatic brain tumor [8]. Astrocytomas, ependymomas, glioblastomas, medulloblastomas, and oligodendrogliomas are different types of secondary tumors. Other forms of brain tumors include hemangioblastomas and rhabdoid tumors [9]. There are many methods for segmenting and detecting brain tumors. Brain tumor segmentation is aimed at distinguishing tumor tissues from normal cells and assessing the nature and extent of tumor regions, which include active tumor tissue, necrotic (dead) tissue, and edema (swelling near the tumor). This is achieved by contrasting abnormal areas with typical tissues [10].

Since they invade surrounding tissues, certain cancers, such as glioblastomas, are challenging to differentiate from normal tissues. As a solution, several picture modalities with contrasts are often used. Diverse pixel intensities, noisy/ill-defined boundaries, and irregular shapes with significant variability are all critical technological challenges in medical image segmentation. Furthermore, because information about the labels of nearby pixels is not included in the classification, segmentation decreases the method's performance [11]. Single-photon emission computed tomography (SPECT) scans, MRI scans, and biopsies are methodologies to detect brain tumors [4]. In medicine, MRI of the brain has gained a lot of importance, as it is an imaging modality that uses nonionizing radiation to generate useful diagnostic images.

Additionally, functional MRI detects variations in blood flow that indicate brain activity. FMRI generates pictures or brain maps of how the brain functions by configuring and operating an advanced MRI scanner so that increased blood flow to active regions of the brain is seen on the MRI scans [12]. Multimodal imaging often necessitates picture alignment since the specimen is typically physically transferred from one imaging equipment to another, or certain modifications in the optical path are required, which might alter the geometrical characteristics of images. With the pictures aligned, the various modalities can be utilized to improve observed object segmentation or better understand the specimen's varied characteristics [13]. A survey conducted by the National Cancer Institute showed that a 10% growth rate could be seen each year in cases related to brain cancer or tumors [14]. The flowchart of the steps followed in brain tumor detection and classification is shown in Figure 1.

The initial stage includes a collection of MRI-based brain image samples that are sent for tumor enhancement. Various filters are used to remove the noise, tumor segmentation using multiple methods such as Otsu and watershed, feature extraction and selection methods to improve classification efficiency using various techniques, and then fusion of all these features for classifying and detecting the tumor in the brain [15]. Different machine and deep learning techniques are capable of recognizing and detecting the tumor inside the brain image. Integrating the Naïve Bayes classifier and recurrent neural networks enhances the accuracy and speed of diagnosis and helps the medical sector have better health outcomes [14]. These techniques are used in diagnostic procedures, treatment protocol development, medication development, personalized medicine, and patient management and care [16]. New methods and technology, such as computational and automated pathology and molecular diagnostics, and many other algorithms, such as firefly and lion optimization, are finding their way into advanced clinical diagnostics, offering some exciting ways to integrate these approaches into healthcare [17].

The key contribution of this work is to detect the tumors inside the brain. Many researchers have already carried out a lot of work in this field using different machine and deep learning techniques, but they have failed due to low accuracy in detecting the tumor. Few studies have used peak signal-to-noise ratio and mean square error as evaluative parameters to see the detection accuracy of tumor inside the brain [18]. They have shown low peak signal-to-noise ratio and a high mean square error, which ultimately affected the accuracy rates of their techniques as a low peak signal-to-noise ratio indicates the lousy quality of image and a high mean square error value indicates the large set of errors. One problem with mean-squared error is that it depends strongly on the image intensity scaling [19]. Suppose a mean-squared error of 100.0 for an 8-bit image (with pixel values in the range 0-255) looks dreadful, but an MSE of 100.0 for a 10-bit image (pixel values in [0,1023]) is barely noticeable [20].

On the other hand, peak signal-to-noise ratio (PSNR) avoids this problem by scaling the MSE according to the image range. PSNR is a good measure for comparing restoration results for the same image so that one image with 20 dB PSNR may look much better than another image with 30 dB PSNR [21]. Hence, we proposed a methodology that used both machine and deep learning algorithms along with different optimization techniques to enhance the accuracy by improving and reducing the peak signal-to-noise ratio and mean square error, respectively. The proposed system also needs some improvement by labeling type of tumor inside the brain so that doctors can easily interpret which

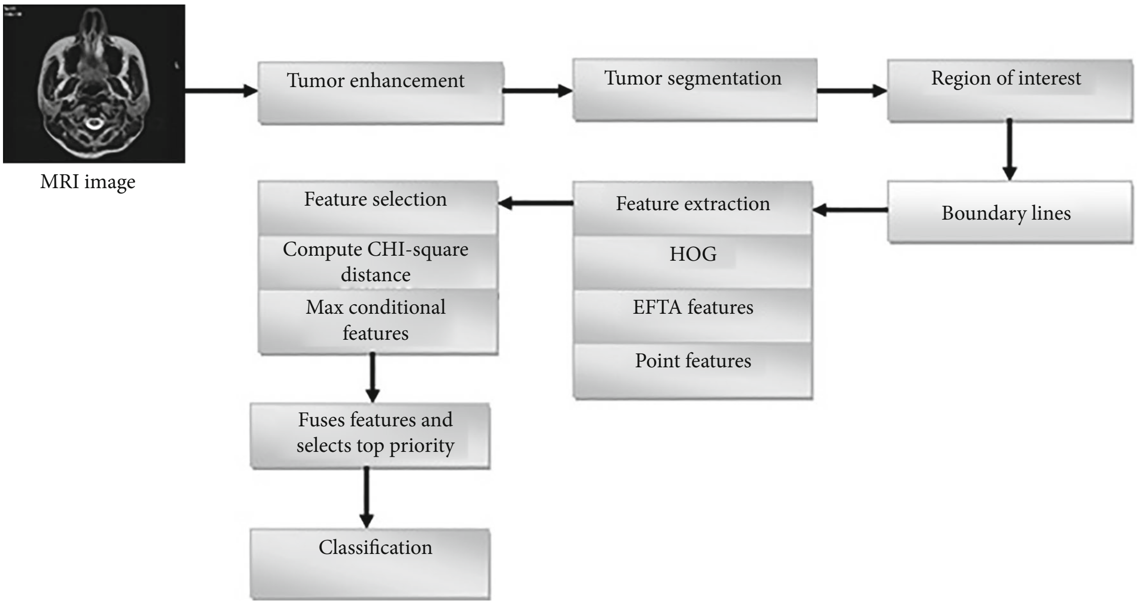


Figure 1: Detection and classification of a brain tumor.

type of brain tumor the patient has and start their medications as early as possible [22, 23].

To achieve the goal of recognizing a brain tumor with the best accuracy detection rate, we first preprocessed the picture with contrast limited adaptive histogram equalization (CLAHE) and threshold segmentation to increase picture visibility. Then, various machine and deep learning-based methods were employed to extract features from preprocessed images. Finally, we classified them using principal component analysis, independent component analysis, Naïve Bayes, and recurrent neural networks [24]. The collected results were eventually optimized using cuckoo search, firefly algorithm, lion optimization, and bat optimization techniques [25, 26].

Finally, based on optimal results, we evaluated the improved findings using assessment measures such as peak signal-to-noise ratio, mean square error, and detection accuracy to find the optimum method for discovering a brain tumor. The algorithm with the greatest PSNR, lowest MSE, and best accuracy rate would be chosen among the other algorithms. Furthermore, the contribution above aids us in providing a better understanding of the machine and deep learning in cancer diagnosis by imaging analysis [27].

Given the enormous number of patients determined to have a tumor and the critical measure of information created during tumor treatment, there is a particular interest in utilizing AI to improve oncologic consideration [20, 28]. This paper puts forward a methodical and experimental study on the machine and deep learning techniques and their utilization in different research areas. In this article, we have provided and implemented machine and deep learning-based algorithms to detect and classify brain tumors using various features.

The parts of this paper are summarized in the following order: related work is presented in Section 2, the methodology is presented in Section 3, experimental results and interpretation are presented in Section 4, prospective perspectives of the machine and deep learning in healthcare are presented in Section 5, and the conclusion is presented in Section 6.

Even with the help of IoMT-enabled MRI devices, we explored our research. The Internet of Medical Things (IoMT) combines medical devices and applications that use network technologies to connect to healthcare information technology systems. Benefits can be like unnecessary hospital visits, and the burden on healthcare systems can be reduced. Now patients can be connected to their physicians and allow medical data transfer over a secure network. IoMT devices are connected to different cloud platforms like Amazon Web Services, on which data is being gathered by IoMT devices can be analyzed and stored. But the amount of data handled by the Internet of Medical Things (IoMT) devices is increasing rapidly as sensitive information is being disclosed. So the privacy and security of the data gathered by IoMT devices being stored or transmitted through the cloud is a major concern these days. Therefore, IoMT can also be called healthcare IoT. The IoMT market consists of several smart devices, like medical/vital monitors and wearables, and it is strictly for healthcare use on the body, in the community, in-home, hospital, or clinic settings and associated telehealth, real-time location, and other services [29].

2. Related Work

Machine and deep learning have shown a fast change in the clinic by deciding the ideal methods of treatment, the necessary dosages, and the period of delivery during the patient's medication. A brain tumor is a dangerous condition that necessitates early detection and specific position techniques [30]. Therefore, machine learning and especially deep learning techniques have drawn considerable attention and sparked interest in recent years for their potential to improve our lives. The growing number of patients who are being identified with tumors and the ample amount of data gathered during the treatment process lead to the need for machine and deep learning to improve oncologic care [28, 31]. Hence, to have more information related to the role of the machine and deep learning in oncology, a section has been provided that presents the technique used by the researchers to detect brain tumor, and based on the gaps found in these algorithms, we have proposed a new methodology [32, 33].

In machine learning, brain tumor segmentation has been performed using a Weiner filter with different wavelet bands and statistical methods [34]. They analyzed the results based on pixel and feature accuracy. For pixel-based accuracy, they compared the foreground, background, error rate, and quality with ground truth annotation, whereas for feature-based accuracy, they extracted local binary patterns using Equations (1) and (2).

$$LBP = \sum_{i=0}^{P-1} s(n_i - G_c)2^i, \tag{1}$$

$$s(x) = \begin{cases} 1, \text{if } x > 0, \\ 0, \text{otherwise,} \end{cases} \tag{2}$$

where P is the number of neighborhood pixels, n_i represents the ith neighboring pixel, and c represents the center pixel.

Another technique to segment the brain was performed using model-based trainable segmentation [34] and a classification system to precisely identify the tumor's location in the MRI of brain tissue. Similarly, Nazir et al. [35] used wavelet subbands to segment brain images and achieved a classification rate of 99.7%. Wavelet transform-based local binary pattern variant features and antilion optimization were also used to develop computer-assisted brain tumor detection map [36]. Pushpa et al. [37] used preprocessing to filter as well as smoothening the image and carried out the segmentation process by using morphological operations to increase the precision in the classification phase. In addition to this, various phases had been involved in brain cancer recognition and categorization, such as preprocessing, cleavage, characteristics extraction, and classification of brain tumors, by utilizing the SVM algorithm [38]. Pugalenthi et al. [39] evaluated and classified the tumor regions into low/high grades based on the analysis carried out with the brain MRI slices. The machine learning technique implemented a sequence of procedures, such as preprocessing, postprocessing, and classification. At the other end, Manogaram et al. [40] suggested an improved orthogonal gamma distribution-based machine learning method for automatically detecting anomalies in the under- and oversegments of brain tumor areas. The experimental system showed the method of orthogonal gamma distribution using Equation (3).

$$f(x) = \begin{cases} \dfrac{x^{p-1}e^{-x}}{\tau_p} \, p > 0, 0 \leq x < \infty, \\ 0 \text{ otherwise,} \end{cases} \tag{3}$$

where p and x are continuous random variables, with the machine learning approach that achieved 99.55% in identifying a brain tumor. Morphological screening, clustering, and Naïve Bayes classifier (NBC) grouping were used to generate clusters of identical and dissimilar patches from the image, which were then classified using the Naïve Bayes classifier by calculating Equation (4).

$$P(H_i \mid D) = \frac{P(H_i)\,P(D \mid H_i)}{P(D)}, \tag{4}$$

where $P(H_i \mid D)$ is the posterior probability, $P(D \mid H_i)$ is the likelihood, $P(H_i)$ is the class prior probability, and $P(D)$ is the detector prior probability for spotting the tumor portion in the brain from magnetic resonance imaging.

Machine learning algorithms have also been applied to detect the edges of a brain tumor from patients' MRI scan of brain images by using some noise removal functions and features of medical images for the diagnosis using balance contrast enhancement techniques. Bahadure et al. [41] improved the performance and reduced the complexity involved in the medical image segmentation process using Berkley wavelet transformation. The authors used the method to extract relevant features from each segmented tissue to improve the accuracy and quality rate of the support vector machine- (SVM-) based classifier. Random Forest and K–nearest neighbor classifier had been used to calculate the area of the tumor region and classified it as benign or malignant using Equation (5).

$$|\tau| = \frac{1}{\sqrt{N}} \sum_p \left| v_1^p \cdots v_n^p, c^p \right|. \tag{5}$$

After transforming it into the quantum state and merging it with the hamming distance, they achieved Equation (6).

$$|\varnothing_n| = \tau \, |\varnothing_{n-1}| = \alpha \sum_{p \in \cap} \left| d_1^p \cdots .d_n^p \, ; v_1^p \cdots .v_n^p, c^p \, ; 1 \right|, \tag{6}$$

where τ is the training set, N refers to the total observation, v^p refers to feature vectors where $(p = 1 \cdots N)$, and c^p is the corresponding class, based on the majority voting method. The main drawback of support vector machine and KNN classifier is that they cannot be used for large datasets. However, despite flaws, these classifiers are also good in high-dimensional space, simple to understand, and used for classification and regression.

On the other hand, deep learning algorithms also classified brain tumor using deep convolution neural network and achieved an accuracy of 98%. Transfer learning approach has also been used to classify brain tumors in magnetic resonance images. Various networks such as VGG16, VGG19, ResNet50, and DenseNet21 obtained the highest classification score of 99.02% using Adadelta. Deep and Emmanuel [11] worked on a fused feature adaptive firefly back propagation neural network for classification, including preprocessing, feature extraction, selection, and combination, and achieved high arrangement precision. Deep convolution neural network has also been used to develop a fully automated brain tumor segmentation and classification model which analyzed MRI images of three different types of tumors in sagittal, coronal, and axial views: meningioma, glioma, and pituitary tumors. It distinguished the brain tumor using Equation (7).

$$(f * g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau).g(x-\tau)d\tau, \qquad (7)$$

where $(f * g)(t)$ are functions that are being convoluted, t is the real number variable of functions f and g, $g(\tau)$ is the convolution of the time function, and τ' is the first derivative of the tau function. The precision of the softmax fully connected layer was utilized to classify the pictures and achieved 98.67% accuracy.

Aorora et al. [42] proposed a method to segment and classify the MRI of the brain as normal or abnormal using the Bhattachraya coefficient and achieved 98.01% accuracy whereas Suneetha and Rani [43] proposed optimized kernel probabilistic C-means algorithm, to identify brain tumors. They also used an adaptive double-window modified mean filter to enlarge the preprocessed image and recurrent neural networks to split the image inputs to recognize the tumor in the MR image and separate the tumor area from the picture.

Recurrent neural network [13] has categorized images to localize the region of the tumor of interest and classified them into four categories using a deep neural network classifier: mild, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma tumors. The classifier was paired with the discrete wavelet transform (DWT) and PCA [44]. Sajid et al. [45] proposed a mixed convolutional neural network for brain tumor segmentation and used various MRI modalities. The proposed procedure by the authors has validated on the BRATS 2013 dataset, yielding scores of 0.86, 0.86, and 0.91 in terms of dice score, sensitivity, and specificity, respectively, for the entire tumor area. Tables 1 and 2 represent the datasets, techniques, and limitations of the work proposed by various researchers.

3. Methodology

This section of the article discusses the approach followed for the detection of brain tumor using various machine and deep learning-based models. Figure 2 shows the steps followed for the process;

The following are the steps followed for the implementation.

The dataset was collected from The Cancer Imaging Archive (TCIA) which is an open-access database and hosts a large archive of medical images of tumor in DICOM format [47].

(i) The dataset included 20 patients' brain MRIs, 40 investigations, and 8798 pictures. It included two MRI tests for each patient, and information on the patient's clinical performance, imaging results, and therapy or intervention changes

(ii) After obtaining the preprocessed image, feature extraction was performed using principal component analysis. It improves visualization, reduces overfitting, and reduces the dimensionality without losing information from any features. Likewise, independent component analysis was used to get rid of unnecessary and redundant data

(iii) The extracted features were later optimized using cuckoo search, lion, and bat optimization (Section 3.3) to smoothen the image in order to provide a high classification rate for the high region of interest in the detection scenario

(iv) After that, the data was divided into two portions in a 7 : 3 ratio: training and testing sets. The training data was utilized to create machine and deep learning models that were supplied with 11 distinct characteristics. Because we may make as many as we want without impacting the train to test split ratio, the dataset is split in this way. K fold cross-validation, on the other hand, is confined to a small data sample

(v) Finally, the data were sent to classification models such as NBC and RNN, with the results being further incorporated with the testing data

(vi) At this point, the model was trained, and with the help of testing data, the results were obtained

(vii) Metrics such as peak signal-to-noise ratio and mean square error rate were used to measure the quality between the original and compressed images, and detection accuracy was used to analyze the performance (Section 3.6). The compressed image will minimize the size without degrading its quality, and it will be easier for the model to detect brain tumors. These metrics were also used to compare the results obtained from the algorithms to show the best technique

The algorithms that were used to evaluate the performance of the PCA+firefly+NBC, ICA+cuckoo+NBC, and ICA+hybrid optimization+RNN to detect the brain tumor are defined below.

3.1. Preprocessing. Preprocessing optimized the picture data by suppressing unintentional distortions, thus enhancing the image quality for subsequent processing. Contrast restricted adaptive histogram equalization or CLAHE is a technique to improve the visibility level of foggy image. It is distinguished from average histogram equalization. The adaptive process computes several histograms, each equivalent to a particular portion of the region, and uses them to regroup the appearance's lightness principles. As a result, CLAHE was used extensively in the proposed work to promote local dissimilarity and refine representations of the boundaries of each appearance area. It reconstructs the picture by transforming each unit with a translation function obtained of adjacent regions. To begin with, the mean and

Table 1: The related work of the machine learning approaches for brain tumor.

Authors	Dataset	Access	Technique	Limitations
Manogaran, G. et al. [40]	MRI dataset	Open	Thresholding, gamma distribution	The proposed work needed to accelerate real-time medical applications and computation time.
Patil, D et al. [36]	BRATS 2015	Open	Local binary pattern, empirical wavelet transform, dynamic fuzzy histogram equalization	The work lacked the accuracy of interpretation.
Arun, N et al. [34]	MRI dataset	Open	Artificial neural network, machine learning	The technique affected the feature extraction because of difficulty in segmentation.
Nazir, M et al. [35]	Harvard dataset	Open	K mean clustering, discrete cosine transform	The proposed study is incapable of correctly classifying malignant brain tissues.
Garg, G et al. [46]	MRI dataset	Open	Principal component analysis, gray level cooccurrence matrix, stationary wavelet transform, Otsu's threshold	The technique needed large dataset for training, had high time complexity, did not work for small dataset and required expensive GPUs which ultimately increased cost to the users.
Kumar, M et al. [21]	MRI dataset	Open	Balance contrast enhancement technique, canny operator, fuzzy c-means	The technique needed to be performed on real medical images of patients in order to solve urgent diagnostic problems of patients.
Bahadure, N et al. [41]	MRI dataset	Open	Berkeley wavelet transformation, support vector machine	The methodology required the combination of more than one classifier and feature selection techniques to improve the accuracy.

Table 2: The related work of the deep learning approaches for brain tumor.

Authors	Dataset	Access	Techniques	Limitation
Jasm, D et al. [7]	MRI dataset	Open	Image mining techniques, neural network	The techniques did not work for video database.
Polat, O et al. [9]	T1-weighted MRI images	Open	VGG16, VGG19, ResNet50, DenseNet21, Adadelta optimizer	The training process of the VGG net was very slow and also had complex architecture.
Deep, A. et al. [11]	BRATS dataset	Open	Neural network, adaptive firefly, kernel principal component analysis	The techniques dealt with computational complexity, time complexity, and feature selection complexity.
Mohsen, H et al. [10]	MRI dataset	Open	DWT, deep neural network, CNN	Training of neural network had been time-consuming as it needed a large-size dataset for training purpose.
Rani, P et al. [13]	Figshare dataset	Open	Deep neural network, R-CNN, ChanVese algorithm	The algorithm segmented the unwanted regions in the brain MRI.
Pernas, F et al. [20]	MRI dataset	Open	Deep convolution neural network, sliding window technique	The proposed work needed prior information about the images and took high computational time.

standard deviation were determined using Equations (8) and (9), respectively [20, 31].

$$m(x, y) = \frac{1}{n\,x\,n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y), \tag{8}$$

where x and y are the points and n is the total observation.

$$\sigma = \sqrt{\frac{1}{n\,x\,n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} (f(x, y) - m(x, y))^2}, \tag{9}$$

where σ stands for standard deviation [48], for segmenting the image, and threshold segmentation, which separates an object from its background, was used. Thresholding is an image segmentation technique in which the pixels of an image are changed to make the image more straightforward to interpret. Thresholding is transforming a color or grayscale image into a binary image, which is essentially black and white. This technique can be expressed as given in Equation (10).

$$T = T\,[x, y, p(x, y)], \tag{10}$$

where T is the threshold value, x and y are the coordinates of the threshold value point, and $p(x, y)$ points are the gray-level image pixels [49]. If $g(x, y)$ is a threshold version of $p(x, y)$ at some global threshold T as shown in Equation (11),

$$g(x, y) = \left\{ \begin{matrix} 1 \text{ if } p(x, y) > T \\ 0 \text{ if } p(x, y) \le 0 \end{matrix} \right\}. \tag{11}$$

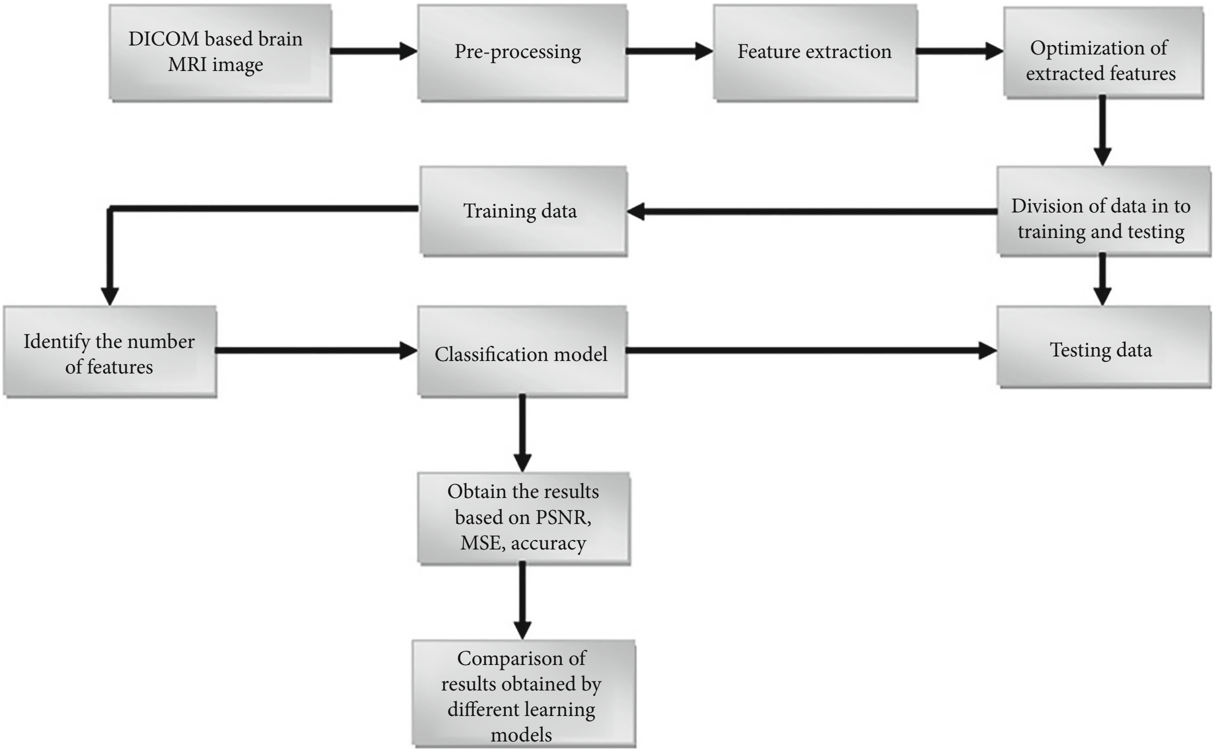

DICOM based brain MRI image
Pre-processing
Feature extraction
Optimization of extracted features
Training data
Division of data in to training and testing
Identify the number of features
Classification model
Testing data
Obtain the results based on PSNR, MSE, accuracy
Comparison of results obtained by different learning models

Figure 2: Proposed system designed for brain tumor detection.

3.2. Feature Extraction. Feature extraction refers to methods that pick and/or merge variables to form functions, thus reducing the amount of data that must be processed while wholly and correctly representing the original data collected. The techniques that were used to remove functionality from the preprocessed images are described below:

3.2.1. Principal Component Analysis. The principal component analysis is a factor analysis method used in image processing to isolate features and apply the characteristics resulting from reducing n-dimensional space. The measures for implementing the PCA method are outlined in Equations (12)–(16). The whole dataset is separated into X and Y. The validation set is handled by Y, while the training set is dealt with by X. The data are arranged into the independent variable's two dimensions. The average of the column is subtracted from each record for each situation [50]. Data standardization is a priority and is calculated using Equation (12).

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}. \tag{12}$$

The covariance of the matrix is calculated using Equation (13).

$$\text{cov}\ (X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \overline{y}). \tag{13}$$

The eigenvalues and eigenvectors on the diagonal of the matrix are evaluated using Equation (14).

$$Av = \lambda v, \tag{14}$$

with eigenvalues on the diagonal and zero values on all other locations. In this equation, any value of λ for which this equation has a solution is known as an eigenvalue of the matrix **A**. Then, the eigenvalue associated with eigenvector v is calculated using Equation (15)

$$\det\ (A - \lambda I) = 0. \tag{15}$$

The eigenvector is used as the feature vector until all of the feature values have been arranged. Preprocessing was used first in the suggested approach, followed by PCA to achieve efficient feature extraction from the feature vector found in the mat register. It was calculated by multiplying the transpose of the original dataset by the transpose of the feature vector using Equation (16).

$$\text{FinalDataset} = \text{FeatureVector}^{T} = \text{StandarizedOriginalDataset}^{T.} \tag{16}$$

3.2.2. Independent Component Analysis. Another algorithm that was used is independent component analysis, which is used for computing the unknown values of random variables. Independent component analysis was created with multivariate data in mind. Principal component analysis and independent component analysis are related in several ways. Data for independent component analysis research may come from several sources, including finance, digital images, paper databases [51]. It breaks down a large dataset into smaller chunks. It initially transforms data to zero-average and then chooses the number of components. Further, it whitens the data to transform the measured patterns x to have a unit variance. Then, on this basis, the random matrix was chosen to implement the orthogonal matrix. Finally, the convergence was carried out, and then, the cycle was repeated. In the ICA model, we used the statistical "latent variables" system and the random variable s_k instead of the time signal by computing the Equations (17) and (18).

$$x_i = a_{j1s1} + a_{j2s2} + .. + a_{jnsn}, \text{for all } j, \tag{17}$$

$$x = As. \tag{18}$$

The latent variables in the ICs remain unclear. A is also undefined in the mixing matrix. As a result, the only observable random vector x is used to estimate A and s, assuming that the number of ICs equals the number of measurable mixtures and that A is square and invertible. Then, after estimating the matrix A, we can compute its inverse, say W, i.e., $W = A^{-1}$, and obtain the independent component simply by

$$S = Wx = A^{-1}x. \tag{19}$$

Hence, it can be said that principal component analysis and independent component analysis have been used to reduce dimensions by creating new uncorrelated variables that maximize the variance and to reveal hidden factors by using non-Gaussian signals.

3.3. Feature Optimization.

3.3. Feature Optimization. Feature optimization is a process in which the number of input variables is reduced so that the computational cost of modeling can be minimized, which will improve the model's performance. The techniques used for feature optimization are briefly explained in detail.

3.3.1. Firefly Optimization Technique. The firefly approach is used to achieve improvements to reduce the function vector dependent on feature vector extraction [42]. Initially, we needed to initialize the objective function using Equation (20).

$$I(r) = \frac{I_s}{r^2}, \tag{20}$$

where $I(r)$ is the amplitude at the source and r is the observer's distance from the source. The light intensity $I(r)$ in the above equation differs according to the square law. Then, we created the initial population of fireflies using Equation (21).

$$x_{t+1} = x_t + \beta_0 e^{-\gamma r^2} + \alpha\varepsilon. \tag{21}$$

The first term in the above equation defines the attractiveness for each x particle, the second term is due to attraction, and the third term is randomization. Further, the light intensity of each of the fireflies was determined to find out the brightness of every firefly by computing Equation (22).

$$I = I_0 e^{-\gamma r^2}, \tag{22}$$

where I is the light intensity. Next, we needed to calculate the attractiveness of the fireflies using Equation (23).

$$\beta = \beta_0 e^{-\gamma r^2}. \tag{23}$$

Then, the movement of firefly i, which is to be attracted to another more attractive firefly j, was determined by applying Equation (24).

$$x_i = x_i + \beta_0 e^{-\gamma r_{i,j}} \left(x_j - x_i\right) + \alpha\varepsilon. \tag{24}$$

Finally, the light intensities of the fireflies were updated and ranked. After ranking the fireflies, the current best solution was selected.

3.3.2. Cuckoo Search. Another algorithm that was used is the cuckoo search, which deals with an optimization algorithm inspired by multiple cuckoo species' brood parasitism, which involves their spawn lying in the shells of other species of birds. Any bird can be engaged in direct combat with intervening cuckoos. Female cuckoos of some populations have evolved to the point that they are now specialized in imitating standards and decorating the spawns of a few chosen host species. Cuckoo search is flawless, similar to the cuckoo's breeding behavior, and can be used to solve a variety of optimization problems. It is based on the idealized rules that each cuckoo lays one egg at a time and deposits it in a nest selected at random [52].

3.3.3. Lion Optimization. The third algorithm used was lion optimization. Lions are the most socially persuaded of all wild species, with high levels of cooperation and antipathy. Lions are a particular focus due to their long-term erotic dimorphism in both community behavior and presence. The lion belongs to the wild felines, having two kinds of social body—residents and migrants. Residents always act in groups which are called prides. The finest clarification in approved iterations for every lion, i.e., the greatest visited location, is obtained and updated gradually throughout the optimization procedure. A pride ground is a zone that contains each associate's stay position. In every pride, designated females aimlessly go stalking. Hunters move near the prey to enclose and clasp it. The rest of the females change towards dissimilar positions in the terrain. Male lions, in arrogance, wander in the area. Females mate with some resident male lions. New males are accepted by their parental pride and develop into nomads when they reach maturity, but their power is less than that of local males. The algorithm performs the initialization of the random populations and the initialization of the probes and lions. Then, each lion particle chooses a random female lion for hunting, and each female lion chooses the best position in the pride. In contrast, the weakest lion in the pride is eliminated from the population and becomes the nomad. Then, for each pride, the immigration rate is evaluated. Finally, the fitness function is considered to select the best females and fill the empty places of the female lions that migrated from the territory

[53]. In this algorithm, every single solution is called "lion." In a N_{var} dimensional optimization problem, a lion is represented as follows in Equation (25).

$$\text{Lion} = \left[x_1, x_2, x_3, \cdots .., x_{N_{\text{var}}}\right]. \tag{25}$$

Cost (fitness value) of each lion is computed by evaluating the cost function, as shown in Equation (26).

$$\text{Fitness value of lion} = f(\text{lion}) = f\left(x_1, x_2, x_3, \cdots ., x_{N_{\text{var}}}\right). \tag{26}$$

The group with the most significant cumulative member penalties is the center, while the other two groups are referred to as the wings. The center of attention for hunters is fake prey (prey). As per Equation (27),

$$\text{Prey} = \sum \text{hunters}\left(x_1, x_2, x_3, \cdots ., x_{N_{\text{var}}}\right) \text{ number of hunters}. \tag{27}$$

If a hunter improves his or her finesses during hunting, prey will flee from hunter and a new position of prey will be achieved as follows in Equation (28).

$$\text{Prey}' = \text{prey} + \text{rand } (0, 1)\, X \,\text{PI}\, X \,(\text{prey} - \text{hunter}), \tag{28}$$

where prey is the current position of prey, hunter is a new position hunter who attacks prey, and PI is the percentage of improvement in fitness of hunter.

3.3.4. Bat Optimization. The final optimization algorithm used was the bat algorithm. The following summarizes the idealization of micro bat echolocation. Each virtual bat flies in a unique direction and speed, with a unique pitch, wavelength, and volume. As it searches for and locates prey, it alters the frequency, loudness, and rate of pulse emission [54]. It is computed by using Equations (29)–(31).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \tag{29}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x^*\right) f_i, \tag{30}$$

$$x_i^t = x_i^{t-1} + v_i^t, \tag{31}$$

where $\beta \in (0, 1)$ is a random vector drawn from uniform distribution and x^* is the current global best solution. A local random walk amplifies the search. Before those criteria are met, the best candidates are selected. The equilibrium between creativity and exploitation can be influenced by adjusting algorithm-dependent parameters in the bat algorithm, using a frequency-tuning technique to manipulate the complex behavior of a swarm of bats.

3.4. Classification. Classification is the method of organizing data into homogeneous categories or classes based on certain common characteristics present in the features. The techniques used for the classification are Naïve Bayes. It easily and quickly predicts the class of the test dataset and recurrent neural network as it remembers each piece of information and is useful in predicting the existence of tumor inside the brain.

3.4.1. Naïve Bayes. The proposed study discussed NBC in the context of supervised learning and achieved high classifying rates to identify pixel units in a testing image in tumor detection [55]. Naïve Bayes shows the probability field that relies on Bayes theorem and is calculated using Equation (32).

$$P(H_i \mid D) = \frac{P(H_i)\, P(D \mid H_i)}{P(D)}, \tag{32}$$

where $P(H_i \mid D)$ is the posterior probability, $P(D \mid H_i)$ is the likelihood, $P(H_i)$ is the class prior probability, and $P(D)$ is the detector prior probability. These classifiers are simple to use, requiring only complete linear parameters in the number of variables, which is a learning system challenge. Naïve Bayes is concerned with the maximum probability that can be achieved by calculating a closed-form appearance that is often involved with the linear method rather than with exclusive iterative estimation as is used with other forms of classification methods.

3.4.2. Recurrent Neural Network. An RNN is a class of ANN where relationships between nodes form a temporal sequence and show temporal dynamic behavior towards data. RNNs are derived classes of neural networks in a direct form and use memory, an internal state of the network to practice variable-length classifications of inputs. The main and most important feature of RNNs is the hidden state, which remembers some information about a sequence. This applies to various processes such as segmentation, image recognition, or speech recognition [56]. We used a recurrent neural network to implement the approach since it remembers all of the information about the calculations. Moreover, it employs the same settings for each input since it produces the same outcome by performing the same job on all inputs or hidden layers. Unlike other neural networks, this decreases the complexity of the parameters. The formula is shown in Equation (33).

$$h(n) = f(h(n-1), x(n); \theta), \tag{33}$$

where $h(n)$ is the current hidden state, $h(n-1)$ is the previous hidden state, $x(n)$ is the current input, and Θ is the parameters of function f. In an RNN, there are two broad sessions of networks with a similar general structure: one for finite instinct and the other for infinite instinct. Both groups demonstrate temporal behavior in a complex manner.

An RNN model is comparable to a convolutional neural network (CNN) or another form of artificial neural network in terms of architecture [11]. To simplify this, a recurrent neural network consists of three layers: an input layer, a hidden layer, and an output layer. However, these layers operate in a detectable order as shown in Figure 3.

Each node in the neural network is connected to each other via weights. These assigned weights are the parameters that have been used in recurrent neural network. These weights are learned when we train the network such that it

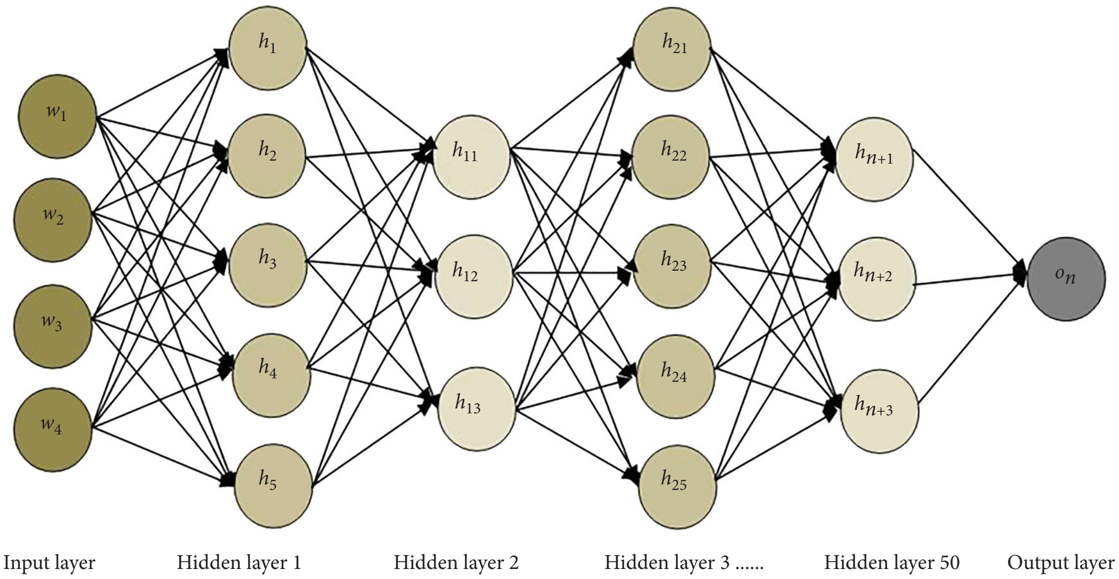


Figure 3: Structure of recurrent neural network.

can tune these parameters. To minimize the error by updating the values of weight, we have used learning rate which behaves as a hyperparameter.

While training the network, input layer is responsible for retrieving the data, which are then preprocessed before being passed to the hidden layer. This layer comprises neural networks, algorithms, and activation functions. A hidden layer of ResNet50 was utilized to extract relevant information from the data along with sigmoid activation function using Equation (34).

$$\sigma = \frac{1}{1+e^{-x}} \text{ where } x \in \sum_{j=1}^{n} I_j W_j. \tag{34}$$

Finally, whether the tumor is present inside the brain is sent to the output layer, which produces the detected result. The RNN procedure is quite variable. The data that traveled through the architecture are looped. For decision-making, each input is dependent on the prior one. Each layer in the network is assigned the same weight and bias by the RNN. As a result, all independent variables become dependent variables.

During back propagation, the parameters were optimized using stochastic gradient descent, which calculated the derivative of the cost function, i.e., $J(h)$: $R^N \longrightarrow R$ with respect to the parameters of the network such as weight of the input brain image (W) and its hidden weight matrix (H), i.e., $\partial J/\partial W$ and $\partial J/\partial H$.

The loop in the RNN ensures the information is preserved in its memory. This is possible by none other than its primary component, which is long short-term memory (LSTM). LTSMs are used to classify, identify, or detect output data based on a series of discrete-time input data. They use gradient descent and back propagation algorithms to minimize error. Such measured states are described as gated states or gated memory called LSTM networks. The equations for the gates in LSTM are shown in Equations (35)–(37).

$$i_n = \sigma(w_i\,[h_{n-1}, x_n] + b_i), \tag{35}$$

$$f_n = \sigma\left(w_f\,[h_{n-1}, x_n] + b_f\right), \tag{36}$$

$$o_n = \sigma(w_o\,[h_{n-1}, x_n] + b_o), \tag{37}$$

where i_n represents the input gate, f_n represents the forget gate, o_n represents the output gate, σ represents the sigmoid function, w_n is the weight for the respective gate neurons, h_{n-1} is the output for the previous LSTM block, x_n is the input at the current timestamp, and b_x stands for the biases for respective gates [30].

3.5. Experimental Setup. The following section provides the details of the dataset used during implementation, parameters used for evaluation, experimental results, and are compared to state-of-the-art strategies.

3.5.1. Dataset Used. This part of the paper discusses the dataset considered for detecting brain tumors using traditional machine and deep learning methods. The dataset under consideration was obtained from The Cancer Imaging Archive (TCIA), an open-access database that houses a vast archive of diagnostic photographs of tumors [57]. TCIA's primary file format is DICOM, as the bulk of the files in the database consist of DICOM-formatted CT, MRI, and nuclear medicine images. DICOM has superior image quality, supports all 65536 shades of color, and has over 2000 attributes. DICOM (.dcm) image files contain patient data such as name, identification number, gender, date of birth, device settings, and image characteristics such as modality, height, bit depth, and proportions [3]. The DICOM header object is defined as an image pixel, plane, an MR/CT image, and patient details [57]. The TCIA dataset is aimed at gaining access to machine and deep learning techniques for detecting tumors within the brain.

Machine and deep learning approaches have proven their robustness around the board. Therefore, in order to assist medical practitioners, we agreed to identify brain tumors using machine and deep learning algorithms. Table 3 outlines the dataset's 18 attributes, and Table 4 defines the age and gender-wise statistical analysis of brain tumor.

The most important prognostic detector in all brain tumor groups is age, as is seen in the features of patients with brain tumors. Metastatic brain tumors become more common as people become older. According to studies, brain tumors affect people of all ages, although they are most common in two age groups: children under 15 and adults 65 and older. The second factor is gender, with men having a higher risk of developing a brain tumor than women. Meningioma, for example, is more prevalent in women than in men. Attributes such as image position, pixel spacing, image orientation, slice thickness, and image type in patients are essential, as if a tumor in the brain is medically suspected; then, the location, size, shape, type, and impact of the tumor in the surrounding areas are evaluated by radiological methods.

Further, the best therapy, surgery, radiation, or chemotherapy is decided based on the obtained results. The (time of echo) difference between the delivery of an RF pulse and the reception of an echo signal is denoted by TE. The repetition time (TR) is the time it takes for two continuous pulses to be emitted in the same order. The trigger time refers to how and when the brain tumor is triggered [58]. Deranged blood vessels are common in cancers and tumors, and these can be seen in dye-enhanced MRI images. Magnetic resonance angiograms or MRAs are a common term for this form of imaging. The most popular form of MRI is one that uses gadolinium. In around one out of every three MRI scans, a contrast bolus agent is used to increase the scan's diagnostic precision [46]. The visibility of inflammation, tumors, blood vessels, and the blood supply of some organs is enhanced by adding contrast to the picture. Features such as size, type, and shape of the tumor are extracted, and the values in the form of pixel spacing, direction, and orientation of the image are used to achieve PSNR, MSE, and detection precision, which are then optimized to realize the best result for brain tumor detection [37].

3.5.2. Simulation Environment. MATLAB 2021a, which allows a screen reader to interact with the command window and build scripts and functions [36], was utilized to apply the proposed approach for detecting the detection of a brain tumor. In addition, MATLAB 2021a includes a statistics and machine learning toolbox for defining, analyzing, and modeling the results, an image processing toolbox for feature extraction, and a deep learning toolbox for developing and applying deep neural networks, including algorithms, pretrained models, and applications, for the execution of machine and deep learning-based algorithms.

3.6. System Evaluation Techniques. Initially, an MRI brain image is taken as an input. After applying the preprocessing technique, the PSNR and MSE values are calculated for different algorithms applied on an input image. PSNR and MSE are used to compare the quality of a reconstructed image based on their values as the higher the PSNR, and the better the image and the lower the MSE, the lower the error. Hence, to test the accuracy of detecting tumor inside the brain, the following parameters are used:

(1) *Peak Signal-to-Noise Ratio*. This is the ratio of the maximum possible power of a signal to the power of corrupting noise that affects the fidelity of its representation. It is calculated by Equation (38)

$$\mathrm{PSNR} = 10.\log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right), \tag{38}$$

where MAX is the maximum possible pixel value of the image, I is the matrix data of an original image, and MSE is the mean square error [59], while implementing the work using MATLAB, we used the function psnr(img,ref) which calculated the peak signal-to-noise ratio for the brain image (img) with ref as an image reference. Initially, mean square error of an image's pixel matrix has been calculated using Equation (34) which is further incorporated in PSNR equation.

(2) *Mean Square Error Rate*. The sum of the squares of the deviations, i.e., the average squared variance between the expected and real values, is calculated by the mean squared error (MSE) or mean squared deviation (MSD). It is measured using Equation (39).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_n\wedge)^2, \tag{39}$$

where $(y_i - y_n\wedge)^2$ is the square of difference between the actual and the detected value [60]. To obtain the mean square error value of a brain image, we used the function $\mathrm{immse}(a, b)$ which calculated the mean square error between a and b using Equation (34).

(3) *Detection Accuracy*. Detection accuracy is characterized as the level of effectively arranged occurrences in which one is used as the grounded truth value to determine the correct output. It is calculated by using Equation (40).

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN}}. \tag{40}$$

The number of true positives, false negatives, and true negatives is represented as TP, FN, FP, and TN, respectively. To produce an effective classifier, the true positive and true negative rates should be closer to 100% [61]. A

Table 3: Description of DICOM brain MRI image dataset.

Attributes	Tag	Value representation
Age range	(0010,1010)	Age string of length 4 bytes
Sex	(0010,0040)	Code string of length 16 bytes
Image position (patient)	(0020,0032)	Decimal string of length 16 bytes
Pixel spacing	(0028,0030)	Decimal string of length 16 bytes
Image orientation (patient)	(0020,0037)	Decimal string of length 16 bytes
Slice thickness	(0018,0050)	Decimal string of length 16 bytes
Echo time	(0018,0081)	Decimal string of length 16 bytes
Inversion time	(0018,0082)	Decimal string of length 16 bytes
Echo train length	(0018,0091)	Integer string of length 12 bytes
Repetition time	(0018,0080)	Decimal string of length 16 bytes
Trigger time	(0018,1060)	Decimal string of length 16 bytes
Sequence variant	(0018,0021)	Code string of length 16 bytes
Scan options	(0018,0022)	Code string of length 16 bytes
Scanning sequence	(0018,0020)	Code string of length 16 bytes
MR acquisition type	(0018,0023)	Code string of length 16 bytes
Image type	(0008,0008)	Code string of length 16 bytes
Photometric interpretation	(0028,0004)	Code string of length 16 bytes
Contrast bolus agent	(0018,0010)	Integer string of length 12 bytes

Table 4: Age wise statistical analysis of brain tumor.

Age	Gender	Type of brain tumor
0-4	Children	Medulloblastoma
5-9	Males	Pilocytic astrocytoma
10-14	Males	Malignant glioma
15-19	Females	Craniopharyngioma
20-34	Females/males	Pituitary tumors/medulloblastoma
35-74	Females	Meningioma

confusion matrix has been developed utilizing (t, y), where t stands for target value and y stands for output value, to generate true positive, true negative, false negative, and false-positive values. The accuracy is then calculated using these numbers.

4. Results and Analysis

Various machine and deep learning algorithms have been used for feature extraction, optimization, and classification to calculate the peak signal-to-noise ratio, mean square error rate, and detection accuracy of brain tumors.

4.1. Experimental Results. Based on the algorithms, three cases were taken to compare the outcomes so that we could find out the best technique for brain tumor detection. Initially, the image underwent the preprocessing state by using the CLAHE technique to improve its contrast.

Figure 4 depicts image preprocessing where the proposed solution can obtain a high strength of the image to normalize the pixels. One of the crucial stages is preprocessing. The preprocess output is fed into the principal component analysis or PCA, which extracts features for the optimization process. Firefly and Naïve Bayes are used for optimization and classification, respectively.

4.1.1. Case 1: Principal Component Analysis, Firefly, and Naïve Bayes Classifier. Figure 5(a) depicts the derived attribute values from principal component analysis. In contrast, Figure 5(b) shows the firefly technique to depict the smoothing of the image and NBC for image recognition and classification. It was smoothing the image which smoothed the pixels and image borders, which can be used in future image recognition systems. The picture classification that was used to detect the tumor region is depicted in Figure 5(c).

In the next scenario, the feature extraction was carried out by independent component analysis, which performs the Gaussian process to reduce the noise and independency among the images, which will extract the features having low variance among the neighborhood pixels. Figure 6(a) shows the preprocessing of the training image using threshold segmentation, which is fed into the feature extraction process. This partitions the image into the various segmentations and uses scaling to obtain meaningful insights, which helps locate the multiple curves and objects in the image.

4.1.2. Case 2: Independent Component Analysis, Cuckoo Search, and Naïve Bayes Classifier. Figure 6(b) shows the smoothing of the image and region of interest. The main region of interest, which has a probability of cancer in the image, is evaluated by the cuckoo search. The smoothing of the image will smooth the pixels and the boundaries of the image, which will lead to a high classification rate using NBC for the high region of interest in the detection scenario.

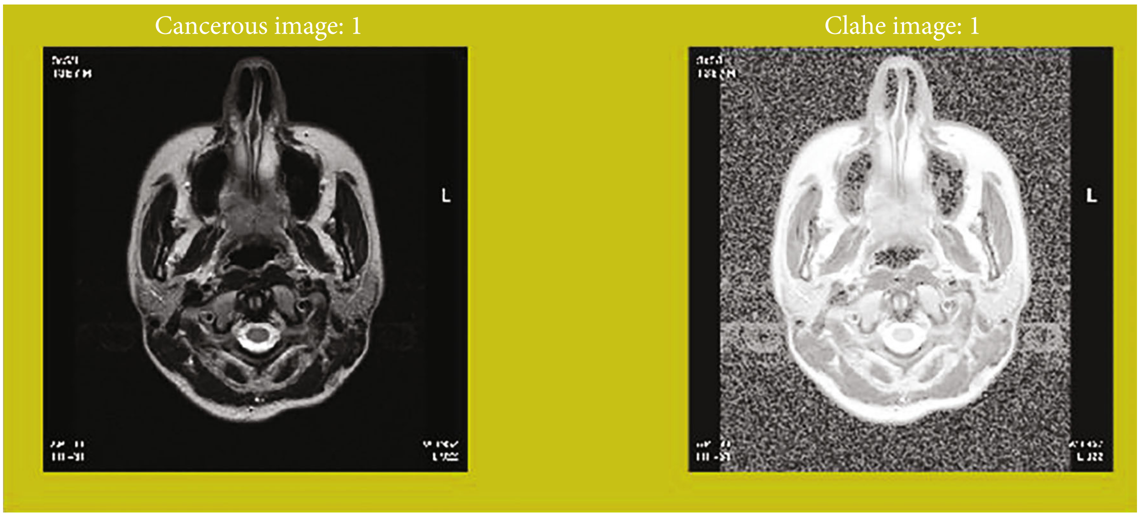


Figure 4: Preprocessing of cancerous image.

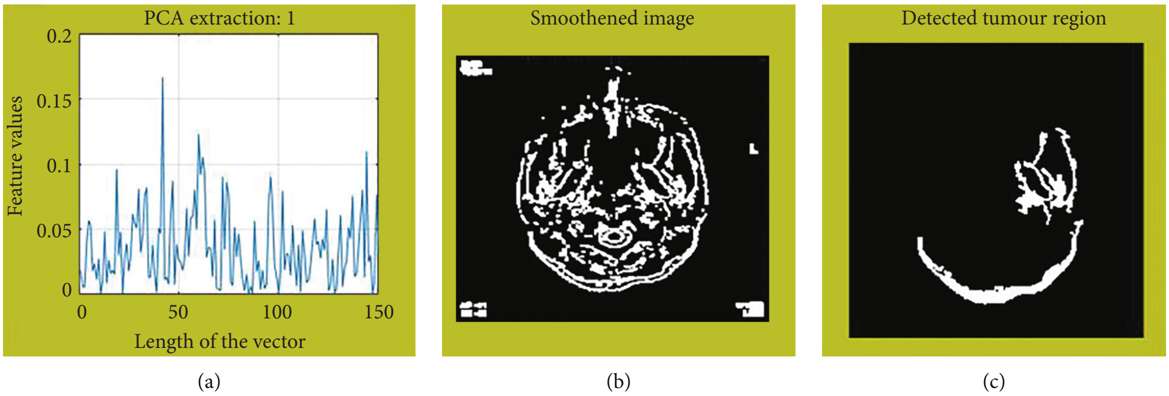


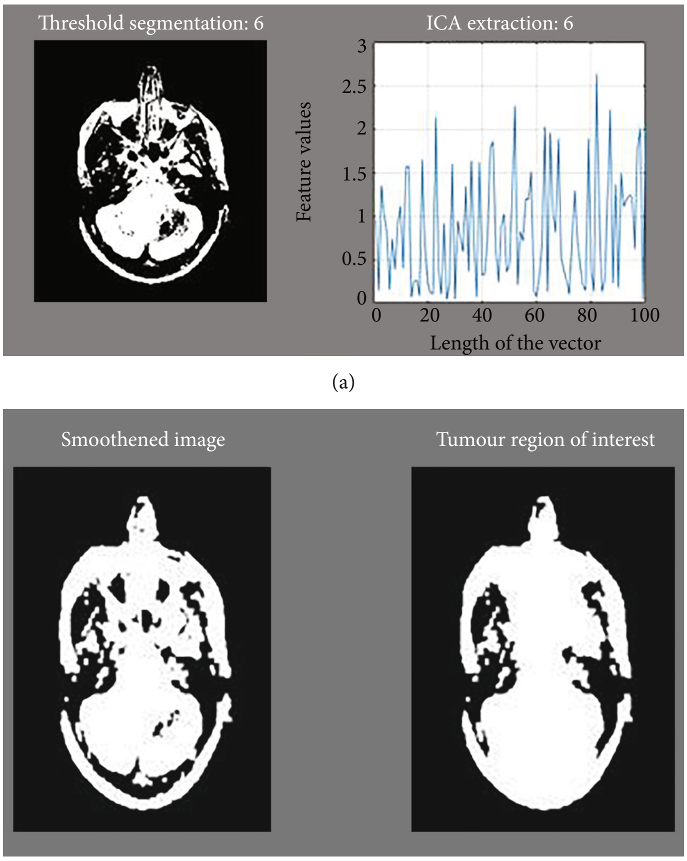
Figure 5: (a) Feature extraction, (b) image smoothing, and (c) brain tumor detection.

Finally, a hybrid optimization, which is the combination of lion and bat optimization, was applied to see the changes in the results and was later followed by a recurrent neural network. Figure 7(a) shows the contrast level enhancement of the image and feature extraction using CLAHE and independent component analysis, which is a significant step in extracting the feature vectors.

4.1.3. Case 3: Independent Component Analysis, Hybrid Optimization (Lion+Bat), and Recurrent Neural Network. The feature vectors were the independent, robust features that had more minor variance and low standard deviations. Figure 7(b) shows the smoothing of the image and the region of interest where there is a probability of detecting a tumor. The classification was carried out using the recurrent neural network classifier. The recurrent neural network classification performed the detection process using deep learning. The deep network and filtrations were completed, and the system achieved high classifications for the high region of interest.

In cases 1–3, Table 5 summarizes the performance of all machine and deep learning-based approaches utilized for feature extraction, classification, and optimization. These algorithms were tested on a few parameters such as peak signal-to-noise ratio, mean square error rate, and sensitivity to identify brain tumors after being trained on five MRI brain images. Table 4 is divided into three groups of algorithms: principal component analysis+firefly+Naïve Bayes, independent component analysis+cuckoo search+Naïve Bayes, and independent component analysis+(lion+bat) optimization+recurrent neural network, with their respective parameters, to find the best algorithm for detecting brain tumors. From case 1, i.e., the PCA+FF+NB group, we obtained optimum peak signal-to-noise ratio values, mean square error rate, and accuracy of 43.76, 3.05, and 96.71, respectively. When the ICA+CS+NB group was employed, the peak signal-to-noise ratio, mean square error rate, and detection accuracy were 31.85, 1.02, and 90.13, respectively. When the recommended model was implemented using ICA+(lion+bat)+RNN, the ideal peak signal-to-noise ratio, mean square error rate, and detection accuracy were 64.81, 1.40, and 98.61, respectively.

When these data were compiled, it was discovered that ICA+hybrid optimization+RNN produced the greatest



(a)

(b)

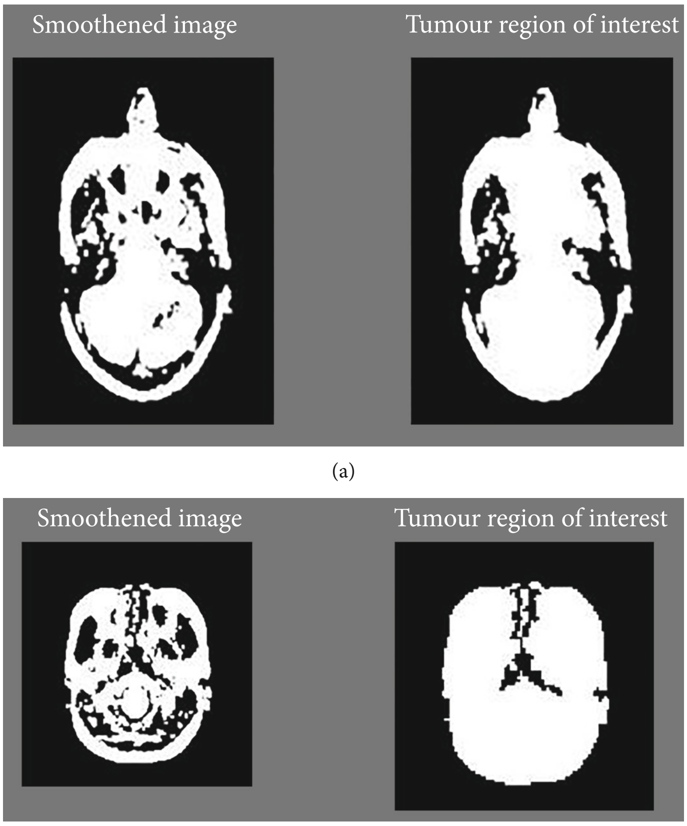
Figure 6: (a) Segmentation and extraction using ICA and (b) smoothing and region of interest.

PSNR and accuracy values of 64.81 dB and 98.61% for all of the pictures and the lowest mean square error for image 4. In a nutshell, CLAHE and thresholding techniques were used for preprocessing; PCA and ICA were used for feature extraction; cuckoo search, firefly, lion, and bat approaches were employed for optimization; NBC and RNN were used for image recognition. The PSNR of 64.81 has been achieved by the group of independent component analysis, hybrid optimization, and an RNN, the mean square error of 1.02 was conducted by the group of ICA, cuckoo search, and NBC, and the accuracy of 98.61 was achieved by the group of ICA, hybrid optimization, and an RNN.

4.2. Comparison with Past Studies. Table 6 shows the preliminary effects of the work instead of state-of-the-art techniques, showing that the proposed work stands out to be more effective than the state-of-the-art techniques in all brain tumor identification categories.

The suggested technique is unique since it achieves better results with high PSNR values of 64.81 dB and a 98.61% accuracy rate in detecting brain tumors, despite the mean square error being somewhat greater than 1.34. This is because the suggested system's characteristics were optimized by utilizing several optimization strategies, including firefly, lion, bat, and cuckoo search, to produce the optimum result for reliably identifying brain cancers. We have also related the planned process with the previous work by comparing the current results with the outcomes achieved by the researchers using machine and deep learning algorithms to detect brain tumor (Table 5). When comparing the present and past methodologies, it was discovered that the researchers' procedures resulted in a low PSNR value and a high MSE value, which limited the detection accuracy rate. As a rule of thumb, a high peak signal-to-noise ratio suggests good image quality, whereas a low peak signal-to-noise ratio suggests bad picture quality. A high mean square error rate, on the other hand, denotes a high error rate, whereas a low mean square error rate denotes a low mistake rate. As a result, we can state that our suggested methodology outperforms the others in terms of PSNR and accuracy.

4.3. Practical Deployment. The suggested model has been linked to a mobile application that allows patients to quickly determine whether or not a tumor is present inside their brain. The proposed mobile application's architecture is depicted in Figure 8. The mobile app is intended to capture the afflicted region of the brain picture, and the data is securely stored on a distant server using the representational state transfer API. This is because remote online storage (Figure 9) is quicker and more dependable. It offers drag-



(a)

(b)

Figure 7: (a) Contrast and feature extraction. (b) Smoothened image and detection.

Table 5: Evaluation parameters of different techniques for brain tumor detection.

Samples	PCA+firefly+Naïve Bayes			ICA+cuckoo search+Naïve Bayes			ICA+hybrid optimization+RNN		
	PSNR (db)	MSE	Accuracy (%)	PSNR (db)	MSE	Accuracy (%)	PSNR (db)	MSE	Accuracy (%)
Image_1	42.21	3.10	96.59	27.89	2.11	89.44	64.55	2.23	97.36
Image_2	42.18	3.15	96.32	28.58	1.02	88.18	57.59	1.61	97.68
Image_3	43.76	3.05	96.71	27.44	3.10	89.19	59.39	2.32	97.11
Image_4	41.75	3.26	96.24	29.79	2.89	90.13	62.41	1.40	98.61
Image_5	42.98	3.16	96.52	31.85	2.60	89.48	64.81	1.95	97.83
Mean	42.57	3.14	96.47	29.11	2.34	89.28	61.75	1.90	97.72
sd	0.79	0.07	0.19	1.76	0.82	0.70	3.18	0.39	0.57

Table 6: Comparative analysis of proposed method with state-of-the-art techniques.

Methods	Peak signal-to-noise ratio	Mean square error rate	Detection accuracy
Kumar, M et al. [21]	15.05	2.03	97.25
Umary, A et al. [38]	17.71	1.34	88.25
Chahal, K et al. [3]	21.40	8.19	98.01
Bahadure, N et al. [41]	42.03	3.07	97.28
Suneetha, B et al. [43]	31.183	4.95	96.32
Proposed work	64.81	1.40	98.61

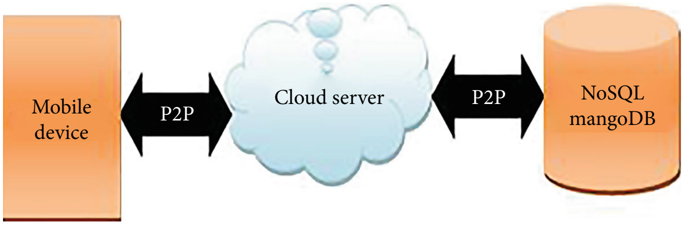


FIGURE 8: Framework of proposed mobile application.

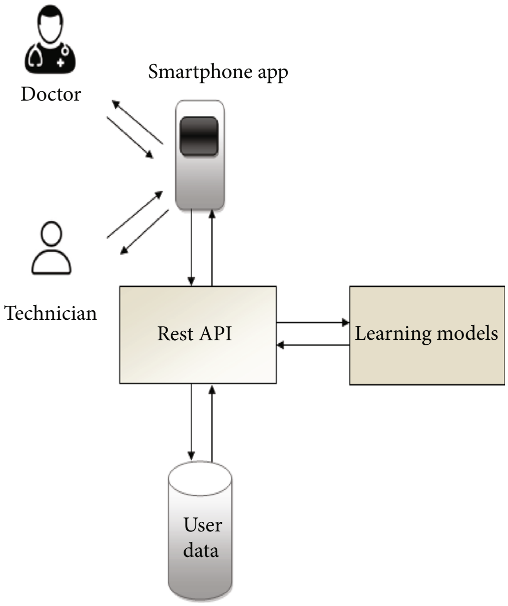


FIGURE 9: Remote server architecture.

and-drop, data compression, data transmission, and data encryption functions, as well as easy-to-use interfaces protected by passwords.

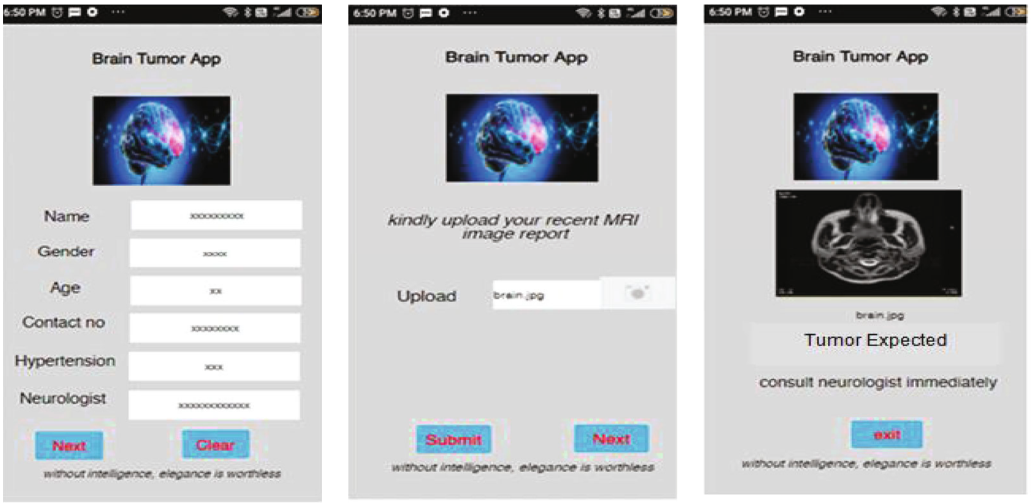
No SQL Mongo DB is used for handling massive user-related data, and point to point (P2P) protocol is used to establish a direct connection. Instead, the mobile application is created by using Android Development and is programmed in Java programming language. This apk file can be accessed by any user. Hence, it is open source, and its execution relies on the receiving and transferring of data to an online remote server.

Doctors and clinical laboratory professionals can use the Smartphone application to detect a tumor inside the brain. The user will utilize a mobile device to capture the MRI brain picture and submit it to the mobile application as input. To identify and classify the tumor inside the brain, the input picture is further processed using several learning models, such as independent component analysis, cuckoo search, and recurrent neural network. The user will receive an output stating that a tumor has been discovered as soon as it is detected. Figures 10(a)–10(c) depict three displays of the brain tumor app, where the patient's information is first required, including name, gender, age, phone number, if they have had hypertension, and the neurologist's name. After correctly filling out their information, the user must click the "Next" button to go to a screen to upload the MRI brain picture. Following the submission of the picture, the suggested technique will begin analyzing the data using learning models to determine whether a brain tumor has been found. The outcome of the user's MRI brain picture input will be displayed later on the third screen, and a message will be displayed along with the image.

Using this information, doctors can start their diagnosis as soon as possible without wasting time to save the patient's life. Furthermore, the application will also help the doctors/technicians locate the small size tumor that is not even clearly visible in the image with the help of CLAHE technique, which improves the visibility level of foggy image to restrict it from growing larger. In a word, this application will assist patients in learning about their brain tumor within a short period after receiving their medical lab data. As a result, they will be able to see professional doctors right away and begin therapy. On the other side, this application will bring confidence to patients who do not have a brain tumor and save them time from unnecessarily attending a hospital.

The security of healthcare data is of the utmost importance in our mobile application. SSL technology is used to encrypt the data in the Smartphone device. The secure socket layer employs a cryptographic scheme that encrypts data using two keys: a public key known to all parties and proprietary or hidden key known only to the message's receiver [62]. The Smartphone application's protection allows patients a better sense of control over their healthcare data's privacy, security, and confidentiality. The protection systems also keep our mobile application's health data protected as well as clean. Though a malicious entity may intercept an app's data across the network if the app misuses SSL, as a result, servers are generally set up with certificates from well-known issuers known as Certificate Authorities to limit fraud.

In most cases, the host platform has a list of well-known CAs that it trusts [63]. As of Android, there have been over 100 CAs that are updated with each version and did not vary from device to device. A CA has a certificate and a private key, much like a server. When a certificate for a server is issued, the CA signs the certificate using its private key. The client can then check that the server has a certificate from a CA that the platform recognizes [64]. Beyond this, there is one more factor, i.e., response time, that directly affects our developed application's performance, as a delay can hamper the performance of the system. Hence, at present, we have computed that the response time taken by our



6:50 PM

Brain Tumor App

Name xxxxxxxxxx
Gender xxxxx
Age xx
Contact no xxxxxxxxx
Hypertension xxx
Neurologist xxxxxxxxxxxxx

Next
Clear
without intelligence, elegance is worthless

(a)

6:50 PM

Brain Tumor App

kindly upload your recent MRI image report

Upload brain.jpg

Submit
Next
without intelligence, elegance is worthless

(b)

6:50 PM

Brain Tumor App

brain.jpg

Tumor Expected

consult neurologist immediately

exit
without intelligence, elegance is worthless

(c)

Figure 10: Interface of the mobile application to detect brain tumor.

application to produce output with proper input and bandwidth is 4-5 seconds.

5. Discussion

The primary goal of this research article is to detect tumors within the brain, for which various techniques such as CLAHE, threshold, ICA, PCA, cuckoo, Naïve Bayes, firefly, bat, cuckoo, and lion have been used throughout the process, from preprocessing to classification. The model has also been integrated into a mobile application to ensure that doctors, clinical technicians, and patients may access it on the go.

On the one hand, identifying brain tumors using mobile applications proves highly beneficial to users. Even so, certain factors such as lens quality, which is responsible for focusing the scattered light that enters the camera onto the sensor, the size of the pixels, which is perhaps one of the most popular specifications of a mobile camera, the aspect ratio, and the size of the sensor, which is critical in determining the quality of a Smartphone image, can all have an impact on the quality of an image. If none of these considerations are taken into account, the patient will receive the incorrect output, potentially fatal to them. Furthermore, the network bandwidth should not be too low for uploading or downloading images and presenting the outcome. Otherwise, it will take longer to identify a brain tumor than the basic technique.

In summary, after collecting it from a mobile device and adequate bandwidth, a clean input image should be addressed for simple access to a mobile application for brain tumor detection to be successful.

6. Future Directions and Limitations of the Research

Machine and deep learning-based techniques deeply optimize the existing methods of anticancer drug research [58]. They have the potential to reduce the cost of care and the ability to merge the large quantity of information collected from the various sources of data to reduce the workload of clinicians, as it is difficult for them to integrate such complex data manually. However, the detections generated by these techniques are evaluated and interpreted by the experts. However, the present research has some limitations [57]. The proposed methodology has been implemented using a single dataset and can be improved by taking multiple datasets, as the paper only focuses on the presence or absence of brain tumor, but it did not detect the type of tumor because of which the current work can also be extended by labeling the type of the tumor that has been seen inside the brain and also by incorporating other techniques of deep learning-based models such as self-defined artificial neural network and convolution neural network can be used to improve the detection rate of a brain tumor. As the mobile application has also been used to detect brain tumors, its main drawback can also be erroneous, which needs to be considered in the future. Moreover, they also have flaws, such as method sophistication, a lack of knowledge about an individual with a specific context, and the possibility that these approaches will include an incorrect diagnosis. As a result, physicians are more likely to administer the wrong care if they lack the experience to spot the error, requiring the use of a vast dataset to train the algorithm and obtain a reliable detection performance.

7. Conclusion

Machine and deep learning are gradually encompassing all phases of our lives, particularly in healthcare. The present work highlights that researchers are quickly gaining a deeper understanding of the challenges and prospects offered by machine and deep learning models as an intelligent system in the area of tumor diagnosis and detection.

Using different learning methods, the potential presence of a tumor, which is a leading cause of death, has been successfully detected. Preprocessing was undertaken for this, and features were removed from the images and then optimized using different techniques. Among all of the methods, the group of independent component analysis, hybrid optimization, and an RNN had the best PSNR of 64.81%, the group of ICA, cuckoo search, and NBC had the best mean square error of 1.02, and the group of ICA, hybrid optimization, and an RNN had the best accuracy of 98.61%.

As healthcare cost is increasing, patients need to keep track of their prescription spending. Since these algorithms are computationally less costly than other approaches, they can be used in hospitals for brain tumor recognition, brain tumor diagnosis, etc. Moreover, the substantial analysis of these algorithms on tumor-based issues supports a strong candidature in controlling a brain tumor at an early stage. When working with medical results, the algorithms used in this work take into account information from a variety of attributes to make a final detection and offer straightforward interpretations of their decisions, making them one of the most valuable tools for assisting physicians in their decisions.

Abbreviations

CLAHE:	Contrast limited adaptive histogram equalization
PSNR:	Peak signal-to-noise ratio
PCA:	Principal component analysis
ICA:	Independent component analysis
FF:	Hybrid optimization
MSE:	Mean square error rate.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] A. Mustaqeem, A. Javed, and T. Fatima, "An efficient brain tumor detection using watershed and threshold based segmentation," *Graphics and signal processing*, vol. 4, no. 10, pp. 34–39, 2012.

[2] N. J. DeNunzio and T. I. Yock, "Modern radiotherapy for pediatric brain tumors," *Cancers*, vol. 12, no. 6, p. 1533, 2020.

[3] K. Chahal and S. Pandey, "A hybrid weighted fuzzy approach for brain tumor segmentation using MR images," *Neural Computing and Applications*, 2021.

[4] A. Mirbeik and N. Tavassolian, "Tumor detection using millimeter-wave technology: differentiating between benign lesions and cancer tissues," *IEEE Microwave Magazine*, vol. 20, no. 8, pp. 30–43, 2019.

[5] M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, and M. R. Khosravi, "Energy-efficient workload allocation in fog-cloud based services of intelligent transportation systems using a learning classifier system," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 1484–1490, 2020.

[6] M. Abbasi, E. M. Pasand, and M. R. Khosravi, "Workload allocation in IoT-fog-cloud architecture using a multi-objective genetic algorithm," *Journal of Grid Computing*, vol. 18, no. 1, pp. 43–56, 2020.

[7] D. Jasm, M. Hamad, and A. Alrawi, "A survey paper on image mining techniques and classification of brain tumor," *Journal of Physics*, vol. 1804, pp. 1–9, 2020.

[8] T. Sharma and K. Sahil Verma, "Intelligent heart disease prediction system using machine learning: a review," *International Journal of Recent Research Aspects*, vol. 4, no. 2, pp. 94–97, 2017.

[9] O. Polat and C. Güngen, "Classification of brain tumors from MR images using deep transfer learning," *The Journal of Supercomputing*, vol. 77, no. 7, pp. 7236–7252, 2021.

[10] H. Mohsen, E. Dahshan, E. Horbaty, and A. Salem, "Classification using deep learning neural networks for brain tumors," *Future Informatics and Computing Journal*, vol. 3, pp. 68–71, 2017.

[11] A. Deep and S. Emmanuel, "An efficient detection of brain tumor using fused feature adaptive firefly back propagation neural network," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11799–11814, 2019.

[12] Q. Eudocia, S. Wendy, M. Clair, and B. Jeffrey, "Report of National Brain Tumor Society roundtable workshop on innovating brain tumor clinical trials: building on lessons learned from COVID-19 experience," *Neuro-Oncology*, vol. 23, no. 8, pp. 1252–1260, 2021.

[13] P. Rani, S. V. Kavita, and G. N. Nguyen, "Mitigation of black hole and gray hole attack using swarm inspired algorithm with artificial neural," *Network Access*, vol. 8, pp. 121755–121764, 2020.

[14] K. Sharma, A. Kaur, and S. Gujral, "Brain tumor detection based on machine learning algorithms," *International Journal of Computer Applications*, vol. 103, pp. 15–20, 2014.

[15] M. Siar and M. Teshnehlab, "Brain tumor detection using deep neural network and machine learning algorithm," *International Conference on Computer and Knowledge Engineering*, pp. 1–4, 2019.

[16] M. A. Z. Chudhery, S. Safdar, J. Huo, H.-U. Rehman, and R. Rafique, "Proposing and empirically investigating a mobile-based outpatient healthcare service delivery framework using stimulus–organism–response theory," *IEEE Transactions on Engineering Management*, 2021.

[17] S. Ghanavati, J. Li, T. Liu, P. Babyn, W. Doda, and G. Lampropoulos, "Automatic brain tumor detection in magnetic resonance images," in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 574–577, Barcelona, Spain, 2012.

[18] G. Tandel, A. Balestrieri, T. Jujaray, N. N. Khanna, L. Saba, and J. S. Suri, "Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm," *Computers in Biology and Medicine*, vol. 122, pp. 103804–103807, 2020.

[19] G. Kaur and A. Oberoi, "Novel approach for brain tumor detection based on Naïve Bayes classification," *Data Management Analytics and Innovation*, vol. 1, pp. 451–462, 2020.

[20] J. Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, and D. González-Ortega, "A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network," *Healthcare*, vol. 9, no. 2, p. 153, 2021.

[21] M. Kumar, P. Mukherjee, K. Verma, S. Verma, and D. B. Rawat, “Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks,” *IEEE Transactions on Network Science and Engineering*, 2021.

[22] G. Rani, M. G. Oza, V. S. Dhaka, N. Pradhan, S. Verma, and J. J. P. C. Rodrigues, “Applying deep learning-based multimodal for detection of coronavirus,” *Multimedia Systems*, pp. 1–12, 2021.

[23] G. Ghosh, K. Verma, D. Anand et al., “Secure surveillance systems using partial-regeneration-based non-dominated optimization and 5D-chaotic map,” *Symmetry*, vol. 13, no. 8, p. 1447, 2021.

[24] A. Onan and S. Korukoğlu, “A feature selection model based on genetic rank aggregation for text sentiment classification,” *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.

[25] M. Sood and S. Verma, “Vinod Kumar Panchal and Kavita "Optimal path planning using swarm intelligence based hybrid techniques" Journal of computational and theoretical nanoscience (JCTN),” *Journal of Computational and Theoretical Nanoscience.*, vol. 16, no. 9, pp. 3717–3727, 2019.

[26] Z. Li, S. Verma, and M. Jin, “Power allocation in massive MIMO-HWSN based on the water-filling algorithm,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 8719066, 11 pages, 2021.

[27] W. Li, Y. Chai, F. Khan et al., “A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system,” *Mobile Network and Applications*, vol. 26, no. 1, pp. 234–252, 2021.

[28] J. Amin, M. Sharif, N. Gul, Y. Mussarat, and S. Shad, “Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network,” *Pattern Recognition Letters*, vol. 129, pp. 1–7, 2020.

[29] Z. Gao, Y. Yang, M. R. Khosravi, and S. Wan, “Class consistent and joint group sparse representation model for image classification in Internet of Medical Things,” *Computer Communications*, vol. 166, pp. 57–65, 2021.

[30] Y. Liu, Y. X. Huang, X. Zhang et al., “Deep C-LSTM neural network for epileptic seizure and tumor detection using high-dimension EEG signals,” *IEEE Access*, vol. 8, pp. 37495–37504, 2020.

[31] G. Ghosh, “Kavita, Sahil Verma, NZ Jhanjhi, “Secure surveillance system using chaotic image encryption technique” 2020, Vol. 993, 012062,” in *IOP Conference Series: Materials Science and Engineering*, vol. 993, no. 1, p. 012062.

[32] A. Onan, “A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 6844–6852, 2015.

[33] A. Onan, S. Korukoğlu, and H. Bulut, “A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification,” *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.

[34] N. Arun, M. Mohammed, S. Mostafa, D. Ibrahim, J. Rodrigues, and V. Albuquerque, “Fully automatic model-based segmentation and classification approach for MRI brain tumor using artificial neural networks,” *Concurrency and Computation: Practice and Experience*, vol. 32, 2020.

[35] M. Nazir, M. Khan, T. Saba, and A. Rehman, “Brain tumor detection from MRI images using multi-level wavelets,” in *International Conference on Computer and Information Sciences*, pp. 1–5, Sakaka, Saudi Arabia, 2019.

[36] D. Patil and S. Hamde, “Automated detection of brain tumor disease using empirical wavelet transform based LBP variants and ant-lion optimization,” *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 17955–17982, 2021.

[37] B. Pushpa and F. Louies, “Detection and classification of brain tumor using machine learning approaches,” *International Journal of Research in Pharmaceutical Sciences*, pp. 1–7, 2019.

[38] A. Umary and H. Kaur, “Automatic brain tumor diagnosis and segmentation: based on SVM algorithm. International Journal of Innovative Technology and Exploring,” *Engineering*, vol. 9, no. 6, pp. 1079–1084, 2020.

[39] R. Pugalenthi, M. Rajakumar, J. Ramya, and V. Rajinikanth, “Evaluation and classification of brain tmor MRI using machine learning technique,” *Control Engineering and Applied Informatics*, vol. 21, pp. 12–21, 2019.

[40] G. Manogaran, P. Shakeel, A. Hassanein, P. Malarvizhi, and G. Chandra, “Machine learning approach-based gamma distribution for brain tumor detection and data sample imbalance analysis,” *IEEE Access*, vol. 7, pp. 12–19, 2019.

[41] N. Bahadure, A. Ray, and H. Thethi, “Image analysis for MRI based brain tumor detection and feature extraction using biologically inspired BWT and SVM,” *International Journal of Biomedical Imaging*, vol. 2017, Article ID 9749108, 12 pages, 2017.

[42] M. Arora, S. Verma, and C. S. Kavita, “A Systematic Literature Review of Machine Learning Estimation Approaches in Scrum Projects,” in *Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing*, P. Mallick, V. Balas, A. Bhoi, and G. S. Chae, Eds., Springer, Singapore, 2020.

[43] B. Suneetha and A. Rani, “A portrayl advance for brain tumor segmentation techniques in magnetic resonance imaging,” *International Journal of Pure and Applied Mathematics*, vol. 119, pp. 1305–1326, 2018.

[44] Z. Sobhaninia, S. Rezeai, A. Noorozi et al., *Brain tumor segmentation using deep learning by type specific sorting of images*, pp. 1–4, 2018.

[45] S. Sajid, S. Hussain, and A. Sarwar, “Brain tumor detection and segmentation in MR images using deep learning,” *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9249–9261, 2019.

[46] G. Garg and R. Garg, “Brain tumor detection and classification based on Hybrid Ensemble Classifier,” *Computer Vision and Pattern Recognition*, pp. 1–18, 2021.

[47] https://wiki.cancerimagingarchive.net/display/Public/Brain-Tumor-Progression.

[48] J. Machiraju and S. Rao, “Pathological brain tumor detection using CLAHE and LS-SVM,” *Test Engineering and Management*, vol. 82, pp. 11323–11332, 2020.

[49] P. Natarajan, N. Krishnan, N. Kenkre, S. Nancy, and B. Singh, “Tumor detection using threshold operation in MRI brain images,” in *IEEE International Conference on Computational Intelligence and Computing Research*, Coimbatore, India, 2012.

[50] S. Gaikwad and M. Joshi, “Brain tumor classification using principal component analysis and probabilistic neural network,” *International Journal of Computer Applications*, vol. 120, no. 3, pp. 5–9, 2015.

[51] G. Sandhya, K. Giri, and T. Satya, “A novel approach for the detection of tumor in brain MR images and its classification via independent component analysis and kernel support vector machine,” *Imaging in Medicine*, vol. 9, pp. 1–5, 2017.

[52] E. George, G. Rosline, and G. Rajesh, “Brain tumor segmentation using cuckoo search optimization for magnetic resonance images,” in *IEEE 8th GCC Conference & Exhibition*, Muscat, Oman, 2015.

[53] N. Mohan, “Tumor detection from brain MRI using modified Sea lion optimization based kernel extreme learning algorithm,” *International Journal of Engineering Trends and Technology*, vol. 68, no. 9, pp. 84–100, 2020.

[54] A. Alhassan and W. Zainon, “BAT algorithm with fuzzy C-ordered means (BAFCOM) clustering segmentation and enhanced capsule networks (ECN) for brain cancer MRI images classification,” *IEEE Access*, vol. 8, pp. 201741–201751, 2020.

[55] H. Zaw, N. Maneerat, and K. Win, “Brain tumor detection based on Naïve Bayes classification,” in *5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, Luang Prabang, Laos, 2019.

[56] R. Suganthe, G. Revathi, S. Monisha, and R. Pavithran, “Deep learning based brain tumor classification using magnetic resonance imaging,” *Journal of Critical Reviews*, vol. 7, pp. 347–350, 2020.

[57] L. Li and J. Wang, “DDIT - a tool for DICOM brain images de-identification,” in *2011 5th International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, Wuhan, China, 2011.

[58] Y. Kumar and M. Mahajan, “5. Recent advancement of machine learning and deep learning in the field of healthcare system,” in *Computational Intelligence for Machine Learning and Healthcare Informatics*, pp. 7–98, 2020.

[59] G. Vishnuvarthanan, M. P. Rajasekaran, N. A. Vishnuvarthanan, T. A. Prasath, and M. Kannan, “Tumor detection in T1, T2, FLAIR and MPR brain images using a combination of optimization and fuzzy clustering improved by seed-based region growing algorithm,” *Wiley Periodicals*, vol. 27, no. 1, pp. 33–45, 2017.

[60] L. Gaur, G. Singh, A. Solanki et al., “Disposition of youth in predicting sustainable development goals using the neuro-fuzzy and random forest algorithms,” *Human-Centric Computing and Information Sciences*, vol. 11, p. 24, 2021.

[61] T. Ruba, R. Tamilselvi, M. Beham, and N. Aparna, “Accurate classification and detection of brain cancer cells in MRI and CT images using nano contrast agents,” *Biomedical and Pharmacology Journal*, vol. 13, pp. 1227–1237, 2020.

[62] A. Mehrdad, M. Black, and N. Yadav, “Security vulnerabilities in mobile health applications,” in *Proceedings of the 2018 IEEE conference on application, Information and Network Security (AINS)*, Langkawi, Malaysia, November 2018.

[63] N. Kaur and S. Verma, “Detection of plant leaf diseases by applying image processing schemes,” *Journal of computational and theoretical nanoscience (JCTN)*, vol. 16, no. 9, pp. 3728–3734, 2019.

[64] S. Ramisetty and S. Verma, “The amalgamative sharp WSN routing and with enhanced machine learning,” *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 9, pp. 3766–3769, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4330701, 9 pages
https://doi.org/10.1155/2022/4330701

Research Article

Question Text Classification Method of Tourism Based on Deep Learning Model

Wanli Luo[1] **and Lei Zhang**[2]

[1]*College of Information and Engineering, Sichuan Tourism University, Chengdu, Sichuan 610000, China*
[2]*Personal Business Department, Sichuan Rural Credit, Chengdu, Sichuan 610000, China*

Correspondence should be addressed to Wanli Luo; luowanli@sctu.edu.cn

Received 21 October 2021; Revised 12 November 2021; Accepted 14 December 2021; Published 5 January 2022

Academic Editor: Mohammad R Khosravi

The Internet of Things applications are diverse in nature, and a key aspect of it is multimedia sensors and devices. These IoT multimedia devices form the Internet of Multimedia Things (IoMT). Compared with the Internet of Things, it generates a large amount of text data with different characteristics and requirements. Aiming at the problems that machine learning and single structure deep learning model cannot effectively grasp the text emotional information in text processing, resulting in poor classification effect, this paper proposes a text classification method of tourism questions based on deep learning model. First, the corpus is trained with word2vec tool based on continuous word bag model to obtain the text word vector representation. Then, the attention mechanism is introduced into the long-short term network (LSTM), and the attention-based LSTM model is constructed for text feature extraction, which highlights the impact of different words in the input text on the text emotion category. Finally, the text features are input into the Softmax classifier to obtain the probability distribution of text categories, and the model is trained combined with the cross entropy loss function. The experimental results show that the average accuracy, recall, and F value are 0.943, 0.867, and 0.903, respectively, which has better classification effect than other methods.

1. Introduction

As one of the main ways of leisure and entertainment after the continuous improvement of China's social economy and people's material living standards, tourism has attracted more and more attention and favor, and "self-help tourism" has become the mainstream of tourism forms. In the process of self-help travel, problems such as route planning, catering, and accommodation and itinerary strategy are easy to occur. With the rapid development of the Internet, tourists mainly obtain tourism information through network query and Q & A. Access to information includes tourism information released by major tourism portals, tourism applications, and other media platforms. This kind of tourism information has the characteristics of popularization and generalization [1]. However, when obtaining tourism information, it is necessary to publish the questions and wait for the reply of other users, which has a delay. Moreover, the tourism Q & A community usually classifies the questions according to the geographical location, which cannot fully cover all kinds of questions. In addition, the traditional tourism Q & A community generally uses manual annotation or machine learning model for problem classification, resulting in low classification efficiency and accuracy, unable to quickly and accurately locate the problem category of tourists, which affects the subsequent information retrieval [2–4]. Therefore, how to automatically classify all kinds of tourism questions quickly and efficiently has become an urgent problem to be solved.

The syntactic and semantic information of tourism question text mainly depends on the text composition and sequence order. On the one hand, the grammar of tourism questions consists of multiple question keywords and some network popular words. The words in the text sequence are modeled to form the low-level subspace structure information of the text sequence [5]. On the other hand, the semantic information and syntactic information of tourism question text come from the text sequence itself. Compared with traditional machine learning technology, the existing deep learning technology can better capture the deep semantic information of the text and solve the error problem caused by manual design features, and the classification accuracy is higher [6]. However, most text classification methods are deep learning models based on a single structure or simply concatenate multiple models. When mining the deep features of text, a large amount of syntax and syntactic information will be lost and redundant information will be added [7]. Therefore, this paper proposes a text classification method of tourism questions using deep learning model. Its innovations are summarized as follows:

(1) In order to overcome the problems of gradient explosion or disappearance problem of recurrent neural network (RNN), the proposed method uses the long-short term memory (LSTM) network to construct the text classification model and inputs the text word vector into the model to complete the feature extraction, so as to ensure the accuracy of tourism question text classification

(2) In order to obtain the different influence weights of different words in emotion classification, a text emotion classification model based on LSTM with attention mechanism is proposed, which focuses on the emotional information of text data and further improves the expression ability of text features

The rest of this article is organized as follows. The second section introduces the relevant research progress in this field; the third section specifically introduces the proposed LSTM text classification model based on the attention mechanism; the fourth section compares with the current text classification model to realize the feasibility of the method proposed in this article and the optimality experiment simulation analysis; Section 5 is the conclusion of this paper.

2. Related Works

At present, there are many researches on the text classification methods of tourism questions at home and abroad. In addition to the early manual annotation methods, the traditional machine learning method is the main method of tourism question text classification in recent years. Early question text classification methods mainly used simple machine learning model to classify and recognize different types of question text. Ref. [8] proposed a text and document classification model of support vector machines (SVM). Different experimental results show that it has high classification accuracy on any kind of data set, but the classification efficiency needs to be improved. Ref. [9] proposed an active learning text question and answer classification method, which can potentially reduce the size of the training data set, but the prediction of model performance in active learning may be affected by statistical deviation, so there is still room to further improve the accuracy of text classification. Ref. [10] proposed a cost sensitive analysis air valve, which is derived by differential evolution algorithm. The experimental results show that the algorithm has high classification accuracy, but the classification accuracy and classification efficiency of complex texts need to be improved.

Deep learning technology has developed rapidly in recent years and has been applied to question text classification tasks and achieved good results. Compared with traditional machine learning technology, it can capture the deep semantic information of text and solve the error problem caused by manual design features and has higher classification accuracy. Ref. [11] uses the classification algorithm of LSTM and convolutional neural network to improve the classification accuracy of problem data sets by changing the vector size and embedding type of combined architecture, but the data sets required for training are large and the training time is long. Ref. [12] evaluated the performance of shallow machine learning and deep learning in text classifiers and text classification embedded in small clinical data sets. Self-training and pretraining word embedding were used as input representation schemes to evaluate logistic regression and long-short term training methods. In the balanced data supported by pretraining embedding, the accuracy of deep learning method was better. Ref. [13] compares the text data classification algorithms of deep learning and traditional machine learning. The results show that the deep learning algorithm has better classification accuracy in some specific cases, but it needs more training data and training time to improve the accuracy. Ref. [14] proposed a unified learning framework of hierarchical cognitive structure learning model, which includes two submodules: attention ordered cyclic neural network and hierarchical two-way capsule. It has good text classification performance, but the simple series structure of the two models is difficult to mine deep-seated text features. Aiming at the shortcomings of the above methods, a text classification method of tourism questions based on deep learning model is proposed, and the attention mechanism is introduced into LSTM network to construct a high-performance text classification model.

3. Proposed Research Methods

3.1. Text Preprocessing. The text of tourism questions is different from the formal and standardized text published by traditional media. The text of tourism questions is usually very short and no more than 130 words at most, including punctuation, slang, abbreviations of specific terms, user nicknames, and other contents. These contents have brought great noise interference to the text emotion classification.

In order to remove unnecessary noise interference, the related technologies in natural language processing are used to preprocess the text. First, this paper uses Jieba word segmentation tool to segment each comment text; then, based on the stop words list provided by Baidu, the stop words are removed, and then the noise is removed. When removing noise, it mainly deals with slang, abbreviations of specific terms, user nicknames, punctuation, and other strings involved.

3.2. Attention-Based LSTM Text Classification Model. The research goal is to solve the problem of text classification, which is mainly divided into three parts: text data representation, text feature extraction, and text classifier. The structure of the attention-Based LSTM text classification model is shown in Figure 1, which is mainly composed of word vector representation part, feature extraction part, and classifier part.

Through the analysis of common text data representation technology, it is decided to use word embedding technology to complete the representation of text data. The word vector is obtained through the word embedding language model. In the feature extraction part, according to the characteristics of text classification corpus, this paper uses the attention-based LSTM model as the feature extraction model. The model uses the LSTM model as the coding model and adds the attention model mechanism to calculate the attention probability, i.e., influence weight, of the text sequence for the overall semantic information, and optimizes the feature vector. In the text classifier part, the logistic regression method is used as the classifier. The logistic regression classifier is simple and efficient and can be easily combined with the feature extraction model.

3.3. Text Word Vector Representation. The corpus is trained with Google's open source tool word2vec model to obtain the vector representation of text words. Word vector can capture the complex mapping from words in corpus to real dimensional vector space. Specify word vector space as $\boldsymbol{\varphi}$, its size is $|\boldsymbol{\varphi}| \times m$, each line in $\boldsymbol{\varphi}$ represents m dimensional word vector of a word, and $|\boldsymbol{\varphi}|$ represents the number of words contained in word vector. A comment text T in the corpus can be expressed as the following sequence:

$$(c_1, c_2, \cdots, c_n), \tag{1}$$

where n represents the number of words in text T; c_i stands for the i $(1 \leq i \leq n)$ word in T. If T is converted into a word vector matrix, first search the word vector corresponding to word c_i in $\boldsymbol{\varphi}$. If it exists, select the corresponding word vector and represent it with $\boldsymbol{C}_i$, otherwise, set the corresponding word vector $\boldsymbol{C}_i = 0$. After finding the word vector corresponding to each word, stack each word vector to form a word vector characteristic matrix $\boldsymbol{C}$, whose size is $n \times m$. Each line of $\boldsymbol{C}$ represents the word vector corresponding to a word in the corpus, which can be expressed as

$$(c_1, c_2, \cdots, c_n) \Rightarrow (\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_n)^T. \tag{2}$$

3.4. Feature Extraction. Text emotion classification is mainly based on the key emotional words expressing views, feelings, and attitudes in the text to judge the text emotion tendency, among which the words with strong emotional color play a key role in judging the text emotion tendency [15, 16]. In order to fully reflect the role of emotional keywords in the process of text emotion classification, this paper proposes a text emotion classification model based on LSTM with attention mechanism. The model adds attention mechanism on the LSTM based network, which mainly distributes the weight of emotional information of words and highlights the impact of different words in the input text on the emotional category of the text [17, 18].

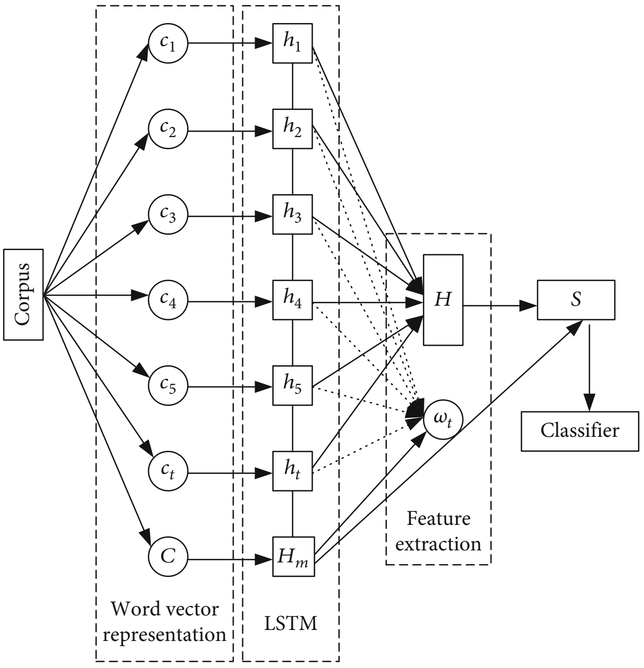


Figure 1: Structure of attention-based LSTM text classification model.

3.4.1. LSTM Network Structure. In this paper, LSTM neural network structure is used as the core component of tourism question text emotion classification model. LSTM neural network structure not only has the advantages of traditional recurrent neural network (RNN), overcomes the problem of RNN gradient explosion or disappearance, but also can effectively process sequence data of arbitrary length and capture long-term dependence of data [19, 20]. The LSTM network structure is shown in Figure 2.

Taking the multifeature representation of words with emotion vector as the input of LSTM, the hidden layer state value corresponding to the input is obtained. The specific calculation of LSTM neural network memory cell is as follows:

$$\begin{aligned}
f_t &= \delta\left(\boldsymbol{\omega}_f[h_{t-1}, \boldsymbol{x}_{ct}] + \boldsymbol{b}_f\right) + \delta\left(\boldsymbol{\omega}_f\left[h_{t-1}, \boldsymbol{x}_{qt}\right] + \boldsymbol{b}_f\right), \\
i_t &= \tanh\left(\boldsymbol{\omega}_i[h_{t-1}, \boldsymbol{x}_{ct}] + \boldsymbol{b}_i\right) + \tanh\left(\boldsymbol{\omega}_i\left[h_{t-1}, \boldsymbol{x}_{qt}\right] + \boldsymbol{b}_i\right), \\
C_t &= f_t \cdot C_{t-1} + i_t * \tilde{C}_t, \\
o_t &= \delta(\boldsymbol{\omega}_o[h_{t-1}, \boldsymbol{x}_{ct}] + \boldsymbol{b}_o) + \delta\left(\boldsymbol{\omega}_o\left[h_{t-1}, \boldsymbol{x}_{qt}\right] + \boldsymbol{b}_o\right), \\
h_t &= o_t * \tanh(C_t),
\end{aligned} \tag{3}$$

where $\boldsymbol{x}_t = [\boldsymbol{x}_{ct}, \boldsymbol{x}_{qt}]$ represents the input at time t; $\boldsymbol{x}_{ct}$ and $\boldsymbol{x}_{qt}$ represent the word meaning vector and emotion vector, respectively; h_t represents the output at time t; i_t represents whether some information in the input door needs to be updated; f_t is the output matrix of the

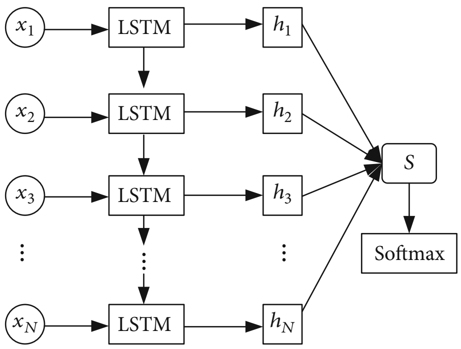


Figure 2: LSTM network structure.

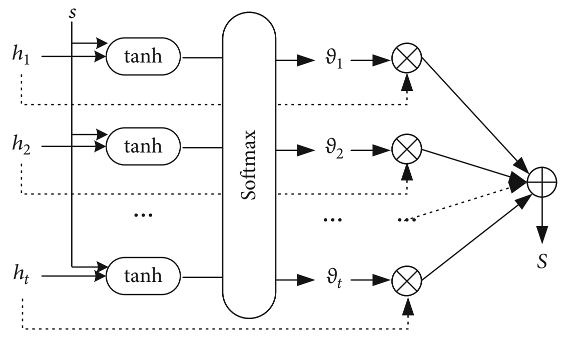


Figure 3: Calculation process of attention mechanism.

forgetting gate; $\boldsymbol{\varpi}$ is the weight matrix; $\boldsymbol{b}$ is the offset vector; δ is sigmoid nonlinear activation function.

3.4.2. Attention Mechanism. The emotional classification of text not only needs to consider the context relationship between words but also needs to consider which words are more prominent in the expression of text emotional classification. Words with greater emotional contribution should be given higher weight or attention [21, 22]. Aiming at the problem that the emotional features cannot be effectively highlighted in the process of text emotional classification, and the proposed method constructs an LSTM text classification model based on attention mechanism, which focuses on the emotional information of text data and further improves the expression ability of text features. In this model, the word emotion influence weight is determined based on the correlation between the output h_t of each hidden layer and the context vector $\boldsymbol{s}$. The calculation process of attention mechanism is shown in Figure 3.

The calculation of attention mechanism can be realized in two steps:

Step 1. Calculate the attention distribution on all input information, that is, take the context vector $\boldsymbol{s}$ and the output h_t of the hidden layer as inputs, enter a single-layer perceptron, and obtain the implicit representation u_t of the result through calculation. The calculation formula is as follows:

$$u_t = \tanh\left(\boldsymbol{\alpha} h_t + \boldsymbol{\beta s}\right), \tag{4}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the weight matrix; h_t is the output of the hidden layer; $\boldsymbol{s}$ is the query vector. Then, $\boldsymbol{\vartheta}_t$ is obtained through softmax operation, which is calculated as follows:

$$\boldsymbol{\vartheta}_t = \text{soft max}\left(u_t\right), \tag{5}$$

where the probability vector composed of $\boldsymbol{\vartheta}_t$ is the emotional attention distribution of the word.

Step 2. Calculate the weighted sum of the input information according to the attention distribution $\boldsymbol{\vartheta}_t$, that is, the attention distribution $\boldsymbol{\vartheta}_t$ represents the correlation between the time t information in the input information vector $\boldsymbol{H}$ and the query $\boldsymbol{s}$ when a query $\boldsymbol{\vartheta}_t$ is given.

The input information is summarized by weighted summation to obtain the attention value. The specific calculation is as follows:

$$S = \sum_{t=1}^{N} \boldsymbol{\vartheta}_t h_t. \tag{6}$$

3.5. Classifier. The text classification model based on attention-based LSTM uses softmax as the output layer for normalization calculation, and combined with the cross entropy loss function, the objective function is expressed as follows:

$$\text{Loss} = -\sum_{i=1}^{K} \boldsymbol{Y}_i \log\left(\boldsymbol{y}_i\right), \tag{7}$$

where K represents the number of texts in the corpus, $\boldsymbol{Y}_i$ represents the real probability distribution vector of the current text category, $\boldsymbol{y}_i$ represents the probability distribution vector of the current text predicted by the classification model, and the dimension of the vector is equal to the number of classification labels. By minimizing the objective function, the classification model can be obtained [23, 24].

The model based on attention mechanism generally includes two parts: one is the calculation process of attention probability distribution, and the other is the calculation process based on the final characteristics of attention distribution. In this model, the attention probability of the output data at time t to the final state is calculated as follows:

$$v_t = \frac{\exp\left(h'_t\right)}{\sum_{i=1}^{N} \exp\left(h'_i\right)}, h'_t = h_t^T \widehat{\boldsymbol{\varpi}} F. \tag{8}$$

Softmax function is the calculation method of attention probability distribution, where N represents the number of input sequence elements; $\widehat{\boldsymbol{\varpi}}$ is the weight matrix; F represents the sum of the final hidden layer state values in each independent direction in the LSTM; h_t represents the sum of the hidden layer state values at time t.

In the proposed model, the final feature F_{final} is obtained based on the attention distribution, and the calculation process is expressed as follows:

Table 1: Experimental environment.

Environmental parameters	Configuration
Operating system	Ubuntu 14.04.5
Development language	Python
Development framework	Tensorflow
Memory	256G
CPU	Intel(R) Xeon(R) CPU E5-2620
GPU	NVIDIA corporation GM200

Table 2: Statistical results of data sets.

Category	Training set	Test set	Validation set	Average text length
Place	1226	525	105	32
Time	1397	598	120	38
Entity	1329	569	114	63
Figures	1215	521	104	75
Description	1691	725	145	22
Character	143	61	12	2534

Table 3: LSTM parameter setting.

Parameter	Value	Parameter	Value
LSTM network layer	1 layer	numClasses	2
Batch_size	128	Dropout	0.7
LSTM_size	256	Loss function	Cross entropy
Learning rate	0.0001	Optimizer	RMSProp optimizer

$$F_{\text{final}} = \sum_{t=1}^{N} v_t h_t. \tag{9}$$

After obtaining the text feature vector F_{final} based on the attention mechanism, the probability distribution of the classification label is calculated through the Softmax function of the output layer. The calculation process is expressed as follows:

$$y = \text{soft max}\left(F^{\prime}_{\text{final}}\right) = \frac{\exp\left(F^{\prime}_{\text{final}(i)}\right)}{\sum_{j=1}^{T}\exp\left(F^{\prime}_{\text{final}(j)}\right)}, \quad F^{\prime}_{\text{final}} = \boldsymbol{\omega}_o F_{\text{final}} \tag{10}$$

where D is the number of category labels; $\boldsymbol{\omega}_o$ represents the weight matrix of the model output layer; $F^{\prime}_{\text{final}(i)}$ represents the i component value in vector $F^{\prime}_{\text{final}}$, and the vector length is equal to the number of classification labels. After the softmax function, the probability distribution y of text category based on the attention mechanism is obtained, and the cross entropy loss is calculated with the real category distribution Y, which is expressed as

$$E(Y, y) = -Y \log (y), \tag{11}$$

Table 4: Classification discrimination confusion matrix.

Real results	Prediction results	
	In category A	Not in category A
In category A	r	l
Not in category A	g	z

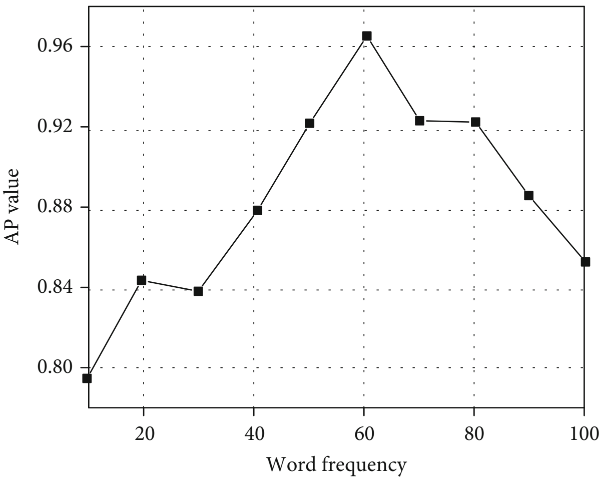


Figure 4: AP value change curve based on word frequency.

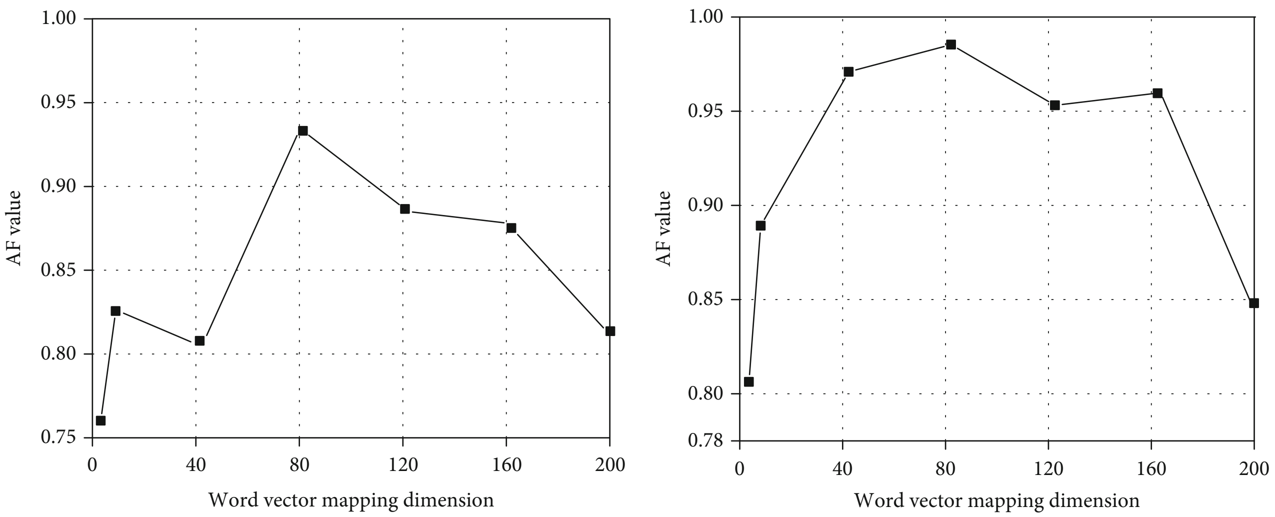
where Y represents the probability distribution of the real category; y represents the probability distribution of the category predicted by the model.

4. Experiment and Analysis

4.1. Experimental Setup

4.1.1. Hardware Environment. The experiment is implemented based on the deep learning framework TensorFlow, which is a deep learning framework based on graph calculation. It uses the data flow between nodes to transfer data and completes the calculation in the nodes. As an open source framework, Tensorflow integrates several models including convolutional neural network, RNN network, and LSTM model [25]. The emergence of Tensorflow framework makes the use of deep learning model simpler and convenient and reduces the difficulty of applying deep learning model [26]. The specific experimental environment is shown in Table 1.

4.1.2. Experimental Data Set. The tourism text data set is used as the experimental data set, which is a user-defined benchmark data set, mainly from tourism websites such as Ctrip, Tuniu, Ma honeycomb, and Tongcheng, including 6 categories of 10000 sample data such as tourism location, time, and people. Before the experiment, the data set needs to be preprocessed such as selection, cleaning, and stop words to reduce errors. In order to verify the effectiveness of the proposed method, 70% of the samples are randomly selected as the training set, the remaining 30% of the samples are used as the test set, and 20% of the samples in the training set are randomly divided as the cross-validation set (the experiment has conducted 6 cross-validation). The statistical results of the data set are shown in Table 2.



(a) Ref. [13]

(b) Proposed method

Figure 5: F value of text classification method for tourism questions with different word vector mapping dimensions.

4.1.3. LSTM Parameter Setting. Based on LSTM neural network structure, a text emotion classification model based on attention mechanism which can express word tag relationship is constructed. The LSTM neural network adopts one-layer network structure, the number of hidden nodes is 256, and the learning rate is 0.0001. The optimization algorithm adopts RMSPropOptimizer optimizer. The specific parameter settings are shown in Table 3.

4.2. Evaluating Indicator. The accuracy P, recall R, and F values are selected as the evaluation indexes, and the classification discrimination confusion matrix is shown in Table 4.

P represents the proportion of samples of real category among the samples predicted to be a category after emotion classification of the test set, that is

$$P = \frac{r}{r+g}. \quad (12)$$

R represents the proportion of a category predicted as a real category in all real categories in the test set, that is

$$R = \frac{r}{r+l}. \quad (13)$$

In order to comprehensively consider the accuracy P and recall R, the weighted harmonic average F of the two is used to measure the final classification effect, that is

$$F = \frac{2 \times P \times R}{P+R}. \quad (14)$$

The task of text emotion classification is oriented to multi classification. Therefore, after calculating the accuracy P and recall R corresponding to each category, the average accuracy (AP), average recall (AR), and average F (AF) corresponding to the three categories are used as the evaluation indexes to measure the performance of emotion classifier.

4.3. Model Training. When training the classification model, the number of iterations is set to 50, and the relationship between AP value and word frequency is shown in Figure 4.

As can be seen from Figure 4, as the word frequency increases, the AP value also increases gradually until the word frequency reaches the optimal value when the word frequency is 60. The AP value exceeds 0.96, and then the AP value decreases with the increase of word frequency. Therefore, when the word frequency is set to 60 in the training of deep learning model, its classification performance is the best.

4.4. Influence of Word Vector Dimension on Model Performance. The word vector mapping dimension plays an important role in the classification accuracy of the model. Therefore, we change the word vector mapping dimension to verify its impact on the text classification accuracy of tourism questions. At the same time, in order to demonstrate the classification accuracy of the proposed method, it is compared with Ref. [13]. The AF values of the two methods in tourism text dataset under different word vector embedding dimensions are shown in Figure 5.

As can be seen from Figure 5, with the increase of word vector mapping dimension, the AF value of the proposed tourism question text classification method first increases rapidly. When the word vector mapping dimension is greater than 80, the AF value stops increasing and begins to decrease, which indicates that too low word vector mapping dimension cannot better map the text to low-dimensional space, and high-dimensional embedding may lead to too sparse vector representation. Therefore, it cannot effectively improve the classification performance and will consume more training time. However, compared with Ref. [13], the AF value of the proposed method is higher as a whole, and when the word vector dimension is in the range of 40~140, the AF value fluctuates less. This is because it adopts LSTM network, which can better map text.

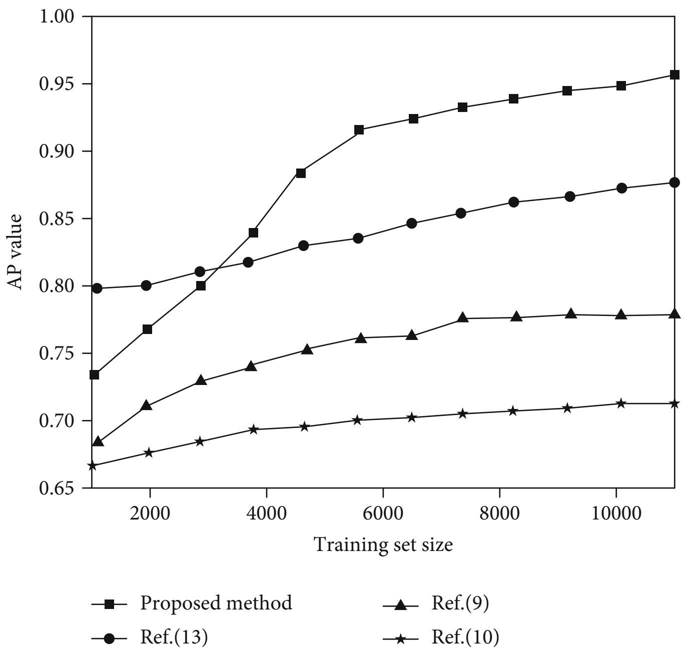


Figure 6: Comparison results of AP values of different methods.

Table 5: Comparison results of classification performance of different methods.

	Ref. [9]	Ref. [10]	Ref. [13]	Proposed method
AP	0.809	0.783	0.878	0.943
AR	0.752	0.664	0.839	0.867
AF	0.779	0.719	0.858	0.903

4.5. Comparison of Classification Performance of Different Methods. According to the size of the training set, the proposed method is compared with the methods in $T = \text{Ref}.$ [9, 10] and [13]. The results are shown in Figure 6.

It can be found from Figure 6 that with the increase of training set, the AP value of various methods tends to be stable. Compared with other methods, the AP value of the proposed method is the highest and close to 0.952. The attention-based LSTM text classification model can effectively improve the classification effect of tourism question text and combined with the cross entropy loss function training model to further ensure the classification performance of the model. Ref. [13] uses a single deep learning model for text classification. Due to the lack of emotional consideration, the AP value is about 0.075 lower than the proposed method. Both Ref. [9] and Ref. [10] adopt traditional methods, so the overall text classification performance is poor when dealing with complex training sets, and the AP value is lower than 0.80.

In addition, the specific data of AP, AR, and AF obtained from the experiments by the four methods are shown in Table 5.

It can be seen from Table 5 that the overall classification performance of the proposed method is the best, and the values of AP, AR, and AF were 0.943, 0.867, and 0.903, respectively. The proposed method uses the LSTM network to extract the depth feature vector, reduces the dimension of the output feature vector, introduces the attention mechanism to highlight the emotional role, significantly improves the classification performance of question text, and proves that it is robust to the emotional classification of tourism text. Ref. [9] uses the active learning model for text classification. The AF value of traditional machine learning is only 0.779, which is 0.124 lower than that of the proposed method. Ref. [10] uses differential evolution algorithm to realize text classification, but the algorithm does not fully analyze the characteristics of tourism text, and the algorithm performance is poor, so the AP value is only 0.783. Ref. [13] classifies text based on a single deep learning model. However, this method lacks emotional consideration, so its AF value is 0.858, and the whole performance needs to be further improved. It can be demonstrated that the proposed method has good text classification ability of tourism questions.

5. Conclusion

Under the background that self-help travel has become the mainstream form of tourism, tourists can obtain information through Q & A from the Internet platform, but there are problems of delay and inaccurate classification in self-help Q & A. Therefore, a text classification method of tourism questions based on deep learning model is proposed. The text word vector obtained based on the continuous word bag model is input into the attention-based LSTM model for feature extraction, and the probability distribution of text category is obtained by Softmax classifier. The proposed method is experimentally analyzed using the tourism text data set, and the results show that the LSTM model can effectively capture the relationship between word vectors. When the word frequency is set to 60 and the word vector dimension is 80, the AP value of the model exceeds 0.96. The introduction of attention mechanism can better highlight the role of emotion and improve the accuracy of text classification of tourism questions. The AP, AR, and AF were 0.943, 0.867, and 0.903, respectively, which were better than other comparison methods. However, the proposed method uses Softmax function for task calculation. In the next research, some acceleration methods, such as hierarchical Softmax and negative sampling technology, can be considered to improve the overall performance of the classification model.

Data Availability

The data included in this paper are available without any restriction.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

We wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper. This research was jointly funded by the Sichuan Science and Technology Program, Grant/Award numbers: 2019ZYZF0169; and the A Ba Achievements Transformation Program, Grant/Award numbers: 19CGZH0006 and 21CGZH0002.

References

[1] C. D. Cottrill, “MaaS surveillance: privacy considerations in mobility as a service,” *Transportation Research Part A: Policy and Practice*, vol. 131, no. 8, pp. 50–57, 2020.

[2] W. Wang and A. Feng, “Self-information loss compensation learning for machine-generated text detection,” *Mathematical Problems in Engineering*, vol. 2021, Article ID 6669468, 7 pages, 2021.

[3] A. Mignan, “A preliminary text classification of the precursory accelerating seismicity corpus: inference on some theoretical trends in earthquake predictability research from 1988 to 2018,” *Journal of Seismology*, vol. 23, no. 4, pp. 771–785, 2019.

[4] F. Zhao, Y. Li, L. Bai, Z. Tian, and X. Wang, “Semi-supervised multi-granularity CNNs for text classification: an application in human-car interaction,” *IEEE Access*, vol. 8, no. 99, pp. 68000–68012, 2020.

[5] S. G. Burdisso, M. Errecalde, and M. Montes-y-Gómez, “A text classification framework for simple and effective early depression detection over social media streams,” *Expert Systems with Applications*, vol. 133, no. 11, pp. 182–197, 2019.

[6] M. Sokolowska, M. Mazurek, M. Majer, and M. Podpora, “Classification of user attitudes in twitter -beginners guide to selected machine learning libraries,” *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 394–399, 2019.

[7] M. M. Mirończuk, J. Protasiewicz, and W. Pedrycz, “Empirical evaluation of feature projection algorithms for multi-view text classification,” *Expert Systems with Applications*, vol. 130, no. 4, pp. 97–112, 2019.

[8] X. Luo, “Efficient english text classification using selected machine learning techniques,” *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, 2021.

[9] A. Varghese, T. Hong, C. Hunter, G. Agyeman-Badu, and M. Cawley, “Active learning in automated text classification: a case study exploring bias in predicted model performance metrics,” *The Environmentalist*, vol. 39, no. 3, pp. 269–280, 2019.

[10] C. Padurariu M. E. Breaban et al., “Dealing with data imbalance in text classification,” *Procedia Computer Science*, vol. 159, pp. 736–745, 2019.

[11] S. Yilmaz and S. Toklu, “A deep learning analysis on question classification task using Word2vec representations,” *Neural Computing and Applications*, vol. 32, no. 7, pp. 2909–2928, 2020.

[12] M. Oleynik, A. Kugic, Z. Kasáč, and M. Kreuzthaler, “Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification,” *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1247–1254, 2019.

[13] A. Varghese, G. Agyeman-Badu, and M. Cawley, “Deep learning in automated text classification: a case study using toxicological abstracts,” *Environment Systems and Decisions*, vol. 40, no. 4, pp. 465–479, 2020.

[14] B. Wang, X. Hu, P. Li, and P. S. Yu, “Cognitive structure learning model for hierarchical multi-label text classification,” *Knowledge-Based Systems*, vol. 218, no. 3, pp. 106876–106887, 2021.

[15] D. Petschke and T. Staab, “A supervised machine learning approach using naive Gaussian Bayes classification for shape-sensitive detector pulse discrimination in positron annihilation lifetime spectroscopy (PALS),” *Section A, Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 947, no. 12, pp. 162742–162742.9, 2019.

[16] B. Zhong, X. Xing, P. Love et al., “Convolutional neural network: deep learning-based classification of building quality problems,” *Advanced Engineering Informatics*, vol. 40, no. 7, pp. 46–57, 2019.

[17] Z. Chen and J. Ren, “Multi-label text classification with latent word-wise label information,” *Applied Intelligence*, vol. 51, no. 2, pp. 966–979, 2021.

[18] Y. Zhu, W. Zheng, and H. Tang, “Interactive dual attention network for text sentiment classification,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8858717, 11 pages, 2020.

[19] K. Purwandari, J. W. C. Sigalingging, T. W. Cenggoro, and B. Pardamean, “Multi-class weather forecasting from twitter using machine learning aprroaches,” *Procedia Computer Science*, vol. 179, no. 4, pp. 47–54, 2021.

[20] A. Mohasseb, M. Bader-El-Den, and M. Cocea, “A customised grammar framework for query classification,” *Expert Systems with Applications*, vol. 135, no. 11, pp. 164–180, 2019.

[21] B. Stasak, J. Epps, and R. Goecke, “Automatic depression classification based on affective read sentences: opportunities for text-dependent analysis,” *Speech Communication*, vol. 115, no. 6, pp. 1–14, 2019.

[22] B. André Sumithra et al., “Text classification to inform suicide risk assessment in electronic health records,” *Studies in Health Technology and Informatics*, vol. 264, no. 3, pp. 40–44, 2019.

[23] T. Henry, D. Banks, D. Owens-Oas, and C. Chai, "Modeling community structure and topics in dynamic text networks," *Journal of Classification*, vol. 36, no. 2, pp. 322–349, 2019.

[24] Y. Baghdadi, A. Bourrée, A. Robert et al., "Automatic classification of free-text medical causes from death certificates for reactive mortality surveillance in France," *International Journal of Medical Informatics*, vol. 131, no. 11, article 103915, 2019.

[25] M. Hashemi, "Web page classification: a survey of perspectives, gaps, and future directions," *Multimedia Tools and Applications*, vol. 79, no. 17-18, pp. 11921–11945, 2020.

[26] F. Beretta, Á. L. Rodrigues, R. Peroni, and J. F. C. L. Costa, "Using UAV for automatic lithological classification of open pit mining front," *REM-International Engineering Journal*, vol. 72, no. 1, Supplement 1, pp. 17–23, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1616725, 16 pages
https://doi.org/10.1155/2021/1616725

Research Article

Heart Failure Detection Using Quantum-Enhanced Machine Learning and Traditional Machine Learning Techniques for Internet of Artificially Intelligent Medical Things

Yogesh Kumar,[1] Apeksha Koul,[2] Pushpendra Singh Sisodia,[1] Jana Shafi,[3] Kavita Verma,[4] Mehdi Gheisari,[5] and Mohamad Bagher Davoodi[6]

[1]*Indus Institute of Technology & Engineering, Indus University, Ahmedabad 382115, India*
[2]*Department of Computer Engineering, Punjabi University, Patiala 147002, India*
[3]*Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdul Aziz University, Wadi Ad-Dwasir 11991, Saudi Arabia*
[4]*Department of Computer Science and Engineering, Chandigarh University, Mohali 140413, India*
[5]*Young Researchers and Elite Club, Parand Branch, Islamic Azad University, Parand, Iran*
[6]*Bounty Company, Iran*

Correspondence should be addressed to Mehdi Gheisari; mehdi.gheisari61@gmail.com

Received 10 October 2021; Revised 30 October 2021; Accepted 1 December 2021; Published 17 December 2021

Academic Editor: Mohammad R Khosravi

Quantum-enhanced machine learning plays a vital role in healthcare because of its robust application concerning current research scenarios, the growth of novel medical trials, patient information and record management, procurement of chronic disease detection, and many more. Due to this reason, the healthcare industry is applying quantum computing to sustain patient-oriented attention to healthcare patrons. The present work summarized the recent research progress in quantum-enhanced machine learning and its significance in heart failure detection on a dataset of 14 attributes. In this paper, the number of qubits in terms of the features of heart failure data is normalized by using min-max, PCA, and standard scalar, and further, has been optimized using the pipelining technique. The current work verifies that quantum-enhanced machine learning algorithms such as quantum random forest (QRF), quantum K nearest neighbour (QKNN), quantum decision tree (QDT), and quantum Gaussian Naïve Bayes (QGNB) are better than traditional machine learning algorithms in heart failure detection. The best accuracy rate is (0.89), which the quantum random forest classifier attained. In addition to this, the quantum random forest classifier also incurred the best results in $F1$ score, recall and, precision by (0.88), (0.93), and (0.89), respectively. The computation time taken by traditional and quantum-enhanced machine learning algorithms has also been compared where the quantum random forest has the least execution time by 150 microseconds. Hence, the work provides a way to quantify the differences between standard and quantum-enhanced machine learning algorithms to select the optimal method for detecting heart failure.

1. Introduction

Quantum computing is a revolutionary concept based on the fundamental principles of nature, i.e., quantum mechanics. With the advancements in physics in the early twentieth century, methods of observation and purity of materials reached the level at which some quantum phenomena became detectable [1], such as regular transistors present in every modern computer or device. It is operated by directing large clouds of carriers of electrical current using engineered materials and quantum-based principles (band structure, localized states, etc.). They produce unusual behaviour for naturally found materials—an ability to precisely control current with current, or current via light, or light via current [2]. Quantum-enhanced machine learning is at the junction of both current research areas: quantum computing and machine learning. It prospects the interaction between machine learning and quantum computing to inquire how outcomes and results of approaches from one area can be utilized to compute the issues of the other field. With a burgeoning amount of data, current machine learning systems rapidly intend to confine classical computational models. In this sense, quantum computational power can offer an advantage in machine learning tasks [3]. Currently, quantum technology is divided into three distinct fields: quantum computing, quantum information, and quantum cryptography. Quantum computation exerts its effect due to the generous permutations that make quantum computers twice as fast as memory fills up with the addition of each qubit. As a result, we require N-bits of binary numbers to describe the N-bits classical bit system. Recent research breakthroughs have made significant contributions to the advancement of machine learning algorithms by using the benefits of quantum computing. However, tremendous work has been done to design and implement quantum versions [4] of ANN. Besides this, they are also based on more natural aspects that are yet to be accomplished [5]. Some authors tried to develop a complete quantum algorithm that could achieve pattern recognition problems [6]. In contrast, others suggested implementing subprograms of traditional machine learning algorithms on a quantum system. The adiabatic quantum-enhanced machine learning method appears to be applicable to some kinds of optimization problems [7, 8].

Despite such tremendous advancements in the medical domain, heart failure has posed an immense threat to changing patients' health. Recent years have witnessed a considerable increase in mortality and morbidity due to heart failure threat to lives [9]. It challenges healthcare providers as it is leading to tremendously high rates of mortality and morbidity. It results from the defects in the myocardium, which further results in the ejection of blood or impairment of ventricular filling. Based on the location, heart failure (HF) can be classified as biventricular left ventricular or right ventricular. It can also be classified as chronic or acute. It is found that females and aged people suffer from HFpEF (heart failure with preserved ejection fraction) [10]. The significant symptoms of heart failure include shortness of dyspnea, orthopnea, nocturnal dyspnea, lethargicness, pedal edema, tachycardia, jugular venous pressure, and S3 gallop [5].

Thus, traditional machine learning and quantum-enhanced machine learning algorithms in healthcare are extensively used to aid the patients and medical staff in many diverse ways. These algorithms have been used for heart failure imaging applications, the contribution of its risk evaluation in different ways, and predicting heart failure detection. Researchers deployed Rubidium-based quantum sensors in one research [7] to detect signals made by atrial fibrillation, a disease that creates an unpredictable and abnormally high heartbeat. They had seen an array of their quantum sensors that can be positioned over the heart for supplying primary data. They also suggested that quantum technologies can increase the clinical results of atrial fibrillation.

In addition to this, the Internet of Medical Things (IoMT) has also been considered to be the wave of the future in the field of healthcare. It is referred to as a collection of medical devices and apps connected to healthcare systems via online computer networks [11]. The Internet of Medical Things (IoMT) comprises smart devices, such as wearables and medical monitors, that are created for healthcare reasons and may be utilized on the human body, at home, in the community, and clinical settings, among other places. Numerous healthcare providers are adopting IoMT apps to enhance treatments and illness management, minimize mistakes, improve patient experience, manage medications, and lower costs [12]. In addition to this, IoMT also reduces the number of needless hospitalizations and the total load on healthcare systems by linking patients directly to their physicians and facilitating the transfer of medical data through a secure network instead of traditional methods. According to a new assessment by Deloitte, its influence on the healthcare business is evident and permanent as it is predicted to increase from $41 billion in 2017 to $158 billion by 2022. Hence, in a nut shell, advancements in wireless communications, sensor networks, mobile devices, big data analysis, and cloud computing, the Internet of Medical Things (IoMT) is transforming the healthcare industry by delivering targeted and personalized medicine and also enable seamless communication of medical data between healthcare providers and patients.

Therefore, it can be said that traditional and quantum-enhanced machine learning algorithms for the Internet of Medical Things (IoMT) aids in making an earlier diagnosis of diseases and identifies predictive characteristics in different pathologic conditions.

To assess where and how traditional and quantum-enhanced machine learning techniques may provide opportunities for detection of heart failure, it is essential to understand the use and applicability of the different techniques such as QKNN, quantum Naïve Bayes, quantum decision tree, and quantum random forest. The primary concern of the presented paper is to highlight the usage and efficiency of the traditional and quantum-enhanced machine learning algorithms along with its comparative analysis based on accuracy, $F1$-score, recall, and precision by using scalar (min-max, standard, and PCA) and pipelining technique.

1.1. Contribution. Quantum computers are based on quantum physics, which allows them to operate far faster than conventional computers without requiring massive hardware systems. Thus, quantum computing can assist computers in achieving faster processing speeds and overcoming more composite problems. This paper puts forward a systematic and experimental study on quantum-enhanced machine learning and its utilization in different research areas. In this article, implemented work has been done to detect heart failure using various attributes with the help of traditional and quantum-enhanced machine learning algorithms. Moreover, we have also shown the comparison between them to have the best algorithm with the highest accuracy rate, $F1$ score, recall, precision, and less computation time. The significant intention of the paper is to highlight the importance of quantum-enhanced machine learning in the healthcare sector. The contributions of the article are as follows:

(1) It provides an inclusive knowledge of quantum-enhanced machine learning

(2) It walks through recent trends and existing applications and systems based on quantum computing

(3) A detailed study of applications of quantum-enhanced machine learning in the healthcare domain is presented

(4) It also shows the implementation of quantum-enhanced machine learning algorithms for the detection of heart failure and compared it with the traditional machine learning algorithms for different parameters such as accuracy, precision, recall, and $F1$-score using a different scale and transformation techniques such as min-max, standard, PCA, and other pipelining technique to optimize the results

2. Process of Traditional Machine Learning and Quantum-Enhanced Machine Learning

The core of machine learning is to train the machine using algorithms that have been executed to handle the data. The traditional methods of machine learning, via its subsets of supervised and unsupervised learning, i.e., deep learning, helps in classifying the images and identifying the patterns and speech, and controls big data, etc. [13]. Nowadays, it becomes necessary to have new approaches to manage, organize, and classify the diverse range of available data [14].

The long-established machine learning has received much heed and investments from different organizations and industries [15]. The industries whose labor is involved in effective data warehouse management are capable of handling a diverse range of data and are very keen to know new approaches to executing this. Quantum-enhanced machine learning is a promising solution that can be used to eradicate such limitations [16]. One of the problems to be resolved in quantum-enhanced machine learning is the vulnerability available in the input data that the proposed methodologies can treat and handle effectively [17]. The critical principle that differentiates quantum-enhanced machine learning in traditional algorithms is how the learning takes place on different data. Unlike conventional algorithms, which mainly focus on design, quantum algorithms are versatile and can effectively solve different problems. Adaptive learning autonomously finds the possible set of behaviors and patterns to solve a complex problem [18].

Although traditional machine learning [19] has been observed to be a flexible and adaptive procedure that can effectively map different patterns, some complex problems still exist that cannot be efficiently solved by the conventional method of machine learning algorithms. Comparing regular machine learning and quantum-enhanced machine learning processes is depicted in Figure 1, which compares the two methods. In the case of traditional machine learning, features are extracted, and then, machine learning algorithms are applied to predict the output. However, in the case of quantum machine learning, the data is subjected to prepare the model where a unitary and complexity system is developed. Then, features are extracted, after which quantum-enhanced machine learning algorithms are applied to predict the outcome. Therefore, the companies whose labor is involved in large record storage organizations are very interested in learning new approaches. Hence, the quantum-enhanced machine learning domain is found to be one of these promising approaches. However, the interest to implement these techniques through quantum computation paves the way to quantum-enhanced machine learning [20].

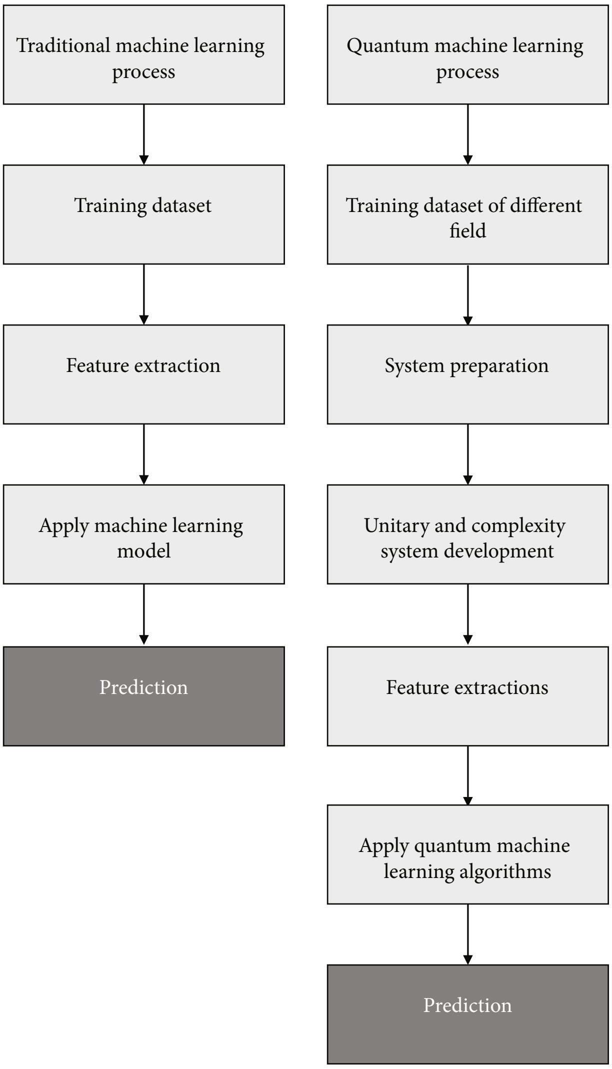

Traditional machine learning process
Training dataset
Feature extraction
Apply machine learning model
Prediction
Quantum machine learning process
Training dataset of different field
System preparation
Unitary and complexity system development
Feature extractions
Apply quantum machine learning algorithms
Prediction

Figure 1: Traditional machine learning and quantum-enhanced machine learning process.

2.1. Working of Quantum-Enhanced Machine Learning. Quantum-enhanced machine learning is a technique developed to augment the algorithms used in regular machine learning. Quantum-enhanced machine learning is a subfield of quantum information processing research [3] that focuses on developing machine learning algorithms capable of learning from data. Quantum computers compute information using the principles of quantum theory, and quantum algorithms are a collection of assertions that execute on these systems [20]. Many quantum algorithms have been designed for machine learning techniques such as neural networks and graphical models. Quantum-enhanced machine learning algorithms mostly rely on the Grover search, which speeds up unordered datasets [21]. Many quantum-enhanced machine learning-based algorithms have been developed for pattern recognition and data extraction [22]. Quantum-enhanced machine learning-based algorithms use quantum computing to solve complex issues for ordinary machine learning algorithms and are executed by acclimatizing established procedures to implement on a potential quantum system [23]. With this pace of advancement, it can be observed that such machines are rapidly used for applications in the coming era, which will then ease the process of analyzing universal information. The rising field also incorporates techniques, namely, robust machine learning methods, that can extend quantum information theory [24]. Quantum-enhanced machine learning is optimizing traditional artificial intelligence systems and is considered one of the future research areas in using machine and deep learning algorithms, as stated in Table 1.

2.2. Role and Platforms of Quantum-Enhanced Machine Learning in Healthcare. Quantum-enhanced machine learning is accelerating its growth and adoption due to its immense potential. Compared to traditional machine learning algorithms, quantum-enhanced machine learning techniques can reduce training time, handle complex network topology, automatically adjust network hyperparameters, perform complex matrix and tensor manipulation at high speeds, and use quantum tunneling to achieve actual objective function goals.

As far as the healthcare industry is concerned, it is using quantum computations to carry a patient-centric concern for medical care clients. Quantum computing and healthcare systems enable hardware solutions that can significantly benefit the healthcare industry in evaluating and treating complicated medical situations [36]. Quantum computing imbues the digital world with quantum physics, enabling computers to process data quicker and solve more complex issues [37]. Additionally, there are several critical concepts associated with quantum computing, including *quantum bits* (*qubits*), which are information units that can exist in either an ON or an OFF state, *quantum superposition*, which allows particles to exist in multiple states and provides tremendous power and flexibility for solving complex problems, *entanglement*, which occurs when pairs or groups of particles are generated, interact, or share spatial proximity, *tunneling* in which a particle goes through what appears to be an energy barrier, and quantum gate that acts on a collection of quantum states called basis states to produce the desired output state.

Machine learning and artificial intelligence are being used with traditional computing resources to interpret CT scans, aid surgical procedures, and analyze big data to develop predictive models of disease [38]. Quantum computing holds no value to medicine without the parallel increase in the availability of clinically meaningful data from numerous sources [9]. Parsons18 [39] predicted that when the first quantum computing devices became operational in roughly 2000, quantum computing applications would shape the future of medical imaging. Quantum computing will be used to analyze diagnostic pictures through the use of artificial intelligence. Not only will picture detail be exponentially increased but physician's interpretation of data will be assisted since effective machine learning can train a quantum computer to detect abnormal discoveries with more precision than the human eye. Second, quantum computing will aid in the development of more effective cancer medicines. Computers are now utilized to manage the myriad of variables involved in developing a radiation plan targeting cancer cells while sparing healthy cells. Machine learning and artificial intelligence have been combined with traditional computer resources to help interpret CT scans, surgical operations, and the analysis of large amounts of data to construct illness prognostic models. Quantum computing is of little use to medicine unless it is accompanied by an increase in the availability of clinically valuable data from various sources. The challenges associated with implementing a healthcare system in which quantum computing analyzes decades of data from billions of clinical encounters and recommends a specific medication to a patient presenting with new-onset depression, cancer, diabetes, or any other condition will be overcome when data from multiple sources, such as existing patient data networks, biobanks, and wearable health devices, are combined.

Thus, quantum computing uses decades of data from billions of clinical encounters and recommends a specific medication for patients presenting with new-onset depression, cancer, diabetes, etc. [40]. Quantum computing can aid in the provision of precise medical imaging and medicines. There are a variety of applications for quantum-enhanced machine learning algorithms, including diagnostic aid, where quantum computing has the potential to improve image analysis, including processing stages such as edge identification and picture matching. Care providers may enhance diagnoses while eliminating the need for recurrent intrusive diagnostic testing, a process known as precision medicine. Precision medicine attempts to customize preventative and treatment methods to the person [41]. Quantum-enhanced machine learning techniques enable earlier, more accurate, and granular risk forecasting, pricing in which quantum algorithms may help superior classification and pattern detection, thereby aiding in discovering abnormal behavior and eliminating fraudulent medical claims. Additionally, in addition to determining the optimal medication, quantum-enhanced machine learning can benefit healthcare systems in a variety of other ways, including the following:

(a) Quantum imaging equipment can provide exact images, enabling the observation of single molecules [14]

Table 1: QML-based different research fields.

Authors	QML algorithm	Research area	Description
Dang et al. [25]	Quantum KNN	Parallel computing	The author used quantum K-nearest-neighbour algorithm to achieve better efficiency in image classification with 83.1% classification accuracy on the Graz-01 dataset and 78% on Caltech-101 dataset.
Lu et al. [26]	Quantum-decision tree	Pattern computation	The author proposed the quantum decision tree model that implemented Neumann entropy in place of Shannon entropy to decide which attribute should be split effectively.
Bang et al. [7]	Quantum C-mean clustering	Diabetes prediction	The authors computed the global optima of the parameters by the enhanced quantum-inspired evolutionary fuzzy C-means (EQIE-FCM) algorithm.
Bharill et al. [27]	Quantum K-mean and quantum fuzzy C -means	Image segmentation	The author proposed four quantum-based clustering algorithms to explore and evaluate the purpose of image segmentation.
Wang et al. [28]	Quantum genetic algorithm	Function optimization	The author proposed a quantum genetic algorithm which is better than the conventional genetic algorithm for computational speed.
Cong et al. [4]	Quantum-CNN	Quantum information theory	The author used quantum-CNN (QCNN) architecture to intertwine the multi-scale entanglement renormalization approach and quantum error correction.
Chen et al. [23]	Quantum-inspired forest	Feature ensembles	The author assigned each principal component a fraction-transition probability where they incorporated the QIS method into random forest and proposed quantum-inspired forest.
Taha et al. [29]	Quantum recurrent neural network	Electro encephalography signals	The author described auto-regressive (AR) model and quantum recurrent neural network (QRNN) and their proposed method, achieving an accuracy of 88.28%.
Wallach et al. [30]	Quantum-neural network	Big data	QNN inherits the basic properties of ANN and contains quantum computing paradigms. It has its application in automated control systems (ACS) and other associative memory devices.
Sosa et al. [31]	Variational quantum classification	Dementia prediction	The authors built a form of variational quantum classification to enable dementia prediction in older people.
Amin et al. [32]	Quantum neural network	COVID-19	For the analysis of COVID-19 pictures, the authors studied quantum machine learning and conventional machine learning methodologies. They achieved 0.94 precision, accuracy, recall, and $F1$-score on POF hospital dataset while 0.96 precision, 0.96 accuracy, 0.95 recall, and 0.96 $F1$-score on UCSD-AI4H dataset.
Gupta et al. [33]	Quantum inspired binary classifier	Diabetes prediction	The authors built a prognostic tool based on the PIMA Indian diabetes dataset to assist physicians in lowering diabetes-related mortality.
Tiwari et al. [34]	Quantum dot synthesis	Healthcare	The authors investigated the basics, synthesis, and applications of quantum dots, emphasizing the healthcare sector.
Martin et al. [35]	Quantum neural network	Healthcare	The authors presented a framework for hybrid quantum machine learning-based health status diagnostics and prognostics.

(b) When combined with quantum computing, machine learning algorithms can assist physicians in understanding therapy outcomes

(c) Machine learning is used to detect irregularities in the human body, while quantum computing aids in the interpretation of therapy outcomes [42]

(d) Radiation beams are utilized in quantum-enhanced machine learning to kill or halt the proliferation of damaged cells [43]

(e) Quantum computers enable clinicians to discover the optimal therapy for each simulation [10]

(f) The capacity of quantum computing to process algorithms has drawn the attention of administrators from a variety of businesses. According to one estimate, the quantum computing business was worth $93 million in 2019 and is expected to grow to $283 million by 2024 [44]

(g) The potential for quantum computing to identify treatments targeting specific forms of cancer significantly adds to its rise in the healthcare business. The amount of money spent on the research and development team, the amount of time spent on research, and the time required for complete radiation analysis will decrease if quantum computing and healthcare systems integrate [13]

Table 2 illustrates the different quantum-enhanced machine learning platforms and their purpose in the healthcare domain, along with their utility of in healthcare.

3. Materials and Methods

This section of the article deals with implementing traditional machine learning algorithms and includes information regarding the dataset considered, and the approaches followed.

3.1. Dataset Description. This section of the paper discusses the dataset that has been considered for detecting heart failure using traditional machine learning and quantum-enhanced machine learning algorithms. The following are the details:

(1) The dataset considered has been collected from the UCI repository

(2) It contains 304 instances with 14 attributes like age, sex, blood pressure, angina, cholesterol, blood sugar, ECG, and maximum heart rate, as shown in Table 3

(3) The target variable is the diseased event to check whether the patient is suffering from heart failure or not

(4) There are no missing values in the considered dataset

Machine learning techniques have proved their robustness in every domain. Thus, we decided to detect heart failure using traditional and quantum-enhanced machine learning algorithms to help medical professionals. Table 3 describes 14 attributes present in the dataset.

The different attributes of the health issues are age which is mainly a significant dangerous cause in budding heart failure. It has been approximated that 82 per cent of the folks who belong to 65 and older are more prone to heart failure. At the same time, the threat of stroke gets doubled every decade after attaining age 55. The second attribute is sex, where males are at higher threat of coronary ailment than premenopausal women. However, if a woman has diabetes, she can have more threat of heart disease than a male. The other factors are chest pain, also known as angina, and is caused when enough oxygen-rich blood does not reach the heart's muscles. The heart patient may feel as if someone is squeezing their chest. The uneasiness can also arise in the shoulders, arms, neck, jaw, or back. The person might also feel digestion issues.

The increased amount of low-density lipoprotein cholesterol, also known as serum cholesterol, on the other hand, is likely to be the cause of arteries collapsing. Uncontrolled blood pressure has the potential to harm the arteries that supply our hearts. When high blood pressure is combined with additional factors like obesity, high cholesterol, or diabetes, the risk of heart failure increases even more. The risk of cardiac arrest is also increased when triglycerides are high. High levels of high-density lipoprotein cholesterol, on the other hand, reduce the risk of a heart attack. A spike in blood sugar levels in the body is another cause of heart failure, which occurs when the pancreas does not generate enough hormones or respond to insulin.

Likewise, achieving maximum cardiac rate is also the cause of heart failure, where the rise in heart failure threat is linked with the escalation of heart rate. For example, it has been observed that the rise in cardiac rate by 10 bpm raises the threat of heart failure by at least 20%. The other factors responsible for heart attack are exercise-induced angina and peak exercise ST-segment. Angina-related discomfort might feel tight, gripping, or squeezing and can range from mild to severe. On the other hand, peak exercise is a treadmill ECG stress test in which pressure is recorded irregularly at a straight or downhill incline when the ST-segment depression is higher than or equal to 1 mm at 60-80 ms.

3.2. Framework. This section of the article discusses the approach followed to functioning traditional and quantum-enhanced machine learning algorithms.

Figure 2 shows the flow that has been followed for the process and its steps are as follows:

Step 1. The dataset was collected from the UCI repository, and its details have already been discussed in the dataset description section.

Table 2: QML-based healthcare platforms [1, 7–9, 23, 38, 45, 46].

Developed platform	Purpose	How to use
PathAI	They developed a machine-learning algorithm to assist pathologists in making precise diagnoses.	Ingenious methods can be developed for different medical treatments, and also, it helps minimize errors in cancer detection.
Enlitic	Artificial intelligence deep learning for actionable insights	It is used to streamline radiology diagnosis and analyses raw and unlabelled medical data such as radiology images and genomics.
Freenome	Earlier cancer detection with AI	Freenome is subjected to discover cancer in its early phases and subsequently helps build new methods of treating cancer.
BioXcel Therapeutics	BioXcel Therapeutics employs AI to re-innovate clinically approved products and medicines.	BioXcel Therapeutics' mission is to reinvent drugs in the immuno-oncology and neuroscience domains. AI is believed to be the only instrument to maximize the value of clinically approved treatments and goods.
BERG Health	Treating rare disease with AI	BERG is an AI-based clinical platform that leverages the treatment by quick mapping of disease, thereby helping develop medicines.
XtalPi	Combining AI-cloud and quantum computing-based digital drug discovery	XtalPi's ID4 platform is an AI-based engine that predicts the chemical and pharmaceutical properties for reinventing drug design and development.
Atomwise	Neural network for clinical trials	Atomwise uses AI technology to configure the most dangerous diseases such as Ebola and multiple sclerosis (MS).
Deep Genomics	Identifying more promising prospects for developing medicines	Deep Genomics is a platform primarily based on AI that can develop drug-related disorders such as neuromuscular and neurodegenerative.
BenevolentAI	Deep learning for targeted treatment	Benevolent is subjected to providing the appropriate treatment to the right patients at the required time by using AI for a better target. Also, it gives unearthed insights with the help of deep learning.
Olive	Automating the most repetitive processes in healthcare	Olive is an AI-based platform built to automate the repetitive task of the healthcare industry and thus saves the precious time of administrators, thereby fostering them to accomplish higher-order tasks.
Qventus	Real-time patient flow optimization	Qventus is a platform that is based on artificial intelligence, which aids in overcoming operational challenges primarily related to emergency rooms and patient safety.
Babylon Health	Increasing access to healthcare	Babylon is an artificial intelligence-based platform that enables patients to access various healthcare-related services via a powerful interactive interface.
CloudMedX	For a better patient journey	CloudMedX provides AI tools to ease the burden by reducing manual work. It works by ingesting structured and unstructured data and then uses this data to get insights by using machine learning algorithms.
Vicarious surgical	Virtual reality-enabled robotics for surgery	Vicarious Surgical aims to improve patient outcomes by fusing virtual reality with robots exhibiting AI technology.
Auris health	AI robots revolutionizing endoscopy	Auris Health is leveraging medical intervention by introducing the Monarch platform. This platform integrates data science and microinstrumentation to bring improvement in endoscopies design and tools.
Intuitive	Pioneering robotic surgery	Intuitive's da Vinci platforms have come up with a pioneering robotic surgery technique in industries.
Microsure	Improving surgical precision	Microsure robots were designed with the ability to aid surgeons with robotic assistance and thereby to overcome human physical limitations.

Step 2. The dataset obtained was preprocessed, which included checking for NULL values and converting string values to float (because string values cannot be directly used in machine learning algorithms), which was not applicable.

Step 3. For implementation, various libraries like time, copy, math, NumPy, matplotlib, sklearn, and most importantly, qiskit for quantum have been used.

Step 4. Then, standardization and transformation of the data have been done with the help of min-max scalar, standard scalar, and PCA scalar to scale all the features. Further, normalization and optimization of the data have been done to obtain better results using the pipelining technique.

Step 5. After this, the data was divided into two parts: training and testing set in the ratio of 8 : 2. The training data were

Table 3: Description of dataset attributes.

Attribute	Description	Input
Age	Age of the patients	Years
Sex	Sex (1 = male; 0 = female)	Float
Trestbps	Resting blood pressure (in mmHg on admission to the hospital)	mm/Hg
CP	Chest pain type– value 1: typical angina – value 2: atypical angina – value 3: nonanginal pain – value 4: asymptomatic	Float
Cholesterol	Serum cholesterol	mg/dl
Fps	(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)	Float
Restecg	Resting electrocardiographic results– value 0: normal – value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) – value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria	Float
Thalach	Maximum heart rate achieved	Binary
Exang	Exercise induced angina (1 = yes; 0 = no)	Int
Old peak	ST depression induced by exercise relative to rest	Continuous
Slope	The slope of the peak exercise ST segment– value 1: upsloping – value 2: flat – value 3: downsloping number of major vessels (0-3) colored by flourosopy	Float
CA	Follow up period number of major vessels (0-3) colored by flourosopy	Float
Thal	3 = normal; 6 = fixed defect; 7 = reversible defect	Float
Target	Whether person suffering through heart disease or not 0 = Normal 1 = suffering	Float

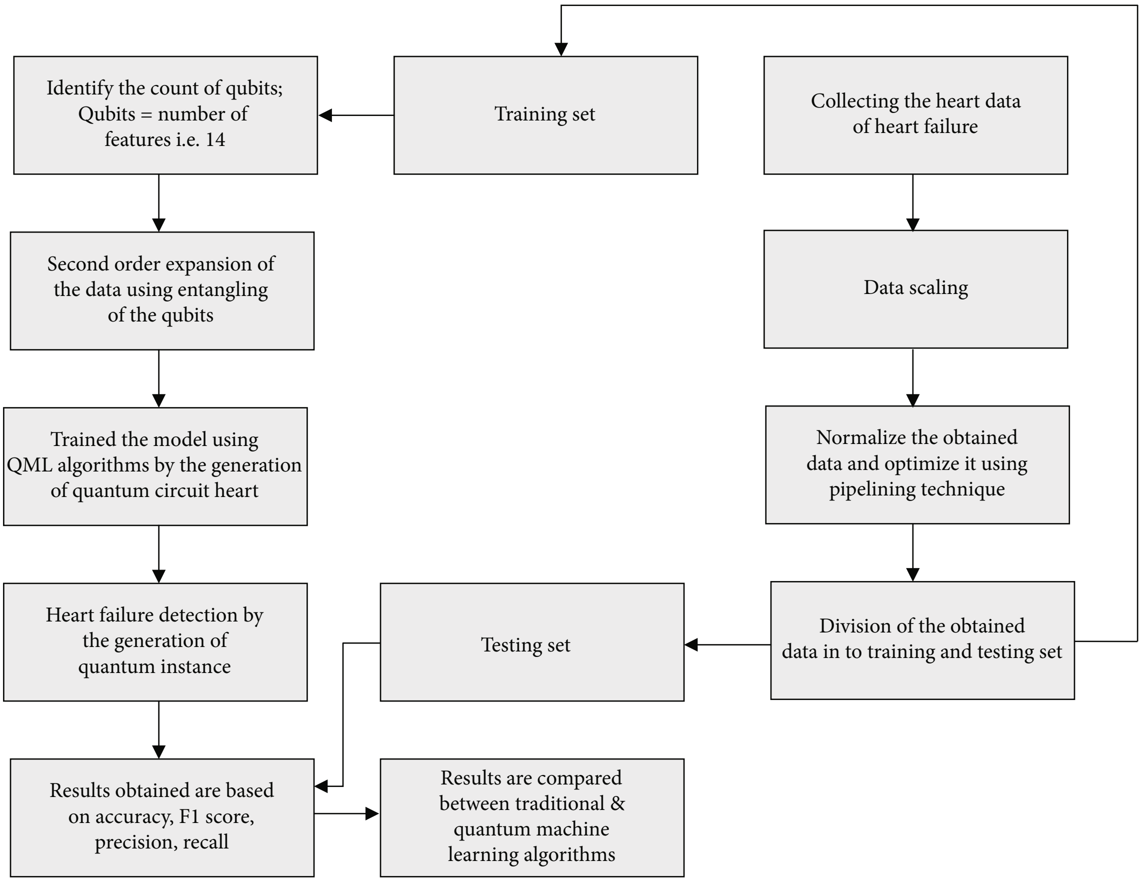

Identify the count of qubits; Qubits = number of features i.e. 14
Training set
Collecting the heart data of heart failure
Second order expansion of the data using entangling of the qubits
Data scaling
Trained the model using QML algorithms by the generation of quantum circuit heart
Normalize the obtained data and optimize it using pipelining technique
Heart failure detection by the generation of quantum instance
Testing set
Division of the obtained data in to training and testing set
Results obtained are based on accuracy, F1 score, precision, recall
Results are compared between traditional & quantum machine learning algorithms

Figure 2: Proposed system design.

used in developing the quantum-enhanced machine learning model.

Step 6. The number of qubits were specified, and the thumb rule to do so is as stated: number of qubits = number of features. So, for our dataset, the numbers of qubits are 14.

Step 7. After this, the mapping of the features is done with the help of entangling qubits to the second-order expansion.

Step 8. Later, the circuit is generated for quantum-enhanced machine learning algorithms in which quantum instances for heart failure detection are formed for the implementation purpose. The model is trained now, and with the help of testing data, the results are obtained using various quality metrics such as accuracy, precision, recall, and $F1$-score.

3.3. Applied Algorithms for Heart Failure Detection. Incorporating quantum algorithms within the machine learning process gives rise to quantum-enhanced machine learning. It commonly uses machine learning algorithms to examine traditional data executed on a quantum processor [47]. Traditional machine learning techniques include the random forest classifier (random decision forests), an ensemble learning approach for classification, regression, and other applications. Using a randomly selected portion of the training data, the random forest classifier generates a series of decision trees. It may be the most popular and widely used AI computation based on its fantastic or spectacular display throughout a vast grouping scope [40]. Random forest algorithm is solved by

$$\text{RFfi}_i = \frac{\sum_{j\in \text{all trees}} \text{normfi}_{ij}}{T}, \tag{1}$$

$$\text{normfi}_i = \frac{\text{fi}_i}{\sum_{j\in \text{all features}} \text{fi}_j}, \tag{2}$$

$$\text{fi}_i = \frac{\sum_{j:\text{node } j \text{ splits on feature } i} \text{ni}_j}{\sum_{k\in \text{all nodes}} \text{ni}_k}, \tag{3}$$

~

$$\text{ni}_j = W_j C_j - W_{\text{left}(j)} C_{\text{left}(j)} - W_{\text{right}(j)} C_{\text{right}(j)}. \tag{4}$$

Here, ni_j means importance of node j, W_j = weighted number of samples reaching node j, C_j = the impurity value of node j, left(j) = child node from left split on node j, right(j) = child node from right split on node j, fi$_i$ = the importance of feature i, RFfi$_i$ = the importance of feature i calculated from all trees in the random forest model, normfi$_i$ = the normalized feature importance for i in tree j, T = total number of trees. K nearest-neighbor classifier is a nonparametric AI strategy that is utilized for characterization just as relapse. It is based on the distances between a question and every model in the information by choosing the predetermined number nearest to the inquiry [48]. It is calculated by using the Euclidean Distance formula, which is shown as

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}, \tag{5}$$

where q and p points are used to calculate the distance at n different positions.

Likewise, there is a choice tree classifier, one of the regulated learning strategies and prescient demonstrating approaches utilized in insights, information mining, and AI. These are among the most mainstream AI calculations to provide their coherence and effortlessness [49]. Decision tree constructs characterization or relapse models as a tree structure. It divides an informative index into smaller subsets, resulting in a tree with choice hubs and leaf hubs as the final result. Decision tree are great instruments for assisting us with picking between a few blueprints [17]. It is calculated by using information gain (IG) and Gini index as shown in

$$\text{IG} = \text{Entropy}(s) - [(\text{Weighted Avg}) * \text{Entropy}(\text{each feature})], \tag{6}$$

$$\text{Entropy}(s) = -P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no}), \tag{7}$$

where s = total number of samples, P(yes) = probability of yes, and P(no) = probability of no,

$$\text{Gini Index} = 1 - \sum_j \text{P}_j^2, \tag{8}$$

where j denotes the number of features.

At the end, there is Naïve Baye's classifier which is a set of classifier algorithm that relies on Baye's theorem. It is used to calculate the conditional probability of a hypothesis being true given that order of information is also true [36]. Equation (9) is as follows

$$P(H_i \mid D) = \frac{P(H_i)\, P(D \mid H_i)}{\sum_j P(H_i)}, \tag{9}$$

where $P(H_i \mid D)$ is the posterior probability, $P(H_i)$ is the likelihood, $P(H_i)$ is the class prior probability, and $P(H_i)$ is the posterior prior probability. On the other hand, there are quantum-enhanced machine learning algorithms used in quantum-enhanced machine learning software as part of a larger implementation such as a quantum random forest classifier that uses quantum trees to select the class and generate random number generators [50]. In quantum K -nearest neighbour, the centroid is detected using the swap gates test between two states of the qubit. Assume that a training set $\boldsymbol{\tau}$ of feature vectors with their associated classifications is provided, together with an unclassified input vector $\overrightarrow{x}$, to select the class c^x for the new input that matches most of its k nearest neighbors [51]. Its formula for training set is shown as

$$|\tau| = \frac{1}{\sqrt{N}} \sum_{p} \left| v_1^p \cdots .v_n^p, c^p \right|. \tag{10}$$

After transforming it in to the quantum state and merging it with the hamming distance, we get,

$$|\varnothing_n| = \tau|\varnothing_{n-1}| = \alpha \sum_{p \in \cap} \left| d_1^p \cdots .d_n^p ; v_1^p \cdots .v_n^p, c^p ; 1 \right|, \tag{11}$$

where $\boldsymbol{\tau}$ is the training set, N refers to feature vectors v^p $(p = 1 \cdots N)$, and the c^p is the corresponding class. Another quantum AI computation is the quantum decision tree classifier, which benefits from a preparation dataset including views about facts that are either collected precisely or received from professionals [17]. The preparation dataset is composed of quantum objects rather than perceptions based on conventional knowledge in a quantum environment. A quantum state, x^i, is spoken to by a d-dimensional property vector with its ascribes $a1$, $a2$, $a3$, $a4$…. an [49]. The quantum state belongs to space S_i and is described as shown in

$$\varnothing = \sum_{j=1}^{m_i} \alpha_{i,j} \left| v_{i,j} \right|, \tag{12}$$

where domain value is represented by $v_{i,j}$, $\varnothing$ is the quantum state and the coefficients $|\boldsymbol{\alpha}_{\boldsymbol{i},\boldsymbol{j}}|^2 = \mathbf{1}$, and at the end, we have quantum Gaussian Naïve Bayes classifier in which the development of a quantum mechanical description returns the simplest form of Baye's theorem [52] using the case of exclusive populations which is shown as

$$P(H_1) = \frac{n(H_1)}{n(H_1) + n(H_2)}, \tag{13}$$

where H defines the exclusive population and datasets D as shown in Equation (9).

3.4. Evaluative Parameters. The performance of the proposed system for heart failure detection has been compared on the grounds of accuracy, $F1$ score, recall, and precision which are described as under [2, 5, 53, 54]:

(i) Accuracy: accuracy is characterized as the level of effectively arranged occurrences which is shown in

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{14}$$

where TP, FN, FP, and TN represent the number of true positives, false negatives, false positives, and true negatives, respectively. To obtain good classifier, the values of TPR and TNR should be nearer to 100%.

(ii) $F1$ score: it is likewise called as F measure. It passes on the harmony between the accuracy and review. It is represented as

$$F1\ \text{score} = \frac{\text{TP}}{\text{TP} + 1/2(\text{FP} + \text{FN})}, \tag{15}$$

where TP = count of true positives, FP = count of false positives, and FN = count of false negatives.

(iii) Recall: the recall is the portion of recovered examples among every important occurrence. It is the incentive between 0.0 for no recall and 1.0 for full recall. It is determined as the quantity of genuine positives separated by the all-out number of genuine positives and bogus negatives. It is represented by

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{16}$$

where TP = number of true positives and FN = number of false negatives.

(iv) Precision: precision is the part of significant cases among the recovered occasions. It is determined as the quantity of genuine positives isolated by the complete number of genuine positives and bogus positives. Equation (17) is represented as:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \tag{17}$$

where TP = count of true positives and FP = count of false positives.

4. Result and Analysis

Results are obtained by applying feature transformation and scaling techniques to normalize the data, so that applied machine and quantum-enhanced machine learning models will treat the data equally. The techniques are min-max scalar, standard scalar, and PCA scalar, from where the results are obtained and are later optimized using the pipelining technique. It is further incorporated into the machine and quantum-enhanced machine learning models for heart failure detection. To begin, the min-max scalar technique is used, which is a complement to Z-scalar normalization. The data is scaled to a defined range, often 0 to 1. They are using this method in comparison to standardization; the expense of having this constrained range results in smaller standard deviations and the suppression of outliers. A min-max scaling is performed using

$$X\text{sc} = X - \frac{X\ \text{min}}{X\ \text{max} - X\ \text{min}}. \tag{18}$$

Min-max scales the data in which the minimum value

Table 4: Heart failure detection using min-max scalar.

Algorithms	Accuracy	$F1$ score	Recall	Precision
QRFC	**0.86**	**0.88**	**0.91**	**0.83**
RFC	0.83	0.85	0.88	0.80
QKNN	0.85	0.86	0.87	0.84
KNN	0.82	0.83	0.84	0.81
QDTC	0.75	0.75	0.76	0.73
DTC	0.77	0.79	0.80	0.76
QGNBC	0.81	0.84	0.87	0.80
GNBC	0.81	0.83	0.86	0.79

Table 5: Heart failure detection using standard scalar.

Algorithms	Accuracy	$F1$ score	Recall	Precision
QRFC	0.85	0.87	0.90	0.85
RFC	0.83	0.84	0.87	0.81
QKNN	**0.88**	**0.89**	**0.90**	**0.87**
KNN	0.85	0.86	0.87	0.84
QDTC	0.75	0.76	0.76	0.73
DTC	0.77	0.79	0.80	0.76
QGNBC	0.83	0.85	0.88	0.81
GNBC	0.81	0.83	0.86	0.79

Table 6: Heart failure detection using PCA.

Algorithms	Accuracy	$F1$ score	Recall	Precision
QRFC	0.82	0.84	0.90	0.79
RFC	0.79	0.81	0.87	0.76
QKNN	**0.84**	0.85	0.88	**0.83**
KNN	0.81	0.82	0.85	0.80
QDTC	0.75	0.90	0.80	0.70
DTC	0.79	**0.92**	0.84	0.73
QGNBC	0.82	0.85	**0.93**	0.80
GNBC	0.80	0.82	0.90	0.76

between the columns became 0, and the maximum value is changed to 1 with other dataset values in between. After scaling using min-max and obtained, the results for different heart failure detection models have been shown in Table 4.

Second, apply the standard scalar to the heart failure dataset, resulting in two variables with values ranging from 10 to 100 and 1000 to 5000. We can compute the biased result by utilizing these predictor values since the variable with the most extensive range will significantly influence the outcome. As a result, it is necessary to normalize the data to a narrow range. Standardization is calculated by subtracting each number from the mean and dividing it by dataset's overall variance. The results of the application of the standard scalar are displayed in Table 5, which was computed using

$$x_{\text{scaled}} = \text{variable value} - \frac{\text{mean}}{\text{standard deviation}}. \tag{19}$$

Then, principal component analysis (PCA), a technique for reducing data dimension, was used to detect the relationships. Additionally, patterns in a dataset may be used to generate a lower-dimensional dataset without sacrificing any information. Thus, PCA analyzes the heart disease dataset to determine the high association between various factors. To preserve the critical data, a final choice is taken to decrease data's massive dimensions. This method is excellent for solving complex data-driven challenges that need the utilization of large datasets. It begins by standardizing the heart disease data, computing the covariance matrix, and calculating the calculated features' eigenvectors and eigenvalues. Finally, it adds the main components and decreases the size of the heart disease dataset (see Table 6).

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}, \tag{20}$$

where standardization equalises the contributions of the many continuous variables to the analysis, subtracting the mean and dividing by the standard deviation for each variable's value, as indicated in Equation (18), maybe done mathematically. After that, the covariance matrix explains the rationale behind the deviation of the input dataset from the mean for each other or the probability of a link. Equation (21) is used to create the matrix.

$$\operatorname{cov}(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \bar{y}), \tag{21}$$

where x and y are variables; they are positively associated if the two variables rise or decrease simultaneously, but they are negatively connected if one increases while the other drops. To determine the primary components of the data, the third step is to compute the eigenvectors and eigenvalues of the resulting covariance matrix. If A is a square matrix, v is a vector, and s is a scalar, then is the eigenvalue associated with eigenvector v to get the new vectors using

$$\det(A - \lambda I) = 0. \tag{22}$$

Finally, the collected data must be recast along the principal component axis. Its objective is to employ feature vectors derived from the covariance matrix's eigenvectors to reorient the data along the original axis to those represented by principle components, a process referred to as principal component analysis. The results obtained after applying it to the dataset are shown in Table 6.

Finally, pipelines are employed to optimize data to achieve faster outcomes. It automates the machine learning workflow by transforming the data sequence and then correlating them together in a model that can be tested and analyzed to determine if a result is positive or negative. We have performed several steps to train the machine and quantum-enhanced machine learning models to optimize the performance using the pipelining technique. It provides flexible

Table 7: Heart failure detection after pipelining.

Algorithms	Accuracy	$F1$ score	Recall	Precision
QRFC	**0.89**	**0.88**	**0.93**	**0.89**
RFC	0.84	0.86	0.90	0.83
QKNN	0.87	0.86	0.92	0.81
KNN	0.82	0.83	0.90	0.78
QDTC	0.81	0.80	0.78	0.85
DTC	0.80	0.79	0.75	0.84
QGNBC	0.85	0.86	0.91	0.82
GNBC	0.84	0.85	0.90	0.81

implementation and performs various tasks such as data collection, cleaning, feature extraction, labelling with dimensionality reduction, and model validation and visualization. The results obtained after applying it to the dataset are shown in Table 7.

On assaying Tables 4, 5, 6, and 7, evaluative parameters such as accuracy, $F1$ score, recall, and precision that have been obtained using quantum-enhanced and traditional machine learning algorithms are compared based on min-max scalar, standard scalar, principal component analysis, and pipelining optimization as shown in Figure 3.

On applying the min-max scaling technique, the quantum random forest classifier showed the highest accuracy rate by 0.86, $F1$ score by 0.88, recall value by 0.91, and precision score by 0.83 as compared to other algorithms. After going through the standard scalar, it can be seen that quantum K nearest neighbor achieved the highest accuracy rate by 0.88, $F1$ score by 0.89, precision value by 0.87, and recall value has been shared by quantum random forest classifier as well as quantum K nearest neighbor by 0.9 each. Likewise, when principal component analysis had been applied, quantum K nearest neighbor achieved highest accuracy rate by 0.84 and precision value by 0.83, decision tree classifier achieved highest $F1$ score by 0.92, and quantum Gaussian Naïve Bayes classifier secured highest recall value by 0.93.

As per the process, these algorithms were further passed through optimization techniques, i.e., pipelining, where the quantum random forest classifier marked the highest values in accuracy, $F1$ score, recall, and precision by 0.89, 0.88, 0.93, and 0.89, respectively. Hence, it can be concluded that quantum-enhanced machine learning algorithms are better than traditional machine learning algorithms in heart failure detection because of their remarkable achievements for all evaluative parameters.

4.1. Comparison with State of the Art Techniques. The comparison shown in Table 8 is based on the processing time taken by the traditional machine learning algorithms and quantum-enhanced machine learning algorithms. For example, the table states that in traditional machine learning algorithms, random forest classifier shows less computation time, i.e., 193 microseconds. In comparison, as K nearest neighbouring classifier shows the worst processing time by 301 microseconds. Likewise, in the quantum-enhanced machine learning algorithm, the quantum random forest classifier executes in 150 microseconds while the quantum decision tree algorithm used 286 microseconds for computation.

Hence, after summing up all these results, including the computational time, it has been observed that after applying the pipelining technique for optimizing the results, quantum random forest stands out to be the best algorithm for detecting heart failure in patients as compared to the other quantum and traditional machine learning algorithms in terms of accuracy, precision, $F1$ score, recall, and computational time. Table 9 shows the preliminary effects of this work as opposed to state-of-the-art techniques, showing that the proposed work stands out to be better than the state-of-the-art techniques in all heart failure identification categories.

5. Future Research Perspective of QML in the Healthcare

With the advancement in technology, the healthcare sector has gained immense popularity and has significantly benefited [8]. These days, quantum-enhanced machine learning plays a vital role in many health-related sectors, in addition to the development of new medical procedures, maintaining and handling patient data and records, and the therapeutics of chronic diseases. In this tech-savvy world, quantum-enhanced machine learning is popularly used to build rationalized administrative processes in medical institutes to map and treat contagious diseases effectively [35].

Quantum computing possesses an intelligent multitier storage capacity solution that helps in enabling the full power of AI in health and medicine [23]. The storage solution is best suited for video applications and also yields great performance in an organization. The multitier storage framework is more pronounced for its immense benefits in healthcare and medicine. Balanced performance, large capacity, and reduced cost are some of the benefits of building a cost-effective approach for the retention of data [57]. Based on analysis done so far, it can be observed that quantum-enhanced machine learning has tremendous applications in the field of the medical department as shown below:

(i) *Medical diagnostics and treatment:* quantum-enhanced machine learning can be advantageous in the field of health and medicine. This field helps in the easy execution of medical services by diluting costs and enhancing the services to patients for medical treatment and examination [30]

(ii) *Heart treatment:* nowadays, QML can build pioneering methods to determine the heart rate through magnetic flux and other more robust imaging techniques. In addition to that, quantum computers will be able to analyse and handle data much more efficiently than old conventional computers, which can be used in CT or magnetic resonance imaging [17]

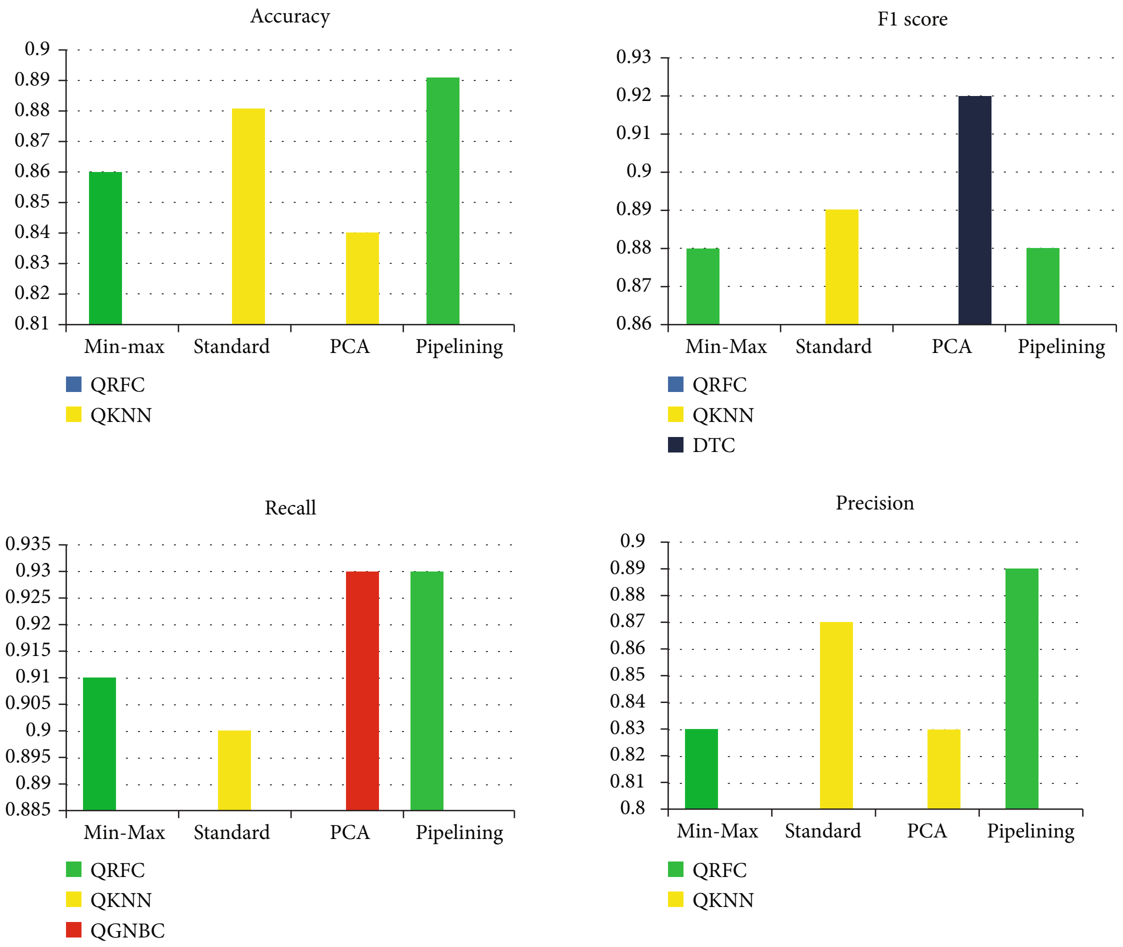


Figure 3: Comparison based on evaluative parameters.

Table 8: Computational time-based comparative analysis between traditional and quantum-enhanced machine learning algorithms.

Traditional machine learning algorithm	Computation time (μs)	Quantum-enhanced machine learning	Computation time (μs)
RFC	**193**	QRFC	**150**
KNN	301	QKNN	245
DTC	292	QDTC	286
GNBC	260	QNGBC	236

Table 9: Comparative analysis of proposed method with state-of-art techniques.

State-of-the-art	Accuracy	*F*1-score	Recall	Precision
Dunjiko, V et al. [10]	0.86	0.71	0.71	0.74
Obiri, D et al. [18]	0.79	0.81	0.80	0.84
Gao, X et al. [55]	0.83	0.85	0.81	0.88
Rajdhan, A et al. [56]	0.85	0.86	0.85	0.88
Proposed work	**0.89**	**0.88**	**0.93**	**0.89**

(iii) *Cancer detection:* cancer research is yet again a different area that is benefitted from the computational power of quantum computing. Clinic experts would be able to treat patients more effectively and timely by enhancing the speed of transferring the data from lab to bedside [58]

(iv) *Biomedical imaging:* enhanced imaging is a significant process that helps in the premature detection of even minute changes in the body, thereby providing other alternative treatment options for patients affected with diseases such as cancer or dementia, which will improve outcomes. Quantum learning has proved to be a promising method in biomedical imaging [59]

(v) *Complex optimization problems:* artificial neural network has resulted to be a precise diagnostic approach in traditional machine learning, which is optimized by varying the specifications of network's framework. These methods of optimization are convenient for quantum computing, where the propensity of "quantum tunnelling" fosters optimization problems to be computed quickly [11, 12, 18, 34, 35]

The two most important applications are quantum-enhanced sampling and discrete optimization. Quantum-enhanced sampling is the process of extracting a slice of a probability distribution from a quantum system, and in finance, discrete optimization is used to maximize the yield of a group of financial properties, which is an optimization challenge, where as in most cases, shallow learning approaches are inaccurate. In addition to this, the study of medical images, logistics, scheduling, climate modeling, weather forecasting, cryptography, and artificial intelligence may benefit from quantum-enhanced machine learning. These use cases collectively contribute significantly to the quadruple goal of healthcare.

Although quantum-enhanced machine learning enhances computing speed and can manage data storage utilizing various techniques, its limits are still readily apparent in practice. The study is confined to a minimally viable solution model. More research in heart disease detection may be required to convert the current model into a quickly deployable services mode, allowing for large-scale use in clinical trials. It also faces several hardware and software challenges [34], including quantum decoherence caused by heat and light, which causes qubits to lose their quantum properties such as entanglement, which results in the loss of stored data. Rotations in quantum computers' logic gates are prone to generate inaccuracy, as any incorrect rotation can result in an error in the output. Likewise, quantum algorithms face the constraint of specific simulations that limit their application.

Hence, the suggested quantum-enhanced machine learning model will need to be integrated with a deep learning framework in the future to improve its performance on comparing with the previously built models and state-of-the-art methodologies.

6. Conclusion

An empirical study has been made with regard to the diverse range of applications of quantum-enhanced machine learning in healthcare. Several authors shared the same idea of using traditional machine learning algorithms to compute a substantial amount of data to compute data intelligently. The primary objective of this study is to have a comparative analysis of standard and quantum-enhanced machine learning algorithms for the prediction of heart failure illness. In the previous work, researchers have used traditional machine learning algorithms in the health care sector, which lacked in terms of accuracy, computation time, and performance as compared to quantum-enhanced machine learning algorithms as they speed up the processing of information, show higher accuracy rate, and also, increase the performance of the system. Thus, in this article, we have performed the experiment by considering various traditional machine learning algorithms and quantum-enhanced machine learning algorithms to make the comparison to depict the best algorithm for heart failure detection. The detailed analysis of the techniques shows that quantum-enhanced machine learning significantly contributes to medical science and healthcare.

We have also shown that the prediction of heart failure as the cause of death can be effectively and more precisely be predicted with the help of QML algorithms. For that purpose, we have performed the scaling and transformation of the results by applying the scaling methods and then optimized it by using pipelining. We have seen a considerable increment in all the performance metrics like accuracy, $F1$ -score, precision, and recall. The large datasets are handled much better by quantum computing, which is its primary objective. Heart rates, temperature, blood pressure, oxygen levels, and other parameters will be monitored by quantum-enhanced machine learning algorithms with the internet of medical things, including smartwatches, fitness wearables, and smartphones. Patients' vital signs, such as blood pressure, heart rate, and electrocardiography (ECG), will be collected and transmitted to a healthcare practitioner using IoMT devices, providing a quick and accurate picture of patient's state of health and well-being. Additionally, the substantial analysis of quantum-enhanced machine learning tools on the diseases and abnormalities soaring rapidly supports a strong candidature in controlling these diseases at an early stage.

Data Availability

The data used to support the findings of this study are available upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

Jana Shafi would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University for supporting this work.

References

[1] M. Amin-Naji, H. Mahdavinataj, and A. Aghagolzadeh, "Alzheimers disease diagnosis from structural MRI using Siamese convolutional neural network," in *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 75–79, Tehran, Iran, 2019.

[2] J. Luyapan, X. Ji, D. Zhu, T. Mackenzie, I. Amos, and J. Gui, "An efficient survival multifactor dimensionality reduction method for detecting gene-gene interactions of lung cancer onset age," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2779–2781, Madrid, Spain, 2018.

[3] P. Rani, S. V. Kavita, S. Verma, and G. N. Nguyen, "Mitigation of black hole and gray hole attack using swarm inspired algorithm with artificial neural network," *Access*, vol. 8, pp. 121755–121764, 2020.

[4] S. Ramisetty, S. Varma, and S. Varma, "The amalgamative sharp wireless sensor networks routing and with enhanced machine learning," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 9, pp. 3766–3769, 2019.

[5] A. Gonsalves, F. Thabtah, R. Mohammad, and G. Singh, "Prediction of coronary heart disease using machine learning: an experimental analysis," in *Proceedings of the 2019 3rd*

International Conference on Deep Learning Technologies, pp. 51–56, Xiamen, China, 2019.

[6] W. Hu, "Comparison of two quantum nearest neighbor classifiers on IBM's quantum simulator," *Natural Science*, vol. 10, no. 3, pp. 87–98, 2018.

[7] J. Bang, S. W. Lee, and H. Jeong, "Protocol for secure quantum machine learning at a distant place," *Quantum Information Processing*, vol. 14, no. 10, pp. 3933–3947, 2015.

[8] X. Tian, Y. Huang, S. Verma et al., "Power allocation scheme for maximizing spectral efficiency and energy efficiency tradeoff for uplink NOMA systems in B5G/6G," *Physical Communication*, vol. 43, 2020.

[9] M. Abdel-Basset, A. Gamal, G. Manogaran, L. Son, and V. H. Long, "A novel group decision making model based on neutrosophic sets for heart disease diagnosis," *Multimedia Tools and Applications*, vol. 79, no. 15–16, pp. 9977–10002, 2020.

[10] K. Wereszczynski, A. Michalczuk, H. Josinski, and A. Polanski, "Quantum computing for clustering big datasets," in *2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE)*, pp. 276–280, Racławice, Poland, 2018.

[11] J. A. Alzubi, A. Yaghoubi, M. Gheisari, and Y. Qin, "Improve heteroscedastic discriminant analysis by using CBP algorithm," in *Algorithms and Architectures for Parallel Processing*, J. Vaidya and J. Li, Eds., vol. 11335 of Lecture Notes in Computer Science, , Springer, 2018.

[12] M. Gheisari, "The effectiveness of schema therapy integrated with neurological rehabilitation methods to improve executive functions in patients with chronic depression," *Health Science Journal*, vol. 10, 2016.

[13] M. Kumar, P. Mukherjee, K. Verma, S. Verma, and D. B. Rawat, "Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks," *IEEE Transactions on Network Science and Engineering*, vol. 25, 2021.

[14] G. Rani, G. Oza, S. Dhaka, N. Pradhan, S. Verma, and J. J. P. C. Rodrigues, "Applying deep learning-based multi-modal for detection of coronavirus," *Multimedia Systems*, vol. 27, 12 pages, 2021.

[15] Z. Lv, L. Qiao, and S. Verma, "AI-enabled IoT-edge data analytics for connected living," *ACM Transactions on Internet Technology*, vol. 21, pp. 1–20, 2021.

[16] M. Arora, S. Verma, C. S. Kavita, and S. Chopra, "A systematic literature review of machine learning estimation approaches in scrum projects," in *Cognitive Informatics and Soft Computing*, P. Mallick, V. Balas, A. Bhoi, and G. S. Chae, Eds., vol. 1040 of Advances in Intelligent Systems and Computing, pp. 573–586, Springer, Singapore, 2020.

[17] X. Gao, A. Amin Ali, H. Shaban Hassan, and M. Anwar, "Improving the accuracy for analyzing heart diseases prediction based on the ensemble method," *Complexity*, vol. 2021, Article ID 6663455, 10 pages, 2021.

[18] M. Rathi and A. Gupta, "Mobile-based prediction framework for disease detection using hybrid data mining approach," *Proceedings of International Conference on Artificial Intelligence and Applications*, , pp. 521–530, Springer, 2021.

[19] Z. Li, S. Verma, and M. Jin, "Power allocation in massive MIMO-HWSN based on the water-filling algorithm," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 8719066, 11 pages, 2021.

[20] F. Ablayev, M. Ablayev, J. Huang, K. Khadiev, N. Salikhova, and D. Wu, "On quantum methods for machine learning problems part II: quantum classification algorithms," *Big Data Mining and Analytics*, vol. 3, no. 1, pp. 56–67, 2020.

[21] Y. Kumar, "Recent advancement of machine learning and deep learning in the field of healthcare system," in *Computational Intelligence for Machine Learning and Healthcare Informatics*, R. Srivastava, P. K. Mallick, S. S. Rautaray, and M. Pandey, Eds., pp. 7–98, De Gruyter, Berlin, Boston, 2020.

[22] S. Qaisar and A. Subasi, "Cloud-based ECG monitoring using event-driven ECG acquisition and machine learning techniques," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 623–634, 2020.

[23] C. Chen and D. Dong, "Superposition-inspired reinforcement learning and quantum reinforcement learning," in *Reinforcement Learning*, pp. 1–5, IntechOpen, 2008.

[24] L. Gaur, G. Singh, A. Solanki et al., "Disposition of youth in predicting sustainable development goals using the neuro-fuzzy and random forest algorithms," *Human-Centric Computing and Information Sciences*, vol. 11, p. 24, 2021.

[25] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, "Image classification based on quantum K-nearest-neighbor algorithm," *Quantum Information Processing*, vol. 17, no. 9, 2018.

[26] S. Lu and S. Braunstein, "Quantum decision tree classifier," *Quantum Information Processing*, vol. 13, no. 3, pp. 757–770, 2014.

[27] N. Bharill, O. Patel, and A. Tiwari, "An enhanced quantum-inspired evolutionary fuzzy clustering," in *2015 IEEE Symposium Series on Computational Intelligence*, pp. 772–779, Cape Town, South Africa, 2015.

[28] H. Wang, J. Liu, J. Zhi, and C. Fu, "The improvement of quantum genetic algorithm and its application on function optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 730749, 10 pages, 2013.

[29] S. Taha and Z. Taha, "EEG signals classification based on autoregressive and inherently quantum recurrent neural network," *International Journal of Computer Applications in Technology*, vol. 58, no. 4, pp. 340–351, 2018.

[30] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery," http://arxiv.org/abs/1510.02855v1.

[31] D. Sierra-Sosa, J. Arcila-Moreno, B. Garcia-Zapirain, C. Castillo-Olea, and A. Elmaghraby, "Dementia prediction applying variational quantum classifier," http://arxiv.org/abs/2007.08653.

[32] J. Amin, M. Sharif, N. Gul, S. Kadry, and C. Chakraborty, "Quantum machine learning architecture for COVID-19 classification based on synthetic data generation using conditional adversarial neural network," *Cognitive Computation*, vol. 13, pp. 1–12, 2021.

[33] H. Gupta, H. Varshney, K. Sharma, N. Pachauri, and P. Verma, "Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction," *Complex & Intelligent Systems*, vol. 7, 2021.

[34] M. Gheisari and M. Esnaashari, "A survey to face recognition algorithms: advantageous and disadvantageous," *Journal Modern Technology & Engineering*, vol. 2, no. 1, pp. 57–65, 2017.

[35] M. Ashourian, M. Gheisar, and A. H. Talkhoncheh, "An improved node scheduling scheme for resilient packet ring network," *Majlesi Journal of Electrical Engineering*, vol. 9, no. 2, p. 43, 2015.

[36] G. Yang, M. A. Jan, A. U. Rehman, M. Babar, M. M. Aimal, and S. Verma, "Interoperability and data storage in internet of multimedia things: investigating current trends, research challenges and future directions," *IEEE Access*, vol. 8, pp. 124382–124401, 2020.

[37] G. Ghosh, G. Kavita, D. Anand et al., "Secure surveillance systems using partial-regeneration-based non-dominated optimization and 5D-chaotic map," *Symmetry*, vol. 13, no. 8, p. 1447, 2021.

[38] S. Verma and S. Mittal, "Implementation and analysis of stability improvement in VANET using different scenarios," *International Journal of Engineering & Technology*, vol. 7, pp. 151–154, 2018.

[39] D. F. Parsons, "Possible medical and biomedical uses of quantum computing," *Neuroquantology*, vol. 9, no. 3, pp. 596–600, 2011.

[40] M. Niemiec, "Error correction in quantum cryptography based on artificial neural networks," *Quantum Information Processing*, vol. 18, no. 6, pp. 1–18, 2019.

[41] J. Suo, L. Wang, S. Yang, W. Zheng, and J. Zhang, "Quantum algorithms for typical hard problems: a perspective of cryptanalysis," *Quantum Information Processing*, vol. 19, no. 6, pp. 1–26, 2020.

[42] Y. Kumar, K. Sood, S. Kaul, and R. Vasuja, "Big data analytics and its benefits in healthcare," in *Big Data Analytics in Healthcare*, A. Kulkarni, Ed., vol. 66 of Studies in Big Data, , pp. 3–21, Springer, 2020.

[43] A. Joseph, T. Hijal, J. Kildea, L. Hendren, and D. Herrera, "Predicting waiting times in radiation oncology using machine learning," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1024–1029, Cancun, Mexico, 2017.

[44] M. Ijaz, M. Attique, and Y. Son, "Data-driven cervical cancer prediction model with outlier detection and over-sampling methods," *Sensors*, vol. 20, no. 10, p. 2809, 2020.

[45] M. F. Ijaz, G. Alfian, M. Syafrudin, and J. Rhee, "Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest," *Applied Sciences*, vol. 8, no. 8, p. 1325, 2018.

[46] A. Gandam, J. S. Sidhu, S. Verma et al., "An efficient post-processing adaptive filtering technique to rectifying the flickering effects," *PLoS One*, vol. 16, no. 5, 2021.

[47] W. Li, Y. Chai, F. Khan et al., "A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system," *Mobile Network and Applications*, vol. 26, no. 1, pp. 234–252, 2021.

[48] J. Bird, A. Ekárt, and D. Faria, "On the effects of pseudorandom and quantum-random number generators in soft computing," *Soft Computing*, vol. 24, no. 12, pp. 9243–9256, 2020.

[49] M. Hammad, A. M. Iliyasu, A. Subasi, E. Ho, and A. A. A. el-Latif, "A multitier deep learning model for arrhythmia detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[50] S. Sharma, "Qeml (quantum enhanced machine learning): Using quantum computing to enhance ml classifiers and feature spaces," vol. 1, 2020, http://arxiv.org/abs/2002.10453.

[51] P. Sharma, K. Choudhary, K. Gupta, R. Chawla, D. Gupta, and A. Sharma, "Artificial plant optimization algorithm to detect heart rate & presence of heart disease using machine learning," *Artificial Intelligence in Medicine*, vol. 102, pp. 101752–101759, 2020.

[52] M. Sood, S. Verma, V. K. Panchal, and Kavita, "Optimal path planning using swarm intelligence based hybrid techniques," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 9, pp. 3717–3727, 2019.

[53] N. Kundu, G. Rani, V. S. Dhaka et al., "IoT and interpretable machine learning based framework for disease prediction in pearl millet," *Sensors*, vol. 21, no. 16, p. 5386, 2021.

[54] T. Reddy, S. Bhattacharya, P. K. R. Maddikunta et al., "Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset," *Multimedia Tools and Applications*, vol. 80, pp. 1–25, 2020.

[55] S. Qaisar and F. Hussain, "An effective arrhythmia classification via ECG signal subsampling and mutual information based subbands statistical features selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, 2021.

[56] A. Rajdhan, A. Agarwal, M. Sai, D. Ravi, and P. Ghuli, "Heart disease prediction using machine learning," *International Journal of Engineering Research and Technology*, vol. 9, no. 4, 2020.

[57] S. Gupta, S. Mohanta, M. Chakraborty, and S. Ghosh, "Quantum enhanced machine learning-using quantum computation in artificial intelligence and deep neural networks: quantum computation and machine learning in artificial intelligence," in *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, pp. 268–274, Bangkok, Thailand, 2017.

[58] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, D. S. Rajput, R. Kaluri, and G. Srivastava, "Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 185–196, 2020.

[59] M. Willsch, D. Willsch, F. Jin, H. Raedt, and K. Michielsen, "Benchmarking the quantum approximate optimization algorithm," *Quantum Information Processing*, vol. 19, no. 7, pp. 1–24, 2020.