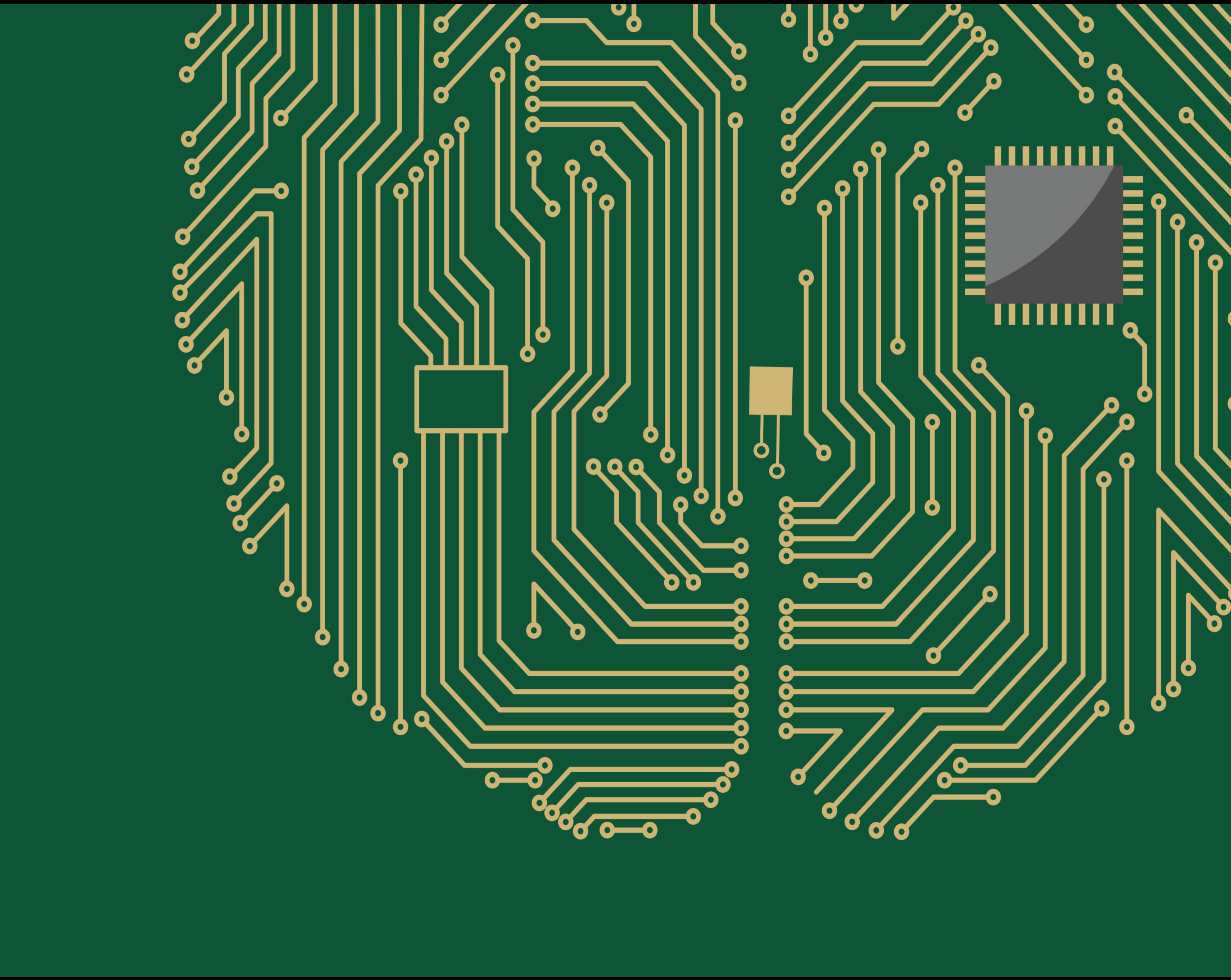
Computational Intelligence and Neuroscience

Computational Overhead vs. Learning Speed and Accuracy of Deep Networks

Lead Guest Editor: Muhammad Ahmad
Guest Editors: Shahzad Sarfraz, Manuel Mazzara, and Salvatore Distefano



Computational Overhead vs. Learning Speed and Accuracy of Deep Networks

Computational Intelligence and Neuroscience

Computational Overhead vs. Learning Speed and Accuracy of Deep Networks

Lead Guest Editor: Muhammad Ahmad
Guest Editors: Shahzad Sarfraz, Manuel Mazzara,
and Salvatore Distefano

Copyright © 2023 Hindawi Limited. All rights reserved.

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

Chief Editor

Andrzej Cichocki, Poland

Associate Editors

Arnaud Delorme, France
Cheng-Jian Lin, Taiwan
Saeid Sanei, United Kingdom

Academic Editors

Mohamed Abd Elaziz, Egypt
Tariq Ahanger, Saudi Arabia
Muhammad Ahmad, Pakistan
Ricardo Aler, Spain
Nouman Ali, Pakistan
Pietro Aricò, Italy
Lerina Aversano, Italy
Ümit Ağbulut, Turkey
Najib Ben Aoun, Saudi Arabia
Surbhi Bhatia, Saudi Arabia
Daniele Bibbo, Italy
Vince D. Calhoun, USA
Francesco Camastra, Italy
Zhicheng Cao, China
Hubert Cecotti, USA
Jyotir Moy Chatterjee, Nepal
Rupesh Chikara, USA
Marta Cimitile, Italy
Silvia Conforto, Italy
Paolo Crippa, Italy
Christian W. Dawson, United Kingdom
Carmen De Maio, Italy
Thomas DeMarse, USA
Maria Jose Del Jesus, Spain
Arnaud Delorme, France
Anastasios D. Doulamis, Greece
António Dourado, Portugal
Sheng Du, China
Said El Kafhali, Morocco
Mohammad Reza Feizi Derakhshi, Iran
Quanxi Feng, China
Zhong-kai Feng, China
Steven L. Fernandes, USA
Agostino Forestiero, Italy
Piotr Franaszczuk, USA
Thippa Reddy Gadekallu, India
Paolo Gastaldo, Italy
Samanwoy Ghosh-Dastidar, USA
Manuel Graña, Spain
Alberto Guillén, Spain
Gaurav Gupta, India
Rodolfo E. Haber, Spain
Usman Habib, Pakistan
Anandakumar Haldorai, India
José Alfredo Hernández-Pérez, Mexico
Luis Javier Herrera, Spain
Alexander Hošovský, Slovakia
Etienne Hugues, USA
Nadeem Iqbal, Pakistan
Sajad Jafari, Iran
Abdul Rehman Javed, Pakistan
Jing Jin, China
Li Jin, United Kingdom
Kanak Kalita, India
Ryotaro Kamimura, Japan
Pasi A. Karjalainen, Finland
Anitha Karthikeyan, Saint Vincent and the Grenadines
Elpida Keravnou, Cyprus
Asif Irshad Khan, Saudi Arabia
Muhammad Adnan Khan, Republic of Korea
Abbas Khosravi, Australia
Tai-hoon Kim, Republic of Korea
Li-Wei Ko, Taiwan
Raşit Köker, Turkey
Deepika Koundal, India
Sunil Kumar, India
Fabio La Foresta, Italy
Kuruva Lakshmanna, India
Maciej Lawrynczuk, Poland
Jianli Liu, China
Giosuè Lo Bosco, Italy
Andrea Loddo, Italy
Kezhi Mao, Singapore
Paolo Massobrio, Italy
Gerard McKee, Nigeria
Mohit Mittal, France
Paulo Moura Oliveira, Portugal
Debajyoti Mukhopadhyay, India
Xin Ning, China
Nasimul Noman, Australia
Fivos Panetsos, Spain

Evgeniya Pankratova, Russia
Rocío Pérez de Prado, Spain
Francesco Pistolesi, Italy
Alessandro Sebastian Podda, Italy
David M Powers, Australia
Radu-Emil Precup, Romania
Lorenzo Putzu, Italy
S P Raja, India
Dr.Anand Singh Rajawat, India
Simone Ranaldi, Italy
Upaka Rathnayake, Sri Lanka
Navid Razmjooy, Iran
Carlo Ricciardi, Italy
Jatinderkumar R. Saini, India
Sandhya Samarasinghe, New Zealand
Friedhelm Schwenker, Germany
Mijanur Rahaman Seikh, India
Tapan Senapati, China
Mohammed Shuaib, Malaysia
Kamran Siddique, USA
Gaurav Singal, India
Akansha Singh, India
Chiranjibi Sitaula, Australia
Neelakandan Subramani, India
Le Sun, China
Rawia Tahrir, Iraq
Binhua Tang, China
Carlos M. Travieso-González, Spain
Vinh Truong Hoang, Vietnam
Fath U Min Ullah, Republic of Korea
Pablo Varona, Spain
Roberto A. Vazquez, Mexico
Mario Versaci, Italy
Gennaro Vessio, Italy
Ivan Volosyak, Germany
Leyi Wei, China
Jianghui Wen, China
Lingwei Xu, China
Cornelio Yáñez-Márquez, Mexico
Zaher Mundher Yaseen, Iraq
Yugen Yi, China
Qiangqiang Yuan, China
Miaolei Zhou, China
Michal Zochowski, USA
Rodolfo Zunino, Italy

Contents

Contents

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9890718, 1 page
https://doi.org/10.1155/2023/9890718

Retraction

Retracted: Magnetic Tile Surface Defect Detection Methodology Based on Self-Attention and Self-Supervised Learning

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Ling, Y. Wu, R. Ali, and H. Zhu, "Magnetic Tile Surface Defect Detection Methodology Based on Self-Attention and Self-Supervised Learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 3003810, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9828465, 1 page
https://doi.org/10.1155/2023/9828465

Retraction

Retracted: Emotion Analysis Model of Microblog Comment Text Based on CNN-BiLSTM

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. Li and S. Yi, "Emotion Analysis Model of Microblog Comment Text Based on CNN-BiLSTM," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1669569, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9810910, 1 page
https://doi.org/10.1155/2023/9810910

Retraction

Retracted: Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. P. Rodrigues, R. Fernandes, A. Aakash et al., "Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5211949, 14 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9897287, 1 page
https://doi.org/10.1155/2023/9897287

Retraction

Retracted: Artificial Intelligence-Based Feature Analysis of Ultrasound Images of Liver Fibrosis

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Xie, S. Chen, D. Jia, B. Li, Y. Zheng, and X. Yu, "Artificial Intelligence-Based Feature Analysis of Ultrasound Images of Liver Fibrosis," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2859987, 9 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9871068, 1 page
https://doi.org/10.1155/2023/9871068

Retraction

Retracted: Machine Learning for Diagnosis of Systemic Lupus Erythematosus: A Systematic Review and Meta-Analysis

Computational Intelligence and Neuroscience

Received 3 October 2023; Accepted 3 October 2023; Published 4 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Zhou, M. Wang, S. Zhao, and Y. Yan, "Machine Learning for Diagnosis of Systemic Lupus Erythematosus: A Systematic Review and Meta-Analysis," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7167066, 14 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9870153, 1 page
https://doi.org/10.1155/2023/9870153

Retraction

Retracted: Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] M. Dong, H. Yu, Z. Sun et al., "Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 8327006, 13 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9867270, 1 page
https://doi.org/10.1155/2023/9867270

Retraction

Retracted: IoT-Based Hybrid Ensemble Machine Learning Model for Efficient Diabetes Mellitus Prediction

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Padhy, S. Dash, S. Routray, S. Ahmad, J. Nazeer, and A. Alam, "IoT-Based Hybrid Ensemble Machine Learning Model for Efficient Diabetes Mellitus Prediction," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2389636, 11 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9860329, 1 page
https://doi.org/10.1155/2023/9860329

Retraction

Retracted: Comparative Analysis of Aesthetic Emotion of Dance Movement: A Deep Learning Based Approach

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Huang, "Comparative Analysis of Aesthetic Emotion of Dance Movement: A Deep Learning Based Approach," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5135495, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9818249, 1 page
https://doi.org/10.1155/2023/9818249

Retraction

Retracted: Study on Vertical Joint Performance of Single-Faced Superposed Shear Wall Based on Finite Element Analysis and Calculation

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] G. Fang, R. Hou, W. Ma, and K. Xu, "Study on Vertical Joint Performance of Single-Faced Superposed Shear Wall Based on Finite Element Analysis and Calculation," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2418874, 8 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 8585839, 10 pages
https://doi.org/10.1155/2023/8585839

Research Article

Impact of Tree Cover Loss on Carbon Emission: A Learning-Based Analysis

Abdul Haleem Butt,[1] **Muhammad Ali Jamshed**,[2] **Ata Ur Rahman**,[1] **Faiz Alam**,[1] **Manoj Shakya**,[3] **Ahmad S. Almadhor**,[4] **and Masoor Ur-Rehman**[2]

[1] *Department of Creative Technologies, Air University, Islamabad, Pakistan*
[2] *James Watt School of Engineering, University of Glasgow, Glasgow, UK*
[3] *Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Nepal*
[4] *College of Computer and Information Sciences, Jouf University, Sakakah, Saudi Arabia*

Correspondence should be addressed to Abdul Haleem Butt; abdulhaleembutt85@gmail.com and Manoj Shakya; manoj@ku.edu.np

Received 21 July 2022; Revised 9 August 2022; Accepted 11 August 2022; Published 1 March 2023

Academic Editor: Muhammad Ahmad

Copyright © 2023 Abdul Haleem Butt et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Describing the processes leading to deforestation is essential for the development and implementation of the forest policies. In this work, two different learning models were developed in order to identify the best possible model for the assessment of the deforestation causes and trends. We developed autoregressive integrated moving average (ARIMA) model and long short-term memory (LSTM) independently in order to see the trend between tree cover loss and carbon dioxide emission. This study includes the twenty-year data of Pakistan on tree cover loss and carbon emission from the Global Forest Watch (GFW) platform, a known platform to get numerical data. Minimum mean absolute error (MAE) for the prediction of tree cover loss and carbon emission obtained through ARIMA model is 0.89 and 0.95, respectively. The minimum MAE given by LSTM model is 0.33 and 0.43, respectively. There is no such kind of study conducted in order to identify the increase in carbon emission due to tree cover loss most specifically in Pakistan. The results endorsed that one of the main causes of increase in the pollution in the environment in terms of carbon emission is due to tree cover loss.

1. Introduction

Global warming is a burning issue causing catastrophic changes and calamities around the world. The increasing occurrence of climate dissipation has been noticed with current global warming that had several biophysical impacts worldwide [1]. One of the main causes of global warming is carbon emission. Sources of carbon emission are both natural and human. Human sources include a variety of man-made actions such as tree logging, forest fires, burning of fossil fuels, cement production, natural gas production, and so on [2, 3].

Hence, the degradation of forests that play an essential role in maintaining a balance in the ecosystem directly adds to global warming. It results in rapid environmental degradation, not only leading to a scarcity of natural resources, decline in quality of life, and long-term public health issues but also inflicting economic losses [4]. Therefore, controlling deforestation can significantly minimize carbon emissions and help improve the environment. Recent studies revealed that the deforestation rate increases with extreme drought and wet years. In another study [5], the authors examined the carbon dioxide emission and climatic effects on major agricultural crop production in Pakistan. The results revealed that the crops such as wheat, maize, sugarcane, cotton, and so on have a constructive association with carbon dioxide emissions. Combating deforestation is being evaluated by cost-effective means of reducing greenhouse gas emissions [6].

Due to the negative impact of carbon emissions on the environment, there is a lot of ongoing research work to find out the efficient methods that have the ability to predict carbon emissions and determine their causes [7–10]. Learning-based techniques have provided new approaches to prediction problems that represent interactions between variables in a deep and layered hierarchy. ML-based techniques like support vector machines (SVMs) and random forest (RF), as well as DL-based algorithms like recurrent neural network (RNN) and LSTM, have attracted lots of attention in recent years because of their applications in a variety of fields [11–14]. In time series forecasting, DL approaches are capable of identifying data structure and pattern, such as non-linearity and complexity [15, 16]. LSTM has been extensively utilized in time series prediction in [17–21]. Autoregressive integrated moving average (ARIMA) is also another forecasting model [22] that predicts the future values based on the past values. ARIMA is the best model for one-step out-of-sample forecasting and is good for the data which consist of linear and short-term dependency (weekly or hourly) [23].

The rapid advancement in the Internet of Things (IoT) would be a future enhancement of this system. There are multiple privacy-related challenges in IoT architecture during communication that can be addressed using blockchain-enabled IoT architecture [24]. Moreover, there is a need for a decentralized auction-based resource allocation mechanism in edge computing-enabled IoT, which would be helpful to make computer resources closer to the devices [25]. To imporved the data quality during the data communication, few studies foucsed on the age of information (AOI) from the prespective of game theory [26]. Game theory is a useful tool to optimize wireless networks by assisting scarce wireless resource allocation, e.g., bandwidth and channels.

Objective of this study is to develop an accessible methodological approach that allows for rapid evaluation of statistical relationships and trends in forest monitoring data using both ML and DL. In this study, two independent learning models were developed using the autoregressive integrated moving average (ARIMA) model and long short-term memory (LSTM). Moreover, this study also focused on understanding the correlation between tree cover loss and carbon emission by taking Pakistan as a case study. Since, there is no such comprehensive effort found in open literature by the authors, it is a novel attempt in this direction. The remaining part of the paper is arranged as follows. The methodology is discussed in Section 2. In Section 3, results are discussed. Section 4 describes the conclusion and future work.

2. Methodology

2.1. Data Collection. Two time series datasets, i.e., tree cover loss and carbon emission in Pakistan, were taken from the GFW platform [27]. Each dataset consists of past 20 years' information on carbon values added to the environment and the tree cover loss from year 2001 to 2020. It should be noted that a more organized dataset comprising other variables of consideration for this study is not available in the open literature.

Figure 1 shows the proposed methodology of the framework. Following the collection of the dataset, the preprocessing stage is carried out to make the data stationary, as shown in Figure 2. The dataset is then divided into training and testing portions in order to train and evaluate the models. The training component of the dataset is used to train the models, and the testing portion is used for evaluation. The MAP and MAPE error evaluation metrics are evaluated for final analysis of the result.

2.2. Data Preprocessing. Figure 2 shows the preprocessing framework. Firstly, we deal with all missing values of the dataset. For any time series forecasting, being stationary is a mandatory property for a statistical model. A series is called stationary if its statistical property does not change with time. To verify this feature in our dataset, we have used the augmented Dickey–Fuller (ADF) test. After applying ADF, it was found that our dataset taken from GFW platform [27] does not fulfill this condition.

In order to make our dataset stationary, we performed a series of transformations such as power log transformation and differencing before applying (ADF) again for the verification. Figure 1 illustrates our adopted methodology in the form of a flowchart. After preprocessing, the dataset is divided into test and training subsets. The training set is used to train the prediction model and the test set is used to evaluate it. The split between the training and test data is kept at 70% and 30%, respectively.

For forecasting, the ARIMA and LSTM models are used to make predictions and are applied on both datasets.

2.3. ARIMA and LSTM Models. The ARIMA model is a generalization of the simpler autoregressive moving average that incorporates the concept of integration. The ARIMA model parameters are as follows:

(i) p: lag order (previously predicted values).

(ii) d: degree of difference.

(iii) q: order of moving average.

An ARIMA model is a time series forecasting model. It incorporates the properties of two autoregression and moving average models, where in autoregression, lags or previously predicted values are known as "autoregression" while lag or previously predicted error is known as "moving average." "Difference" is to make time series stationary (also known as integrated stationary time series version).

First step is to determine appropriate hyperparameters of ARIMA, p, h, and q, accurately to predict the behavior of the time series. Then, these hyperparameters are fitted into the training data. Finally, the model fitting residuals are analyzed to check whether the model assumptions are satisfied [6].

This study also utilizes LSTM model, which is a special type of RNN and is able to deal with long-term time dependencies [28]. There are many types of LSTM models that can be used for specific type of time series forecasting

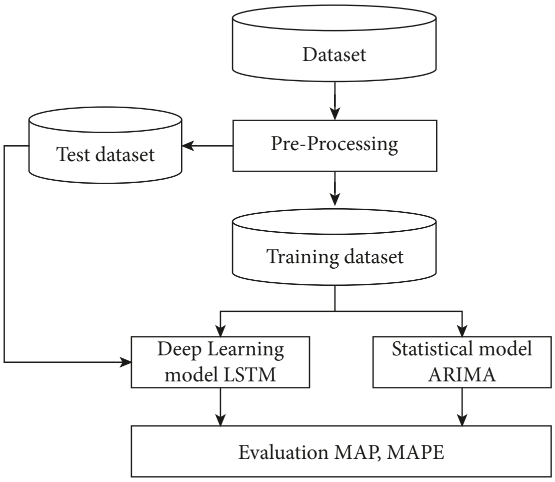


Figure 1: Hierarchical representation of the proposed methodology used to carry out forecasting.

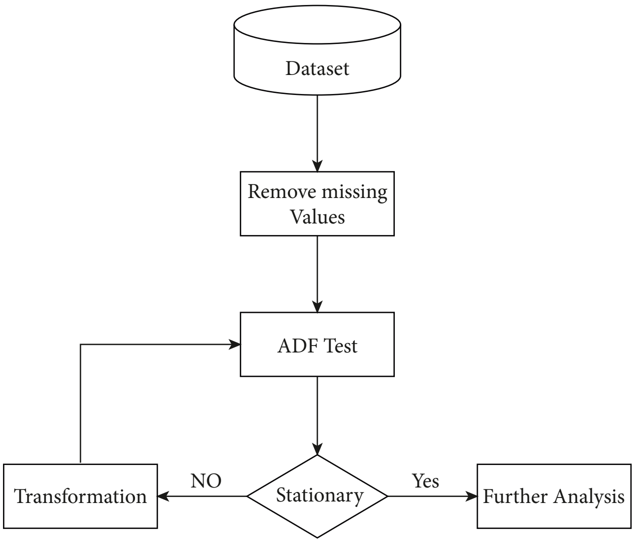


Figure 2: Hierarchical illustration of data preprocessing framework.

problem. In univariate LSTM, single series of observation is required to learn from the past values.

On the contrary, multivariate LSTM makes use of two or more kinds of parallel time series information to learn from the past observations. The basic architecture of univariate LSTM is shown in Figure 3. Basic LSTM network consists of cells that store the data. These cells resemble a transport line that connects one module to another conveying data from the past and gathering the present values [29]. For LSTM, three-layer-based architectures is proposed in this study with a dropout probability of 0.3 and zero non-trainable parameter. The lagged value of time series is used to predict future value with 40,901 learnable parameters.

The LSTM is applied on both of the parameters of carbon emission and tree cover loss with same layers and parameters, and the model for each dataset is trained with 50 epochs. Two separate univariate LSTM models are applied to two different time series variables, i.e., tree cover loss and carbon emission. The models are not only validated based on the difference between observed and predicted values also known as residuals but also exploited for future prediction of tree cover loss association with carbon emission for next three years with upper confidence level of 80% and lower confidence level of 90%.

3. Results

3.1. Stationary Time Series. Being stationary is an essential condition in time series analysis. Most of the time series models assumed that each point is independent of one another. To check this feature in our time series, ADF is applied. Moreover, to make the time series information stationary, differencing and log transformation are performed. The result of ADF test is shown in Table 1. It can be noted that the value of P is less than 0.05. This clearly indicates that after applying log transformation and differencing, we have obtained a stationary time series suitable for further prediction analysis.

3.2. Tree Cover Loss with ARIMA. ARIMA results for tree cover loss are shown in Figures 4 and 5. The model is trained with multiple order of ARIMA. The minimum MAE for tree cover loss is 0.95 with ARIMA (1, 1, 1). Similarly, the minimum MAE for tree cover loss is 1.4 and 1.2 with ARIMA (1, 2, 1). Values of p, d, and q are cross-checked using auto ARIMA function in Python. Figure 4 depicts the train (actual) and test (predicted) data showing that the data predict some trends and are at a right scale. In Figure 5, data for past twenty years (2001–2020) are utilized to train the model for prediction of tree cover loss for next five years. The results indicate an increase in tree cover loss in the coming years.

3.3. Carbon Emission with ARIMA. Figures 6 and 7 show the carbon emission results using ARIMA. The model is trained with multiple order of ARIMA. The minimum MAE of 1.20 and mean absolute percentage error (MAPE) of 1.24 are obtained for carbon emission forecasting with ARIMA order of (1, 1, 1) and (1, 2, 1), respectively. The twenty-year data, from 2001 to 2020, are used for the model training. The same model is further exploited to carry out forecasting carbon emission over next five years. The results indicate an increase in the carbon emission for this coming period.

The two results also exhibit that increase in the carbon emission has significant and similarly directed impact on the tree cover loss.

3.4. Tree Cover Loss with LSTM. LSTM has also been applied on the same dataset to compare the results with ARIMA model for tree cover loss. Figures 8 and 9 demonstrate the results for this study. Training and testing loss decay per epoch is shown in Figure 8 while Figure 9 illustrates the tree cover loss prediction for next three years, i.e., 2023–2025.

3.5. Carbon Emission with LSTM. Figures 10 and 11 show the results of LSTM-based carbon emission. The LSTM model is trained up to 50 epochs for both tree cover loss and carbon

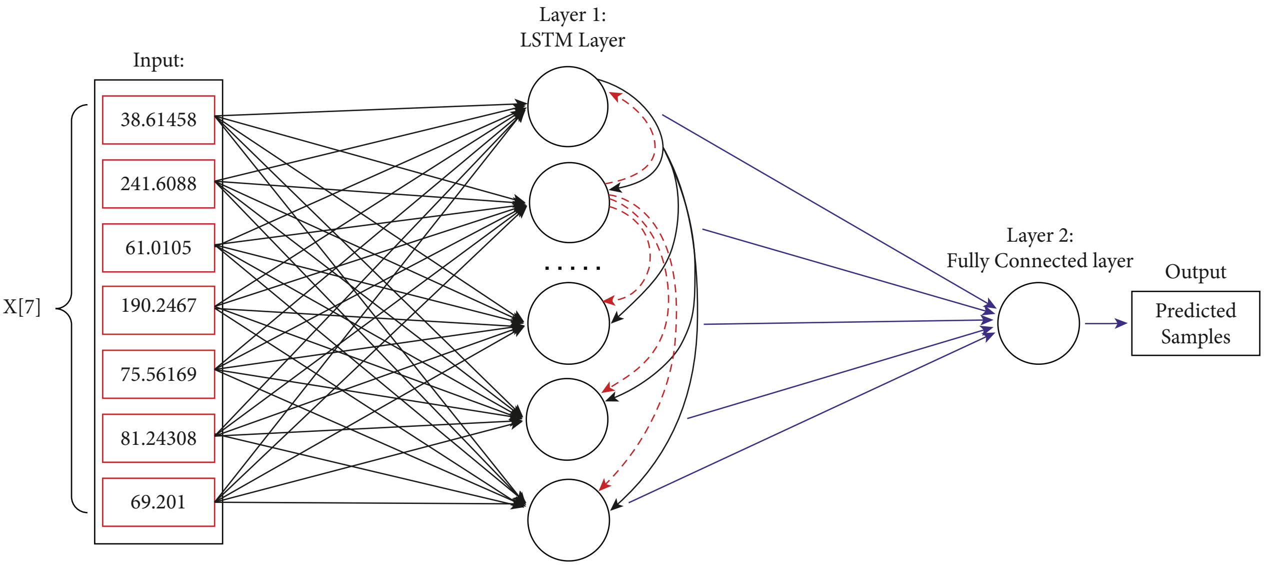


Figure 3: Architecture of basic LSTM network for univariate prediction.

Table 1: ADF test results.

Tree cover loss		Carbon emission	
Test statistic	−4.1812	Test statistic	−3.7851
P value	0.00078	*P* value	0.00532
Lag used	0.00000	Lag used	0.00000
No. of abbreviations used	18.0000	No. of abbreviations used	15.0000
Critical value (1%)	−3.8590	Critical value (1%)	−2.0521
Critical value (5%)	−3.0420	Critical value (5%)	−2.6710
Critical value (10%)	−2.6609	Critical value (10%)	−1.7219

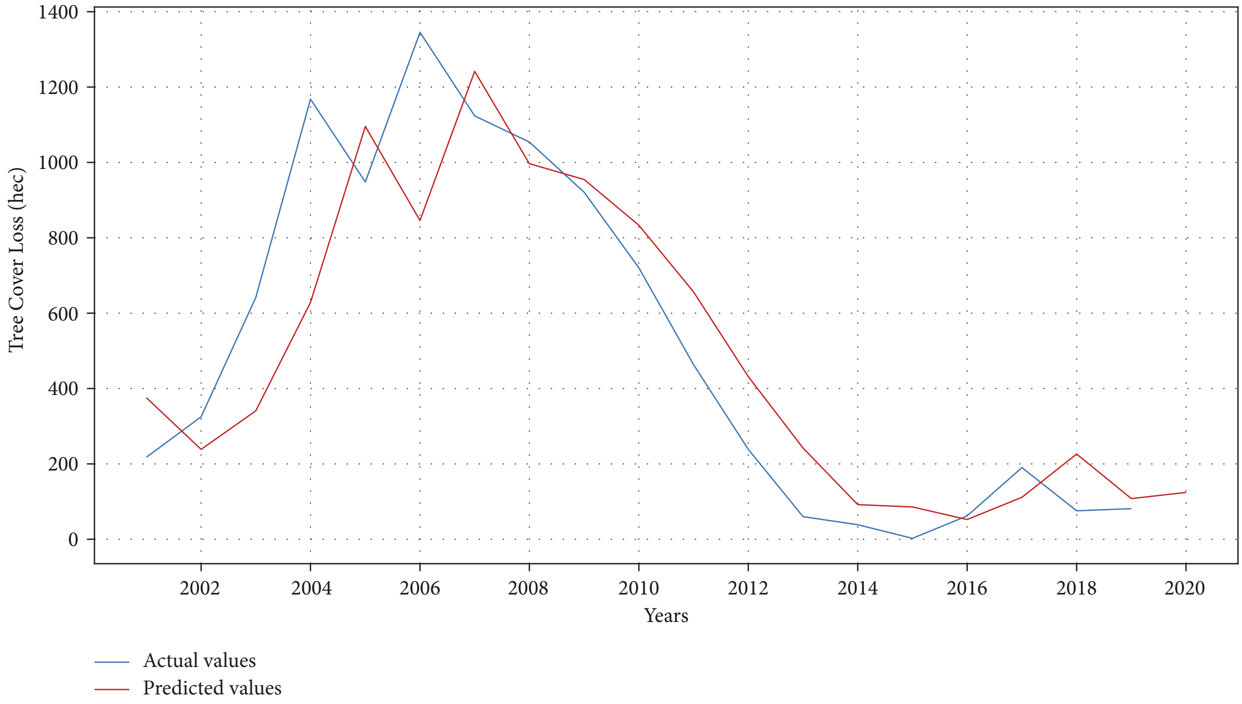


Figure 4: Actual versus predicted tree cover loss using ARIMA model.

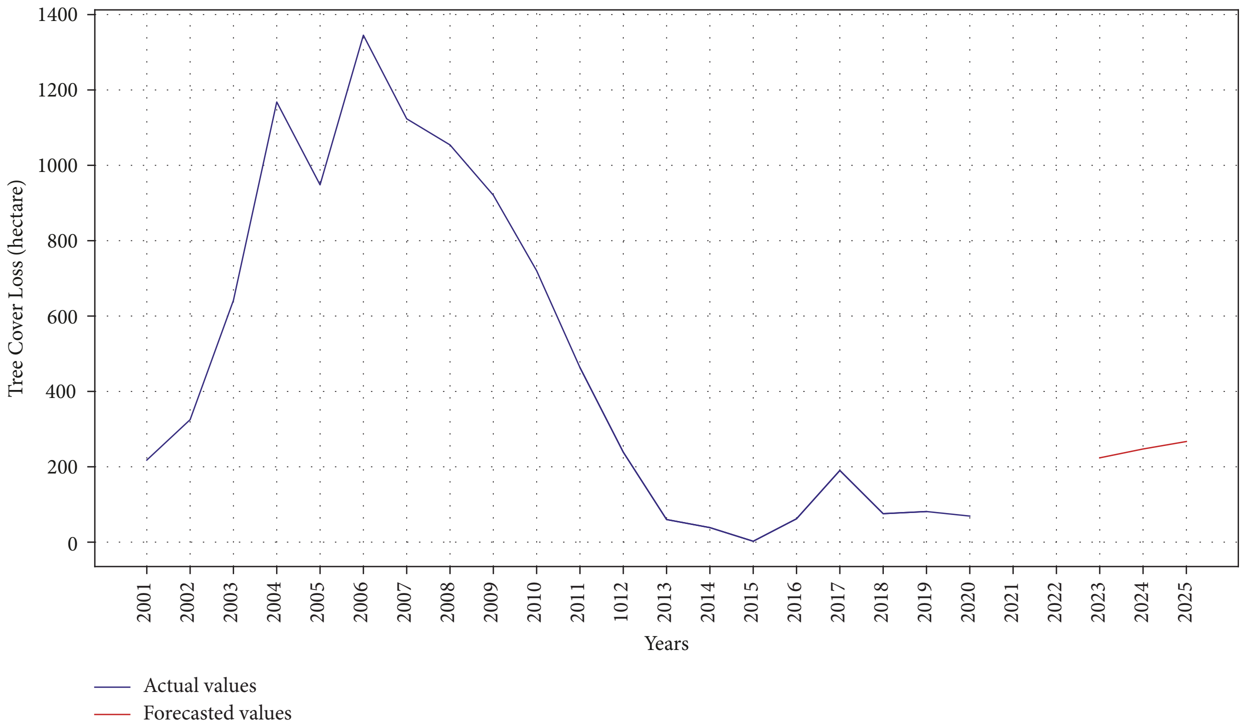


Figure 5: Train versus future predicted tree cover loss using ARIMA model.

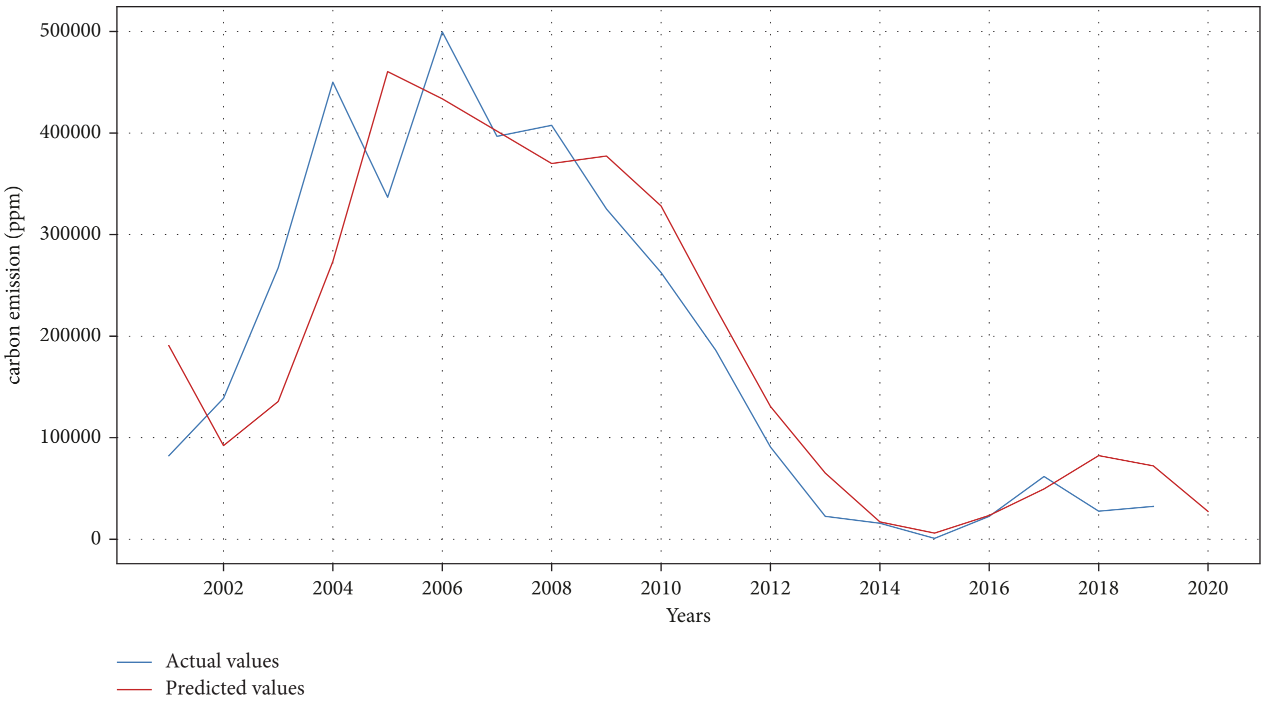


Figure 6: Actual versus predicted carbon emission using ARIMA model.

emission. It is observed from Figure 10 that the train and testing loss is decreasing over time after each epoch while using LSTM. The results indicate that a linear correlation exists between the carbon emission and tree cover loss. Future predictions of the LSTM model also reflect an increasing pattern of the carbon emission.

3.6. Comparison of ARIMA and LSTM. The ARIMA model works by filtering high-frequency noise from data, detecting local patterns based on linear dependencies, and predicting future trends [30]. In addition, the ARIMA model converts tree cover losses and carbon emission features into special temporary variables before matching them and only

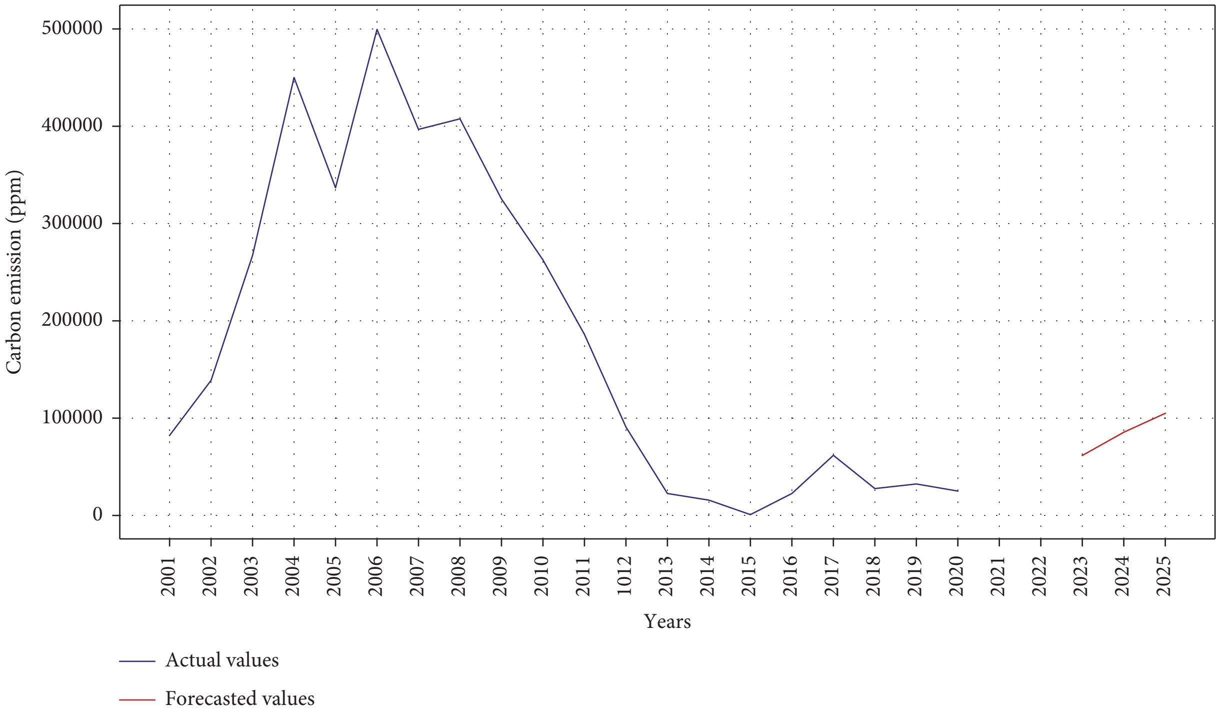


Figure 7: Actual versus predicted carbon emission using ARIMA model.

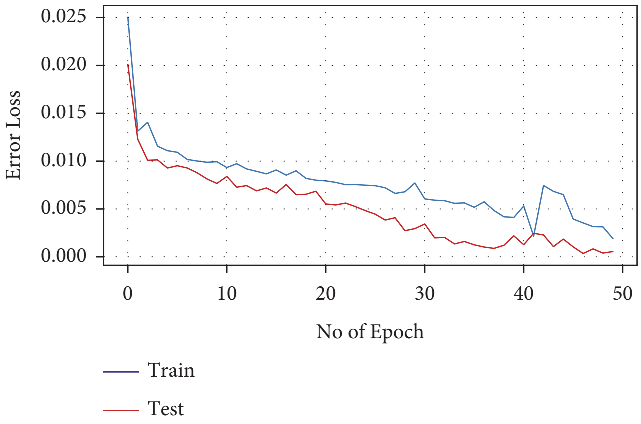


Figure 8: Loss decay per epoch for tree cover loss using LSTM.

considers the linear portion of the series [31]. The ARIMA model is simple and forthright and only requires to adjust the values of p, d, and q. The ARIMA model, however, is unable to deal with the non-linear relationship between the tree cover loss and carbon emissions.

On the contrary, the neural network such as LSTM can deal with both linear and non-linear patterns [32]. LSTM is a type of RNN that is meant to learn temporal patterns, capture non-linear dependencies, and preserve relevant memory for a longer period of time, resulting in achieving more accurate predictions [33].

Working of the two approaches, ARIMA and LSTM, for tree cover loss and carbon emission prediction is investigated using mean average precision (MAP) and MAPE evaluation matrices. Results are summarized in Table 2. It is observed that MAP obtained through LSTM is 0.33 while MAPE is 0.25. For carbon emission, LSTM gives MAP of 0.43 and MAPE of 0.40.

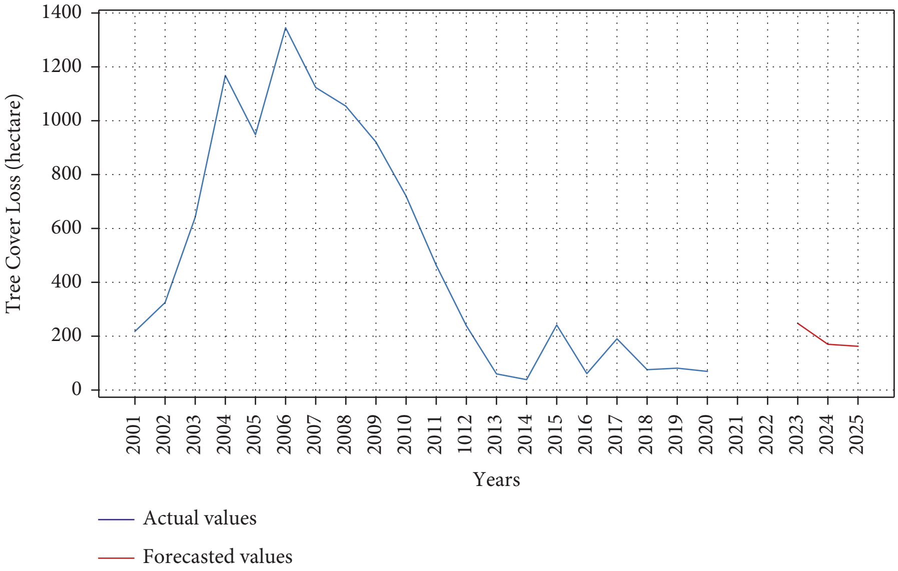


Figure 9: Actual versus predicted tree cover loss using LSTM.

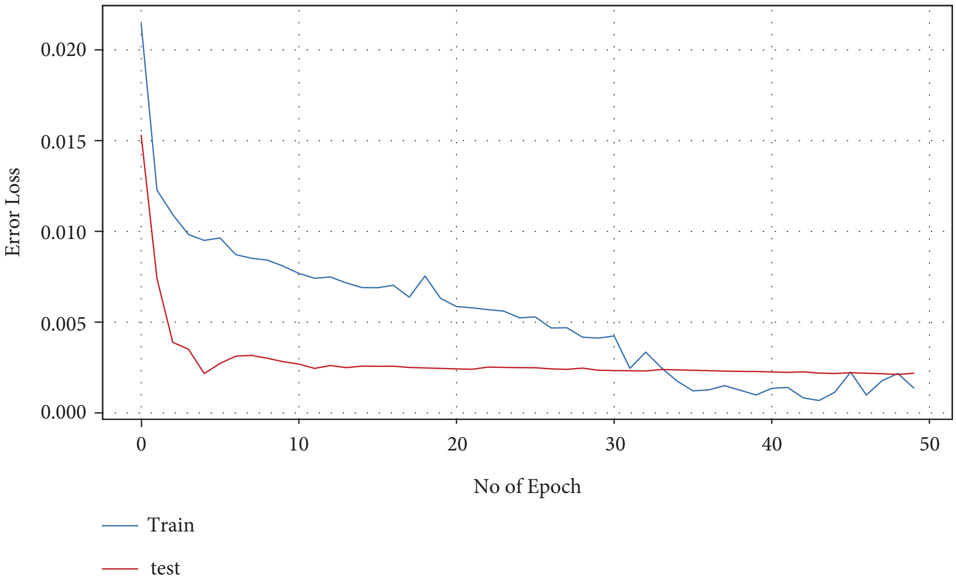


Figure 10: Loss decay per epoch for carbon emission using LSTM.

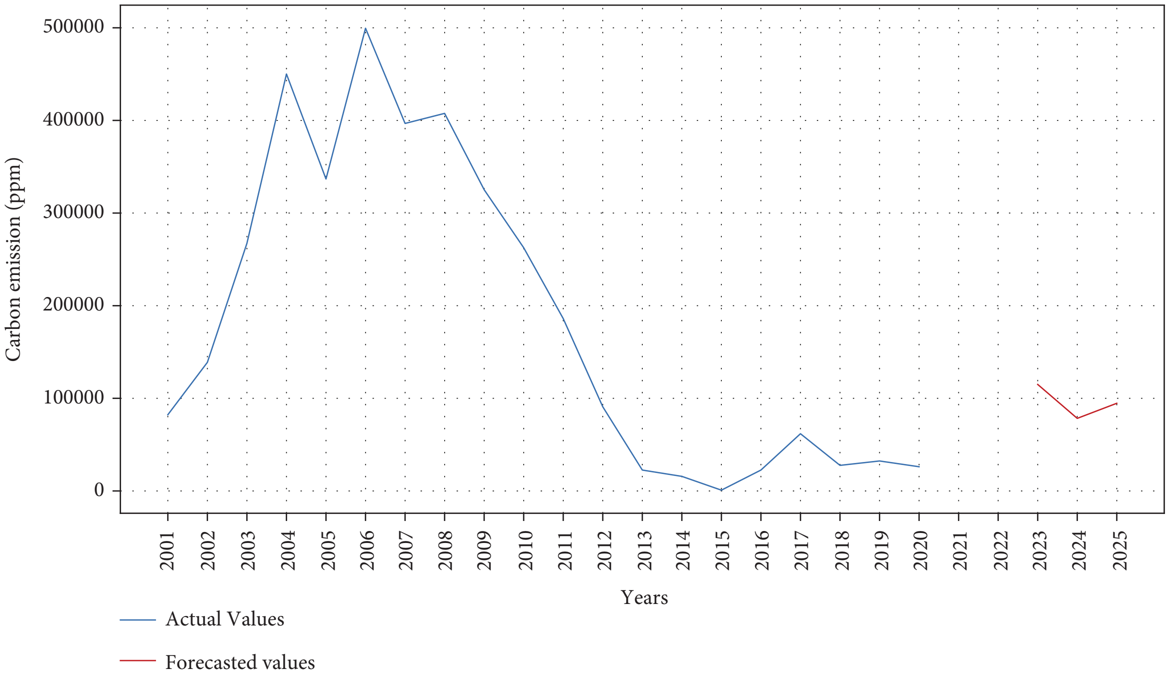


Figure 11: Actual versus predicted carbon emission using LSTM.

Table 2: Comparison of ARIMA and LSTM for tree cover loss and carbon emission prediction.

	Tree cover loss		Carbon emission	
Models	MAP	MAPE	MAP	MAPE
ARIMA (1, 2, 1)	1.34	4.45	1.24	1.20
ARIMA (1, 1, 1)	0.95	4.35	0.89	5.65
ARIMA (1, 2, 2)	3.81	9.10	3.58	8.21
LSTM	0.33	0.25	0.43	0.40

The results clearly show that LSTM has performed better than the ARIMA model in estimating and predicting tree cover loss and carbon emission for the analyzed data, hence ratifying that LSTM architecture is more suitable for time series prediction than ARIMA. The results also reflect the strong reliance and linear relationship between the tree cover loss and carbon emission.

4. Conclusion

In this work, a detailed analysis of tree cover loss and carbon emission data is carried out using ARIMA and LSTM techniques. 20-year data, from 2001 to 2020, are utilized to train and test the models and get predictions for next 5 years. The relationship between the two environmental factors is also established. The results have shown that temporal variations in the trend component of both carbon emission and tree loss cover are remarkably associated with each other. It has established that increase in the tree cover loss directly affects carbon emission in the atmosphere. Carbon emission could be one of the significant causes of the tree cover loss and deforestation.

Working of LSTM is found to be more vigorous in these prediction studies. Though very significant, this study is limited by the limited data availability. Both ARIMA and LSTM models showed the same trends. However, LSTM is a model that can learn the long-term dependencies, and it can remember the information that is processed in the model for a very long time [22]. In terms of computational time, the ARIMA models consume more time when using the rolling forecast method, and it is unfeasible to train new models when the orders of p, d, and q increase [34]. LTSM models take significantly less time to train, and once trained, constant predictions can be obtained, while ARIMA models need to be retrained.

Future aspects of this study include better understanding of carbon emission impact and control by considering more factors such as wood fuel, fire, and timber harvest. Inclusion of more factors and parameters can improve the overall prediction accuracy of the models while providing a broader understanding of causes of carbon emission. If more variables that contribute to carbon emissions and deforestation are taken into account, the work described in this paper will be more sophisticated.

Data Availability

The data used to support the findings of this study are available publicly on Global Forest Watch.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by Innovative and Collaborative Research Grant under Pakistan-UK Education Gateway

(ICRG-2020) (project no. 310366) (Deforestation in Pakistan: Combating through Wireless Sensor Networks (DePWiSeN)).

References

[1] X. Yuan, R. Hamdi, F. U. Ochege, A. Kurban, and P. De Maeyer, "The sensitivity of global surface air temperature to vegetation greenness," *International Journal of Climatology*, vol. 41, no. 1, pp. 483–496, 2021.

[2] P. Prosperi, M. Bloise, F. N. Tubiello et al., "New estimates of greenhouse gas emissions from biomass burning and peat fires using modis collection 6 burned areas," *Climatic Change*, vol. 161, no. 3, pp. 415–432, 2020.

[3] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Ur-Rehman, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in internet of things: a review," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5482–5494, 2022.

[4] W. Qiu, J. Zhang, H. Wu, M. Ahmad, and M. Ahmad, "The role of innovation investment and institutional quality on green total factor productivity: evidence from 46 countries along the "belt and road"," *Environmental Science and Pollution Research*, vol. 29, no. 11, pp. 16597–16611, 2022.

[5] A. Rehman, H. Ma, I. Ahmad, and M. I. Ahmad, "Examining the carbon emissions and climate impacts on main agricultural crops production and land use: updated evidence from Pakistan," *Environmental Science and Pollution Research*, vol. 29, no. 1, pp. 868–882, 2022.

[6] X. Liu, Z. Song, and Z. Song, "A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: from classical data mining to deep learning," *Renewable and Sustainable Energy Reviews*, vol. 119, Article ID 109632, 2020.

[7] M. Usman, R. Kousar, M. S. A. Makhdum, M. R. Yaseen, A. M. Nadeem, and A. M. Nadeem, "Do financial development, economic growth, energy consumption, and trade openness contribute to increase carbon emission in Pakistan? an insight based on ardl bound testing approach," *Environment, Development and Sustainability*, pp. 1–30, 2022.

[8] S. Ullah, N. . M. Syed, T. Gang et al., "Recent global warming as a proximate cause of deforestation and forest degradation in northern Pakistan," *PLoS One*, vol. 17, no. 1, Article ID e0260607, 2022.

[9] S. Grunwald, "Artificial intelligence and soil carbon modeling demystified: power, potentials, and perils," *Carbon Footprints*, vol. 1, no. 1, p. 6, 2022.

[10] N. L. Harris, . E. Goldman, C. Gabris et al., "Using spatial statistics to identify emerging hot spots of forest loss," *Environmental Research Letters*, vol. 12, no. 2, Article ID 024012, 2017.

[11] A. Y. Dawod and M. A. Sharafuddin, "Assessing mangrove deforestation using pixel-based image: a machine learning approach," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3178–3190, 2021.

[12] A. Zulfiqar, M. M. Ghaffar, M. Shahzad et al., "AI-Forest-Watch: semantic segmentation based end-to-end framework for forest estimation and change detection using multi-spectral remote sensing imagery," *Journal of Applied Remote Sensing*, vol. 15, no. 02, Article ID 024518, 2021.

[13] K. Taghipour, M. Heydari, Y. Kooch, H. Fathizad, B. Taghizadeh-Mehrjardi, and R. Taghizadeh-Mehrjardi, "Assessing changes in soil quality between protected and degraded forests using digital soil mapping for semiarid oak forests, Iran," *Catena*, vol. 213, Article ID 106204, 2022.

[14] D. Dominguez, L. d. J. del Villar, O. González-Rodríguez, and M. González-Rodríguez, "Forecasting amazon rain-forest deforestation using a hybrid machine learning model," *Sustainability*, vol. 14, no. 2, p. 691, 2022.

[15] R. V. Maretto, L. M. G. Fonseca, N. Jacobs, T. S. Korting, H. N. Parente, and L. L. Parente, "Spatio-temporal deep learning approach to map deforestation in amazon rainforest," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 771–775, 2021.

[16] S. Birim, I. Kazancoglu, S. K. Mangla, A. Kazancoglu, and Y. Kazancoglu, "The derived demand for advertising expenses and implications on sustainability: a comparative study using deep learning and traditional machine learning methods," *Annals of Operations Research*, pp. 1–31, 2022.

[17] J. Brownlee, "Time series prediction with lstm recurrent neural networks in python with keras," *Machine Learning Mastery*, vol. 2016, 2016.

[18] F. A. Gers, J. Cummins, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[19] S. Schmidhuber and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[21] T. R. R. D. S. R. M. L. K. B. S. S. S. Gadekallu, D. S. Rajput, M. P. K. Reddy et al., "A novel PCA-whale optimization-based deep neural network model for classification of tomato plant diseases using GPU," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1383–1396, 2021.

[22] R. Zhang, H. Song, Q. Chen, Y. Wang, S. Wang, and Y. Li, "Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China," *PLoS One*, vol. 17, no. 1, Article ID e0262009, 2022.

[23] A. K. a. H. D. A. A. S. Muhammad, "Forecasting pet food item stock using ARIMA and LSTM," in *Proceedings of the 4th International Conference on Computer and Informatics Engineering (IC2IE)*, 2021.

[24] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su, and H. Chen, "Resource allocation and trust computing for blockchain-enabled edge computing system," *Computers & Security*, vol. 105, Article ID 102249, 2021.

[25] T. Wang, Q. Yang, X. Shen, T. R. Gadekallu, W. Wang, and K. Dev, "A privacy-enhanced retrieval technology for the cloud-assisted internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4981–4989, 2022.

[26] Z. Han, Y. Yang, W. Wang, L. Zhou, T. N. Nguyen, and C. Su, "Age efficient optimization in UAV-aided VEC network: a game theory viewpoint," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2022, pp. 1–10, 2022.

[27] G. f. watch, *Global Forest Watch Pro*, Forest Watcher and forest Atlases, [Online], 2014.

[28] F. Succetti, A. Rosato, R. Araneo, and M. Panella, "Deep neural networks for multivariate prediction of photovoltaic power time series," *IEEE Access*, vol. 8, pp. 211490–211505, 2020.

[29] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *Proceedings of the 17th IEEE international conference on machine learning and applications (ICMLA)*, pp. 1394–1401, 2018.

[30] Y.-w. Wang, Z.-z. Jiang, and Y. Jiang, "Comparison of autoregressive integrated moving average model and generalised regression neural network model for prediction of

haemorrhagic fever with renal syndrome in China: a time-series study," *BMJ Open*, vol. 9, no. 6, Article ID e025773, 2019.

[31] T. Petukhova, D. Ojkic, B. McEwen, R. Poljak, and Z. Poljak, "Assessment of autoregressive integrated moving average (arima), generalized linear autoregressive moving average (glarma), and random forest (rf) time series regression models for predicting influenza a virus frequency in swine in ontario, Canada," *PLoS One*, vol. 13, no. 6, Article ID e0198313, 2018.

[32] X. Wang and M. Meng, "A hybrid neural network and arima model for energy consumption forecasting," *Journal of Computers*, vol. 7, no. 5, pp. 1184–1190, 2012.

[33] J. Gu, L. Liang, H. Song et al., "A method for hand-foot-mouth disease prediction using geodetector and lstm model in guangxi, China," *Scientific Reports*, vol. 9, no. 1, pp. 17928–18010, 2019.

[34] P. K. A. Y. K. R. Kumar, "Three stage fusion for effective time series forecasting using Bi-LSTM-ARIMA and improved DE-ABC algorithm," *Neural Computing & Applications*, vol. 2022, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9871068, 1 page
https://doi.org/10.1155/2023/9871068

Retraction

Retracted: Machine Learning for Diagnosis of Systemic Lupus Erythematosus: A Systematic Review and Meta-Analysis

Computational Intelligence and Neuroscience

Received 3 October 2023; Accepted 3 October 2023; Published 4 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Zhou, M. Wang, S. Zhao, and Y. Yan, "Machine Learning for Diagnosis of Systemic Lupus Erythematosus: A Systematic Review and Meta-Analysis," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7167066, 14 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7167066, 14 pages
https://doi.org/10.1155/2022/7167066

RETRACTED

Research Article

Machine Learning for Diagnosis of Systemic Lupus Erythematosus: A Systematic Review and Meta-Analysis

Yuan Zhou , Meng Wang, Shasha Zhao, and Yan Yan

Department of Dermatology, Plastic Surgery Hospital, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing, China

Correspondence should be addressed to Yan Yan; yanyan@psh.pumc.edu.cn

Received 22 June 2022; Revised 31 July 2022; Accepted 3 August 2022; Published 22 November 2022

Academic Editor: Muhammad Ahmad

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

Background. Application of machine learning (ML) for identification of systemic lupus erythematosus (SLE) has been recently drawing increasing attention, while there is still lack of evidence-based support. *Methods*. Systematic review and meta-analysis are conducted to evaluate its diagnostic accuracy and application prospect. PubMed, Embase, Cochrane Library, and Web of Science libraries are searched, in combination with manual searching and literature retrospection, for studies regarding machine learning for identifying SLE and neuropsychiatric systemic lupus erythematosus (NPSLE). Quality Assessment of Diagnostic Accuracy Studies (QUADA-2) is applied to assess the quality of included studies. Diagnostic accuracy of the SLE model and NPSLE model is assessed using the bivariate fixed-effect model, and the data are pooled. Summary receiver operator characteristic curve (SROC) is plotted, and area under the curve (AUC) is calculated. *Results*. Eighteen (18) studies are included, in which ten (10) focused on SLE and eight (8) on NPSLE. The AUC of SLE identification is 0.95, the sensitivity is 0.90, the specificity is 0.89, the PLR is 8.4, the NLR is 0.12, and the DOR is 73. AUC of NPSLE identification is 0.89, the sensitivity is 0.83, the specificity is 0.83, the PLR is 5.0, the NLR is 0.20, and the DOR is 25. *Conclusion*. Machine learning presented remarkable performance in identification of SLE and NPSLE. Based on the convenience for inclusion factor collection and non-invasiveness of detection, machine learning is expected to be widely applied in clinical practice to assist medical decision making.

1. Introduction

Systemic lupus erythematosus (SLE) is an autoimmunity-mediated, chronic, and refractory connective tissue disease (CTD) with multiple systems involved. It usually occurs in women aged 20 to 40 years old, and the ratio of incidence between male and female is 1 : 9. The prevalence of PLE varies from 1 to 10 per 100000 people in different countries and regions, and the incidence in coloured people is higher than that in white people [1–3].

Commonly used clinical diagnostic criteria included American College of Rheumatology (ACR) criteria [4], Systemic Lupus International Collaborating Clinics (SLICC) criteria [5], and European League Against Rheumatism/American College of Rheumatology Classification (EULAR/ACR) Criteria [6] for Systemic Lupus Erythematosus. Sensitivity and specificity reported in different cohorts range from 0.84 to 0.95 [7]. However, early identification and diagnosis for SLE are still difficult due to the heterogeneity of its clinical and laboratorial indicators. The multiple organ damage would aggravate over time, making early recognition and diagnosis of SLE important [8]. Diagnosis of neuropsychiatric systemic lupus erythematosus (NPSLE) currently follows the ACR criteria [9], which is mainly based on clinical symptoms that have already occurred, and occurrence of those symptoms typically indicates highly active NPSLE with high mortality, presenting a challenge to the early identification for NPSLE. Machine learning (ML) refers to a technology to make computer simulate or implement human learning activities, which can make full use of information via algorithms to obtain hidden, effective, and understandable knowledge from massive data, so as to build predictive models [10]. Recently, ML has shown excellent pattern-recognizing capability and has gradually affected clinical

RETRACTED

decision making in multiple fields, including rheumatic immunology [11, 12].

Its value for SLE and NPSLE identification and diagnosis is particularly brought into focus. However, the diagnostic accuracy varies in different models, and there is also lack of evidence-based support. We conducted this meta-analysis to identify the value of machine learning in the recognition of SLE and NPSLE and to explore which predictors are more clinically significant, so as to provide reference for future development of diagnostic systems and models.

2. Methods

The systematic review and meta-analysis were conducted in strict accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [13] statement and had been registered on PROSPERO (registration no. CRD42022329180).

2.1. Definition. In this study, SLE is defined as patients meeting the ACR [4] or SLICC [5] criteria, and NPSLE is defined as patients meeting the NPSLE ACR [9] criteria.

2.2. Hypothesis. Can machine learning applications play a significant role in the identification of systemic lupus erythematosus (SLE) and neuropsychiatric systemic lupus erythematosus (NPSLE)?

2.3. Literature Search and Study Selection. PubMed, Embase, Cochrane Library, and Web of Science were searched from inception to March 2022, via combination of medical subject headings and free words, for studies that applied ML for identification of SLE and NPSLE. Manual searching and literature retrospection were also conducted. Search items in PubMed included "Lupus Erythematosus, Systemic," "Systemic Lupus Erythematosus," "Lupus Erythematosus Disseminatus," "Libman-Sacks Disease," "machine learning," "Deep learning," "Transfer Learning," "Ensemble Learning," "artificial intelligence," and "Prediction model," with the language restricted to English.

Patients who have symptoms of SLE or NPSLE were included. Exclusion criteria were as follows:

(1) Patients who had history of other cerebral diseases.

(2) Unable to participate in relevant clinical and laboratory tests.

(3) Concomitant with other CTDs. Eligible randomized controlled trails (RCTs), case-control studies, cross-sectional studies, nested case-control studies, and cohort studies were all included.

(4) Studies with the participants less than 30 in training set of the model or without modeling were excluded.

2.4. Literature Screening and Data Extraction. All identified articles were imported to Endnote. The titles and abstracts were browsed following duplicate removal to exclude irrelevant studies. Full texts of the remaining articles were read, according to the inclusion and exclusion criteria, to screen out eligible studies. Extracted data included name of first author, publication date, sample size, types of models, indices of modeling, and outcome measures. Outcome measures included sensitivity (SEN), specificity (SPE), positive likelihood ratio (PLR), negative likelihood ratio (NLR), diagnostic odds ratio (DOR), summary receiver operator characteristic curve (SROC), area under the curve (AUC), and clinical application value. Literature screening and data extraction were conducted by two reviewers (Yuan Zhou and Meng Wang) independently, and any disagreements were settled via discussion with a third reviewer (Shasha Zhao).

2.5. Quality Assessment. Quality assessment of included studies was performed by two reviewers using Quality Assessment of Diagnostic Accuracy Studies (QUADAS-2) criteria [14], which contain four (4) domains in terms of risk of bias: patient selection, index test, reference standard, and flow and timing. Each domain was assessed, and the results were pooled to grade an included study as "low risk," "high risk," or "unclear risk." Disagreements were resolved by a third reviewer to reach a consensus.

2.6. Statistical Analysis. Statistical analysis was performed using STATA 15.0. A grouping analysis was processed based on different types of machine learning algorithms. C-indices with 95% confidence intervals (95% Cis) of the prediction models were pooled. Then, the diagnostic accuracy of ML for SLE and NPSLE was evaluated using the bivariate fixed-effect model. Outcomes that were included in the model contained point-estimated values of SEN, SPE, PLR, NLR, and DOR, with their 95% CIs provided. The SROC was plotted, and AUC with its 95% CI was calculated. Deek's funnel plot was applied to assess the publication bias, and Q and I^2 statistics were used for heterogeneity test. I^2 greater than 50% indicated significant heterogeneity. p value less than 0.05 indicated statistical significance.

3. Results

3.1. Study Selection and Risk of Bias Assessment. Sixteen hundred and eighty-one (1681) articles were identified through initial search, and 1226 remained after removing duplicates. Eleven hundred and sixty-six (1166) ineligible articles were excluded after browsing the abstracts and titles, and full texts of the remaining sixty (60) articles were read. Finally, a total of eighteen (18) studies were included, in which ten (10) [15–24] focused on SLE and the remaining eight (8) [25–32] on NPSLE. The study selection process is shown in Figure 1, and the characteristics of included studies are shown in Tables 1 (for SLE) and 2 (for NPSLE). Among the eighteen (18) studies, fifteen (15) were published in recent five (5) years, and thirteen (13) were published in recent three (3) years, which revealed that this field might be an emerging hotspot and innovative.

RETRACTED

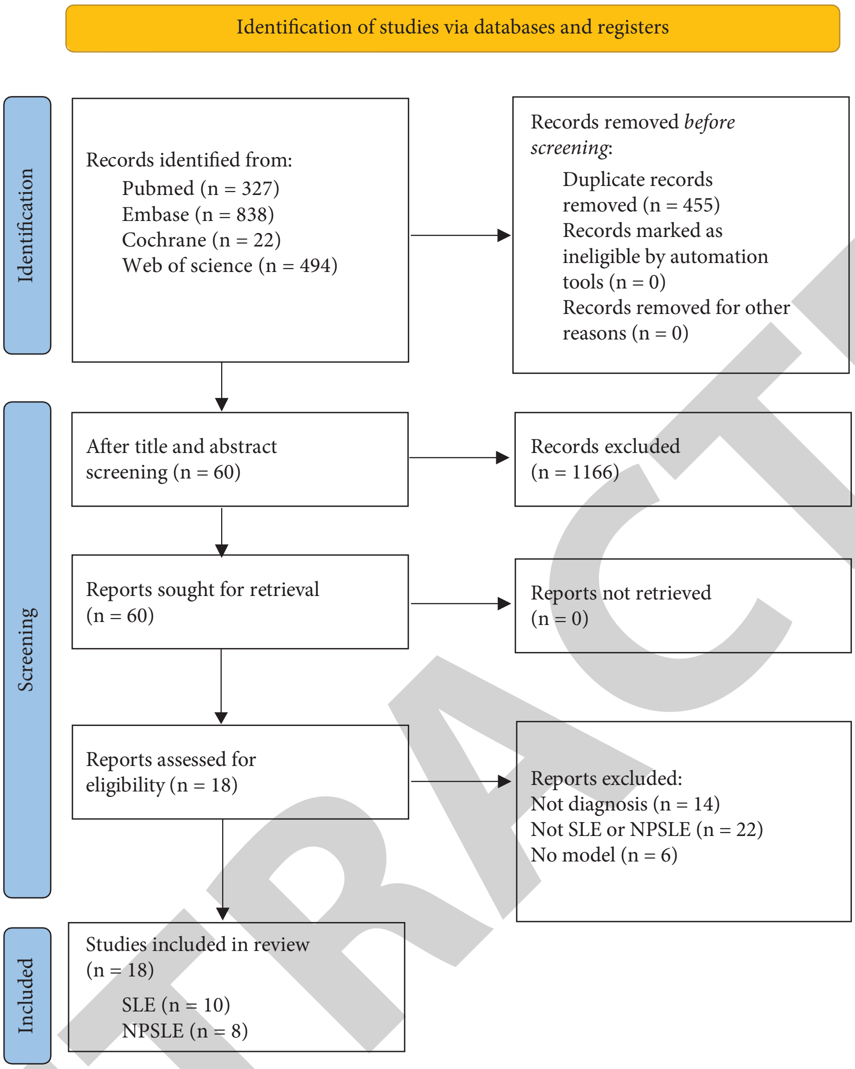

Identification of studies via databases and registers
Identification
Records identified from:
Pubmed (n = 327)
Embase (n = 838)
Cochrane (n = 22)
Web of science (n = 494)
Records removed *before screening*:
Duplicate records removed (n = 455)
Records marked as ineligible by automation tools (n = 0)
Records removed for other reasons (n = 0)
Screening
After title and abstract screening (n = 60)
Records excluded (n = 1166)
Reports sought for retrieval (n = 60)
Reports not retrieved (n = 0)
Reports assessed for eligibility (n = 18)
Reports excluded:
Not diagnosis (n = 14)
Not SLE or NPSLE (n = 22)
No model (n = 6)
Included
Studies included in review (n = 18)
SLE (n = 10)
NPSLE (n = 8)

FIGURE 1: Flowchart of the identification, inclusion, and exclusion of studies.

Risk of bias assessment for included studies was conducted according to QUADAS-2 criteria, via RevMan 5.3 software. The results are shown in Figure 2.

High risk of bias might be inevitable because most of the included studies ($n = 13$) were retrospective case-control design studies, and studies on NPSLE had limited sample size. Clinical practicability of included studies was graded as low risk, suggesting considerable clinical value of our study. We divided the included studies into a SLE subgroup and NPSLE subgroup for heterogeneity test and publication bias assessment, as shown in Figures 3 and 4, respectively.

3.2. Diagnostic Performance of ML for SLE. There are 10 studies included in meta-analysis for ML in SLE identification, with 15 different models and 19631 participants. Among the studies, 5 are from registration databases and 5 from retrospective case-control studies, with sufficient sample size. The results of analysis are presented in Figures 5(a), 6(a), 7(a), and 8(a). The AUC is 0.95 [95% CI (0.93, 0.93)], the sensitivity is 0.90 [95% CI (0.85, 0.93)], the specificity is 0.89 [95% CI (0.86, 0.92)], the PLR is 8.4 [95% CI (6.2, 11.4)], the NLR is 0.12 [95% CI (0.08, 0.17)], and the DOR is 73 [95% CI (40–134)]. According to Figure 7(a), based on a hypothesis of PLR = 10 and NLR = 0.1, there are still 8 algorithms that could competently distinguish SLE patients and non-SLE patients, in which 3 algorithms are in critical state. Figure 8 reveals the post-test probability of ML for SLE diagnosis, which indicated that assuming that the pre-test probability is 50%, the post-test probability of ML for SLE diagnosis is 0.89, and the probability of being diagnosed as non-SLE is 0.1.

3.3. Diagnostic Performance of ML for NPSLE. There are 8 studies included in meta-analysis for ML in NPSLE identification, with 18 different models and 569 participants. All the studies are from retrospective case-control studies, with limited sample size. The results of analysis are presented in Figures 5(b), 6(b), 7(b), and 8(b). The AUC of NPSLE

RETRACTED

Table 1: Characteristics of the included studies for SLE.

Author	Year	Country	Sample	Prediction	Definition	Sample number of training set (event/all)	K-fold cross validation	External validation	Model
Murray et al.	2019	USA	Registration data	Diagnosis	ACR	583/16767	10	Yes	LR
Ceccarelli et al.	2021	Italy	Case-control	Diagnosis	EULAR/ ACR	173/306	No	NA	LR, SVM, DT
Barnado et al.	2022	USA	Registration data	Diagnosis	NA	121/249	5	Yes	RF, XGB
Jorge et al.	2019	USA	Registration data	Diagnosis	ACR/ SLICC	66/200	No	Yes	LR
Adamichou et al.	2020	Greece	Registration data	Diagnosis	NA	401/802	10	Yes	LR
Huang et al.	2009	China	Case-control	Diagnosis	ACR	32/118	No	Yes	DT
Samundeswari et al.	2018	India	Case-control	Diagnosis	ACR	200/400	10	NA	ANN, SVM
Dai et al.	2010	China	Case-control	Diagnosis	ACR	50/99	10	Yes	KNN
Matthiesen et al.	2021	Portugal	Case-control	Diagnosis	NA	64/128	10	Yes	PLS
Turner et al.	2017	USA	Registration data	Diagnosis	ACR	272/562	5	Yes	ANN, RF, NB, SVM, Word2Vec

identification is 0.89 [95% CI (0.86, 0.92)], the sensitivity is 0.83 [95% CI (0.79, 0.87)], the specificity is 0.83 [95% CI (0.76, 0.88)], the PLR is 5.0 [95% CI (3.4, 7.3)], the NLR is 0.20 [95% CI (0.15, 0.27)], and the DOR is 25 [95% CI (13–47)]. Based on a hypothesis that PLR = 10 and NLR = 0.1, there are 3 ML models that could competently distinguish NPSLE patients and non-NPSLE patients. ML also presented excellent diagnostic performance for NPSLE.

4. Discussion

In this study, we reviewed studies that applied ML to diagnose SLE and NPSLE and conducted a meta-analysis. This is the first meta-analysis performed to evaluate the performance of ML for SLE identification, with high clinical significance. ML is the combination of statistics and computer science, which can make full use of information and obtain veiled, effective, and understandable knowledge from massive data to reveal connections between the data so as to build prediction models. ML typically falls into two categories: supervised learning and unsupervised learning [33–35]. It can assist clinicians in decision making via its remarkable pattern-recognizing capability and has shown excellent performance in identification for inflammatory diseases, cardiovascular diseases, and brain diseases [36, 37].

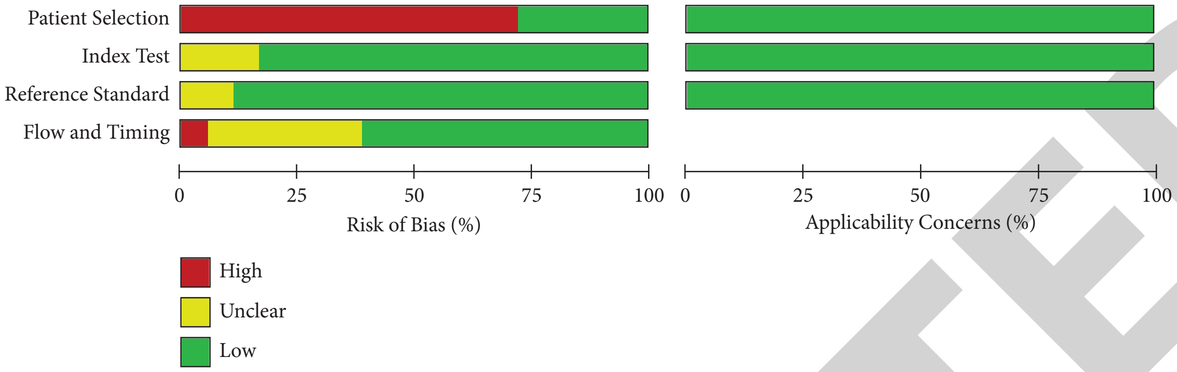
There are various ML models designed for SLE, with sufficient participants. Types of the models did not affect the diagnostic accuracy. Commonly applied methods such as random forest (RF) and logistic regression (LR) are used because most of the models included more than 10 variables. RF could produce highly accurate classifiers for various types of data and could evaluate the importance of variables when determining categories, so that it could produce unbiased estimates for generalized errors [38, 39]. This could relatively ensure the accuracy of multivariate modeling. LR is a machine learning method designed to solve classification problems. It is a predictive analysis based on probability distribution [40]. LR is less likely to overfit, although that might occur in high-dimensional datasets. The training time of LR is shorter than that of most complex algorithms (such as artificial neural network) due to the simplicity of probability interpretation; therefore, it has relatively practical diagnostic performance [41, 42]. The rest of the models, such as SVM, DT, and ANN, are also applied, with comparatively remarkable diagnostic performance. Among the 10 studies, 8 used K-fold cross validation and 8 used external validation. Accuracy of the models is reliable. Most of the models included clinical and laboratorial data as variables and added extra variables on the basis of ACR, EULAR, and SLICC criteria. The risk factors are ordered based on the model itself, which provided more directions and basis for us to improve the SLE classification criteria. On the other hand, there are 3 studies [20, 22, 23] that performed analyses for blood polypeptides and lipids, and 1 study [21] distinguished SLE patients from normal people through skin imaging examination. All of these studies yielded decent results and provided more directions for early identification of SLE in clinical practice.

There are limited number of studies that focused on ML for NPSLE, and MRI results are applied in these studies for learning and modeling so that most of the studies applied support vector machine (SVM) [43, 44]. ML has been widely applied in imaging diagnosis for many diseases such as brain metastases, retinopathy, and so on and has been statistically validated by meta-analysis [45, 46]. The optimization of SVM takes into account the minimization of empirical risk and structural risk so that it is stable. From a geometric point of view, the stability of SVM is reflected in that it requires the largest margin when constructing a hyperplane decision boundary; therefore, there is plenty of space between the boundaries to contain test samples, which is more suitable for solving image problems [47]. There are 5 studies that

RETRACTED

Table 2: Characteristics of the included studies for NPSLE.

Author	Year	Country	Sample	Prediction	Definition	Sample number of training set (event/all)	K-fold cross validation	External validation	Model	MRI machine	Tesla	Slice thickness (mm)
Tan et al.	2021	China	Case-control	Diagnosis	ACR	23/39	10	NA	Bagging, DT, SVM, AdaBoost, PCA + AdaBoost, VAE + SVM, PCA + DT, PCA + bagging, PCA + SVM, GA + SVM, MARL + GA + SVM	NA	3	NA
Li et al.	2020	China	Case-control	Diagnosis	ACR	23/39	3	NA	BL-SVM	SIGNA, General Electric Medical Systems	3	5
Zhuo et al.	2019	China	Case-control	Diagnosis	ACR	31/63	Leave-one-out CV	NA	SVM	Non-invasive 3D ASL-MRI; GE MR750, GE Healthcare	3	NA
Sun et al.	2010	China	Case-control	Diagnosis	ACR	27/54	NA	Yes	Decision tree	NA	NA	NA
Simos et al.	2019	Greece	Case-control	Diagnosis	EULAR	41/72	5	NA	SVM	Resting-state fMRI, Siemens Vision/Sonata scanner (Erlangen, Germany)	1.5	3
Inglese et al.	2022	Netherlands	Case-control	Diagnosis	ACR	68/136	NA	Yes	SVM	Philips Healthcare, Best, Netherlands	3	NA
Luo et al.	2022	China	Case-control	Diagnosis	ACR	36/83	NA	Yes	SVM	MAGNETOM Verio scanner (Siemens Healthcare)	3	6
Simos et al.	2020	Greece	Case-control	Diagnosis	EULAR	44/83	5	Yes	Random forest	Resting-state functional MRI, (Vision/Sonata, Siemens/Erlangen)	1.5	3

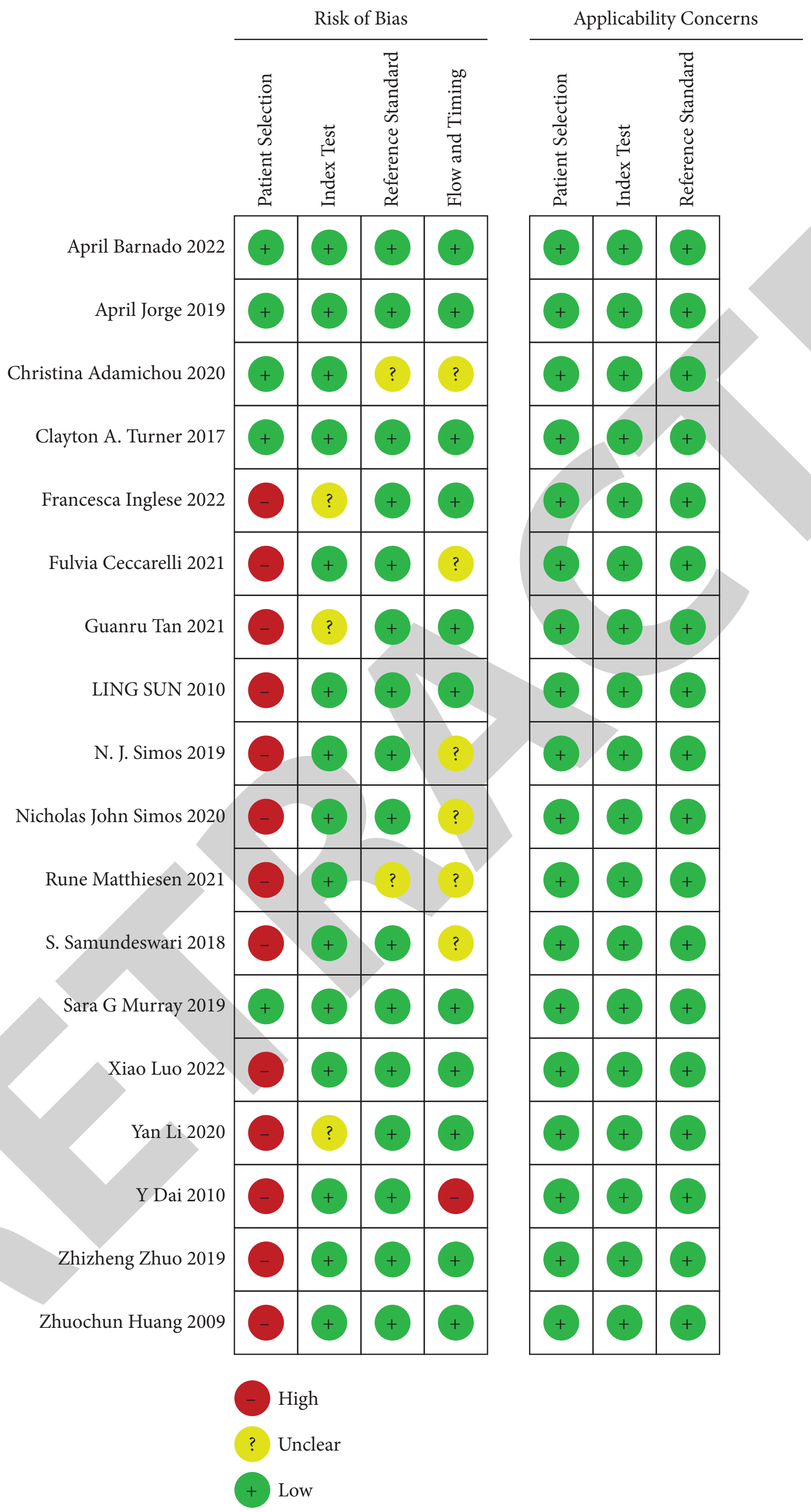


(a)

Figure 2: Continued.

RETRACTED

RETRACTED



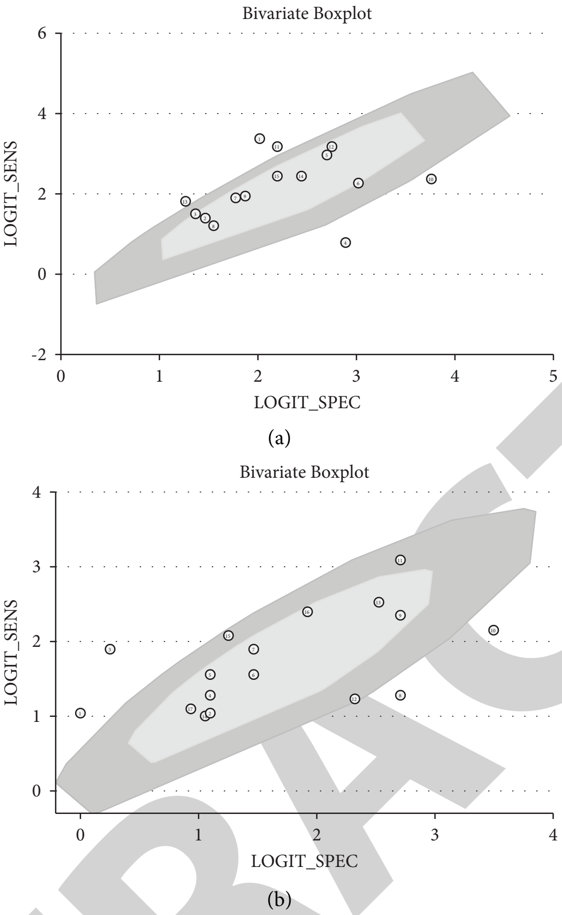
	Risk of Bias				Applicability Concerns		
	Patient Selection	Index Test	Reference Standard	Flow and Timing	Patient Selection	Index Test	Reference Standard
April Barnado 2022	+	+	+	+	+	+	+
April Jorge 2019	+	+	+	+	+	+	+
Christina Adamichou 2020	+	+	?	?	+	+	+
Clayton A. Turner 2017	+	+	+	+	+	+	+
Francesca Inglese 2022	–	?	+	+	+	+	+
Fulvia Ceccarelli 2021	–	+	+	?	+	+	+
Guanru Tan 2021	–	?	+	+	+	+	+
LING SUN 2010	–	+	+	+	+	+	+
N. J. Simos 2019	–	+	+	?	+	+	+
Nicholas John Simos 2020	–	+	+	?	+	+	+
Rune Matthiesen 2021	–	+	?	?	+	+	+
S. Samundeswari 2018	–	+	+	?	+	+	+
Sara G Murray 2019	+	+	+	+	+	+	+
Xiao Luo 2022	–	+	+	+	+	+	+
Yan Li 2020	–	?	+	+	+	+	+
Y Dai 2010	–	+	+	–	+	+	+
Zhizheng Zhuo 2019	–	+	+	+	+	+	+
Zhuochun Huang 2009	–	+	+	+	+	+	+

– High
? Unclear
+ Low

(b)

Figure 2: Quality evaluation of the included literature.

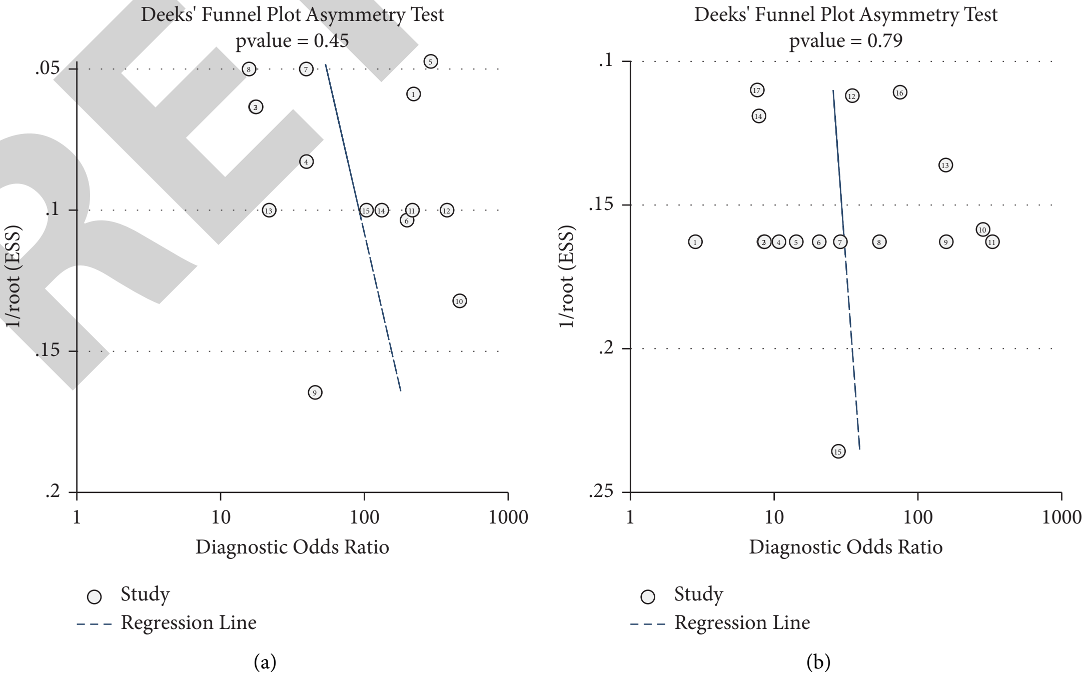
RETRACTED



(a)

(b)

Figure 3: Heterogeneity analysis: (a) for SLE; (b) for NPSLE.

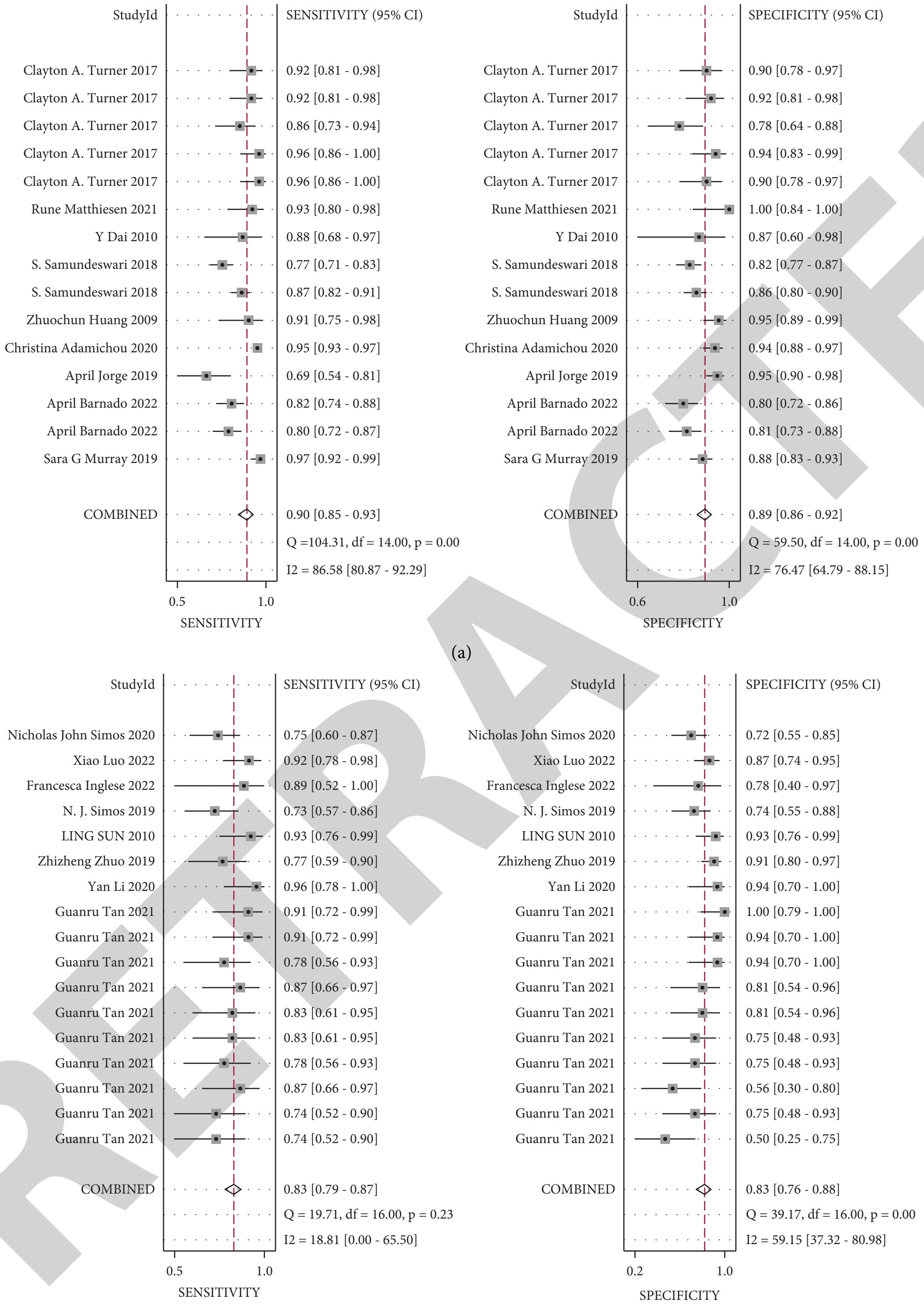


(a)

(b)

Figure 4: Publication bias: (a) for SLE; (b) for NPSLE.

RETRACTED



(a)

(b)

Figure 5: Diagnostic performance of machine learning: (a) for SLE; (b) for NPSLE.

applied K-fold cross validation and 4 studies that applied external validation to improve accuracy of the models, and SVM tended to overfit when the sample size was too small. Xiao et al. compared the results of ML with those of two senior radiologists and found the former to be more competent, which further improved the efficacy of ML. However, in the model training process, hundreds of different brain functional areas need to be identified and analyzed one by one, and then are sorted by the degree of influence, and the parts with greater influence are selected

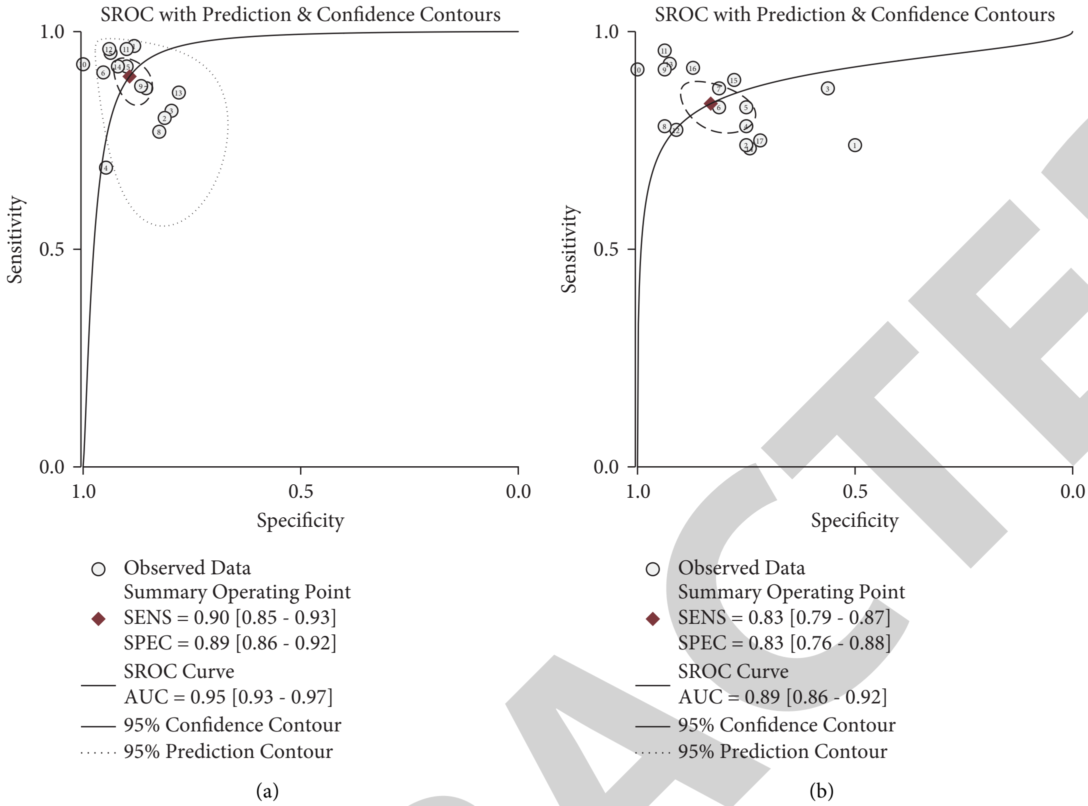


FIGURE 6: The summary receiver operating characteristic (SROC) curve for the diagnostic performance of machine learning: (a) for SLE; (b) for NPSLE.

RETRACTED

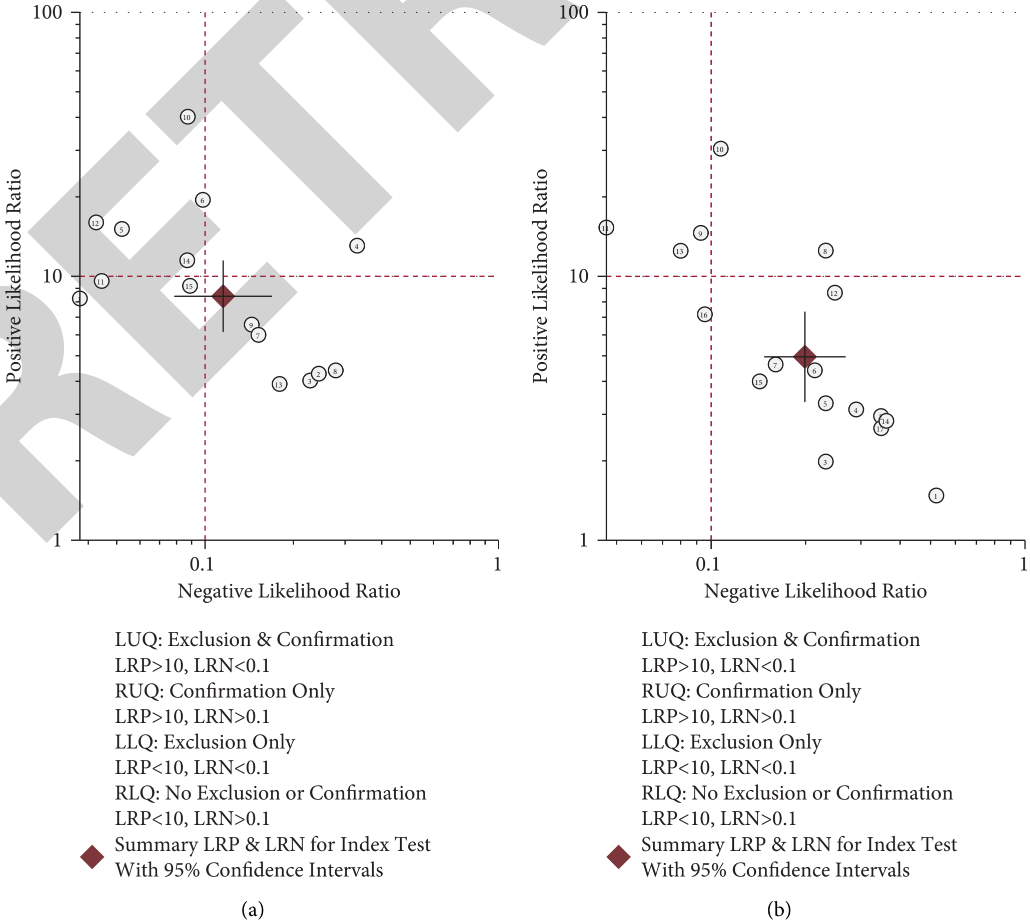


FIGURE 7: Likelihood matrix diagram of clinical application of machine learning: (a) for SLE; (b) for NPSLE.

RETRACTED

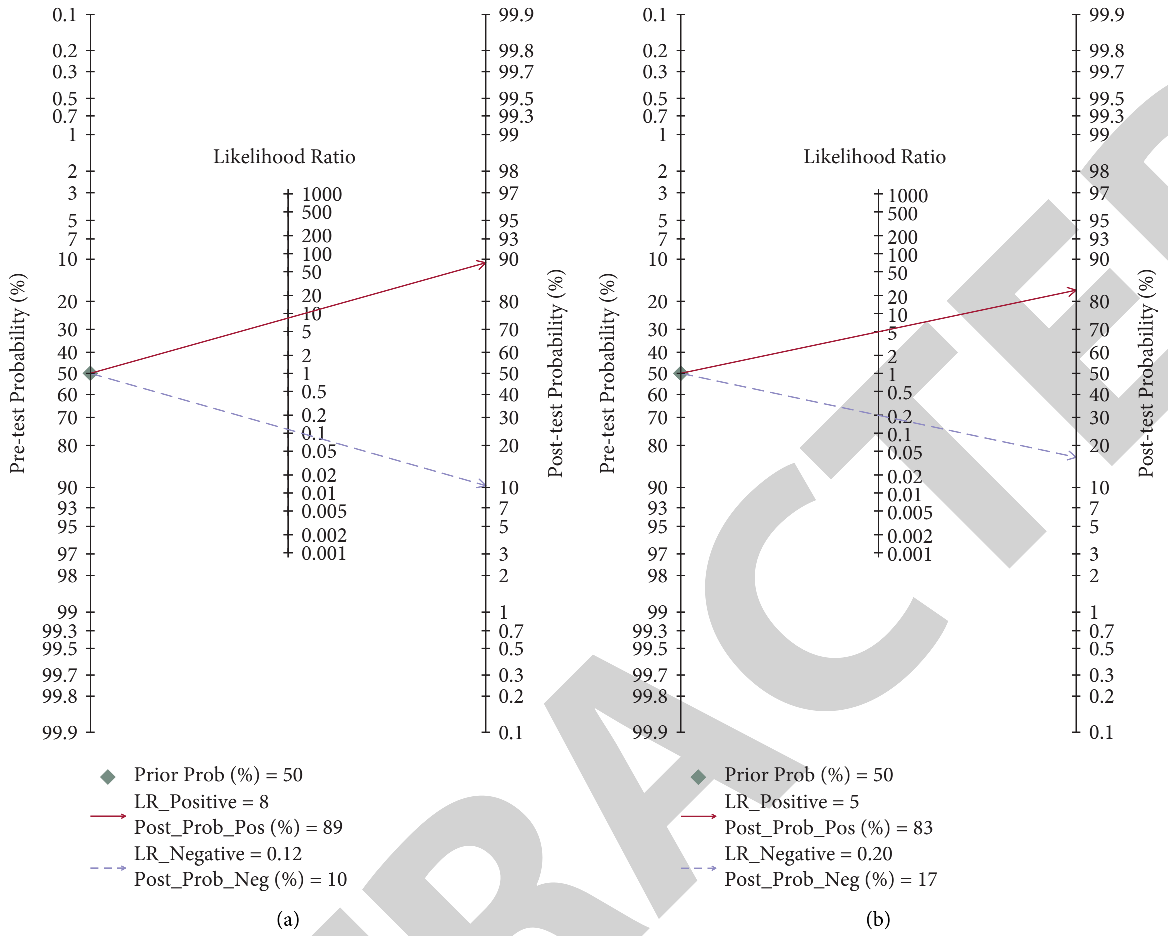


Figure 8: Fagan's diagram of the clinical application of machine learning: (a) for SLE; (b) for NPSLE.

for modeling, which often requires long time of training. With the wide and deep application of ML in imaging, it is necessary to improve the algorithm, increase the training efficiency, and reduce the time of training.

It is worth mentioning that in the process of searching the literature, we found that there are a large number of studies based on molecular and genetic levels, using machine learning methods to identify SLE risk genes or SLE-related antibodies. [48–55]. Several studies applied ML to evaluate the activity and prognosis of SLE [56–61] and to assist in classification of lupus nephritis [62, 63]. All these studies yielded satisfying results. Apart from all these, systematic literature reviews have been applied in different fields, such as rough set exploration system [64], machine learning methods for cyber security [65], and meta-learning for algorithm selection [66]. To sum up, ML is expected to play a variety of roles in clinical practice, and more relevant studies are needed.

4.1. Limitations. Though a comprehensive search was conducted in PubMed, Embase, Cochrane, and Web of Science, the number of included studies is small. Secondly, there is significant heterogeneity among the studies in terms of variable selection. We look forward to including more clinically significant, noninvasive, and easily collectible variables to further refine the model. At the same time, studies that focused on ML for NPSLE had limited sample size. It is difficult to recruit patients in countries with small populations so that most of the studies are from China, leading to lack of multiracial comparison. However, as NPSLE clinically represents the high activity and lethality of SLE, it is useful and necessary to conduct relevant meta-analysis. More ML models are needed to identify NPSLE in different races. Lastly, due to the clinical practice and the nature of the diagnostic experiment itself, the literatures included in this study inevitably lack prospective studies.

5. Conclusion

Machine learning (ML) is expected to be widely applied in clinical practice to assist in medical decision making. ML techniques can be used for the identification of systemic lupus erythematosus (SLE) and neuropsychiatric systemic lupus erythematosus (NPSLE). These techniques are considered an effective auxiliary method for the diagnosis of these diseases. These techniques gained great attention of the researchers, but there is still lack of evidence-based support. Systematic review and meta-analysis were conducted to evaluate the diagnostic accuracy and application prospect of

RETRACTED

ML techniques. Four libraries (PubMed, Embase, Cochrane Library, and Web of Science) were searched for the collection of relevant articles regarding machine learning for identification or diagnosis of SLE and NPSLE. Diagnostic accuracy of the SLE and NPSLE models was assessed using the bivariate fixed-effect model. Out of eighteen (18) studies, ten (10) were related to SLE and eight (8) to NPSLE. The AUC of SLE identification is 0.95, the sensitivity is 0.90, and the specificity is 0.89. AUC of NPSLE identification is 0.89, the sensitivity is 0.83, and the specificity is 0.83. It is concluded that ML played a significant role in identification of SLE and NPSLE.

Data Availability

The datasets used 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.

Authors' Contributions

Yuan Zhou and Yan Yan were responsible for conception and design. Yan Yan provided administrative support. Yuan Zhou was responsible for provision of study materials and patients. Yuan Zhou, Meng Wang, and Shasha Zhao were responsible for collection and assembly of data and data analysis and interpretation. All authors wrote the manuscript and approved the final version of the manuscript.

Acknowledgments

This research project was supported by Beijing Natural Science Fundation (No. 7202171).

References

[1] M. R. W. Barber, C. Drenkard, T. Falasinnu et al., "Global epidemiology of systemic lupus erythematosus," *Nature Reviews Rheumatology*, vol. 17, no. 9, pp. 515–532, 2021.

[2] C. S. Lau and A. Mak, "The socioeconomic burden of SLE," *Nature Reviews Rheumatology*, vol. 5, no. 7, pp. 400–404, 2009.

[3] K. A. Owen, A. C. Grammer, and P. E. Lipsky, "Deconvoluting the heterogeneity of SLE: the contribution of ancestry," *The Journal of Allergy and Clinical Immunology*, vol. 149, no. 1, pp. 12–23, 2022.

[4] M. C. Hochberg, "Updating the American College of Rheumatology revised criteria for the classification of systemic lupus erythematosus," *Arthritis & Rheumatism*, vol. 40, no. 9, p. 1725, 1997.

[5] M. Petri, A. M. Orbai, G. S. Alarcón et al., "Derivation and validation of the Systemic Lupus International Collaborating Clinics classification criteria for systemic lupus erythematosus," *Arthritis & Rheumatism*, vol. 64, no. 8, pp. 2677–2686, 2012.

[6] M. Aringer, K. Costenbader, D. Daikh et al., "European League against rheumatism/American College of Rheumatology classification criteria for systemic lupus erythematosus," *Arthritis & Rheumatology*, vol. 71, no. 9, pp. 1400–1412, 2019.

[7] Ö. Dahlström and C. Sjöwall, "The diagnostic accuracies of the 2012 SLICC criteria and the proposed EULAR/ACR criteria for systemic lupus erythematosus classification are comparable," *Lupus*, vol. 28, no. 6, pp. 778–782, 2019.

[8] A. Doria, M. Zen, M. Canova et al., "SLE diagnosis and treatment: when early is early," *Autoimmunity Reviews*, vol. 10, no. 1, pp. 55–60, 2010.

[9] M. H. Liang, M. Corzillius, S. Cheol Bae et al., "The American College of Rheumatology nomenclature and case definitions for neuropsychiatric lupus syndromes," *Arthritis & Rheumatism*, vol. 42, no. 4, pp. 599–608, 1999.

[10] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[11] M. Jiang, Y. Li, C. Jiang, L. Zhao, X. Zhang, and P. E. Lipsky, "Machine learning in rheumatic diseases," *Clinical Reviews in Allergy and Immunology*, vol. 60, no. 1, pp. 96–110, 2021.

[12] K. J. Kim and I. Tagkopoulos, "Application of machine learning in rheumatic disease research," *Korean Journal of Internal Medicine (Korean Edition)*, vol. 34, no. 4, pp. 708–722, 2019.

[13] A. Liberati, D. G. Altman, J. Tetzlaff et al., "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration," *PLoS Medicine*, vol. 6, no. 7, Article ID e1000100, 2009.

[14] P. F. Whiting, A. W. Rutjes, M. E. Westwood et al., "QUADAS-2: a revised tool for the quality assessment of diagnostic accuracy studies," *Annals of Internal Medicine*, vol. 155, no. 8, pp. 529–536, 2011.

[15] S. G. Murray, A. Avati, G. Schmajuk, and J. Yazdany, "Automated and flexible identification of complex disease: building a model for systemic lupus erythematosus using noisy labeling," *Journal of the American Medical Informatics Association*, vol. 26, no. 1, pp. 61–65, 2019.

[16] F. Ceccarelli, M. Lapucci, G. Olivieri et al., "Can machine learning models support physicians in systemic lupus erythematosus diagnosis? Results from a monocentric cohort," *Joint Bone Spine*, vol. 89, no. 3, Article ID 105292, 2022.

[17] A. Barnado, A. M. Eudy, A. Blaske et al., "Developing and validating methods to assemble systemic lupus erythematosus births in the electronic health record," *Arthritis Care & Research*, vol. 74, no. 5, pp. 849–857, 2022.

[18] A. Jorge, V. M. Castro, A. Barnado et al., "Identifying lupus patients in electronic health records: development and validation of machine learning algorithms and application of rule-based algorithms," *Seminars in Arthritis and Rheumatism*, vol. 49, no. 1, pp. 84–90, 2019.

[19] C. Adamichou, I. Genitsaridi, D. Nikolopoulos et al., "Lupus or not? SLE Risk Probability Index (SLERPI): a simple, clinician-friendly machine learning-based model to assist the diagnosis of systemic lupus erythematosus," *Annals of the Rheumatic Diseases*, vol. 80, no. 6, pp. 758–766, 2021.

[20] Z. Huang, Y. Shi, B. Cai et al., "MALDI-TOF MS combined with magnetic beads for detecting serum protein biomarkers and establishment of boosting decision tree model for diagnosis of systemic lupus erythematosus," *Rheumatology*, vol. 48, no. 6, pp. 626–631, 2009.

[21] S. Samundeswari, V. Ramalingam, B. Latha, and S. Palanivel, "Pattern classification techniques for the classification of cutaneous manifestations of systemic lupus erythematosus," *Pakistan Journal of Biotechnology*, vol. 15, no. 2, pp. 333–337, 2018.

[22] Y. Dai, C. Hu, L. Wang et al., "Serum peptidome patterns of human systemic lupus erythematosus based on magnetic bead separation and MALDI-TOF mass spectrometry analysis," *Scandinavian Journal of Rheumatology*, vol. 39, no. 3, pp. 240–246, 2010.

[23] R. Matthiesen, C. Lauber, J. L. Sampaio et al., "Shotgun mass spectrometry-based lipid profiling identifies and distinguishes between chronic inflammatory diseases," *EBioMedicine*, vol. 70, Article ID 103504, 2021.

[24] C. A. Turner, A. D. Jacobs, C. K. Marques et al., "Word2Vec inversion and traditional text classifiers for phenotyping lupus," *BMC Medical Informatics and Decision Making*, vol. 17, no. 1, p. 126, 2017.

[25] G. Tan, B. Huang, Z. Cui, H. Dou, S. Zheng, and T. Zhou, "A noise-immune reinforcement learning method for early diagnosis of neuropsychiatric systemic lupus erythematosus," *Mathematical Biosciences and Engineering*, vol. 19, no. 3, pp. 2219–2239, 2022.

[26] Y. Li, Z. Ge, Z. Zhang et al., "Broad learning enhanced (1)H-mrs for early diagnosis of neuropsychiatric systemic lupus erythematosus," *Computational and Mathematical Methods in Medicine*, vol. 2020, Article ID 8874521, 12 pages, 2020.

[27] Z. Zhuo, L. Su, Y. Duan et al., "Different patterns of cerebral perfusion in SLE patients with and without neuropsychiatric manifestations," *Human Brain Mapping*, vol. 41, no. 3, pp. 755–766, 2020.

[28] L. Sun, H. Chen, C. Hu et al., "Identify biomarkers of neuropsychiatric systemic lupus erythematosus by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry combined with weak cation magnetic beads," *Journal of Rheumatology*, vol. 38, no. 3, pp. 454–461, 2011.

[29] N. J. Simos, G. C. Manikis, E. Papadaki, E. Kavroulakis, G. Bertsias, and K. Marias, "Machine learning classification of neuropsychiatric systemic lupus erythematosus patients using resting-state fMRI functional connectivity," in *Proceedings of the IEEE International Conference on Imaging Systems and Techniques (IST) IEEE International School on Imaging*, Abu Dhabi United Arab Emirates, December 2019.

[30] F. Inglese, M. Kim, G. M. Steup-Beekman et al., "MRI-based classification of neuropsychiatric systemic lupus erythematosus patients with self-supervised contrastive learning," *Frontiers in Neuroscience*, vol. 16, Article ID 695888, 2022.

[31] X. Luo, S. Piao, H. Li et al., "Multi-lesion radiomics model for discrimination of relapsing-remitting multiple sclerosis and neuropsychiatric systemic lupus erythematosus," *European Radiology*, vol. 32, no. 8, pp. 5700–5710, 2022.

[32] N. J. Simos, S. I. Dimitriadis, E. Kavroulakis et al., "Quantitative identification of functional connectivity disturbances in neuropsychiatric lupus based on resting-state fMRI: a robust machine learning approach," *Brain Sciences*, vol. 10, no. 11, p. 777, 2020.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] R. C. Deo, "Machine learning in medicine," *Circulation*, vol. 132, no. 20, pp. 1920–1930, 2015.

[35] Z. Obermeyer and E. J. Emanuel, "Predicting the future - big data, machine learning, and clinical medicine," *New England Journal of Medicine*, vol. 375, no. 13, pp. 1216–1219, 2016.

[36] S. J. Al'Aref, K. Anchouche, G. Singh et al., "Clinical applications of machine learning in cardiovascular disease and its relevance to cardiac imaging," *European Heart Journal*, vol. 40, no. 24, pp. 1975–1986, 2019.

[37] N. Foulquier, P. Redou, C. Le Gal, B. Rouviere, J. O. Pers, and A. Saraux, "Pathogenesis-based treatments in primary Sjogren's syndrome using artificial intelligence and advanced machine learning techniques: a systematic literature review," *Human Vaccines & Immunotherapeutics*, vol. 14, no. 11, pp. 2553–2558, 2018.

[38] L. Blanchet, R. Vitale, R. van Vorstenbosch et al., "Constructing bi-plots for random forest: Tutorial," *Analytica Chimica Acta*, vol. 1131, pp. 146–155, 2020.

[39] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones et al., "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 20170387, 141 pages, 2018.

[40] I. Pinal-Fernandez and A. L. Mammen, "On using machine learning algorithms to define clinically meaningful patient subgroups," *Annals of the Rheumatic Diseases*, vol. 79, no. 10, 2020.

[41] J. H. Chen and S. M. Asch, "Machine learning and prediction in medicine - beyond the peak of inflated expectations," *New England Journal of Medicine*, vol. 376, no. 26, pp. 2507–2509, 2017.

[42] F. Cabitza, R. Rasoini, and G. F. Gensini, "Unintended consequences of machine learning in medicine," *JAMA*, vol. 318, no. 6, pp. 517-518, 2017.

[43] D. Shen, G. Wu, and H. I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, no. 1, pp. 221–248, 2017.

[44] J. G. Lee, S. Jun, Y. W. Cho et al., "Deep learning in medical imaging: general overview," *Korean Journal of Radiology*, vol. 18, no. 4, pp. 570–584, 2017.

[45] S. J. Cho, L. Sunwoo, S. H. Baik, Y. J. Bae, B. S. Choi, and J. H. Kim, "Brain metastasis detection using machine learning: a systematic review and meta-analysis," *Neuro-Oncology*, vol. 23, no. 2, pp. 214–225, 2021.

[46] B. J. Mortazavi, N. S. Downing, E. M. Bucholz et al., "Analysis of machine learning techniques for heart failure readmissions," *Circulation: Cardiovascular Quality and Outcomes*, vol. 9, no. 6, pp. 629–640, 2016.

[47] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.

[48] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321–332, 2015.

[49] M. K. Breitenstein, V. J. Hu, R. Bhatnagar, and M. Ratnagiri, "Approaching neural net feature interpretation using stacked autoencoders: gene expression profiling of systemic lupus erythematosus patients," *AMIA Jt Summits Transl Sci Proc*, pp. 435–442, 2019.

[50] P. Budde, H. D. Zucht, S. Vordenbäumen et al., "Multiparametric detection of autoantibodies in systemic lupus erythematosus," *Lupus*, vol. 25, no. 8, pp. 812–822, 2016.

[51] J. C. Almlöf, A. Alexsson, J. Imgenberg-Kreuz et al., "Novel risk genes for systemic lupus erythematosus predicted by random forest classification," *Scientific Reports*, vol. 7, no. 1, p. 6236, 2017.

[52] J. Imgenberg-Kreuz, J. C. Almlöf, D. Leonard et al., "Shared and unique patterns of DNA methylation in systemic lupus erythematosus and primary sjögren's syndrome," *Frontiers in Immunology*, vol. 10, p. 1686, 2019.

[53] M. H. Bahari, A. Azemi, M. M. Mirsalehi, M. Mahmoudi, and M. Khadime, "Recognizing pathogenic antibodies in SLE using general regression neural networks," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 4, pp. 2855–2864, 2012.

RETRACTED

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9890718, 1 page
https://doi.org/10.1155/2023/9890718

Retraction

Retracted: Magnetic Tile Surface Defect Detection Methodology Based on Self-Attention and Self-Supervised Learning

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Ling, Y. Wu, R. Ali, and H. Zhu, "Magnetic Tile Surface Defect Detection Methodology Based on Self-Attention and Self-Supervised Learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 3003810, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3003810, 10 pages
https://doi.org/10.1155/2022/3003810

RETRACTED

Research Article

Magnetic Tile Surface Defect Detection Methodology Based on Self-Attention and Self-Supervised Learning

Xufeng Ling,[1] **Yapeng Wu**,[2] **Rahman Ali**,[3] **and Huaizhong Zhu**[1]

[1]*Shanghai Normal University Tianhua College AI School, Shanghai, China*
[2]*Key Laboratory of Intelligent Infrared Perception, Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai, China*
[3]*University of Peshawar, Peshawar 19120, Khyber Pakhtunkhwa, Pakistan*

Correspondence should be addressed to Yapeng Wu; wuyapeng@mail.sitp.ac.cn

Received 8 April 2022; Revised 29 June 2022; Accepted 6 July 2022; Published 3 August 2022

Academic Editor: Muhammad Ahmad

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

As the core component of permanent magnet motor, the magnetic tile defects seriously affect the quality of industrial motor. Automatic recognition of the surface defects of the magnetic tile is a difficult job since the patterns of the defects are complex and diverse. The existing defect recognition methods result in difficulty in practical application due to the complicated system structure and the low accuracy of the image segmentation and the target detection for the diversity of the defect patterns. A self-supervised learning (SSL) method, which benefits from its nonlinear feature extraction performance, is proposed in this study to improve the existing approaches. We proposed an efficient multihead self-attention method, which can automatically locate single or multiple defect areas of magnetic tile and extract features of the magnetic tile defects. We also designed an accurate full-connection classifier, which can accurately classify different defects of magnetic tile defects. A knowledge distillation process without labeling is proposed, which simplifies the self-supervised training process. The process of our method is as follows. A feature extraction model consists of standard vision transformer (ViT) backbone, which is trained by contrast learning without labeled dataset that is used to extract global and local features from the input magnetic tile images. Then, we use a full-connection neural network, which is trained by using labeled dataset to classify the known defect types. Finally, we combined the feature extraction model and defect classification model together to form a relatively simple integrated system. The public magnetic tile surface defect dataset, which holds 5 defect categories and 1 nondefect category, is used in the process of training, validating, and testing. We also use online data augmentation techs to increase training samples to make the model converge and achieve high classification accuracy. The experimental results show that the features extracted by the SSL method can get richer and more detailed features than the supervised learning model gets. The composite model reaches to a high testing accuracy of 98.3%, and gains relatively strong robustness and good generalization ability.

1. Introduction

The magnetic tile is an important component of the motor. Its quality will affect the performance of the motor, and the defects of the magnetic tile will lead to the decline of the performance of the motor, thus affecting the service life of the motor. If the defective magnetic tile is used in industrial robots or other industrial products, it will cause huge losses. We aim to develop a method to automatically detect the defects of the magnetic tiles in this study. The effect of traditional manual defect testing methods is easily influenced by individual experience and other subjective factors, which bring low consistency and efficiency. On the other hand, machine visual detection methods have the advantages of higher automation, good consistency, and noncontact measurement; therefore, they gradually become the mainstream methods of surface defect detection. Magnetic surface defect detection methods can be categorized into traditional image processing methods and deep learning methods [1, 2]. Zhu et al. [3] proposed a magnetic tile surface defect detection algorithm based on improved homomorphic filtering and the Canny algorithm to solve the problems of uneven illumination, low contrast, and grinding texture of magnetic tile images. The experimental results show that this method has a good effect on the surface defect detection of magnetic tile and high detection accuracy, and is suitable for many types of surface defects of magnetic tile. Zhang et al. [4] proposed a method to perform the visual detection of line defects on the surface of micromagnetic tile by using the adaptive static mask, masking the contour of micromagnetic tile, nonlinear anisotropic diffusion equation, and suppressing the surface texture of micromagnetic tile. The experimental results show that the algorithm can accurately extract the line defects in the surface image of magnetic tile, and the detection accuracy reaches to 94.6%. Ma et al. [5] proposed a method based on K-means clustering to segment the break defects of magnetic tile surface, conducted in-depth research on the selection of light source and K-means clustering algorithm, and used two algorithms to segment the image of magnetic tile surface. The experimental results show that the K-means clustering algorithm can correctly segment the break defects of the magnetic tile surface. The traditional image processing methods have the good systematic and strong logical characteristics, and achieve a good effect on many specific targets, but they are highly dependent on illumination, and show less robust and less ideal effects in practical applications.

Since 2012, the deep learning method has achieved continuous breakthroughs, VGG, GoogleNet, ResNet, and other networks that have been proposed by researchers, and the classification accuracy has been improved, especially the image visualization and semantic analysis algorithms [6, 7] continuously improves the deep learning technologies. Many defect detection frameworks using machine vision were applied in practice [8]. Convolutional neural network (CNN) has gradually become the leading method for recognition and detection tasks, and the deep learning method has also been deeply applied in magnetic tile detection. Xie et al. [9] proposed a defect segmentation and classification method based on U-Net, and they extracted the defect feature through the U-Net coding part, classified the defect using the feature, and then, output the segmented defect region through the decoding part. The experimental results show that the classification accuracy of this method is 98.9%, which meets the high-precision requirements of industrial production, and provides a new idea for the automation of magnetic tile surface quality detection. Guo et al. [10] proposed a defect detection algorithm based on the masked region with convolution network (Mask R-CNN), which mainly solved the problem that the traditional defect detection algorithm failed to accurately segment defects. Through the steps of image preprocessing, residual network ResNet-50, constructing feature pyramid network, regional proposal network (RPN), fully convolutional network for semantic segmentation (FCN), and fully connected layer for prediction, the algorithm realizes strong generalization ability and accurately segments the defects on tile images with complex surface texture, uneven illumination, and low contrast. Zhang et al. [11] proposed a deep convolutional generative adversarial network (GAN) using the Gaussian mixture model to generate magnetic tile images with defects, which solves the problems of difficult collection of magnetic tile defect samples, uneven number of different defect samples, single defect type, and so on. Based on GAN, the algorithm regards the input noise potential space of the generated image as the Gaussian mixture model, and can generate magnetic tile defect images with good quality and rich defect types. Zhang and Wang [12] proposed a magnetic tile surface quality recognition system based on the convolutional neural network. The magnetic tile target is segmented from the collected image and normalized to obtain the standard image. The multiscale ResNet-18 is used as the backbone network to design the recognition system; a novel in-class mix-up operation is designed to improve the generalization ability of the system to samples; and the recognition accuracy of the system can reach 97.9%. Aiming at the problems of unclear magnetic tile imaging, low contrast, complex texture background, and many types of defects, Hu et al. [13] divided the types of magnetic tile surface defects into three categories and designed defect extraction methods, respectively, according to the different characteristics of the three types of surface defects. The experimental results show that this method can accurately and quickly extract the defect region on the surface of magnetic tile, and the detection accuracy is 93.5%. Li et al. [14] proposed a magnetic ring surface defect detection method based on the masked image. The magnetic ring surface defects are divided into two categories. According to the imaging characteristics of the two types of surface defects and their relationship with the background, the corresponding defect extraction methods are designed to use mask technology. Finally, many online experiments are carried out on samples with different illumination, specifications, defect types, and sizes. The experimental results show that this method has strong robustness, which can accurately and quickly extract the defects in each region of the magnetic ring surface image, and the detection accuracy reaches 95.3%.

Compared with traditional machine vision methods, the main advantage of deep learning methods is that they can automatically extract latent features without doing any manual feature engineering jobs like traditional methods, and have better robustness and adaptability in practical applications [15–17]. At present, the deep learning methods of magnetic tile image defect detection are all supervised learning methods. Nowadays, the supervised learning is the most common machine learning method. Given the dataset labeled manually in advance, the machine can learn to map the input data to the labeled target. Supervised learning has two disadvantages: first, it needs enough labeled training datasets including images, text, and voice, which mostly needs to be built manually. The data annotation process is a time-consuming and expensive process. In some areas, such as the medical field, obtaining a big labeled dataset is a challenging job. Second, data annotation will lead to information loss. For example, an image contains very rich information, including background information and secondary target information in addition to labeling objects. A single training task only extracts the information related to the training target in the image and ignores other valuable information. To solve the shortcomings of the supervised learning method, a single-stage defect recognition method,

RETRACTED

RETRACTED

which integrates defect detection and defect classification together of magnetic tile, is proposed in this study, a similar job what our team did in another research [18]. This method uses a deep neural network model based on the self-supervised learning methodology.

The key contributions of the proposed method include the following:

(i) An efficient multihead self-attention mechanism is proposed, which automatically locates single or multiple defect areas of magnetic tile and extracts features of the magnetic tile defects

(ii) The design of an accurate full-connection classifier can accurately classify different defects of magnetic tile defects.

(iii) A knowledge distillation process without labeling is proposed, which simplifies the self-supervised training process.

(iv) A flexible backbone network selection methodology is proposed, which can select suitable networks such as ViT and ResNet without changing the whole network structure.

In this study, the self-supervised learning network is used to train the encoder to do the feature extraction job. The backbone network is ViT. We make full use of the existing pretrained models to evaluate the classification effects of the target task. If there exists any pretrained model, which meets the requirements, then it can be directly used; otherwise, we use the self-supervised learning method to fine-tune the pretrained model. We also train a fully connected linear classifier MLP to do the classification task.

The rest of the study is structured as follows. Section 2 describes in the proposed research methodology. Section 3 realizes the proposed methodology by performing experiments and analyzing the results, and Section 4 concludes the research work did in this study.

2. Proposed Research Methodology

Since 2020, the self-supervised learning has increasingly attracted people's attention and become the most promising development direction at present [19]. Self-supervised learning is a supervised learning without human participation. It is not unsupervised learning [20–22]. With the exquisite design, it realizes self-supervised learning from the relationship of training data. Labels are generated from input data and are usually generated by the heuristic algorithm.

At present, the self-supervised learning of machine vision can be divided into two types: generative self-supervised learning and discriminant self-supervised learning [20, 23]. The main methods of generative self-supervised learning are VAE and GAN. The main idea is to train a model with the goal of learning the internal relationship of image pixels and then mask a part of the image, requiring the model to reconstruct or restore the obscured part of the image. This type of task is relatively difficult, because the model can reconstruct the image at the pixel level only when it has rich detailed information. The typical method of discriminant self-supervised learning is contrastive learning. Contrastive learning is to train a feature extraction model by automatically constructing similar and dissimilar instances. Through this model, similar instances are close in the projection of feature space, while dissimilar instances are far away in the projection space. The key points are how to construct similar and dissimilar instances, how to construct a model structure, and how to prevent model collapse. In general, compared with generative self-supervised learning, contrastive learning is less difficult.

At present, there are many kinds of self-supervised learning methods [24, 25] that can be roughly divided into two categories. The first one is generic self-supervised learning, such as scene occlusion removal, depth estimation, optical flow estimation, and image correlation point matching. The second one is discriminant self-supervised learning with the typical methods, such as solving jigsaw puzzles, motion propagation, rotation prediction, MoCo, and SimSam. The method in this study mainly refers to the method of Caron et al. [24] and belongs to the second category.

2.1. Workflow of the Proposed Methodology. To solve a coming downstream task such as an image classification task, the first step is data preparation, and an appropriate dataset for the task is to be carefully chosen. The second step is the model collection, and each modal $M_1, M_2, \ldots, M_n$ in the model zoo is taken out. The third step is KNN evaluation [26], and each existing pretrained model is evaluated by the KNN method to find whether there exists a model whose KNN classification accuracy exceeds 80%. If there is no pretrained model that meets the requirement, then we must do the step 4, using the self-supervised learning method to retrain a new model to improve its feature extraction performance, and go to the fifth step; on the contrary, if there exists model M_K that meets the requirement, then we do not need to retrain a new model; we can then do the fifth step; and the pretrained model M_K, which has relatively good feature extraction performance, will be fine-tuned to improve its classification ability. The fine-tuning job is to train an MLP (a fully connected linear classifier) attached to the feature extraction model, to complete the downstream classification. The process of the method is shown in Figure 1. The sixth step is result analysis to do research on the method.

2.2. The Backbone Network (ViT). The backbone network of contrastive learning in this study is the ViT [27]. Compared with the convolutional neural network CNN, the attention-based ViT method can achieve better results in image recognition, mainly because ViT can realize self-supervised learning. The model design of ViT in this study is basically consistent with the transformer architecture in the initial natural language processing [28]. This has three very significant advantages: first, the model setting is simple. Second, it is similar to transformer and has good scalability. Third, it is out of the box with efficient implementation. The

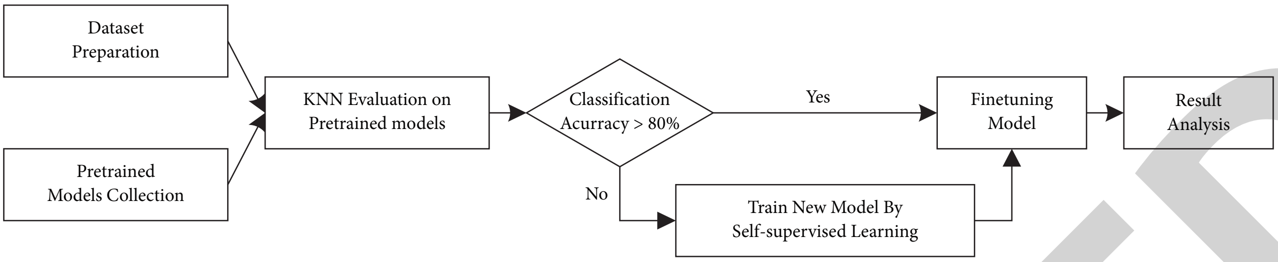


Figure 1: Workflow of the proposed method.

architecture of ViT model is mainly composed of five parts: image patch segmentation, image patch embedding, position embedding, sequence of self-attention modules, and a fully connected layer MLP that acts as a classification header. First, the input image is divided into $N \times N$ image patches, and each image patch is embedded into a vector (called image block embedding) through a fully connected network. Each image patch can be represented as a word, and the input image can be represented as a $N \times N$ word sequence. Then, we add an additional learnable token (called CLS), which may synthesize the information of the whole sequence at the head of the sequence. The task of image classification is transformed into the semantic recognition task. Next, we design the ViT network with sequence length L, dimension D, and K attention modules cascaded. Finally, there is a full-connection layer MLP, which realizes the classification function as a downstream task. The network structure is shown in Figure 2.

In this study, the input image size is 224×224 pixels; the patch is 16×16 pixels; and the image is split into 14×14 patches. There are 12 multi-self-attention head blocks cascaded to form the backbone network, and the main job of the backbone network is feature extraction. Each block holds 12 heads; the embedding dimension is 768; and the sequence length is 197. The MLP is for the downstream classification task.

2.3. Contrast Learning. The contrastive learning network consists of the student network and the teacher network. The student network and the teacher network are two ViT networks with the same structure but different parameters. For the input image x, the random image change transforms the image x into $X1$ and $X2$, respectively. $X1$ is input to the student network, and $X2$ is input to the teacher network. The k-dimensional feature outputs from the student network are normalized by softmax (taking the temperature parameter t as the denominator in the feature dimension) to obtain the probability distribution $P1$. The probability of the output of the teacher network is $P2$. To avoid the network collapse in the process, two skills, namely, averaging and sharpening, are adopted. Cross entropy is used to calculate the similarity between the student network output and the teacher network output. Because it is the input of different transformations of the same image, the error of the two network output distributions $P1$ and $P2$ is required to be small enough. The error loss is gradient backpropagated to the student network, while the teacher network does not. The update of the teacher network parameters is completed by an exponential moving average of the student network parameters. The model structure is shown in Figure 3.

2.4. Implementation Process. The student network g_{θ_s} (the parameter is θ_s) needs to be trained to get the same results as the output of the given teacher network g_{θ_t} (the parameter θ_t). Thus, for an input image $\mathbf{X}$, the $\mathbf{K}$-dimensional output probability distribution of the two networks is denoted as P_s and P_t. The probability distribution $\mathbf{P}$ can be obtained by softmax normalization of the network G output, that is, formula (1)as follows:

$$P_s(x)^i = \frac{\left(\exp\left(g_{\theta_s}(x)^i/\tau_s\right)}{\sum_{k=1}^{K}\exp\left(g_{\theta_s}(x)^k/\tau_s\right)}. \tag{1}$$

Here, τ_s is a temperature parameter that controls the steepness of the output distribution. The smaller the value of τ_s, the steeper it is; and the larger its value, the smoother it is. Given teacher network on the premise of g_{θ_t}, we match the output distribution of the two networks by minimizing the cross entropy.

$$\min H\left(P_t(x), P_s(x)\right). \tag{2}$$

Here, $H(a,b) = -a\log b$. Next, we describe formula (2) in detail. First, we intercept and deform the input image, construct different views, and form the view set v. This collection includes 2 global views (x_{g1}, x_{g2}) and 8 local views $(x_{l1}, x_{l2}, \ldots, x_{l8})$. The global view is only available to the teacher network; all views are available to the student network. The local-global similarity is formed in this way, and the training process is to minimize the loss as described in formula (3)as follows:

$$\text{loss} = \min \sum_{x \in x_1^g, x_2^g} \sum_{x' \in V x' \neq x} H\left(P_t(x), P_s(x')\right). \tag{3}$$

The image resolution is set to 224×224, and the size of the global view is set to (0.4–1) to cover a large area of the input image, while the size of the local view is set to (0.15–0.4) to cover only a small area of the input image. The parameters of the network can be trained out by minimizing the formula (3) of random gradient descent θ_s.

2.5. Candidate Pretrained Model Selection. The pretrained model has two very valuable characteristics: first, the features extracted by the self-supervised ViT method can have image

RETRACTED

RETRACTED

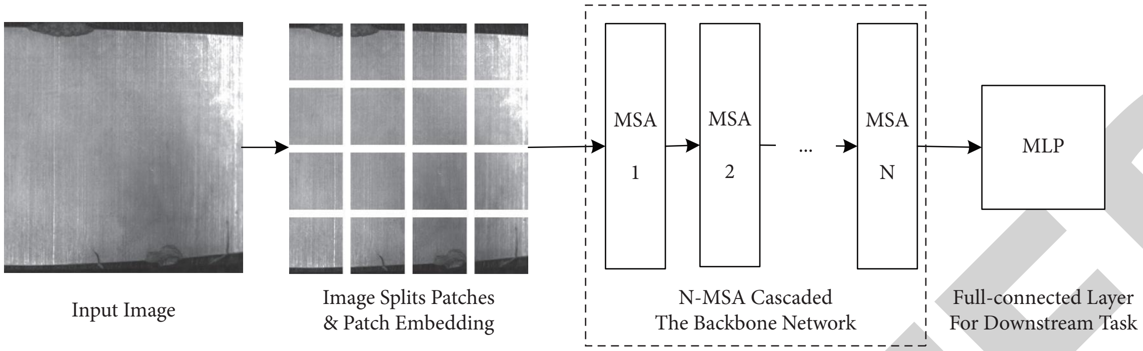


Figure 2: Network structure of ViT, and MSA denotes multi-self-attention head block.

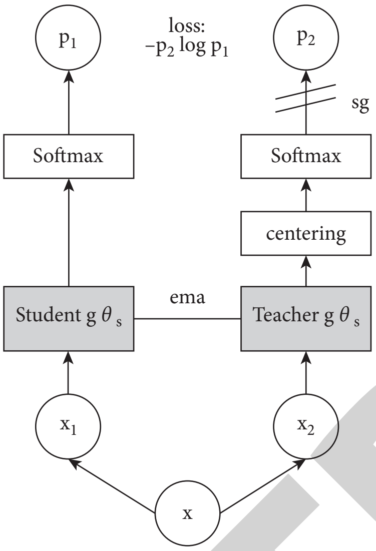


Figure 3: Architectural design of self-supervised DINO model; SG indicates no reverse gradient propagation; and EMA indicates exponential moving mean update.

segmentation ability, including scene layout and object boundary. This information can be obtained by extracting the self-attention features of the last layer. Second, the features extracted by self-supervised DINO-ViT do not need any fine-tuning, linear classifier, or data enhancement. Only the most basic KNN method can achieve a good classification effect, and the top-1 accuracy on ImageNet [29] can reach 78.3%. This study proposes a self-supervised learning model based on the combination of contrastive learning and self-attention mechanism for magnetic tile recognition.

There are a series of open-source pretrained models based on ImageNet on GitHub. Some models are listed in Table 1 [29]. These models have good performance in ImageNet and can be directly used or after training and tuning. Table 1 contains self-supervised feature extraction models with different parameters, in which “Layers” refers to the layers of transformer; “Dimension” refers to the working dimension of transformer; “Number of heads” refers to the number of heads with attention, similar to the number of convolution kernels in CNN; “Token number” refers to the length of embedded code; and “Parameter quantity” refers to the total parameters of the model, excluding the parameters of the MLP full-connection layer.

Table 1: List of parameters of pretrained model.

Model	Layers	Dimension	Heads	Tokens	Parameters (M)
ResNet-50	—	2048	—	—	23
ViT-S/16	12	384	6	197	21
ViT-S/8	12	384	6	785	21
ViT-B/16	12	768	12	197	85
ViT-B/8	12	768	12	785	85

Table 2: Magnetic tile defect dataset details.

Category	Blowhole	Crack	Break	Fray	Uneven	Free
Number	115	114	85	32	103	952

Each existing pretrained model is validated to find out whether it can meet the requirement. For ViT-S/8 & ViT-6/8, 8 stands for small image patch with 8*8 pixels, with the input image 224*244; it has longer patch sequence of 785 and has 4 times heavier computation than ViT-S/16 and ViT-B/16, respectively; and meanwhile, the classification accuracy is improved only by 1–2%. Finally, we choose ViT-B/16 to do extensive experiments, so it has the balance between performance and computation.

3. Experimental Results and Analysis

The SSL-ViT model proposed in this study is trained, verified, and tested on the open data of magnetic tile. The experiment is completed on a server configured with four RTX 2080Ti GPUs. The server operating system is CentOS 7.6 and memory is 128G. The program is implemented in Python language and is completed on the Python 1.60 deep learning platform. The specific experimental work includes selection of magnetic tile image dataset, image data preprocessing, model training, model optimization, model test, comparative experiment, and so on.

3.1. Data Preparation. The experimental dataset in this study is the magnetic tile surface defect dataset. The dataset was collected by Huang from the Institute of Automation,

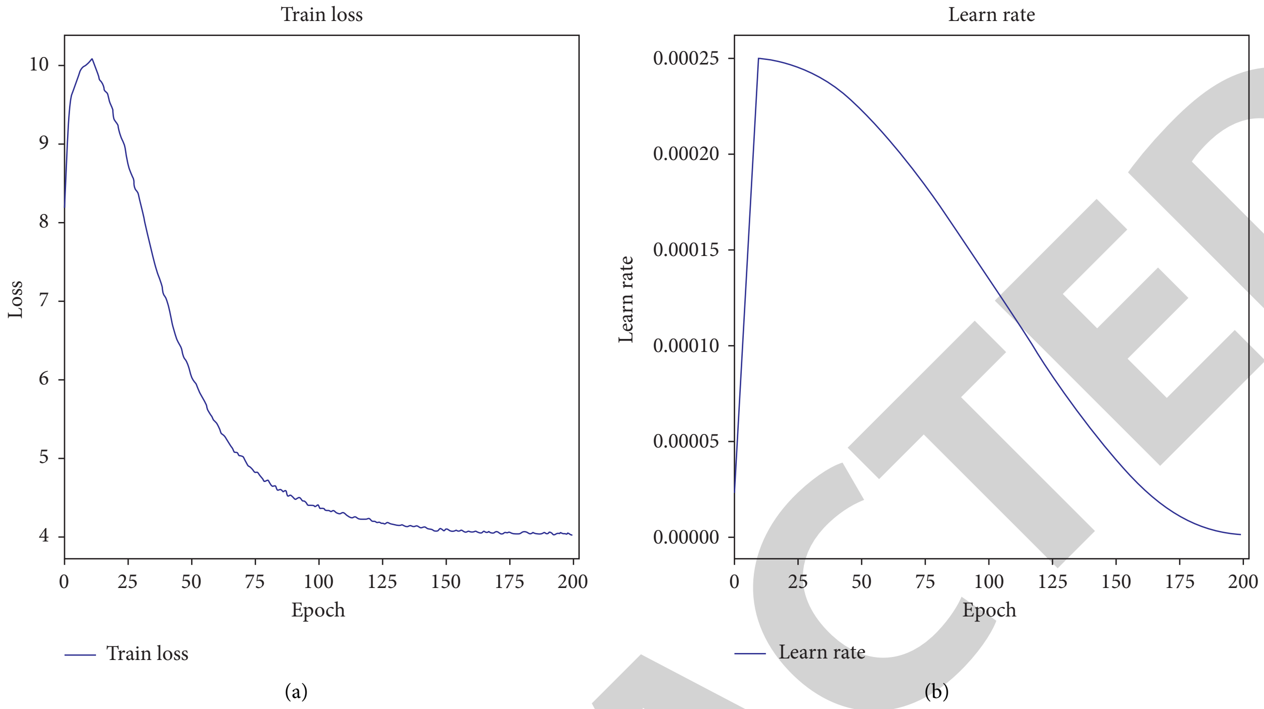


Figure 4: Schematic diagram of training loss and learning rate, and the pretrained model is ViT-B/16.

Chinese Academy of Sciences, in order to detect and identify five defects in industrial magnetic tile. The defects in the dataset are blowhole, crack, break, fray, and unevenness. Each defect has dozens to hundreds of gray images, and the size of each gray image is also inconsistent. We randomly divided the dataset into training set, verification set, and test set, with a ratio of 8 : 1 : 1.

The dataset is shown in Table 2. It can be seen from the table that a large number of pieces, i.e., 952, are without defects. The fray category of defect data is less, and other categories are relatively balanced. The number of data samples is small; the large dataset needs to be enhanced through data enhancement; and then, we can reduce the dependence of the model on irrelevant attributes, and improve the robustness and generalization ability of the model. An online data augment method is used to increase the samples; each sample of the dataset is translated, resized, flopped, and stretched to produce more samples. Since the experimental platform is PyTorch, many methods in torchvision package such as RandomResizedCrop, RandomHorizontalFlip, RandomRotation, and RandomAffine are used to achieve online data augmentation operation.

Table 3: Estimation of KNN classification accuracy, N means, and N neighbors.

N	5 (%)	10 (%)	20 (%)	30 (%)
Classification accuracy	96.9	94.7	86.5	81.2

3.2. Retraining of Pertained Model. The existing pretrained model is trained on ImageNet and has good image feature extraction functions. Because ImageNet images are mainly images in nature, the pretrained model has a strong ability to extract the characteristics of natural images. As an industrial product, the visual characteristics of magnetic tile are significantly different from natural objects, and the direct use of the pretrained model may result in poor effects. However, this does not mean that the pretrained model is worthless, and the basic visual feature extraction function is very valuable. We import the pretrained model and then train it with the magnetic tile dataset, so that the model can extract the magnetic tile features well.

We can judge a pretrained model as whether it can do a good feature extraction job for a certain dataset such as the magnetic tile dataset by simply using KNN (K-nearest neighbor, an automatic classification method) validation [11]. Experiments show that if the classification accuracy of KNN validation of a pretrained model on a certain dataset is above 80%, the feature extraction effect of the pretrained model is good, and the next job is to fine-tune the model to achieve better downstream classification task. If the KNN validation gets a low classification accuracy score, the pretrained model cannot extract features well from the task of the certain dataset, and the next job is to choose another pretrained model or retrain a new model to improve its feature extraction ability. In this study, the KNN classification accuracy of the pretrained model is 67.35%, which cannot meet the requirements; therefore, the pretrained model cannot be used and need to be retrained. The pretrained model is retrained over the magnetic tile dataset to optimize its feature extraction ability. The training epoch is set to 200. The optimizer is set as ADM, and the learn rate is set as the cosine optimization scheme. The training loss and learn rate are shown in Figure 4.

RETRACTED

RETRACTED

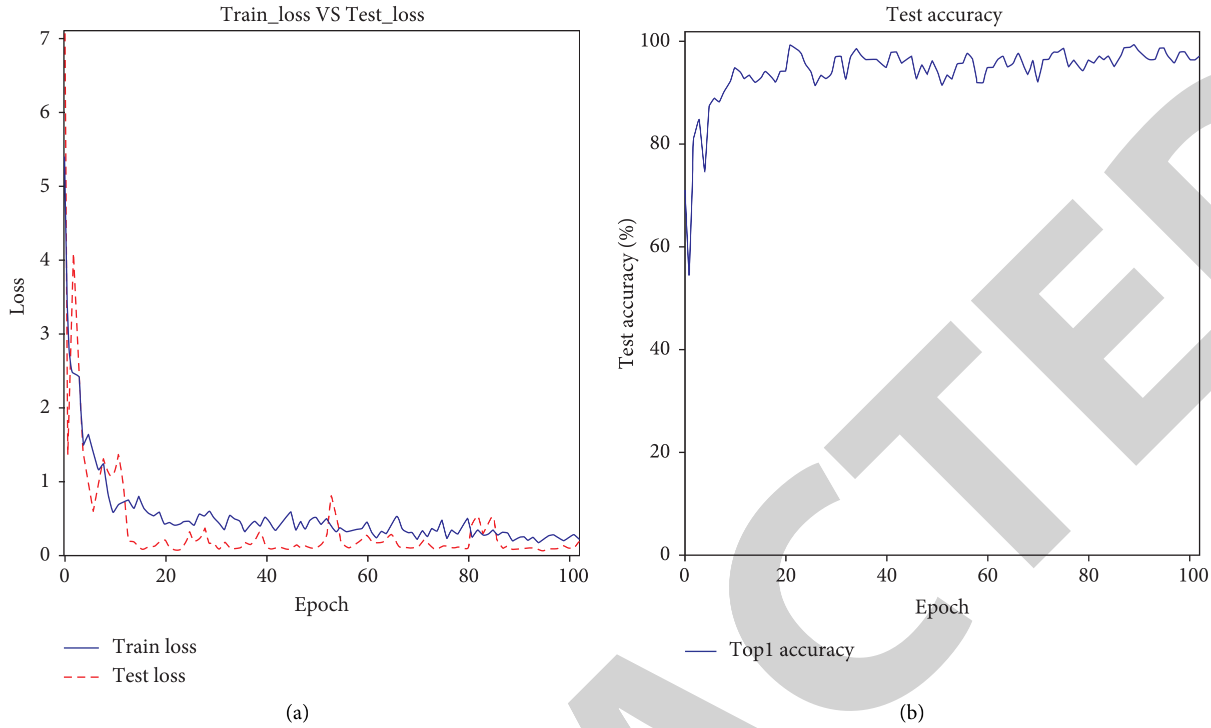


Figure 5: Linear classification training, testing loss, testing accuracy, and the pretrained model are ViT-B/16.

During the retraining of the pretrained model with the magnetic tile data, the training error change curve is shown in Figure 4(a). With the increase in epoch times, the training error decreases. When the epoch reached 100, the training error would basically converge. When the epoch reached 200, the training error would converge to about 3.9. Figure 4(b) shows the change curve of learning rate. The warm-up linearly increases in the first 30 cycles (this is a skill to improve training efficiency) and then gradually decreases.

3.3. KNN Estimation and MLP.

3.3. KNN Estimation and MLP. When the training period exceeds 100, the decline speed of training error will slow down; when the training period exceeds 150, the training error will tend to converge; then, the retraining of the pretrained model is basically completed; and the feature extraction network is optimized. Next, we use KNN to evaluate the feature extraction ability of the model. The results are shown in Table 3.

Because the number of samples in the dataset is relatively small, when the N increases, the classification accuracy of KNN decreases. Generally, we take $N = 10$ as the reference basis, and the classification accuracy reaches 94.7%. Therefore, we have reasons to believe that the trained model has a good ability of magnetic tile feature extraction.

Next, we train a fully connected network MLP to classify features, that is, applying the linear classifier training to realize the classification function. This is a downstream task and a training process of supervised learning. The ratio of training set, verification set, and test set is 8 : 1 : 1; epoch is set to 100; and the training process is shown in Figure 5. We can see that the change curve of training error and verification error is shown in Figure 5(a). The loss value of training set and test set gradually decreases with the increase in epoch. When the epoch value is greater than 40, the training error hardly decreases; the verification error also changes very little; and the change curves of classification accuracy of training set and the verification set are shown in Figure 5(b). With the increase in epoch times, the accuracy of training set and verification set increases. Finally, the accuracy of verification set is 98.25%.

We train a feature extraction network to realize image feature extraction through self-supervised learning and then learn a linear classifier through a supervised learning method to complete the downstream classification task. This hybrid model of self-supervised learning combined with supervised learning not only achieves a good classification effect, but also improves the universality of the model, reducing the threshold of practical application, and has a broad application prospect.

3.4. Comparative Test and Analysis. We conducted a comparative experiment and compared it with the Swin-ViT, ViT, and ResNet-50 using supervised learning. We found that if we do not do region extraction and other work, and directly use the supervised learning Swin-ViT, T2T-ViT, and ResNet50 networks to classify magnetic tiles, the classification accuracy of all the supervised methods does not exceed 80.0%, as shown in Table 4. Because the defects of the magnetic tile are open and have no fixed mode. In this case, if the defect region is not extracted in advance, the traditional supervised learning method is difficult to accurately extract

Table 4: Comparison of SLL method with other supervised methods.

N Nearest-neighbor method	RestNet-50	ViT-base	Swin-ViT-base	SSL method
Top-1 accuracy	63.0%	67.7%	78.5%	98.3%
Epoch	200	200	200	200

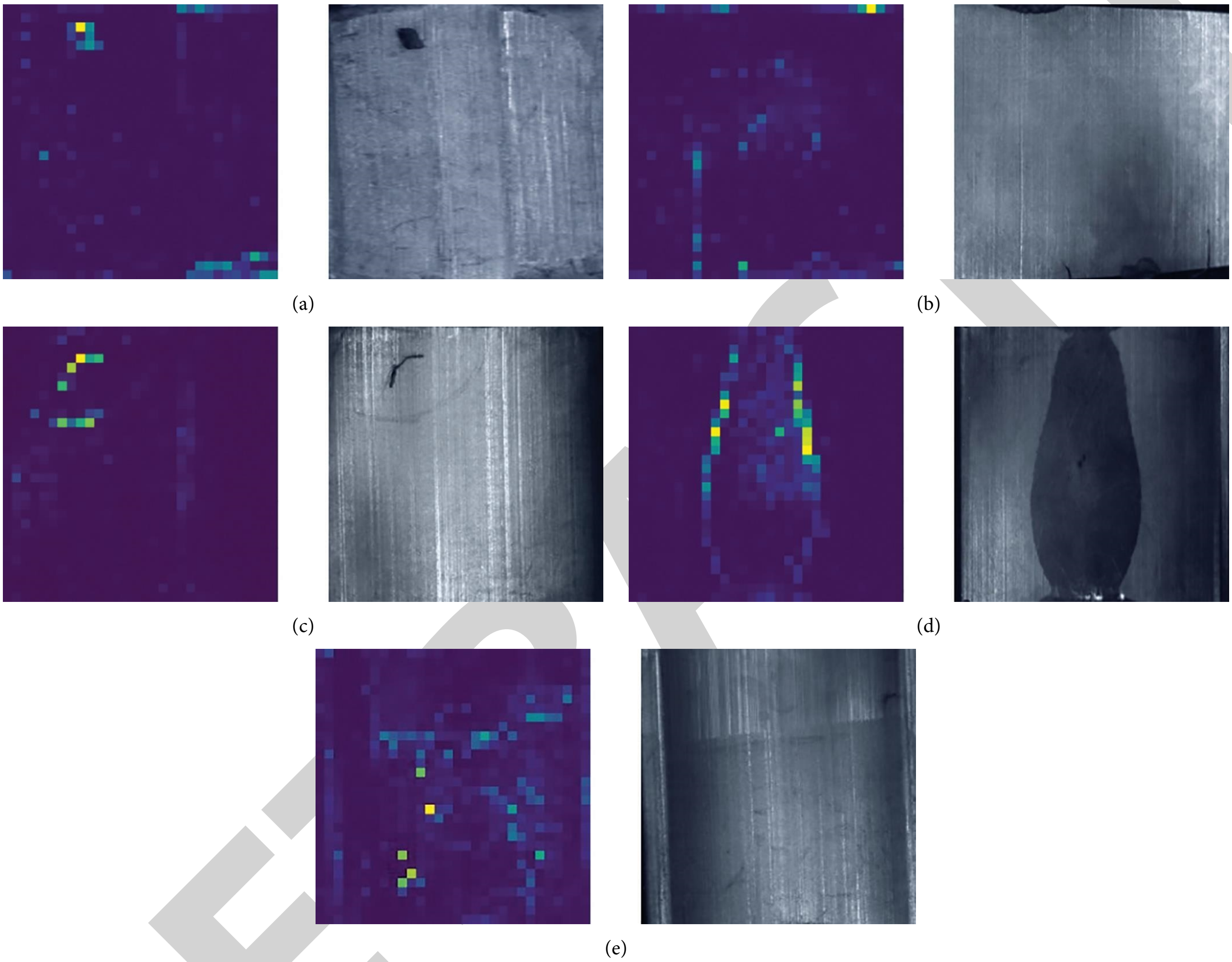


Figure 6: Original images of various defects and self-attention images. Different colors denote different attention scale; yellow denotes high attention; bright green denotes common attention; and dark green denotes low attention. (a) Blow hole. (b) Break. (c) Crack. (d) Fray. (e) Uneven.

RETRACTED

features of the magnetic tile, and it is difficult to achieve high classification accuracy.

3.5. Attention Map Analysis. The model has multiple attention heads. The experimental model in this study has six attention heads, and each attention head focuses on different attention regions. We note that the attention area of each attention head is very similar to human visual characteristics. The areas that the attention heads pay attention to are those with very rich information. Further observation and research show that the defect areas will be noticed by one or more attention heads. This is the advantage of self-supervised learning. It is not affected by label data, and the learned features are very comprehensive. Self-supervised learning has a better effect on open tasks than on closed tasks. We carefully analyzed the model feature extraction and attention map, which is shown in Figure 6.

In Figure 6, the left side is the attention map and the right side is the original image. Figure 6(a) is a blowhole defect; Figure 6(b) is a crack defect; Figure 6(c) is a break defect; Figure 6(d) is a fray defect; and Figure 6(e) is an unevenness defect. The feature extraction method of the self-supervised model is richer and more detailed, which can automatically pay attention to the location of defect attention, and find the abnormal region location. For complex industrial vision, using a self-supervised method can have a good effect.

The self-attention maps contain information about image segmentation. Different attention heads can process

RETRACTED

different semantic regions of an image, even if they are occluded. ViT of supervised learning cannot deal with messy objects well, both qualitatively and quantitatively. On the contrary, self-supervised ViT can deal with these objects. Even if the scene is very complex, the defect area of the image can be segmented from the complex nondefect background by multiple attention heads. We believe that the SLL method pays attention to the abnormal areas of the image and separates the abnormal region from the normal region. But sometimes, abnormal areas are nondefect area, and in this case, we need a downstream MLP to do further nonlinear classification to solve this problem.

4. Conclusions

To solve the difficult task of magnetic tile defect detection of industrial product quality control job, we use self-supervised learning method for feature extraction and combined supervised learning method for the downstream classification task, to detect the defects of the magnetic tile. This method will be able to undertake a single-stage magnetic tile defect detection. The open-source magnetic tile surface defect dataset, which contains five defect categories, is used in this research. The experimental results show that the self-supervised learning method has unique advantages in feature extraction. First, multiple self-attention heads can automatically locate different defect locations. Second, the model holds image segmentation information of complex scenes. Third, the model has strong feature extraction and generalization ability. The above characteristics show that the self-supervised method can extract global semantic features and local detail features. The model uses the fully connected network to complete various downstream classification tasks. Since the above method is flexible and convenient, it holds practical application value for many industrial product quality inspections. In addition, the self-supervised learning model has a good application prospect in the field of industrial images. The dataset we use to train and test our method in this study is relatively not big enough, and we need to collect more samples of magnetic tile defects to develop a larger dataset to train and test our method in the next study. The future work is to test the self-supervised learning model in multiple industrial scenes, develop a self-supervised learning feature extraction model with excellent generalization, develop a lightweight model that can be applied in smart manufacturing, and also find out whether the industrial conditions like temperature and humidity affect the results of this study. A practical magnetic tile defect detection system will be widely used in smart manufacturing and bring huge economic benefits.

Data Availability

Data can be provided on the request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (42075134).

References

[1] Shockletti, "Review on the application of surface defect detection," *Electronic Technology*, vol. 49, no. 8, pp. 189–191, 2020.

[2] Q. F. Jin and L. Chen, "A Survey of Surface Defect Detection of Industrial Products Based on a Small Number of Labeled Data," 2022, https://arxiv.org/list/cs.CV/new.

[3] Z. X. Zhu, L. Zhao, H. Li, and H. R. Wang, *Research on Magnetic Tile Surface Defect Detection Algorithm Based on Improved Homomorphic Filtering and Canny Algorithm*, Laser & Optoelectronics Progress, Germany, 2021.

[4] L. B. Zhang, J. F. Li, and J. M. Shen, "Study on visual detection method of surface linear defects on micro-magnetic tile," *Journal of Optoelectronics - Laser*, vol. 30, no. 9, pp. 951–959, 2019.

[5] X. D. Ma, R. B. Yuan, and H. F. Li, "A segmentation method of magnetic tile defect image based on k-means clustering," *Software Guide*, vol. 18, no. 12, pp. 180–186, 2019.

[6] H. Naeem, B. Guo, M. R. Naeem, F. Ullah, H. Aldabbas, and M. S. Javed, "Identification of malicious code variants based on image visualization," *Computers & Electrical Engineering*, vol. 76, pp. 225–237, 2019.

[7] F. Ullah, M. R. Naeem, L. Mostarda, and S. A. Shah, "Clone detection in 5G-enabled social IoT system using graph semantics and deep learning model," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3115–3127, 2021.

[8] S. A. Singh and K. A. Desai, *Automated Surface Defect Detection Framework Using Machine Vision and Convolutional Neural Networks*, Intell Manuf, 2022.

[9] J. Xie, J. M. Yao, Q. Yao, and Z. X. Yan, "Segmentation and recognition of magnetic tile surface defects based on deep learning," *Chinese Journal of Liquid Crystals and Displays*, vol. 36, no. 5, pp. 713–722, 2021.

[10] L. Y. Guo, H. Y. Duan, W. W. Zhou et al., "Surface Defect Detection Algorithm of Magnetic Tile Based on Mask R-CNN," *Computer Integrated Manufacturing Systems*, no. 5, pp. 1393–1400, 2022, https://kns.cnki.net/kcms/detail/11.5946.tp.20210809.1131.004.html.

[11] J. Zhang, J. Xie, F. M. Liang, X. Y. Xu, and J. J. Dong, "Research on generation method of magnetic tile defect image based on improved DCGANs network," *Journal of Chinese Computer Systems*, vol. 42, no. 3, pp. 589–594, 2021.

[12] J. A. Zhang and J. T. Wang, "Magnetic tile surface quality recognition based on multi-scale convolution neural network and within-class mixup operation," *Journal of Computer Applications*, vol. 41, no. 1, pp. 275–279, 2021.

[13] H. Hu, J. F. Li, and J. M. Shen, "Detection methods for surface micro-defection on small magnetic tile based on machine vision," *Journal of Mechanical* & *Electrical Engineering*, vol. 36, no. 2, pp. 117–123, 2019.

[14] J. F. Li, Z. X. Zhang, and J. M. Shen, "Study on surface defect extraction of magnetic ring based on masking technology," *Journal of Optoelectronics - Laser*, vol. 28, no. 7, pp. 732–741, 2017.

[15] H. B. Shi, B. B. Zhang, and Q. M. Zhang, "A multitarget visual attention based algorithm on crack detection of industrial

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9897287, 1 page
https://doi.org/10.1155/2023/9897287

Retraction

Retracted: Artificial Intelligence-Based Feature Analysis of Ultrasound Images of Liver Fibrosis

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Xie, S. Chen, D. Jia, B. Li, Y. Zheng, and X. Yu, "Artificial Intelligence-Based Feature Analysis of Ultrasound Images of Liver Fibrosis," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2859987, 9 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2859987, 9 pages
https://doi.org/10.1155/2022/2859987

RETRACTED

Research Article

Artificial Intelligence-Based Feature Analysis of Ultrasound Images of Liver Fibrosis

Youcheng Xie,[1,2] Shun Chen,[1] Dong Jia,[1,2] Bin Li,[1,2] Ying Zheng,[1] and Xiaohui Yu[1]

[1]*Department of Gastroenterology, The 940 Hospital of Joint Logistic Support Force of PLA, Lanzhou 730050, China*
[2]*The First Clinical Medical School, Gansu University of Chinese Medicine, Lanzhou 730000, China*

Correspondence should be addressed to Xiaohui Yu; yuxiaohui528@126.com

Received 26 May 2022; Revised 20 June 2022; Accepted 24 June 2022; Published 30 July 2022

Academic Editor: Muhammad Ahmad

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

Liver fibrosis is a common liver disease that seriously endangers human health. Liver biopsy is the gold standard for diagnosing liver fibrosis, but its clinical use is limited due to its invasive nature. Ultrasound image examination is a widely used liver fibrosis examination method. Clinicians can diagnose the severity of liver fibrosis according to their own experience by observing the roughness of the texture of the ultrasound image, and this method is highly subjective. Under the premise that artificial intelligence technology is widely used in medical image analysis, this paper uses convolutional neural network analysis to extract the characteristics of ultrasound images of liver fibrosis and then classify the degree of liver fibrosis. Using neural network for image classification can avoid the subjectivity of manual classification and improve the accuracy of judging the degree of liver fibrosis, so as to complete the prevention and treatment of liver fibrosis. Therefore, the following work is done in this paper: (1) the research background, research significance, research status at home and abroad, and the impact of the development of medical imaging on the diagnosis of liver fibrosis are introduced; (2) the related technologies of deep learning and deep convolutional network are introduced, and the indicators of liver fibrosis degree assessment are constructed by using ultrasonic image extraction features; (3) using the collected liver fibrosis dataset to conduct model evaluation experiments, four classic CNN models are selected to compare and analyze the recognition rate. The experiments show that the GoogLeNet model has the best classification and recognition effect.

1. Introduction

Liver disease is a major clinical problem, and the ability to accurately diagnose it is critical in the treatment of patients. For example, there are two types of liver diseases: localized lesions and diffuse lesions. In focal liver disease, the lesions are concentrated in a small area of liver tissue, while in diffuse lesions, the lesions are located throughout the liver. The most common kind of liver disease, with major consequences for public health, is called diffuse liver disease [1]. Liver fibrosis is the most common type of diffuse liver disease, which refers to the pathological process of liver connective tissue hyperplasia caused by chronic diseases or other reasons and eventually liver cirrhosis [2]. Liver fibrosis is reversible. If liver fibrosis is detected and treated in time, it can still return to normal. Otherwise, it will be difficult to cure once it develops into liver cirrhosis [3]. The quantitative staging system for liver fibrosis can not only diagnose whether there is liver fibrosis in time and reduce the risk of liver fibrosis patients transforming into cirrhosis but also provide a certain reference for doctors to observe the effect of drug treatment and facilitate doctors to adjust the dose of drugs; it has important clinical significance. According to the Chinese Medical Association's guidelines for the prevention and treatment of viral hepatitis, fibrosis can be divided into five stages: $S0$, $S1$, $S2$, $S3$, and $S4$. Among them, $S0$ represents no liver fibrosis, $S4$ represents liver cirrhosis, and $S1$–$S3$ in the middle represent different degrees of liver tissue lesions [4]. Liver biopsy is the gold standard for clinical diagnosis of the severity of liver fibrosis, but biopsy is a traumatic examination that causes not only pain but also bleeding, and many patients are reluctant to undergo biopsy. Moreover, it may cause misdiagnosis due to sampling error and these shortcomings limit its application in the diagnosis of liver fibrosis [5]. Ultrasound image examination has the characteristics of noninvasiveness, convenience, high repeatability, no radiation, and low price and is one of the widely used liver examination methods [6]. After fibrosis occurs in some tissue parts of the liver, the absorption and reflection properties of ultrasound in these tissue parts will also change. The ultrasound signal reflected by the liver tissue is converted by the ultrasound diagnostic apparatus, and the resulting ultrasound image of the liver will also change. These changes mainly include roughening of the liver parenchyma, thickening of the hepatic vein wall, and sharpening of the liver capsule margin [7]. In the ultrasound images of normal rat livers and liver ultrasound images of cirrhotic rats, it can be seen that the liver parenchyma becomes rough. However, other changes, such as thickening of the hepatic vein wall and sharpening of the capsular margin, are not evident in the picture. This is because in the actual ultrasound examination process, due to the limitations of equipment and the proficiency of the operating doctor, the characteristics of some fibrotic lesions, such as the hepatic vein wall and the edge of the capsule, are difficult to reflect in the ultrasound image. Therefore, clinicians mainly make the diagnosis of fibrosis based on the roughness of liver ultrasound images. Doctors can diagnose the severity of liver fibrosis based on their own experience by observing the thickness of the echoes in the ultrasound images, combined with the shear wave velocity value [8]. This method is highly subjective and can only diagnose hepatic fibrosis qualitatively. Not only can it not subdivide the 5 stages of fibrosis, it is also insensitive to the effect of drug treatment, and it may cause misdiagnosis and delay the treatment of patients. There is a clinical need for an automatic diagnosis system that can extract the characteristics of the echo thickness of the liver from ultrasound images and perform quantitative staging of fibrosis. Moreover, it is required that the quantitative staging of fibrosis is as fine as possible, and it is better to divide 5 stages more accurately, which can not only improve the diagnostic accuracy of fibrosis but also has great significance for the observation of drug treatment effects. Under the premise that artificial intelligence technology is widely used in medical image analysis, this paper uses the convolutional neural network analysis to extract the characteristics of ultrasound images of liver fibrosis and then classify the degree of liver fibrosis. Using neural network for image classification can avoid the subjectivity of manual classification and improve the accuracy of judging the degree of liver fibrosis, so as to complete the prevention and treatment of liver fibrosis. Therefore, the work done in this paper has great practical significance.

2. Related Work

Clinically, the main role of various auxiliary examinations is to perform fibrosis staging for chronic liver disease to determine whether there is liver fibrosis and its severity. After the diagnosis of liver fibrosis or even liver cirrhosis, clinicians will initiate screening and monitoring procedures and formulate a comprehensive treatment plan [9]. These methods can also follow up patients who have been diagnosed with fibrosis, predict whether patients with liver cirrhosis will develop significant portal hypertension, and can also be used to evaluate the efficacy of patients after antiviral and liver fibrosis treatment, providing evidence for the formulation of clinical treatment plans [10]. Ultrasound is currently one of the most commonly used examination techniques for evaluating chronic liver disease [11]. The advantages are that it is noninvasive, objective, reproducible, real-time, simple, and inexpensive and is not affected by the metal machinery placed in the patient, which has been widely accepted. Ultrasound can now provide information on liver morphology and hemodynamics as well as tissue physical properties such as liver stiffness or elastic modulus, making it one of the most important noninvasive methods for evaluating liver fibrosis in recent years with the extensive development of new technologies such as ultrasound elastography and contrast-enhanced ultrasound [12]. There are three main methods for noninvasive assessment of liver fibrosis by ultrasound elasticity: transient elastography (TE), real-time tissue elastography imaging (RTE), and point quantification and two-dimensional shear wave elastography (pSWE and 2DSWE). Acoustic radiation force impulse (ARFI) is applied in both pSWE and 2DSWE [13]. All technicians performing elastic ultrasound operations need to go through a relevant learning curve, master the corresponding basic knowledge, have experience in conventional ultrasound operations, and undergo specialized elastic ultrasound training. During the examination, the patient needs to hold his breath. Holding his breath in a resting state is the guarantee for obtaining the best measurement value. Deep inhalation, Valsalva breathing, and deep exhalation will all change the hepatic venous pressure and affect the elasticity value of liver stiffness [14]. TE is a technology that indirectly measures the liver stiffness value by measuring the propagation velocity of the shear wave excited by the vibration of the patient's skin by a low-frequency mechanical source, thereby reflecting the degree of liver fibrosis. Vibration-controlled transient elastography (VCTE) produced by FibroScan company in the United States is currently the most commonly used assessment method for liver fibrosis [15]. In addition, VCTE has been validated in large-scale clinical trials of various liver diseases, and it is recommended to use VCTE in hepatitis B patients rather than serological indicators for liver fibrosis assessment [16]. RTE is to apply an internal or external dynamic or static pressure to the tissue. Under the action of elasticity, biomechanics, and other factors, the displacement of the tissue before and after the compression will change. By calculating the change of this displacement and using ultrasonic imaging methods and digital image processing techniques, tissue stiffness can be estimated using strain elastography [17]. HITACHI has developed a piece of software for quantitative analysis of tissue diffusion, selects ROI in the elastic image, and divides the color in the region into 256 levels. The horizontal and vertical coordinates represent the number of color levels and

RETRACTED

RETRACTED

frequency distribution of strain values, respectively. Strain histograms were used to evaluate 11 parameters of the liver based on the amount of fibrotic features extracted from image texture features, and then the liver fibrosis index (LFI) was analysed according to these 11 parameters, which is used to reflect the degree of liver fibrosis. RTE can avoid the intrahepatic large vascular structure, and it is its advantage to select the liver parenchyma for imaging and quantitative detection. Moreover, obese patients and patients with ascites can be imaged, which has certain clinical application prospects [18]. SWE includes point quantification pSWE and 2DSWE. Using acoustic radiation pulses to generate transverse shear waves at different depths in the body, the propagation speed of shear waves in different soft tissues is different, so by measuring the propagation speed, the elastic value of tissue hardness can be directly calculated, and the color carrying the information can also be the codes which are superimposed on 2D anatomical images, allowing qualitative and quantitative detection of regions of interest [19]. In addition to the above ultrasound methods for evaluating liver fibrosis, CT perfusion imaging is also a new method to indirectly estimate the degree of liver fibrosis through noninvasive evaluation of liver perfusion imaging [20]. A number of studies have shown that liver CT perfusion imaging can reflect changes in blood perfusion in liver fibrosis and early cirrhosis, and the changed perfusion parameters are related to the stage of liver fibrosis, which is of great value for the early diagnosis and treatment of liver fibrosis [21]. Compared with methods using manual feature extraction, deep learning technology can automatically extract regions of interest and features of target images without manual specification and can significantly improve model performance with the help of massive data [22]. The past 2-3 years have seen a dramatic increase in the number of papers applying deep learning techniques to ultrasound disease classification. At the same time, fusing hand-designed features with features computed by deep learning can further improve the performance of the model [23, 24].

3. Methods

3.1. Deep Learning. Deep learning is a kind of machine learning, which is an advanced technical means. It completes practical tasks by simulating the connection mode of human nerves. It generally includes several connection layers, and each connection layer is mathematically related to each other. This method is usually based on a large amount of data. By continuously nonlinearizing the data, it establishes abstract high-level features to fit a more accurate model, which is also called hierarchical learning and deep network learning. The perceptron is also the neuron of the model. One neuron corresponds to one output, but the real scene is more complicated. People combine multiple perceptrons to have a perceptron model with multiple outputs. However, the single-layer perceptron model can only be applied to the case where the data are linearly separable, and the data processing in the actual task is linearly inseparable, so the single-layer perceptron model greatly limits its application. From this, researchers began to build a multilayer perceptron model. With each additional model layer, it gets increasingly harder to fit the model. The multilayer perceptron model may now be trained more quickly, thanks to the invention of the backpropagation technique. The ANN was born at this time, which consists of an input layer, an output layer, and several hidden layers. The basic structure of the ANN is shown in Figure 1. The input layer receives incoming data and is made up of neurons that pass the received data to other layers. The number of elements in the input layer should be equal to the number of variables in the dataset, that is, the number of attributes in the dataset. The hidden layer is between the input layer and the output layer, which contains a large number of neurons. As the network continues to train, the weights are updated and the prediction ability is stronger. The output layer integrates the predicted features, which is mainly different from the model type. If the model is a classification network, the output will have multiple values according to the task. If the model is a regression network, the output will have one value according to the task. The weight of a neuron refers to the strength of the connection between two neurons, and the weight is usually initialized to a small value, such as a normally distributed value between −1 and 1.

Deep learning is a learning method that uses deep ANN to solve practical tasks. Compared with shallow network models, it has a layered structure and can use multiple layers of nonlinear changes to abstract features. However, at the same time, its training difficulty is also greatly improved. In the process of deep learning network model training, there are often situations such as overfitting or underfitting. The liver fibrosis degree assessment task in this paper adopts a modeling method based on DNN. Through a large number of nonlinear transformations in the modeling process, an abstract feature map can finally be obtained, that is, the features learned by the network; in the training process, various methods are used to avoid overfitting and underfitting.

3.2. Basic Structure of the Convolutional Neural Network. CNN is a common structure in deep learning framework, and its model is the most widely used in deep learning. At the same time, it has a very prominent performance in classification and recognition tasks, so this paper also uses the CNN model combined with the theory of transfer learning to complete the classification and recognition of liver fibrosis. It gives an effect of learning, or the influence of the experience of learning on the completion of other activities. For liver fibrosis images, the input layer of the CNN is passed into the network, the surface damage data features are extracted through one or more layers of convolution pooling and other operations, and the extracted features are then entered into one or more full connection layers. Finally, the damage category label is output through the output layer.

3.2.1. Convolutional Layer. In a CNN structure, there are usually multiple convolutional layers. The low-level convolutional layer usually obtains the overall information of the image, such as edges and lines, but as the number of

RETRACTED

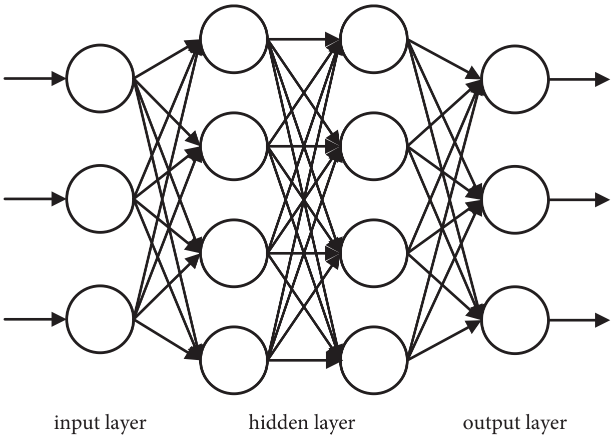


Figure 1: Topological structure of artificial neural network.

convolutional layers increases, more complex and more advanced features can be extracted through continuous iteration. The operation of extracting features from the input image in the convolution layer is called the convolution operation. The relationship between the input and output of the convolution operation is expressed as

$$Y = \frac{1 - B + 2P}{S}. \tag{1}$$

It can be seen from the formula that the output feature map size Y is related to the convolution kernel size B, the convolution kernel single moving step S, the input image size, and whether it is filled.

The above formula is used for a $5 * 5$ image. When using a $3 * 3$ convolution kernel for convolution operation, assuming that the step size of a single movement of the convolution kernel is set to 1, the calculation formula of the feature map size after output convolution is

$$\begin{cases} P = \text{valid} \longrightarrow Y = \dfrac{5 - 3}{1} + 1 = 3, \\ \\ P = \text{same} \longrightarrow Y = \dfrac{5 - 3 + 2}{1} + 1 = 5. \end{cases} \tag{2}$$

Through the analysis, it can be seen that different choices of filling methods can keep the input image size unchanged or reduce the input image size. For CNNs, the convolutional layer usually adopts P = same filling method to maintain the image size. The change of image size is achieved through the pooling layer. It can be seen from the analysis that the stride and the size of the convolution kernel together determine the size of the feature map. The computing speed of the network is determined by the sparse connection and weight sharing between the convolution kernels. Sparse connections are relative to ANNs. In the ANN, each neuron in the current layer is connected to all the neurons in the front and back layers. The drawback is that each link has a weight that must be memorized and stored. Increasing the depth of the network model increases the difficulty of training and converging the network, as well as the amount of memory required to execute the task. In CNN, each neuron simply saves weights in its own structure to address the ANN problem. For example, a convolution kernel of size $3 * 3$ only has to store its own 9 weights; hence, the number of connections between each neuron and the preceding layer is solely linked to its own size. The term "sparse connection" refers to the fact that the convolution kernel has less connections than a fully linked system. For the same convolution kernel, the weights utilized in the convolution operation will be the same in each round of training and the internal weights will only be modified after one iteration has been finished. For a convolution kernel, the weights of each convolution remain the same over each iteration. This is known as weight sharing.

3.2.2. Pooling Operation. In theory, all feature maps extracted by convolution can be used as the input of the network model, but the result of this is that the increase in the amount of parameters leads to an increase in the amount of calculation, so in the convolution operation when done, a pooling operation is required. By setting the step size and window size, the network further reduces the size of the network and the amount of parameters without changing the number of channels and network weights, so as to divide the entire data through pooling and solve several convolutions. When the layer performs feature extraction, a large amount of computation and parameters are generated, thereby reducing the dimension of the image. The addition of pooling reduces the overfitting of the network to a certain extent and increases the image processing speed, thereby improving the performance of the network model. Commonly used pooling methods are max pooling and average pooling. Among the two pooling methods, max pooling is the most commonly used pooling method. In practical applications, in order to further alleviate overfitting, some pooling layers also use overlapping pooling, which means that the pooling windows are allowed to overlap to form overlapping average pooling and overlapping maximum pooling, which effectively reduces the error rate.

3.2.3. Fully Connected Layer. To put it simply, the full-connected layer uses the fundamental matrix operation to get its result: a one-dimensional vector. It is used to categorize and identify characteristics of a subject. Because of its completely linked function, it frequently has to keep a large number of parameters and uses a large amount of storage space. The convolutional layer can be used instead of the fully connected layer to retain the image's spatial structure while reducing memory usage due to the benefits of the convolutional layer.

3.2.4. Activation Function. For complex data such as images and audio, the neural network must have the ability to solve nonlinear tasks in order to extract and learn complex features from these data and introduce activation functions to add nonlinear characteristics to the network. Let the model have the effect of fitting the data well. Commonly used activation functions are sigmoid, Tan*h*, and ReLU. The sigmoid function expression is

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{3}$$

The sigmoid function nonlinearly maps the input continuous real numbers between 0 and 1. When the sigmoid function is infinitely close to 0 or 1, the function will be saturated, that is, the gradient will disappear, resulting in the inability to calculate the weight, the parameters cannot be updated, and the querying new data cause neural network training to fail. Therefore, the sigmoid function is not used in the probability output of the final CNN.

Tan*h* function is also called hyperbolic tangent function, and its function expression is

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{4}$$

The emergence of the hyperbolic tangent function, centered at 0, solves the problem that the output value of the sigmoid function is always greater than 0, but the gradient function still has a saturation phenomenon, that is, the problem of gradient disappearance still exists, when the input value of the function is greater than the fixed value or when it is less than the fixed value, the output value of the function will be fixed at about 1 and −1, which will lead to a small derivative value of the function. At the same time, when the network parameters are updated, according to the chain derivation rule of the derivative, with the network layer increases, the gradient vanishing phenomenon still exists.

The functional expression of the ReLU function is

$$f(x) = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{5}$$

When the input of the ReLU function is negative, the corresponding function value of the function is 0. When the input value is not negative, the function is a zero-crossing point and a positive proportional function with a slope of 1. Because the ReLU function has a faster convergence speed and with the input when the value increases, the gradient value remains unchanged and the problem of gradient saturation will not occur. Therefore, in most neural networks, ReLU is usually selected as the activation function. However, the problem with the ReLU function is that when the input value is negative, the corresponding function value is always 0. At this time, the gradient saturation will occur and the neurons in the network cannot be activated. However, this problem can be circumvented by setting the learning rate parameter.

3.3. Training of the Network Model

3.3.1. Dropout Strategy. In the classification and recognition network model of liver fibrosis degree, the phenomenon of overfitting is manifested in that the deviation of accuracy between the training set and the test set of liver fibrosis images is too large, so that the network model can hardly be used. The proposal of network sparsification effectively prevents the overfitting phenomenon in the training process. During training, some input nodes or output nodes are randomly disabled. Dropout can be regarded as adding an auxiliary layer with uncertainty to the network, which can reduce the training parameters of the network and improve the generalization ability of the model.

3.3.2. Cross-Entropy Damage Function. The output layer is used for image classification and recognition, and the input images can be classified into corresponding categories in the form of probability. The loss function, also known as the cost function, is used to evaluate the error between the true value of the model and the output value. As the loss value decreases, the stability of the model gradually increases. In this experiment of classifying and identifying the degree of liver fibrosis, the softmax cross-entropy loss function was used to calculate the error. The loss for each batch is

$$J(x, \hat{y}) = -\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n}\left[y_{ji} log \hat{y}_{ji} + \left(1 - y_{ji}\right)\log\left(1 - \hat{y}_{ji}\right)\right], \tag{6}$$

where m is the total number of samples and n is the number of label categories. If only one iteration of the network model is performed, the loss values obtained by all batches are connected to obtain the loss curve in the training process.

3.3.3. Parameter Optimization Method. The process of network training is the process of updating the weights of the network iteratively, which is also the process of parameter optimization. From the previous section, we know that the network performance is evaluated by the loss function. The better the network performance, the lower the loss value, and the reduction of the loss value is the result of the continuous optimization of the network. Commonly used network optimization methods include mini-batch gradient descent method, momentum technology, and adaptive time estimation method. The mini-batch gradient method solves the frequent update and fluctuation of traditional gradient descent algorithm, resulting in large fluctuations in loss function and network instability and convergence. However, the learning rate of mini-batch gradient descent is fixed, and the same learning rate makes it difficult to converge when the training sample data are sparse. The same momentum technique also faces the problem of nonconvergence of the model caused by the fixed learning rate. The adaptive time estimation method can adjust the learning rate without human participation in the network training process. This method of automatic learning rate adjustment can not only shorten the iteration time required for network training but also improve the learning effect. It can prevent the learning rate from disappearing and reduce the fluctuation range of the loss function. Obviously, compared with other algorithms, Adam is more suitable as a parameter optimization method for liver fibrosis classification and identification experiments.

3.3.4. Batch Normalization Algorithm. As the depth of the network model built by deep learning increases, the training

RETRACTED

RETRACTED

process of the model becomes more and more complicated and the convergence rate of the network model decreases. Therefore, the batch normalization (BN) algorithm is introduced. For a complex network model, some of the neurons in the hidden layer can also be regarded as input, so the activation value of each hidden layer neuron in the deep neural network can be preprocessed by whitening.

In order to select the optimal network model for experiments, this paper selects four classic models for experimental comparison, including the AlexNet model, the VGGNet-16 model, the VGGNet-19 model, and the GoogleNet model.

3.4. Feature Extraction Based on RTE Images. We extracted 11 image features to quantify RTE images, directly obtaining each elastography parameter. The ranges of 11 parameters reflecting liver fibrosis in real-time elastography are shown in Table 1. In addition, we also considered the patient's age and gender as characteristics. Therefore, for each sample, we collected a total of 13 features.

As can be seen from Table 1, the value range of each parameter is different and cannot be directly used to train the neural network model. Therefore, we first normalized the data. For continuous-valued scalars, we normalize them so that each parameter value is in the range [0, 1]. For example, the age of the patient is usually between 0 and 100 years. If the patient is 55 years old, the standard value should be 0.55; the standardization formula is

$$\overline{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{7}$$

where x_i refers to the value of the input variable, $x_{\min}$ is the minimum value within the data range, and $x_{\max}$ is the maximum value within the data range. In addition, for binary variables, 0 and 1 are used to represent features, for example, gender is 1 for female and 0 for male.

4. Experiment and Analysis

4.1. Data Sources and Evaluation Indicators. The research subjects were from a multicenter real-time elastography project of hepatitis B fibrosis. A total of 780 patients with chronic hepatitis *B* and cirrhosis participated in this study. The study protocol was approved by the relevant institutional independent ethics committee, the purpose of the study was explained to the patients, and written informed consent was obtained. The inclusion criteria of patients were as follows: (1) they had received histopathological examination of the liver; (2) they had a history of hepatitis *B* or HBsAg positive for more than 6 months and were still HBsAg positive; (3) they had undergone an ultrasound examination. The exclusion criteria were as follows: other types of hepatitis other than viral hepatitis B, metabolic diseases, fatty liver, and drug-induced and alcoholic liver injury. Ultimately, a total of 640 samples were selected for analysis. The study subjects included 480 male and 260 female patients, ranging in age from 22 to 70 years.

Table 1: Liver fibrosis ultrasound real-time tissue elastography parameter index.

Attributes	Attribute value range	Label
Strain mean	53.28–140.35	*X*1
Standard deviation	28.45–74.75	*X*2
Blue field percentage	0.72–61.87	*X*3
Complexity	9.16–58.89	*X*4
Consistency	0–0.05	*X*5
Contrast	64.20–348.95	*X*6
Correlation	0.95–0.99	*X*7
Equality	3.18–4.05	*X*8
Clutter	0.06–0.25	*X*9
Skewness	0.15–1.22	*X*10
Kurtosis	1.96–4.13	*X*11

According to the classification and identification of the degree of liver fibrosis in this subject, the accuracy of the evaluation index is selected for evaluation, and the formula is

$$\text{Acc} = \frac{\text{number of correctly predicted samples}}{\text{total number of samples}}. \tag{8}$$

4.2. Best Fit Model Selection Experiment. In order to obtain the most suitable network model, four network model structure schemes were selected in this experiment, namely, AlexNet, VGG-16, VGG-19, and GoogLeNet. By comparing and analyzing the recognition accuracy values of different network models on the test set, the optimal network model structure is selected. Table 2 shows the recognition results of different network models. From the table, we can see that the recognition accuracy of the GoogLeNet model is better than other models, so the optimal network model selected in this paper is GoogLeNet.

4.3. Optimal Model Optimization Experiment. Through the experimental analysis of the optimal model selection, it can be seen that GoogLeNet is the optimal model, but different iterations, batches, and learning rates can lead to changes in the output of the model. In order to verify how many training batches, how many learning rates, and how many network iterations, the GoogLeNet recognition effect is the best and the performance of the network model is verified by controlling variables, so as to select the optimal liver fibrosis degree recognition model.

4.3.1. Batch Size. In order to verify the influence of different parameters on the classification and identification of wind turbine blade surface damage and then select the optimal GoogLeNet network model parameters, we first conduct batch size experiments. Ideally, the batch size of the network model should be the iterative experiment using the entire training set. However, if the batch size is too large, it will not only affect the network performance, but also increase the computer burden. In this experiment, 16, 32, and 64 were selected as the batch size of the experiment, and the test results were analyzed. Figure 2 shows the model test results for different batch sizes.

RETRACTED

Table 2: Damage identification results of different network models.

Network structure	Acc
AlexNet	0.3895
VGG-16	0.6728
VGG-19	0.8676
GoogLeNet	0.9529

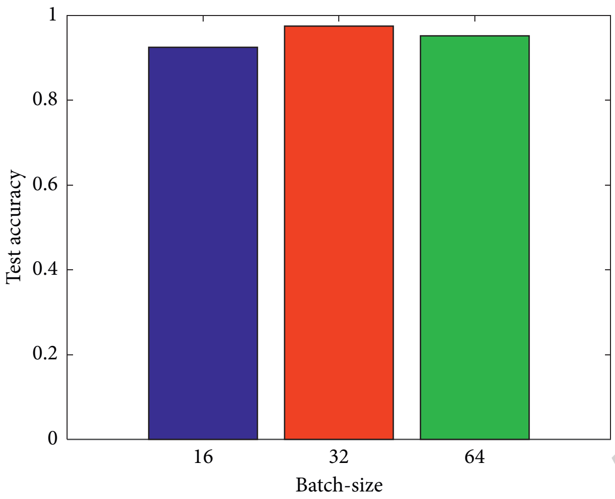


Figure 2: Comparison of model accuracy under different batch sizes.

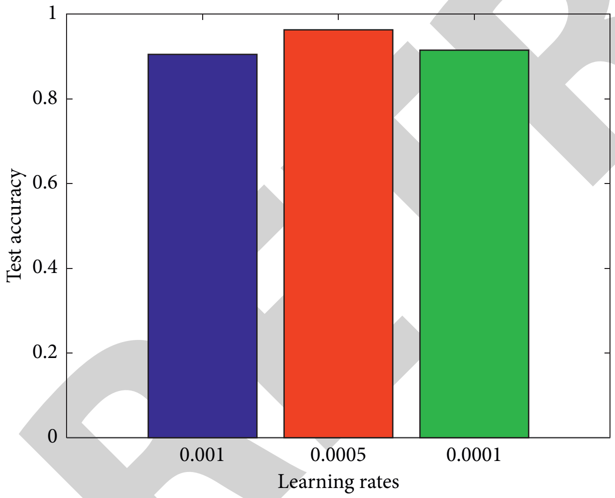


Figure 3: Comparison of model accuracy under different learning rates.

Through the analysis of Figure 2, it can be seen that the introduction of large batches will reduce the performance of the network and the accuracy of 32 batches is higher than that of other batches. Therefore, batch size 32 is selected as the parameter in this experiment.

4.3.2. Learning Rate Setting. In order to explore the influence of different learning rates on the model recognition results, three different learning rates were set, respectively, and the accuracy of the test set under different learning rates was compared, as shown in Figure 3.

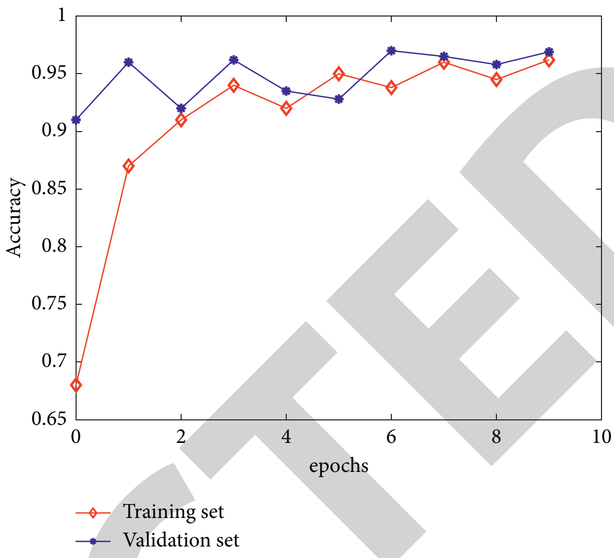


Figure 4: Training set and validation set accuracy curve.

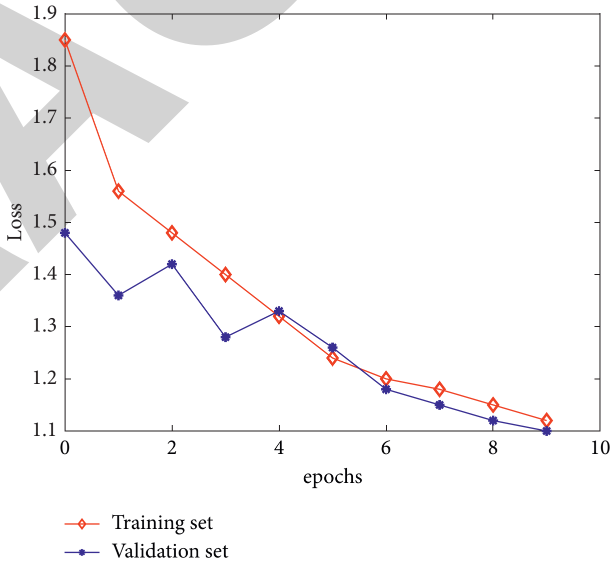


Figure 5: Training set and validation set loss curve.

Similarly, through the analysis of Figure 3, it is found that the recognition result of 0.001 learning rate is the lowest, which cannot meet the experimental requirements. The accuracy rate of 0.0005 is the highest, so the learning rate of 0.0005 is finally selected as the parameter of the model.

4.3.3. The Number of Iterations Experiment. In order to further optimize the model, we collected the accuracy and loss during the running of the model, that is, the accuracy and loss of the training set and validation set, as shown in Figures 4 and 5.

From Figure 4, it can be seen that, with the increase of the number of iterations, the accuracy of the network is constantly improving and the loss is also decreasing, but the curve does not change significantly after 8 iterations. 8, 9,

RETRACTED

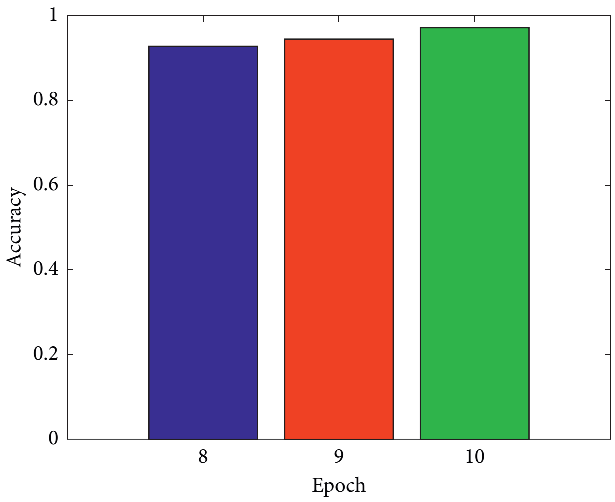


Figure 6: The accuracy of the model under different iterations.

and 10 were selected as the number of iterations, and the most suitable parameters for the model were selected by analyzing the accuracy of the test set under different iterations. Figure 6 shows the model test results for three iterations. It can be clearly seen that when the iteration is 10, the recognition effect of the network model is the best.

5. Conclusion

Liver fibrosis is a pathological process of abnormal connective tissue proliferation in the liver, usually caused by various pathogenic factors. Liver fibrosis will occur in the process of liver repair and healing of any liver injury, but if the injury cannot be removed in time, the development of liver fibrosis to a later severe stage will lead to liver cirrhosis, which in turn leads to serious complications such as liver cancer. At present, the diagnosis of liver diseases mainly includes liver biopsy, serum examination, and imaging examination. MRI has advantages in observing and evaluating superficial organs, cartilage, and other tissues and has multiplanar imaging capabilities, but it is expensive and cannot be dynamically checked in real time; compared with CT, it has insufficient spatial resolution and radiation damage, and MRI cannot perform real-time dynamic examination. Ultrasound imaging technology has the advantages of nondestructive, real-time, convenient, and no ionizing radiation and is suitable for clinical screening and follow-up, especially for the diagnosis of liver fibrosis and early cirrhosis, and its advantages are obvious. For the analysis of medical images, the main method is manual interpretation, which depends on the subjective experience and knowledge level of doctors, which is prone to subjective misjudgment and low efficiency. In this context, this paper extracts the characteristic parameters of liver fibrosis ultrasound images and then uses neural network to evaluate the degree of liver fibrosis and completes the following work: (1) it introduces the research background, research significance, research status at home and abroad, and the influence of the development of medical imaging on the diagnosis of liver fibrosis; (2) the related technologies of deep learning and deep convolutional network are introduced, and the indicators of liver fibrosis degree assessment are constructed by using ultrasonic image extraction features; (3) using the collected liver fibrosis dataset to conduct model evaluation experiments, four classic CNN models are selected to compare and analyze the recognition rate. The experiments show that the GoogLeNet model has the best classification and recognition effect.

Data Availability

The datasets used 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] R. G. Barr, G. Ferraioli, M. L. Palmeri et al., "Elastography assessment of liver fibrosis: society of radiologists in ultrasound consensus conference statement," *Radiology*, vol. 276, no. 3, pp. 845–861, 2015.

[2] M. Kanamoto, M. Shimada, T. Ikegami et al., "Real time elastography for noninvasive diagnosis of liver fibrosis," *Journal of Hepato-Biliary-Pancreatic Surgery*, vol. 16, no. 4, pp. 463–467, 2009.

[3] P. Bedossa, D. Dargère, and V. Paradis, "Sampling variability of liver fibrosis in chronic hepatitis C," *Hepatology*, vol. 38, no. 6, pp. 1449–1457, 2003.

[4] W. C. Yeh, S. W. Huang, and P. C. Li, "Liver fibrosis grade classification with B-mode ultrasound," *Ultrasound in Medicine and Biology*, vol. 29, no. 9, pp. 1229–1235, 2003.

[5] S. M. Martinez, G. Crespo, M. Navasa, and X. Forns, "Noninvasive assessment of liver fibrosis," *Hepatology*, vol. 53, no. 1, pp. 325–335, 2011.

[6] D. Meng, L. Zhang, G. Cao, W. Cao, G. Zhang, and B. Hu, "Liver fibrosis classification based on transfer learning and FCNet for ultrasound images," *IEEE Access*, vol. 5, pp. 5804–5810, 2017.

[7] C. Aubé, "Imaging modalities for the diagnosis of hepatic fibrosis and cirrhosis," *Clinics and research in hepatology and gastroenterology*, vol. 39, no. 1, pp. 38–44, 2015.

[8] L. Y. Xue, Z. Y. Jiang, T. T. Fu et al., "Transfer learning radiomics based on multimodal ultrasound imaging for staging liver fibrosis," *European Radiology*, vol. 30, no. 5, pp. 2973–2983, 2020.

[9] C. P. Selinger and R. W. Leong, "Noninvasive liver fibrosis assessment: why does the APRI not work for hepatitis B?" *Hepatitis Monthly*, vol. 11, no. 7, pp. 556-557, 2011.

[10] R. A. Standish, E. Cholongitas, A. Dhillon, A. K. Burroughs, and A. P. Dhillon, "An appraisal of the histopathological assessment of liver fibrosis," *Gut*, vol. 55, no. 4, pp. 569–578, 2006.

[11] G. Ferraioli, V. W. S. Wong, L. Castera et al., "Liver ultrasound elastography: an update to the world federation for ultrasound in medicine and biology guidelines and recommendations," *Ultrasound in Medicine and Biology*, vol. 44, no. 12, pp. 2419–2440, 2018.

[12] C. F. Dietrich, J. Bamber, A. Berzigotti et al., "EFSUMB guidelines and recommendations on the clinical use of liver

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9860329, 1 page
https://doi.org/10.1155/2023/9860329

Retraction

Retracted: Comparative Analysis of Aesthetic Emotion of Dance Movement: A Deep Learning Based Approach

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Huang, "Comparative Analysis of Aesthetic Emotion of Dance Movement: A Deep Learning Based Approach," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5135495, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 5135495, 10 pages
https://doi.org/10.1155/2022/5135495

RETRACTED

Research Article

Comparative Analysis of Aesthetic Emotion of Dance Movement: A Deep Learning Based Approach

Ya Huang

College of Music, Hunan International Economics University, Changsha, Hunan 024321, China

Correspondence should be addressed to Ya Huang; v11314020@stu.ahu.edu.cn

Received 23 May 2022; Revised 20 June 2022; Accepted 24 June 2022; Published 21 July 2022

Academic Editor: Muhammad Ahmad

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

Dance is a unique art with the human body movement as the main means, but dance is not limited to the human body movement itself. Like any art, dance is the product of human social behavior and a romantic behavior of human thoughts and emotions in the virtual world. Dances with different characteristics will also reflect different aesthetics, different cultural psychology, different living styles, and emotional trajectories of different times and different nationalities. People rely on the image of dance artists to develop and inherit the profound ideological connotation and philosophy of life. Viewers may form their own diversified and unique aesthetic characteristics. In the new era, in order to better promote the development, communication, and dissemination of dance art, it is very necessary to analyze and explore the connotation and aesthetic characteristics of dance art. Only through specific movements can the value and ideological connotation of works be expressed. Therefore, this paper comparatively analyzes dance movement aesthetic emotion based on deep learning. Experimentations are performed to systematically analyze the models from various perspectives. Findings of the evaluation show that CAP and CNN are effective models that can successfully extract high-level emotional features. The method proposes and effectively selects the best models among the five standard models based on key features and is, therefore, suitable in predicting the dancer's emotion and for the analysis of the dance movement in the future.

1. Introduction

Among all the art categories, dance is the oldest art [1]. Beauty is the first element of dance art, whereas dance art is mainly expressed through human body movements [2]. In primitive society, the ancient ancestors learned to dance before they had no words and perfect language. They expressed their feelings through actions, gestures, and facial expressions. It is evident that dance was the main way of life and entertainment for our ancestors [1]. Therefore, art historians call dance the "mother of art." Dance is an art form having strong comprehensiveness. The elements of dance need to be integrated to convey the artistic beauty and information contained in it to the audience. There are thousands of varieties of dances in the world, which are hard to count [3]. However, in terms of its social function, it can be divided into two categories: life dance and art dance. Life dance refers to the dance that has been popular among people for a long time and is closely related to people's daily life. Artistic dance refers to the artistic practice completed by professional or amateur dancers through artistic creation and stage performance. According to different themes, genres, and capacities, artistic dance can be roughly divided into two categories: dance drama and dance [4]. Each category is different according to the style, content, and style, as well as some specific varieties. Although there are various kinds of dance art; all contain temporal, dynamic, and static combinations of the rhythmic human movement. Dance covers direct, strong, and generalized lyricism to synthesize music, art, and other artistic factors [5]. Dance action is not only the basic material of the dance language but also the material premise for the visibility of the dance language. A dance work, in terms of the form, is the repetition, development, change, connection, and cooperation of various movements. Among the elements of dance, action is the most important. Without human body action, there is no

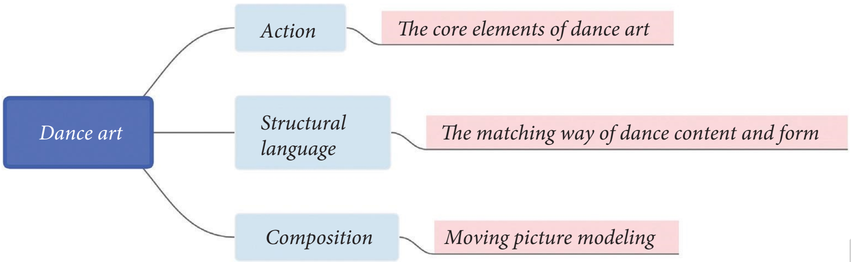

Dance art
Action
The core elements of dance art
Structural language
The matching way of dance content and form
Composition
Moving picture modeling

Figure 1: The main linguistic elements are covered by the art of dance.

dance art. Because the expression of the dancer's emotion, the display of the artistic conception of the work, and the shaping of the dance image always run through the dance movement [6]. The artistic language of all kinds of dance is composed of some relatively stable factors. The main linguistic elements covered by the art of dance are shown in Figure 1.

Dance action comes from life and nature, but it is very different from live-action and natural movement. Dance is an artistic human action as the movement of dance and the creative thinking of the artists esthetically represent a certain purpose of the performance [7]. Dance is a programmed and visualized language formed through artistic processing, refinement, transformation, and evolution. A key element of dance art is the infectious expression to convey emotion and meaning. Hence, not all human body movements can be called dance movements. Dancing also needs music and clothing to achieve better results [8]. The meaning of dance action can be divided into narrow sense and broad sense. In a narrow sense, dance action refers to the dynamic action in the process of movement, including single action and process action. According to the function and role, dance movements can be divided into expressive movements and illustrative movements. Life is full of rhythm, rather, rhythm comes from life [9]. The rhythm of life varies with different regions and other conditions like manifestations as fast, slow, and comfortable. This difference in the rhythm of life can be expressed through the rhythm of dance. Dance is a kind of expressive art, which is long in lyricism and clumsy in the narrative [10]. Dance in the expression of the character's thoughts and feelings. With dance, the coordinated, and rhythmic movements of body parts, feelings, and scenarios are expressed in a seemingly way. Dance can more directly reveal a variety of emotions and a complex psychological state.

In social life, people's various expressions and movements have become dynamic dance expressions with different styles after artistic processing, refining, and beautification [11]. It is from observation and experience that dance creators master the characteristics of natural expressions of various characters in life, and form dance expressions with strong appeal through artistic imagination and artistic creation [12]. This paper proposes a dance movement aesthetic emotion analysis method based on deep learning and verifies the method through a number of experiments. The experimental results show that the model can well predict the emotion of the dance movement, which lays a foundation for the emotion analysis of the dance movement in the future.

2. Related Work

Dance is the evocative art carrying sensual and artistic expression [13]. As dance is a diversified subject, it exists in every culture and covering key aspects of culture, it is pertinent to discuss elements, forms, and aesthetics of dance, hence the following subsections. Machine learning [14] is the emerging technology used in various domains like health [15, 16], social work [17], and decision making [18]. Details about dance contents, structure, and forms are elaborated in the following subsections.

2.1. The Matching Way of Dance Content and Form. Dance work is achieved if the formal elements such as dance movements and skills exist in isolation. Only through the ingenious conception and a reasonable structure of dance, artists convey feelings and ideas. A successful dance creation process necessitates that dancers ought to have rich life accumulation and subjective perception. Dancers should summarize, refine, process, and transform the social life materials they have mastered to form a theme that is suitable for the dance performance [19]. On this basis, we should also endow the theme with corresponding dance forms. This requires the work to be structured according to the performance characteristics of dance art, so as to turn it into a dance form that can be easily perceived by the audience [20]. The structural process of dance works is the process of finding the appropriate dance form from content according to the theme to be expressed in works, so as to make content and the form of dance reach a perfect fit [21]. Therefore, once the theme of the dance is determined, the structure is an important step in creation. Different dance works with different themes have different ways of structures. Dance structures are based on the experience of creating some excellent dance and dance theater works. On the basis of composition, dance has the following main types, as shown in Figure 2.

The chronological structure of the dance is based on the natural sequence of events and the emotional development of the characters to arrange actions and dance scenes [22]. Its main artistic feature is to organize the activities of the characters in space in the order of the trend of the time, emphasizing the continuous development of the characters'

RETRACTED

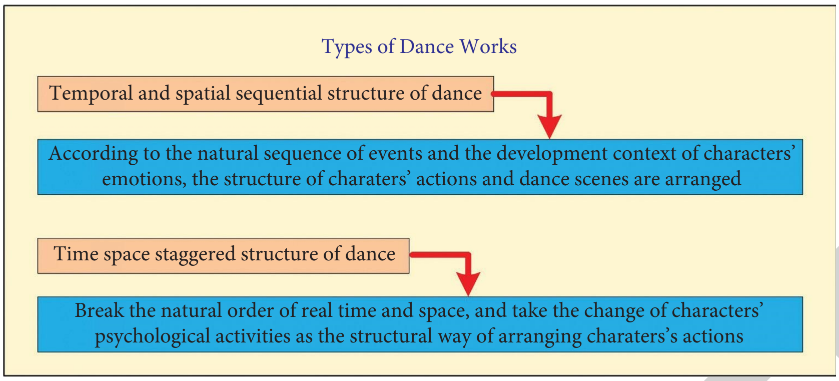

Types of Dance Works
Temporal and spatial sequential structure of dance
According to the natural sequence of events and the development context of characters' emotions, the structure of charaters' actions and dance scenes are arranged
Time space staggered structure of dance
Break the natural order of real time and space, and take the change of characters' psychological activities as the structural way of arranging charaters's actions

Figure 2: Types of dance works.

thoughts and feelings, or plot events. Dance works are in the order of "beginning, development, climax, and ending," with clear levels and clear division of scenes [23]. Because the structure of this kind of dance is clear and in line with the logic of the development of real life, the content of the dance is easy to be understood and accepted by the audience. The structure has always been adopted by most dance choreographers and directors. It is why some people call it a traditional structure [24]. The interlaced structure of dance is a way to break the natural sequence of real-time and space and use the change of the characters' psychological activities as the thread of arranging the characters' actions and unfolding the plot events. This kind of structure combines the scenes of different times and spaces according to certain artistic ideas, so as to show a broader and rich life content in the limited stage, time, and space. This will also help to describe and show the complex and profound inner spiritual world of the characters. This kind of structure has strong artistic expression and can mobilize the audience's artistic imagination and aesthetic creativity in dance appreciation [25]. Besides, such structure aids in expanding the scope of the dance theme performance, and in-depth depiction of the character's inner world.

Generally speaking, the structure of dance should be restricted by the theme and content. Works with different themes and contents have different structural ways. Lyrical dance often expresses and describes different emotions of dancers by comparing and changing the rhythm of dance movements [26]. Narrative dance or dramatic dance drama needs to arrange the action of the characters and the formal structure of the dance scene according to the development process of the plot and events of dance works. Its structure is generally composed of beginning, development, climax, and end [27]. The ingenious and clumsy structure has a great relationship with whether dance works can perfectly display content and have artistic charm. Therefore, in order to pursue the overall beauty of dance works, excellent dance choreographers and directors have always attached great importance to the structure of the dance.

2.2. Moving Picture Modeling. Dance is a moving plastic art, and the dance form is a visual and an intuitive image in a flowing state. In a dance performance, various formations and pictures formed by the dancers' dance posture and moving lines are the dynamic structure of dance in time and space. The composition of dance is a major part of dance modeling and an important factor in the composition of dance works [28]. Whether it is a solo dance, group dance, or dance drama, actors should move in a certain direction and route in the stage space. Dance composition refers to the movement line (constantly changing the dance route) and pictures modeling the dancer in the stage space in dance works, that is, dance composition is composed of the movement line and the picture of the dancer [29]. Composition is both an expression of meaning and a process of form creation. Dance composition not only makes the stage picture varied, but also can express different thematic ideas and shape different dance images. Moreover, proper dance composition is required to create different momentum, atmosphere, and formal beauty of dance. Stage space motion line, also known as "stage scheduling", refers to the trajectory of dancers moving in stage space. Different types of stage space movement lines can give the audience different feelings [30]. Choreographers usually use different lines to express different contents based on this. The folk dances of all ethnic groups are based on the dance venue and the ideas contained in dance composition, which is mainly reflected in dance formation. Like the composition of other arts, the composition of dance art has laws to follow and needs to follow specific ideas and principles [31]. The structure belongs to the formal category of art and is generally restricted by the content of art. Due to the different contents of dance works, the stage scheduling is also different. In the treasure house of the traditional dance art, there are many examples of exquisite dance compositions, but they cannot be used at the same time. The creation of dance works must be creatively applied according to the content of specific works. The dance movement and the feeling structure is shown in Figure 3.

Only through emotional investment can dance have life and soul and better move the audience. Dance can show the most realistic feelings in people's hearts. Dance is the purest art. A good dancer can move the audience with the most delicate emotion, thus reflecting the deep connotation of the whole dance. Dance is an art form that expresses emotion through the continuous movement of movements [32]. The emotion of dance has the functions of promotion,

RETRACTED

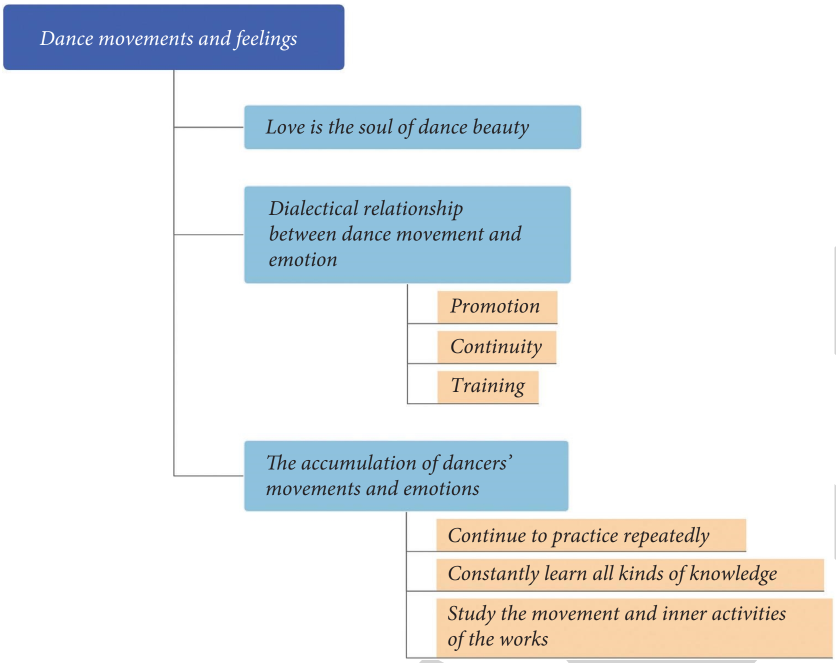

Dance movements and feelings
Love is the soul of dance beauty
Dialectical relationship between dance movement and emotion
Promotion
Continuity
Training
The accumulation of dancers' movements and emotions
Continue to practice repeatedly
Constantly learn all kinds of knowledge
Study the movement and inner activities of the works

Figure 3: The dance movement and the feeling structure.

RETRACTED

continuity, and training at the same time. It can stimulate the potential energy of actors. In addition to acting training, emotion can best improve an actor's artistic cultivation and personal quality. Through the movement training, the emotion is injected into it, making the dance movement full of life. The feeling cannot be embodied and expressed without action, and action loses the ability of expression and the value of art without emotion [33]. Movement is the main external factor of dance, which can be seen directly by the naked eye, while emotion is hidden in the process of the dance movement that is slowly expressed through a delicate performance. Emotion in dance penetrates into the hearts of dancers and audiences. Emotion in dance is the amplification of emotion in real life, and action is also the artistic processing of action in real life [34]. Through this emotional amplification and action processing, the art of dance is rich and colorful, and full of life.

2.3. Aesthetics of Dance Art. In a sense, it can be said that the main way to evaluate whether a dance work can achieve success is to see whether the action displayed is artistically appealing or not. Therefore, dance movement is in the most important position of the constituent elements of dance art [35]. The expression of dance emotion, the shaping of characters, and the display of dance artistic conception in dance art should be realized through action. The movements are selected and refined by the choreographer in daily life and/or through learning. The movements are endowed with distinctive image characteristics and emotional factors. The dance movements have a certain sense of beauty. Dance art should pay attention to dynamic beauty, which is determined by the movement requirements of dance art [36]. Although this will limit the dancer's rhythm to some extent, only in this way we can fully express the theme content and internal emotion of dance. Rich aesthetic dance movements can enrich the aesthetic characteristics of dance. With the traction of movements, it can more naturally open the dance plot and then make it more attractive to people's attention.

Lyricism is one of the main aesthetic and artistic features of dance art. Emotional expression is the basis of art. Any work separated from emotional expression cannot form art. Dance is an art form that expresses people's thoughts and feelings and reflects the form of production and life by human body movements that have been organized, refined, and artistically processed. Dance art contains people's strong emotional factors [37]. From a certain point of view, among the major art forms, such as music, poetry, and dance, the effect of expressing emotion through dance is the most ideal. Because the emotions that cannot be expressed by language or words can be expressed by dance, which can not only express people's deep feelings but also give people enough imagination space. Feelings are rich, intuitive, and uncertain. They can be intuitive and perceived because dance is the perceptual form and externalization of people's mental activities, which is closely connected with people's expressions and movements. In addition, feelings have mutual integration and interoperability, which is also the psychological basis for art appreciation and art creation. It can build a bridge of aesthetic feelings between dance creator, performer, and audience, between movement and feelings [38]. Expression and action are inseparable from people's emotional expression. Therefore, human action has unique advantages in expressing thoughts and emotions and can express people's emotional experiences and mental states incisively and vividly. The beauty of rhythm refers to the beauty presented by the strength and speed of dance action. With the beauty of rhythm, the aesthetic characteristics of

RETRACTED

dance art are exhibited. In short, the beauty of dance is the premise of the beauty of rhythm. Only by perfectly integrating and coordinating the leaping, relaxing, and rotating rhythms of dance movements can dance art render the atmosphere and express and vent emotions [39]. Dance art can form its rhythmic beauty only through the changes of expression and movement and the rhythm with ups and downs.

The appreciation and transmission of beauty is the greatest success of the artwork of dance, which sprouts naturally through the living image and the exquisite movement of nature. At the same time, dance can also drive the plot forward, gradually expressing the rich emotional factors and natural characteristics contained therein in a comprehensive and full manner. It can convey the experience and feeling of beauty to the audience and give them a wide space to imagine the beauty, prompting the viewer to produce an emotional resonance and deep and unique physical and mental feelings from deep inside.

3. Methodology

Like musing, dance has the appealing structure formed from tempo, meter, and rhythm. Aesthetic emotion of the dance movement is an important aspect of dance which needs to be properly analyzed. This research work proposes an effective method for analyzing the models used for the analysis of emotion in the dance movement.

3.1. Relevant Technical and Theoretical Basis. The language model is a statistical model to calculate the probability of any word order in the text. It is a basic subject of NLP. The neural network language model has only one hidden layer. The maximum penalty likelihood estimation of the random gradient rise method is used to train the network parameters, and softmax is used to calculate the probability of word order. Traditional encoding usually refers to distributed feature representations obtained by learning through neural network language models that can represent complex contextual information in a large-scale corpus in a non-sparse manner. In order to effectively focus on specific local information and tap deeper into feature information, an attention mechanism is proposed. The attention mechanism models the global dependence of input and output so that the model can learn feature information in different representation subspaces, so as to generate more semanticist representations. The calculation formula of the attention mechanism is as follows:

$$\text{Ateention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathbf{V}. \tag{1}$$

In terms of parallelism, the multi-head attention model, like CNN, does not rely on the previous calculation and can be well parallelized, which is better than RNN. The sequence-to-sequence model is a supervised learning algorithm in which the input is a sequence of tokens, and the generated output is another sequence of tokens. However, when encoding, all semantic information is encoded into a fixed and an unified semantic feature vector and then decoded, which will cause a great loss of semantic information, especially word order information, which greatly reduces the decoding performance of the model. Therefore, researchers use the attention mechanism to break the limitation of the traditional encoder-decoder structure that encoding and decoding only depend on the vector of a fixed length and size. The sequence-to-sequence structure with the attention mechanism is shown in Figure 4. The codec network with the attention mechanism weights and sums these inputs by saving the intermediate process output of the encoder and training the attention model, and associates the output sequence with the attention model when decoding the output. The introduction of the attention mechanism enables the model to be associated with the required context information during encoding and decoding, and it strengthens this association in the process of iterative learning.

Parsimonious Bayes is widely used in sentiment classification for assigning the category with maximum probability to a given review for classifying documents. According to the Bayes' rule, it is expressed by the following formula:

$$P\left(c_j \mid d\right) = \frac{P\left(c_j\right)P\left(d \mid c_j\right)}{P(d)}, \tag{2}$$

where c_j represents class and d data. The probability equation is smoothed so that it ends up with the following calculation:

$$\widehat{P}\left(f \mid_i c_j\right) = \frac{1 + n_{ij}}{m + \sum_{k=1}^{m} n_{kj}}. \tag{3}$$

Maximum entropy classification follows the principle of maximum entropy, which means that the probability distribution that best represents the current state of knowledge is the most entropic, given precisely stated a priori data.

$$\begin{aligned} P\left(c_j \mid d\right) &= \frac{1}{\pi(d)} \exp\left(\sum_{i=1}^{m} \lambda_{i,c_j} F_{i,c_j}\left(d, c_j\right)\right), \\ F_{i,c_j}(d, x) &= \begin{cases} 1, & \text{if } n_i > 0 \text{ and } x = c_j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4}$$

After several iterations, the parameters are adjusted to maximize the entropy of the training data distribution. Model fusion can be realized on the basis that different models usually do not produce the same test error on a test set. The mathematical expression for the expectation of the squared error after the model fusion is as follows:

$$\mathbb{E}\left[\left(\frac{1}{k}\sum_{i} \varepsilon_i\right)^2\right] = \frac{1}{k^2}\mathbb{E}\left[\sum_{i}\left(\varepsilon_1^2 + \sum_{j \neq i} \varepsilon_i \varepsilon_j\right)\right]. \tag{5}$$

When the error is completely correlated, that is, $C = V$, the mean square error is reduced to V, so the model average is not helpful. When the error is completely uncorrelated, that is, $C = 0$, the expectation of the square error of the fusion model is only W. This means that the expectation of integration square error decreases linearly with the increase of the integration scale. In other words, on average, the fusion performs at least,

RETRACTED

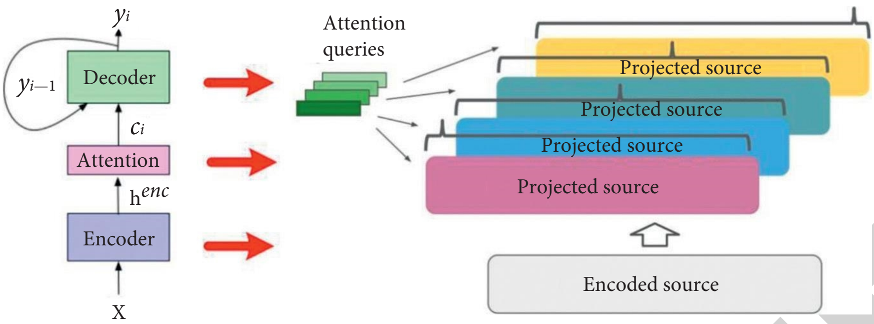


Figure 4: Schematic diagram of the attention-based sequence-to-sequence model.

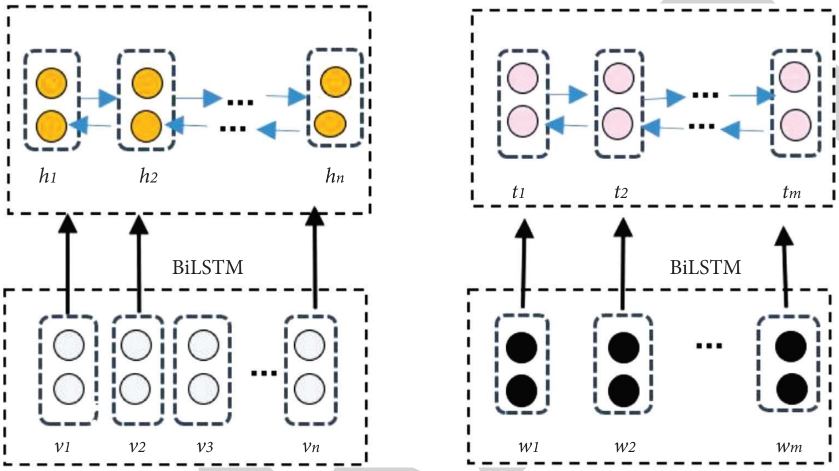


Figure 5: The network structure of the coding layer of the TOAC model.

as well as any of its members, and if the error of members is independent, the fusion model will perform better than its constituent members.

Table 1: The experimental parameters.

Parameter name	Value
Number of iterations	3
K-max pooling	7
Capsule amount	8
Network dimension	12
Batch size	32
Hidden layer dimension	48

3.2. Improving the Sentiment Analysis of Attentional Mechanisms. In this paper, we propose to integrate the attention mechanism with the capsule network to solve the dance emotion classification problem by proposing a network model based on an improved attention mechanism for a target-oriented emotion analysis task. The various components of the model include a text encoding layer, an attention interaction layer, and a capsule classification layer. The role of the encoding layer is to convert content and targets into model inputs. Using the bidirectional encoder representations from transformers (BERT) as an example, the input is first tagged as a fragment. The target is modeled to obtain vector representation and finally fed into a two-way long short term memory (LSTM) network. The network structure of the coding layer of the TOAC model is shown schematically in Figure 5. The coding layer does not consider target information and content information together, so the design pays attention to the interaction layer to realize the correlation and interaction between the two.

Specifically, the three matrices of inputs Q, K, and V are copies of the encoding matrix corresponding to content or target, corresponding to the three important components of attention, i.e., query, key, and real value, having the following computational procedure.

$$\text{attention}\ (Q, K, V) = \operatorname{softmax}\left(f_{\text{att}}(Q, K)\right)V, \tag{6}$$

where f_{att} is the probability alignment function. In this article, the scaled dot product is used:

$$f_{\text{att}}(Q, K) = \frac{QK^{\mathrm{T}}}{\sqrt{d_k}}. \tag{7}$$

Target coding and content coding achieve interactive correlation by exchanging V values. The attenuating attention calculation proposed in this paper is introduced below. First, we find the target center position in the content text by string matching. We set the hyperparameter n as the window size parameter, which indicates half of the window size. After that, the text constructs the local information

Table 2: Findings of the comparative analysis.

Model ID	Positive			Negative		
	P	R	$F1$	P	R	$F1$
1	0.938	0.911	0.924	0.914	0.941	0.927
2	0.936	0.924	0.930	0.925	0.937	0.931
3	0.922	0.924	0.923	0.924	0.923	0.923
4	0.953	0.936	0.931	0.936	0.926	0.931
5	0.927	0.894	0.923	0.901	0.957	0.928

mask, and the relevant mathematical expressions are as follows:

$$M_i = \begin{cases} E, & |i - P| \leq n, \\ \dfrac{n}{|i - P|} * E, & |i - P| > n. \end{cases} \tag{8}$$

The primary capsule layer after several layers of repetition is the network layer for the initial extraction of multihead attention output. Since the dimensionality of each capsule is large at this time, the k-max-pooling operation is used to obtain the largest k-dimensional feature in each capsule, and the final capsule representation is obtained by linear variation and vector compression, at which time the number of words in the vector of the output capsule network is summed with the number of words in the target. In this paper, we finally obtain the output capsules by classifying the capsule layer. According to the definition of the capsule network, the model length of its vector is expressed as the probability of that classification, and the final category is selected for the category represented by the capsule with the longest mode. In this paper, the loss of each capsule j is evaluated and calculated as follows:

$$L_j = Y_j \max\left(0, m^+ - \|v_i\|\right)^2 + \lambda\left(1 - Y_j\right)\max\left(0, -m^- + \| v_i \|\right)^2. \tag{9}$$

Using SEP lexical chunks as splicing separators, the model can recognize that facts and conjectures belong to two paragraphs of the text. The generation module uses the native until as the basic framework. The probability calculation based on spark softmax is shown in the following formula:

$$p_i = \begin{cases} \dfrac{e^{s_i}}{\sum_{j \in \Omega_k} e^{s_j}}, & i \in \Omega_k, \\ 0, & i \notin \Omega_k. \end{cases} \tag{10}$$

At this time, the inequality for the existence of cross-entropy is as follows:

$$\begin{aligned} \log\left(\sum_{i=1}^{n} e^{s_i}\right) - s_{\max} &= \log\left(1 + \sum_{i \neq t} e^{s_i - s_{\max}}\right), \\ &\geq \log\left(1 + (n-1)e^{s_{\min} - s_{\max}}\right). \end{aligned} \tag{11}$$

4. Experiments and Results

This paper was designed to compare multiple classification models. Among them, the plain Bayesian use and the maximum entropy model use 10543 action emotions as features. In order to compare the performance difference between conventional and pretraining-based models, the model fusion is not taken into account in this paper. Although all of these models are valid, a combination of different machine learning methods will yield better scores. Details about the experimental setup are presented in the following lines. The verification set and the test set are separated from the training set, with a proportion of 20%. If the training index is not significantly improved after the batch verification set is 100. This paper tested the following five models on MioChnCorp-2. The benchmark model is a fine-tuning BERT model with all trainable parameters. A direct output vector using a pretrained model is then fed into a feedforward neural network classifier with all parameters trainable. Each vector obtained from the pretrained model is clustered using a capsule network, which is then fed into a feedforward neural network classifier with untrained pretrained model parameters. The experimental parameters were set as shown in Table 1.

The dynamic representation, pooling, and splicing of local features are further extracted using CNN and then fed into a feedforward neural network classifier. The splicing clustering features and local features are then fed into a feedforward neural network classifier with untrained pretrained model parameters. The results of the measurement are shown in Table 2 and Figure 6.

Comparing model 2 and model 3 in Table 2, it can be found that CAP and CNN can successfully extract high-level features, but CAP is more effective as it has 0.7% higher accuracy than CNN. This means that the fine-tuning model may not be necessary. It can be found that fine-tuning the language model will even reduce the optimal result by 0.5%, which may be harmful to the extraction of high-level semantic features. The pretraining language model represented by BERT has a significant impact on the text classification task and significantly improves the performance. However, there are some obvious shortcomings, especially in terms of more training time than traditional fine-tuning models. The classification accuracy of each model on the dataset is shown in Table 3 and Figure 7.

RETRACTED

RETRACTED

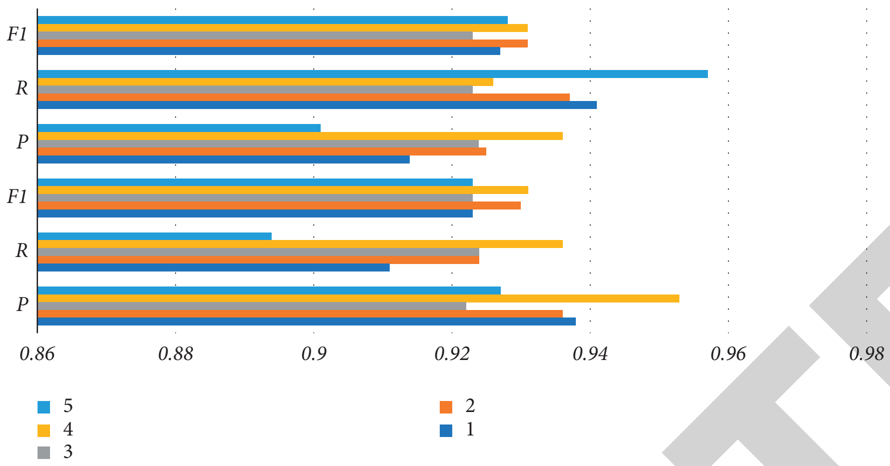


Figure 6: Results of the comparative measurements.

Table 3: The classification accuracy of each model on the dataset.

	Model	Accuracy	Macro-F1
Baseline	TransCap	0.739	0.701
	MemNet	0.703	0.641
	CNN-ASP	0.725	0.653
	TOCA-W2V	0.765	0.719
BERT	BERT-PT	0.781	0.751
	BAT	0.794	0.765
	AEN-BERT	0.800	0.763
	TOCA-BERT	0.814	0.783

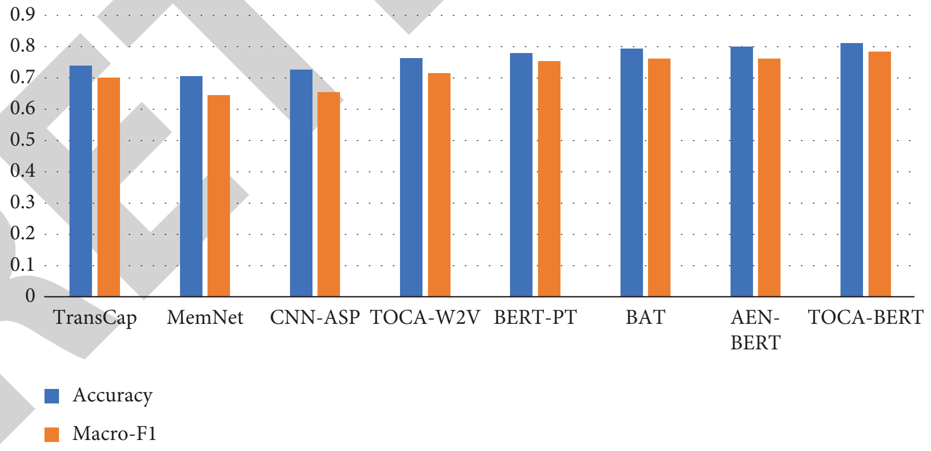


Figure 7: The classification accuracy of each model on the dataset.

Among all the attention mechanism models, ATAE-LSTM has the worst classification performance, even inferior to the traditional SVM method. The performance advantage of the remaining models with more network depth is obvious, which shows that increasing the network depth has a positive effect on the multi-input classification model ability.

5. Conclusion

Emotion is the soul of dance. It is precise because of the soul that dancers can create dance movements that are more appropriate to the soul. Through the movement training, the emotion is injected into it, making the dance movement full of life. A dancer through performance shows the joys and sorrows of life to the audience. When emotion in the dance performance reaches the extreme, some actions like the technical skills are burst out. Movement is the main external factor of dance, which can be seen directly by the naked eye, while emotion is hidden in the process of dance movements which are slowly expressed through a delicate performance. The fundamental tasks in the field of dance emotion have received a lot of attention from researchers and scholars in China and abroad. Based on these, this paper proposes a dance movement aesthetic emotion analysis method based on deep learning and verifies the method through a number of experiments. This paper takes the dance movement emotion analysis task as the entry point, explores the modeling approach from three perspectives: model fusion, pretraining model tuning, and multi-input modeling. The model is adjusted by the counter samples, and the reasons for the misclassification are analyzed by observing the decision interpretation of the misclassification samples. Finally, counter samples are used to improve the performance of the model and enhance the interpretability of the model. The research constructs a self-explanatory generation module to explain the decision behavior of the model. The results show that this method not only improves the robustness of the model but also improves the interpretability of the model.

The effectiveness of the modeling method can be proved by experiments, but the black box of deep neural networks cannot be avoided. In future, the research work will be extended to cover the compressed language model based on knowledge distillation by enhancing the training dataset, and to explore new attention mechanisms to accelerate convergence of the model. In addition, the follow-up research will focus on exploring interpretable methods based on confrontation samples and combined attention mechanisms.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

References

[1] A. L. Kaeppler, "Dance ethnology and the anthropology of dance," *Dance Research Journal*, vol. 32, no. 1, p. 116, 2000.

[2] S. W. Stinson, D. Blumenfield-Jones, and J. Van Dyke, "Voices of young women dance students: an interpretive study of meaning in dance," *Dance Research Journal*, vol. 22, no. 2, p. 13, 1990.

[3] M. De Ree, G. Mantas, and A. Radwan, "Key management for beyond 5G mobile small cells: a survey," *IEEE Access*, vol. 7, pp. 59200–59236, 2019.

[4] M. Banks and K. Oakley, "The dance goes on forever? Art schools, class and UK higher education," *International Journal of Cultural Policy*, vol. 22, no. 1, pp. 41–57, 2016.

[5] L. Guarino, "Is dance a sport?: a twenty-first-century debate," *Journal of Dance Education*, vol. 15, no. 2, pp. 77–80, 2015.

[6] N. A. Oparina, U. V. Nedelnitsyna, I. D. Levina, O. V. Maltseva, and M. G. Kaitandjyan, "Folk dance as a means of formation and creative education of primary school children personality," *PalArch's Journal of Archaeology of Egypt/Egyptology*, vol. 17, no. 6, pp. 731–742, 2020.

[7] S. S. Kwan, "When is contemporary dance?" *Dance Research Journal*, vol. 49, no. 3, pp. 38–52, 2017.

[8] M. J. del Barrio-Tellado, L. C. Herrero Prieto, and C. Murray, "Audience success or art for art's sake? Efficiency evaluation of dance companies in the United States," *Nonprofit Management and Leadership*, vol. 31, no. 1, pp. 129–152, 2020.

[9] E. Brannigan, "Dance and the gallery: curation as revision," *Dance Research Journal*, vol. 47, no. 1, pp. 5–25, 2015.

[10] A. M. Wisang, "Cultural co-modification of jaranan turangga yakso dance in jaranan festival intrenggalek region," *Journal of Advances in Social Science and Humanities*, vol. 5, no. 1, pp. 555–561, 2019.

[11] S. Jadhav, M. Joshi, and J. Pawar, "Art to smart: an automated bharatanatyam dance choreography," *Applied Artificial Intelligence*, vol. 29, no. 2, pp. 148–163, 2015.

[12] H. Hamroeva, "National and cultural lines in Uzbek dance," *European Journal of Molecular & Clinical Medicine*, vol. 8, no. 1, pp. 1038–1046, 2021.

[13] R. Duerden, "Dancing in the imagined space of music," *Dance Research*, vol. 25, no. 1, pp. 73–83, 2007.

[14] R. Ali, S. Lee, and T. C. Chung, "Accurate multi-criteria decision making methodology for recommending machine learning algorithm," *Expert Systems with Applications*, vol. 71, pp. 257–278, 2017.

[15] R. Ali, J. Hussain, M. H. Siddiqi, M. Hussain, and S. Lee, "H2RM: a hybrid rough set reasoning model for prediction and management of diabetes mellitus," *Sensors*, vol. 15, no. 7, pp. 15921–15951, 2015.

[16] R. Ali, M. H. Siddiqi, M. Idris et al., "GUDM: automatic generation of unified datasets for learning and reasoning in healthcare," *Sensors*, vol. 15, no. 7, pp. 15772–15798, 2015.

[17] R. Ali, M. Afzal, M. Hussain et al., "Multimodal hybrid reasoning methodology for personalized wellbeing services," *Computers in Biology and Medicine*, vol. 69, pp. 10–28, 2016.

[18] R. Ali, M. Afzal, M. Sadiq et al., "Knowledge-based reasoning and recommendation framework for intelligent decision making," *Expert Systems*, vol. 35, no. 2, Article ID e12242, 2018.

[19] A. Oke, "Keeping time in dance archives: moving towards the phenomenological archive space," *Archives and Records*, vol. 38, no. 2, pp. 197–211, 2017.

[20] R. Ni Made, "Deconstructing ideologies behind rodat dance in Kepaon Village, Bali, Indonesia in the global era," *ASIA LIFE SCIENCES The Asian International Journal of Life Sciences*, vol. 28, no. 1, pp. 17–29, 2019.

[21] N. Safitri and R. A. A. E. Nugroho, "Stimulation dance creations art on gross motor development children aged 5-6 Years in islamic Al-huda TK semarang," *Indonesian Journal of Early Childhood Education Studies*, vol. 6, no. 1, pp. 39–42, 2017.

[22] J. Zazulak, M. Sanaee, A. Frolic et al., "The art of medicine: arts-based training in observation and mindfulness for fostering the empathic response in medical residents," *Medical Humanities*, vol. 43, no. 3, pp. 192–198, 2017.

[23] S. Buigut and O. Amaize, "Determinants of theatre, dance, and art museum attendance in the United Arab Emirates," *Journal of Heritage Tourism*, vol. 15, no. 6, pp. 612–625, 2020.

[24] A. Kussanova, B. Tleubayeva, L. Nikolayeva, and A. Shankibayeva, "DIRECTOR'S interpretation of Kazakh

RETRACTED

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9818249, 1 page
https://doi.org/10.1155/2023/9818249

Retraction

Retracted: Study on Vertical Joint Performance of Single-Faced Superposed Shear Wall Based on Finite Element Analysis and Calculation

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] G. Fang, R. Hou, W. Ma, and K. Xu, "Study on Vertical Joint Performance of Single-Faced Superposed Shear Wall Based on Finite Element Analysis and Calculation," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2418874, 8 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2418874, 8 pages
https://doi.org/10.1155/2022/2418874

RETRACTED

Research Article

Study on Vertical Joint Performance of Single-Faced Superposed Shear Wall Based on Finite Element Analysis and Calculation

Gao-Ni Fang, Rong-Cheng Hou, Wei Ma, and Kai Xu

School of Civil Engineering, Anhui Jianzhu University, Hefei 230601, China

Correspondence should be addressed to Wei Ma; mawei@ahjzu.edu.cn

Received 1 June 2022; Revised 18 June 2022; Accepted 21 June 2022; Published 18 July 2022

Academic Editor: Muhammad Ahmad

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

In this paper, a new type of single-faced superposed shear wall system is proposed based on the single-faced superposed shear wall structure. The nonlinear finite element analysis of a new splicing form of single-faced superposed shear wall components is carried out by using the method of combining experimental research and ABAQUS finite element software analysis. The impact of noncolumn on the connection performance of vertical seam is studied. The analysis shows no significant difference in the effect of eliminating concealed column on the hysteretic performance and bearing capacity of the single-faced superposed shear wall with the joint. The single-faced composite shear wall with two different splicing forms presents bending-shear failure. The single-faced composite shear wall model without concealed column design can also maintain good overall performance and seismic behavior.

1. Introduction

The composite plate shear wall structure is an integral shear wall structure formed by taking the prefabricated wall as the template and pouring concrete in the inner cavity. Compared with the ordinary reinforced concrete wall, the composite shear wall has the advantages of light weight, short construction period, and good benefits. Because of its good overall performance and shear capacity, it has become the focus of attention of Chinese and foreign scholars [1–4].

Considering the structural difference between domestic and foreign superposed shear walls, many researchers did many tests to survey their axial and eccentric compression performances, out-of-plane flexural performances, and shear connector performances [5–8], whereas rare studies involved their seismic behaviors.

Most parts of China have seismic fortification requirements, so it is very important to consider the seismic performance of composite shear walls. Zhang et al. [9] conducted low-frequency cyclic horizontal loading tests at 6 horizontal connection nodes of double-faced superposed shear walls and 2 cast-in-situ nodes. Their test results indicate that the vertical joint of double-faced superposed shear walls had seismic behavior similar to the cast-in-situ nodes. However, the seismic behavior of the horizontal joint was not mentioned. Zhao et al. [10] carried out the same tests on 3 pieces of double-faced superposed shear walls, with results as shown below. When the depth-width ratio was 2.0 and the axial compression ratios were 0.2 and 0.1, increasing the area of the longitudinal joint bar effectively controlled the deformation in the bottom joint area and the damage concentration on the wall, resulting in a full and strong energy dissipation loop. Based on this result, they pointed out that joint bars can be added to improve the joint performance of superposed shear walls. H. Li and B. Li [11] surveyed the forces on 9 pieces of reinforced concrete shear walls under cyclic loading, proposed a multispring macro finite element model of the reinforced concrete shear wall, obtained a more suitable constitutive model reflecting the relationship between load and displacement, and provided the applicable calculation formula.

Because of the structural stability, industrial production, and many other advantages, the single-sided laminated plate shear wall with an insulation layer has attracted the attention of many scholars. Einea et al. [12] found that the overall performance is equivalent to that of the integrally cast concrete wall by studying the solid concrete or metal truss reinforcement as the shear transfer mode. Benayoune et al.

RETRACTED

[13] found that when the specimen is damaged, only the top or bottom concrete is damaged, and the integrity is good. Ma et al. [14] tested the seismic behavior of such shear wall, obtaining the following results: the failures of single-faced superposed slab shear wall and holistically cast-in-situ shear wall were both bending-shear failures; in the failure stage, the former structure had a little poor ductility but performed well in energy dissipation. Their research did not investigate the performance of single-faced superposed slab shear walls that were horizontally joined. However, the horizontal joint of superposed walls is inevitable in practical engineering since the dimensions of the prefabricated walls are restricted by the modes of transportation and hoisting. Ma et al. [15] got two wall slabs jointed horizontally by using embedded columns and tested the structure's seismic behavior. The results reveal that the failure form of single-faced superposed slab shear walls jointed by embedded columns and that of a holistic single-faced superposed slab shear wall were both bending-shear failures. The two structures had similar seismic characteristics properties such as hysteretic performance, ductility, and energy dissipation capacity. However, the embedded columns reduce the assembly efficiency, so there is still room for optimization of the connection form of vertical connection joints of composite plate shear walls.

This paper proposes a new type of single-faced superposed slab shear walls with joints but without embedded columns, and based on applicable tests, a finite element model is established and analyzed numerically [16–19]. Furthermore, the seismic behavior of the joint is analyzed based on a comparison of the joints' bearing capacity and failure form under low-frequency cyclic loading.

2. Test Overview

2.1. Test Design. In this test, the authors designed a test piece PJ-1 and a test piece ZT-1, with design details as presented in Figure 1. The former was two pieces of single-faced superposed slab shear walls jointed horizontally, with columns embedded at the joints, while the latter was a holistic piece of single-faced superposed slab shear wall without joint. The two test pieces had the same physical dimensions and were both composed of three parts: the base, single-faced superposed slabs, and the top beam. The mechanical properties of the main materials of the reinforcement and concrete are listed in Tables 1 and 2.

2.2. Loading Scheme. This test was carried out in the Anhui Provincial Key Laboratory of Building Structure and Underground Engineering. The designed axial compression ratio was 0.1. An oil Jack was applied to the top parts of the test pieces with a constant vertical load of 600 kN. The top beams were under low-frequency horizontal cyclic loading from a 100 T electro-hydraulic servo actuator to do pseudo-static tests. The loading device is illustrated in Figure 2.

The loading protocols in Figure 3 were formulated as per the *Standard for Test Method of Concrete Structures*, and the loading was under mixed control of load and displacement. In the first stage, the load is loaded through load control. The horizontal force is loaded from 50 kN and added to the specimen yield with 25 kN as the level difference. The load cycle of each level is once. After the specimen yield, the load is controlled by displacement. The multiple of the horizontal displacement value measured by the external displacement meter on the top of the shear wall and the built-in displacement meter in the system is selected for loading step by step, and each level of displacement is cycled three times. When the test piece is loaded to its own destruction or the load value of the test piece drops to 85% of the maximum load value, the test is completed.

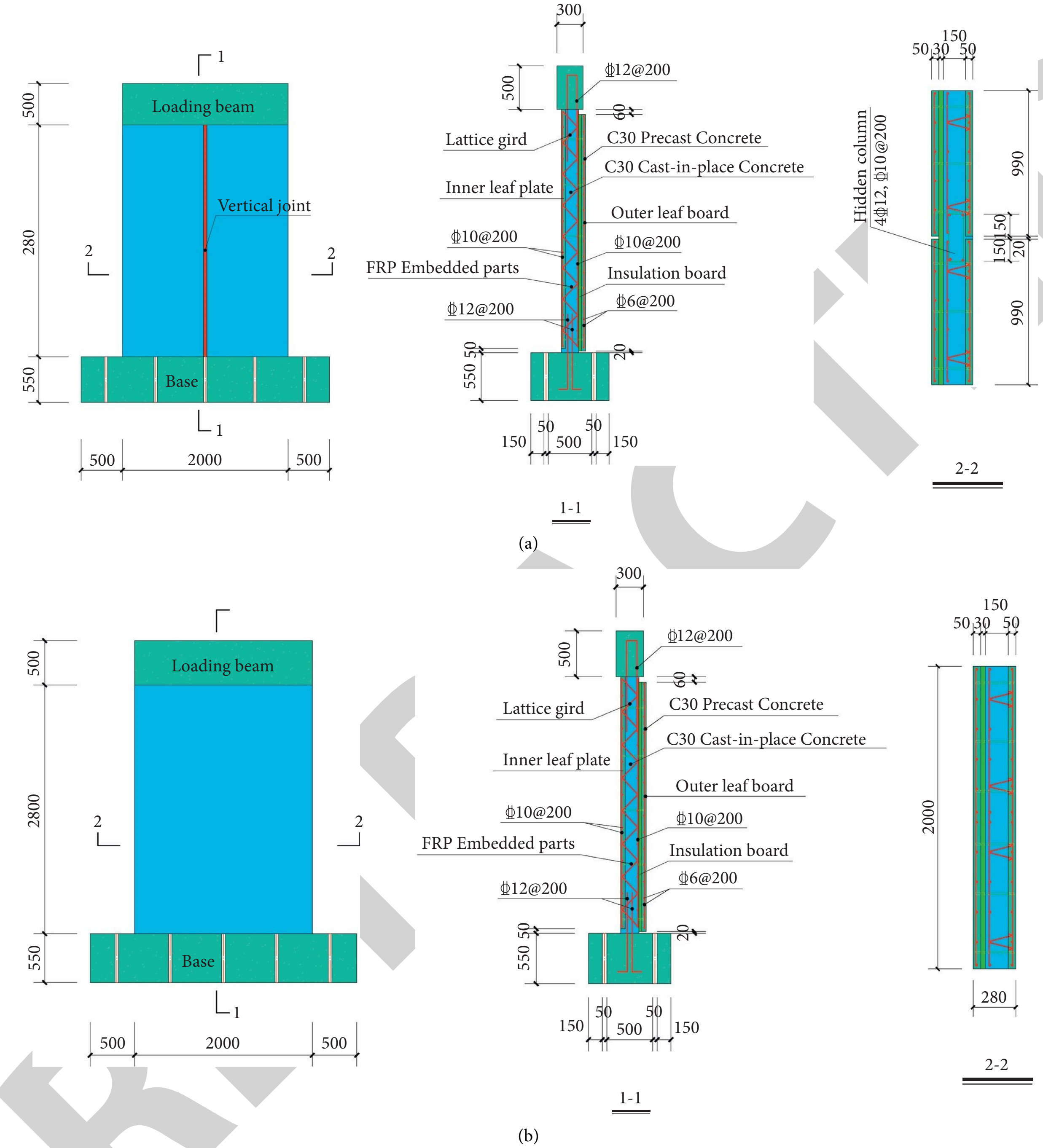
2.3. Test Results. The failures on the test pieces are depicted in Figure 4. The wall edge showed horizontal cracks, as seen on test pieces PJ-1 and ZT-1. As the load increased, some horizontal cracks gradually extended and formed inclined bending-shear cracks. The inclined cracks ran at 45° roughly and finally formed X-shaped cracks. As observed from the development process and distribution of the cracks, the test pieces bent first and then sheared, forming bending-shear failures. The upper parts of the walls were almost intact, and the concrete at two ends of the base became weak and peeled off. During the entire loading on test piece PJ-1, there is no slip in the joint of the concealed column all the time, and the crack development is relatively sufficient, indicating that the joint structure of concealed column is reasonable, which can effectively splice two single-sided superimposed shear walls to form an overall common stress.

As for the hysteretic curves of the test pieces in Figure 5, their envelope areas were both in a straight line in the initial stage of loading and nearly coincided before cracking, complying with the force characteristics in the elastic stage. After yielding, the envelope areas enlarged gradually, and the test pieces' energy dissipation capacity increased somewhat. Over a comparison, it can be seen that the hysteretic curves of test pieces PJ-1 and ZT-1 almost coincided. This result implies that a vertical joint had a small impact on the hysteretic performance of a single-faced superposed slab shear wall. At the same time, the embedded columns enabled the jointed single-faced superposed slab shear walls to form a whole part subjecting to a holistic force.

3. Calculation for the Finite Element Analytical Model

3.1. About the Model. According to the test results, the authors raised a test piece PJ-2 with the same design as PJ-1 but without embedded columns. Based on this new design, two shear-wall finite element models were established by using the Abaqus FEA.

3.2. Parameters of the Model. Given the loading protocols and the cyclic loading on concrete, the constitutive model of the concrete material adopted the concrete damaged plasticity model contained in the Abaqus FEA. Tensile and compression damage factors were used to depict the degradation in the initial elasticity of concrete and the damage accumulation [20]. The material properties of the model



(a)

(b)

Figure 1: Physical dimensions and reinforcements of the test pieces. (a) PJ-1. (b) ZT-1.

Table 1: Mechanical properties of reinforcement materials.

Pieces number	Reinforcement type	Diameter (mm)	Yield strength f_y (MPa)	Tensile strength f_u (MPa)	Elongation A (%)
PJ-1	HRB400	10	492.2	525.1	11.9
	HRB400	12	544.2	668.2	12.5
ZT-1	HRB400	10	535.6	668.5	20.01
	HRB400	12	487.3	641.2	19.03

RETRACTED

Table 2: Mechanical properties of concrete materials.

Pieces number	Part	Cubic strength f_{cu} (MPa)
PJ-1	Prefabricated part	33.5
	Cast-in-place part	33.3
ZT-1	Prefabricated part	33.5
	Cast-in-place part	31.8

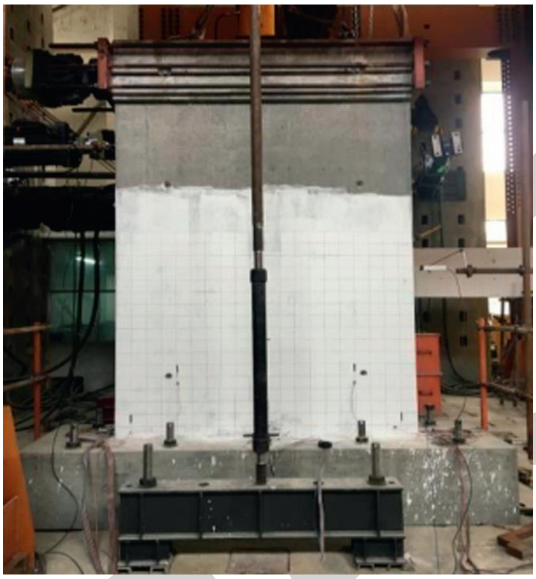


Figure 2: Loading device for the test pieces.

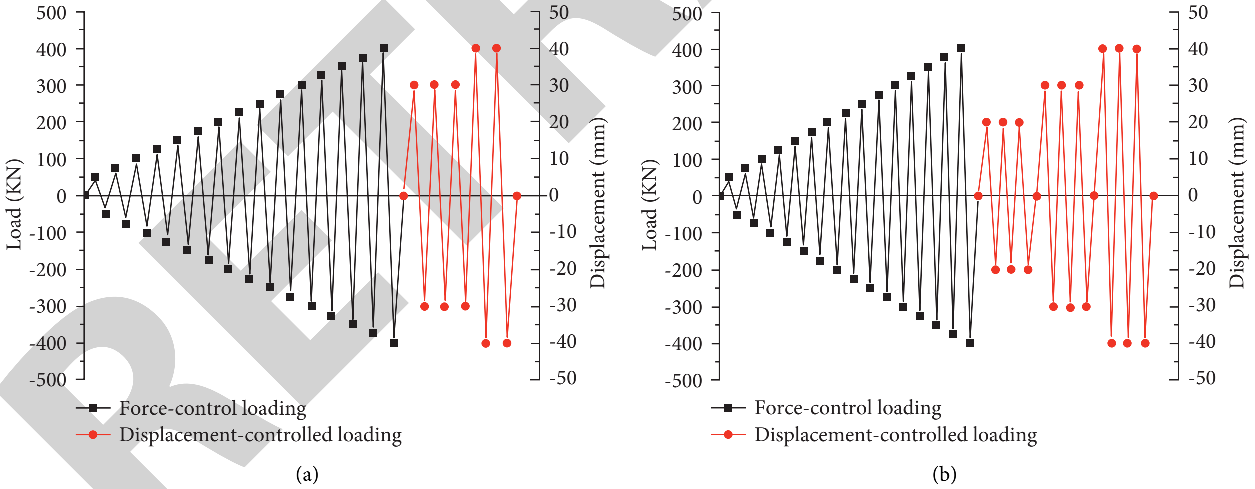
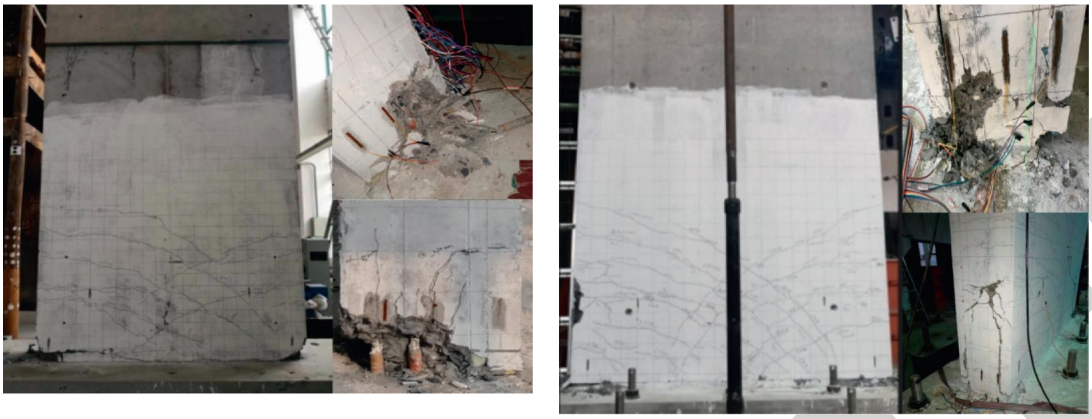


Figure 3: The loading protocols. (a) PJ-1. (b) ZT-1.

were input as per the *Code for Design of Concrete Structures* (GB50010–2010). In the finite element model, the concrete adopted the C3D8R solid element, the dilation angle was 30°, the flow eccentricity was 0.1, $(f_{b_0}/f_{c_0}) = 1.16$ (f_{b_0} is the biaxial compressive strength of concrete, and f_{c_0} is the uniaxial compressive strength), the Poisson's ratio was 0.2, the elastic modulus was 3×10^4 MPa, the ratio of second stress invariants on the tensile and compression meridians was 0.667, and the viscosity coefficient was 0.005. The reinforcement adopted the T3D2 truss element. The experimental values of materials expressed the constitutive relation of concrete in the test, where the yield strength f_y of steel was 530 MPa, the Poisson's ratio was 0.3, and the elastic modulus was 2×10^5 MPa. As the forces on the top beam and the base were not considered in the test, the two parts were designed with infinite rigid beams under ideal conditions.

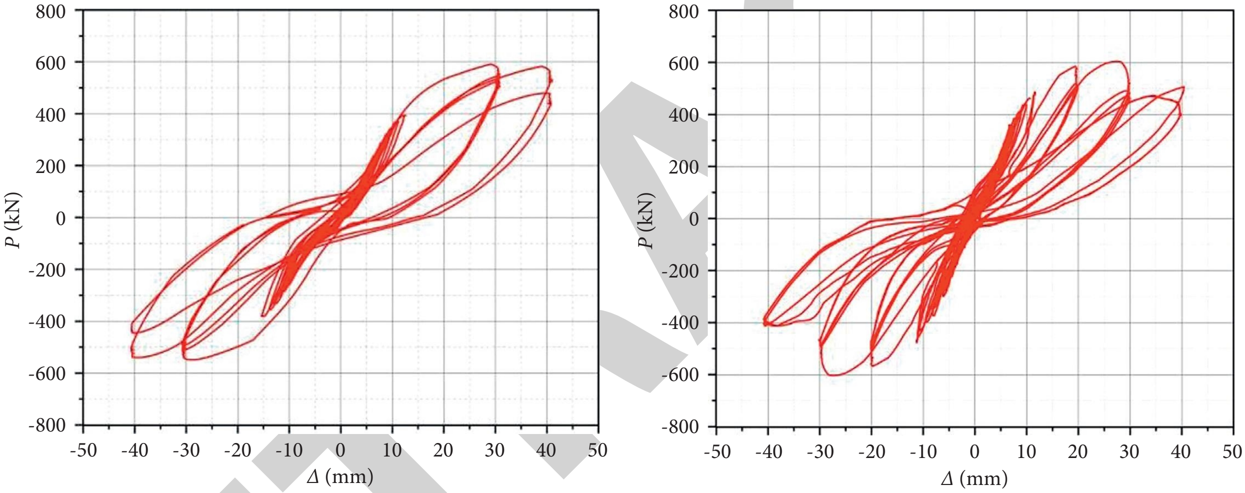
The coupling points were arranged in the center of the bottom beam and fixed on the "load" module. The reinforcement got in contact with and constrained by the concrete by an embedding range constraint order. The

RETRACTED



(a) (b)

Figure 4: Failures on the test pieces. (a) PJ-1. (b) ZT-1.



(a) (b)

Figure 5: Hysteretic curves of the test pieces. (a) PJ-1. (b) ZJ-1.

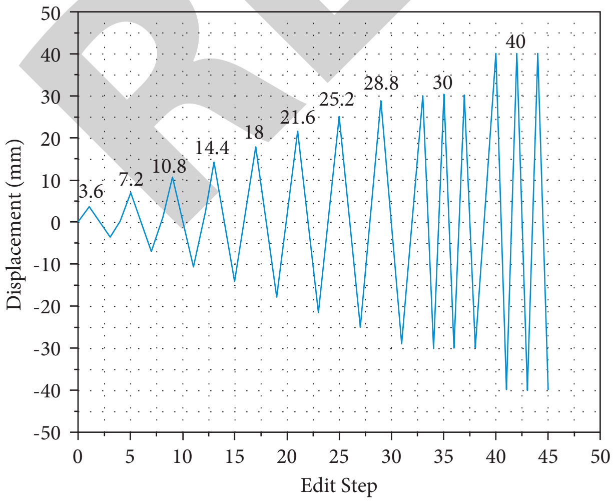


Figure 6: The loading protocols.

superposed interface between the prefabricated member and cast-in-situ concrete was in binding contact. As this model was mainly used to investigate the seismic behavior of the shear wall, the interfaces between the shear wall and the loading beam and between that and the base were also arranged in binding contact. An FRP connection was set between the outer leaf board and the load-carrying wall of the test piece and bore no force all the way. For this reason, frictional contacts (friction coefficient: 0.6 [9]) were designed between the thermal insulation board and the outer leaf board concrete and between that and the post-cast intracavity concrete. The normal force transmission between the new and old superposed concrete faces was by hard contact so that the applied force vertical to the interface was fully transmitted in the cross section.

The model was designed with three analytical steps: initial step, step 1, and step 2. In the initial step, relevant boundary conditions were set. In step 1, 600 kN axial

RETRACTED

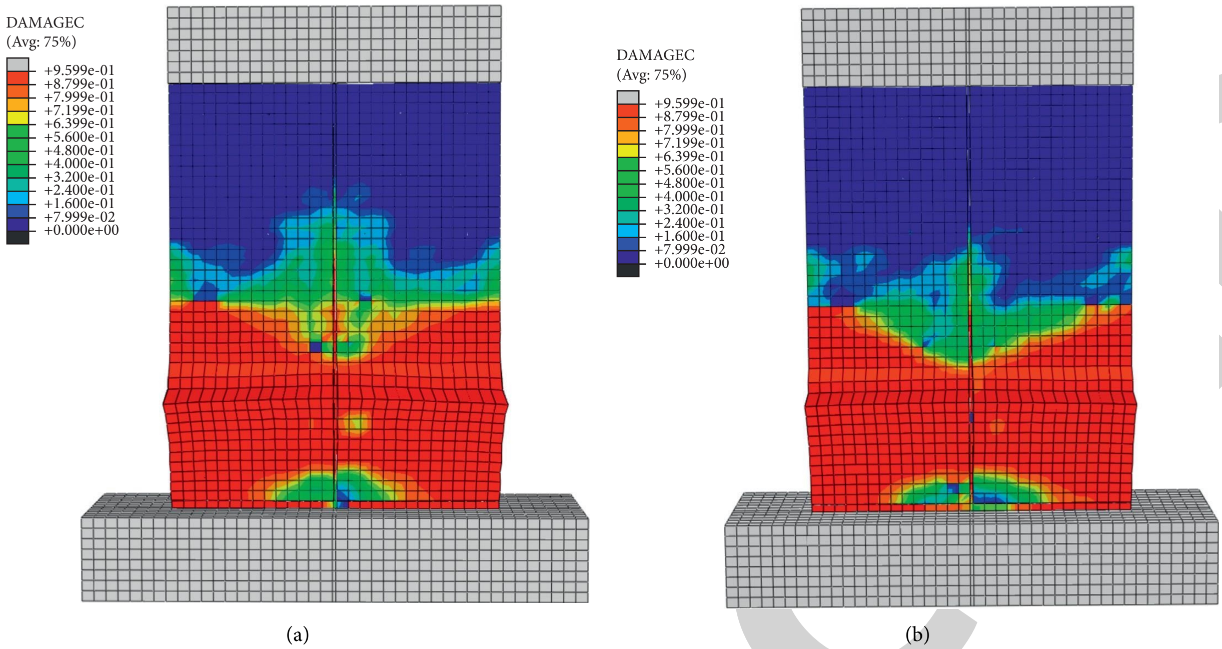


Figure 7: The cloud chart of concrete damage. (a) The design with embedded columns (PJ-1). (b) The design without embedded columns (PJ-2).

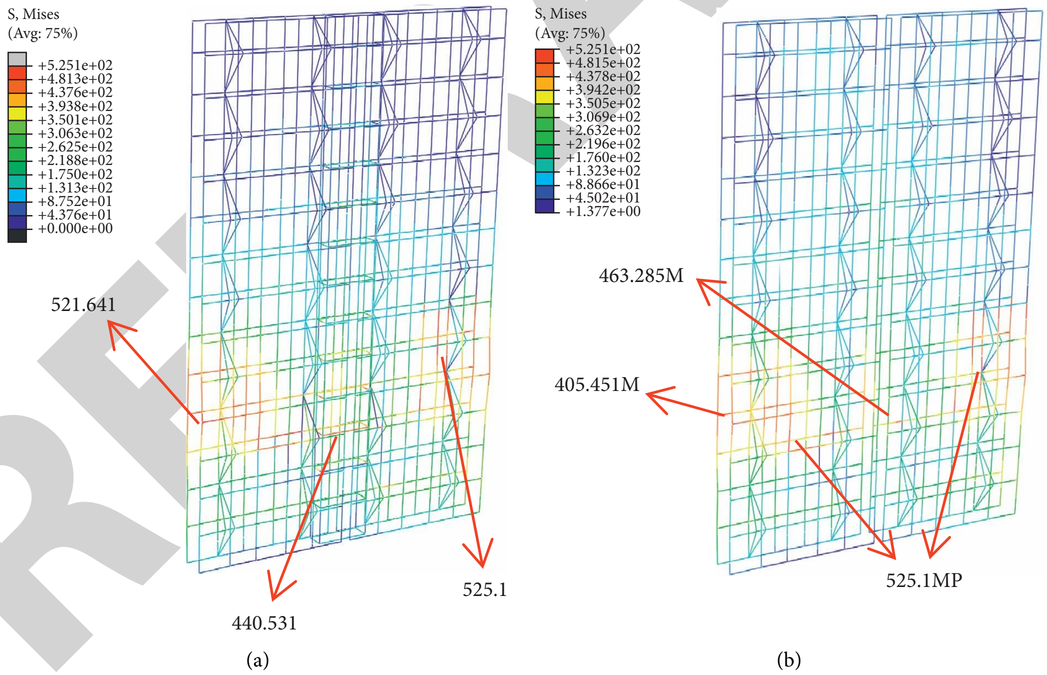


Figure 8: The cloud chart of reinforcement stress. (a) The design with embedded columns (PJ-1). (b) The design without embedded columns (PJ-2).

pressure was loaded at the coupling point in the center of the top beam. In step 2, the loading time was set as per the loading protocols for the test, and the load-displacement mixed control loading was simulated and converted into displacement loading based on the model's amplitude function. After conversion, the loading protocols are illustrated in Figure 6.

4. The Simulated Results and Analysis

4.1. Damage. Figures 7 and 8 display the concrete analysis results of the finite element model. As can be observed from the concrete damage cloud chart, the concrete damage at the joint of the shear wall model without embedded columns was greater than that with embedded columns. However, the two designs

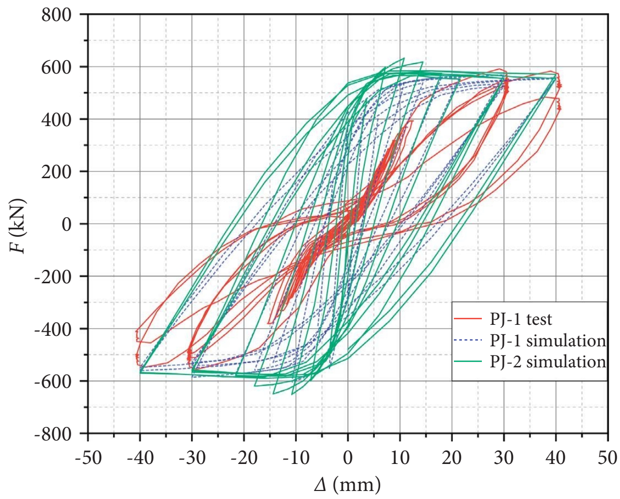


Figure 9: Comparison of the hysteretic curves.

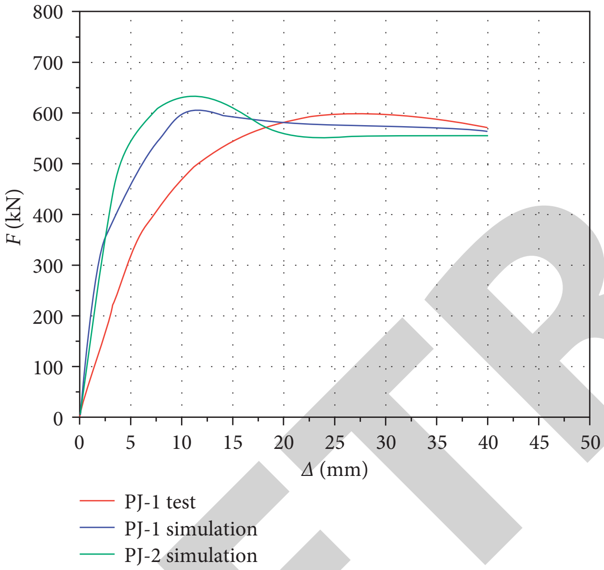


Figure 10: Comparison of the skeleton curves.

show almost identical holistic failure forms. According to the cloud chart of reinforcement stress, in both of the two models, the reinforcement stress on the two sides was the highest, gradually decreased toward the joint, and had no interruption and sudden change at the joint. This result demonstrates that the reinforcement meshes of both two models (with and without embedded columns) can form holistic force-bearing bodies. The reinforcement stress at the joint of test piece PJ-1 was 440.531 MPa, while that at the joint of test piece PJ-2 was 463.285 MPa, with a difference of 5.2% from the former. This result reveals that the design with or without embedded columns had no great impact on the model, and the model without embedded columns still had a good holistic performance.

4.2. Hysteretic Performance. As presented in the hysteretic curves and skeleton curves (Figures 9 and 10) of the two test pieces obtained by simulated calculation, the area enveloped by the hysteretic loop of PJ-2 was slightly larger than that of PJ-1. The reason was that, without embedded columns, the concrete at the vertical joint showed failures under the action of the horizontal load, so the test piece was divided into two small shear walls bearing forces continuously. At this point, the test piece has lost overall rigidity, gained ductility, and gained energy dissipation capacity. Moreover, the PJ-2's ultimate bearing capacity became a little higher than the PJ-1's. These results verify that the design without embedded columns is feasible.

As also presented in the following figures, the skeleton curves of PJ-1 and PJ-2 had similar trends. The simulated curves of the two in the elastic stage almost coincided because no significant changes occurred in the mechanical properties of the two shear walls in this stage. After entering the elastic-plastic stage with continuous loading, the reinforcement inside the test piece yielded gradually, and both the two models experienced obvious rigidity degradation. However, PJ-2's rigidity degraded more significantly than PJ-1's due to the concrete failures at the joint of PJ-2. The skeleton curves tended to be gentle in the plastic stage, which suggests that both structures had good ductility (without a large difference) after subjecting to the ultimate load.

5. Conclusion

In this study, the test and ABAQUS FEA were used to make finite element analysis on new single-faced superposed slab shear walls, obtaining the following conclusions:

First, the design with or without embedded columns has no significant impact on the bearing capacity and hysteretic performance of single-faced superposed slab shear walls horizontally jointed. Using the two designs, the single-faced superposed slab shear walls showed bending-shear failures. This result indicates that the horizontally jointed single-faced superposed slab shear walls with loading beam can still have good holistic performance and seismic behavior in practical engineering, even without embedded columns.

Second, the design without embedded columns helps simplify the construction process and improve assembly efficiency while ensuring the structure's bearing capacity.

Third, the optimized design was just simulated in this study and still requires further experiments in the future to verify its rationality and consistency.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Fund of Education Department of Anhui Province under Grant KJ2020A0478, Science and technology plan for Housing and Urban Rural Construction in Anhui Province (2021-YF20).

RETRACTED

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1759542, 12 pages
https://doi.org/10.1155/2022/1759542

Research Article

Efficient Detection Method of Pig-Posture Behavior Based on Multiple Attention Mechanism

Li Huang, Lijia Xu, Yuchao Wang, Yingqi Peng, Zhiyong Zou, and Peng Huang

College of Mechanical and Electrical Engineering, Sichuan Agriculture University, Ya'an 625014, China

Correspondence should be addressed to Lijia Xu; xulijia@sicau.edu.cn

Received 5 May 2022; Revised 7 June 2022; Accepted 13 June 2022; Published 16 July 2022

Academic Editor: Muhammad Ahmad

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

Due to the low detection precision and poor robustness, the traditional pig-posture and behavior detection method is difficult to apply in the complex pig captivity environment. In this regard, we designed the HE-Yolo (High-effect Yolo) model, which improves the Darknet-53 feature extraction network and integrates DAM (Dual attention mechanism) of channel attention mechanism and space attention mechanism, to recognize the posture behaviors of the enclosure pigs in real-time. First, the pig data set is clustered and optimized by the K-means algorithm to obtain a new anchor frame size. Second, the DSC (Depthwise separable convolution) and h-switch activation function are innovatively introduced into the Darknet-53 feature extraction network, and the C-Res (Contrary residual structure) unit is designed to build Darknet-A feature extraction network, so as to avoid network gradient explosion and ensure the integrity of feature information. Subsequently, DAM integrating the spatial attention mechanism and the channel attention mechanism is established, and it is further combined with the Incep-abate module to form DAB (Dual attention block), and HE-Yolo is finally built by Darknet-A and DAB. A total of 2912 images of 46 enclosure pigs are divided into the training set, the verification set, and the test set according to the ratio of 14 : 3:3, and the recognition performance of HE-Yolo is verified according to the parameters of the precision P, the recall R, the AP (i.e., the area of P-R curve) and the MAP (i.e., the average value of AP). The experiment results show that the AP values of HE-Yolo reach 99.25%, 98.41%, 94.43%, and 97.63%, respectively, in the recognition of four pig-posture behaviors of standing, sitting, prone and sidling of the test set. Compared with other models such as Yolo v3, SSD, and faster R–CNN, the mAP value of HE-Yolo is increased by 5.61%, 4.65%, and 0.57%, respectively, and the single-frame recognition time of HE-Yolo is only 0.045 s. In the recognition of images with foreign body occlusion and pig adhesion, the mAP values of HE-Yolo are increased by 4.04%, 4.94%, and 1.76%, respectively, while compared with other models. Under different lighting conditions, the mAP value of HE-Yolo is also higher than that of other models. The experimental results show that HE-Yolo can recognize the pig-posture behaviors with high precision, and it shows good generalization ability and luminance robustness, which provides technical support for the recognition of pig-posture behaviors and real-time monitoring of physiological health of the enclosure pigs.

1. Introduction

China is the world's largest producer of pigs and a major consumer of pork. According to statistics, 337.42 million pigs were sold in the first half of 2021, a year-on-year increase of 34.2% compared with 2020. The 2019 China statistical yearbook shows that China's per capita meat consumption is 26.9 kg/person, and the consumption proportion of pork in meat products is as high as 60%. This shows that pig breeding is the most important industry in China's animal husbandry, which is of great significance to national life, stabilizing prices, ensuring the smooth operation of the economy, and stabilizing the overall social situation. Most of China's pig breeding industry is still dominated by periodic manual supervision, with high labor intensity, low work efficiency, poor timeliness, and easy to make mistakes. It is imperative to use the information and intelligent technical means to detect the behavior and posture of pigs in real time.

At present, many scholars applied CNN (convolutional neural networks) to recognize multi-category targets [1–5] in different fields. In pig-posture behavior recognition, Gao et al. [6] used a 3D convolutional network to recognize the aggressive pig behaviors, but some interference factors such as pig adhesion, different lightings, and complex behaviors resulted in a weak generalization ability of CNN. Zhou et al. [7] designed a pig face detection method with an attention mechanism [8–10] combined with a feature pyramid structure, which made the network more focused on the effective information area of the images, but the gradient dispersion appeared with the deepening of the number of the network layers. Yang et al. [11] used CNN to recognize the individual pig-drinking behaviors, but the method showed the complex calculation of feature selection and relied on high hardware. Zheng et al. [12] used the faster R–CNN to recognize the standing, lying, and the other behaviors of sows, but the recognition results of different scales of targets were quite different. The R–CNN series networks designed in the literature [13–15] generated sample feature maps through convolution processing of input images, and then classify them. The experimental results showed that these networks were not very effective to recognize the multi-scale small targets. The Yolo series networks designed in the literature [16–18] converted the positioning of the target frames into a regression problem, which showed high recognition accuracy and short recognition time. Among them, the Darknet-53 feature extraction network of Yolo v3 was characterized by a large calculation, unstable gradient, and incomplete feature information extraction, and the feature information of each region of the feature map was given the same weight. In fact, the contribution degrees of feature information of some areas such as pig-limbs, pig-abdomen, and pig-trunk were far greater than that of pig-head, pig-feces and pig-pigsty. Therefore, it is considered to introduce DAM (i.e., dual attention mechanism) to focus more on the target areas.

To realize the real-time recognition of the pig-posture behaviors in the enclosure environment, a DSC (i.e., depthwise separable convolution) structure and h-switch activation function are introduced into the Darknet-53, and C-Res (i.e., contrary residual structure) unit is designed. Further, DAM is combined with Incep-abate to form the DAB (i.e., dual attention block), so as to reduce the computational complexity and the hardware cost of the network, and further reduce its false recognition rate. The rest remainder of this study is organized as follows. In Section 2, the pig mages are acquired and preprocessed, and the building process of HE-Yolo model is described. In Section 3, Experiments are conducted using the pig mages with four pig-posture behaviors in an enclosure environment, and the results are analyzed and discussed. Finally, the conclusions of this study are drawn in Section 4.

The detection method proposed in this paper can be used to infer the movement state of captive pigs. Based on this technology, the active state of pigs in a fixed time can be judged, and the judgment index of pig activity degree can be generated, which provides effective help for the follow-up behavior research and physiological condition analysis of pigs.

2. Materials and Methods

2.1. Experimental Image Preparation

2.1.1. Subsection. The pig images are acquired from a pig farm in Guang'an city of Sichuan province (10 : 00–14 : 00 on August 14th, 2021, with cloudy weather) and another pig farm in Leshan city of Sichuan Province (10 : 00–14 : 00 on August 14th, 2021, with sunny weather). Canon EOS580D anti-shake camera is 0.5~2.0 m away from the pig-body to take photos of pig-trunk of different scales. The sizes of the two pig farms are 10 m × 5 m × 0.7 m and 8 m × 6 m × 0.8 m, with 2~10 pigs in each column. A total of 46 pigs in three columns in a nursery period of 24~54 days and a fattening period of more than 70 days are selected as the study objects.

The postural behaviors of pigs vary greatly, which are usually divided into four categories: standing, sitting, prone, and sidling. A total of 30 segments of video data of pigs are manually screened, with each segment of 10~100 s, and 728 images with a resolution of 1920 × 1080 are obtained by frame-by-frame processing. To expand the number of images and improve the universality of the network, we perform -10° flip, $+10^{\circ}$ flip, mirror flip, and random luminance adjustment on the images to expand the data set (shown in Figure 1). After expansion, a data set of 2912 images is obtained.

2.1.2. Devising the Image Data. Take the pig-limb, the pig-abdomen, and the pig-trunk as the dividing points and do not mark the pig images in which any two of three parts do not appear within the marking range. The open-source tool labeling [19] is used to label the images with four pig-posture behaviors, namely "standing", "sitting", "prone" and "sidling". According to the general division strategy [20], the labeled extended dataset is divided into the training set of 2040, the validation set of 436, and the test set of 436 according to the ratio of 14 : 3:3, and with the resolution of 1920 × 1080 × 3. The images are further scaled and converted into the network with a resolution of 416 × 416. After labeling all images, the coordinates of the marked parts are transformed to obtain the coordinates of the scaled corresponding parts, as shown in Figure 2.

2.2. Building HE-Yolo. The Yolo series network is mainly composed of a pooling layer and convolution layer, and the resolution of the input image is 416 × 416. The input image is processed by a series of 3 × 3 depthwise convolution and 1 × 1 pointwise convolution, pooling, and up-sampling operations to output the feature map. Darknet-53 consists of 1 × 1 pointwise convolution and 3 × 3 depthwise convolution layers, and the batch normalization layer and Leaky ReLU concatenated after each convolutional layer are to prevent overfitting. The convolutional layer, batch normalization layer, and Leaky ReLU activation function together form the basic convolution unit DBL of Darknet-53, with a total of 53 DBL units. To predict the multi-scale small targets, Yolo v3 uses anchor boxes of 9 sizes. According to the idea of a feature pyramid network, Yolo v3 designs 3 sizes of network

Original image -10° flip +10° flip Mirror flip Brightness adjustment

Figure 1: The image data after expansion.

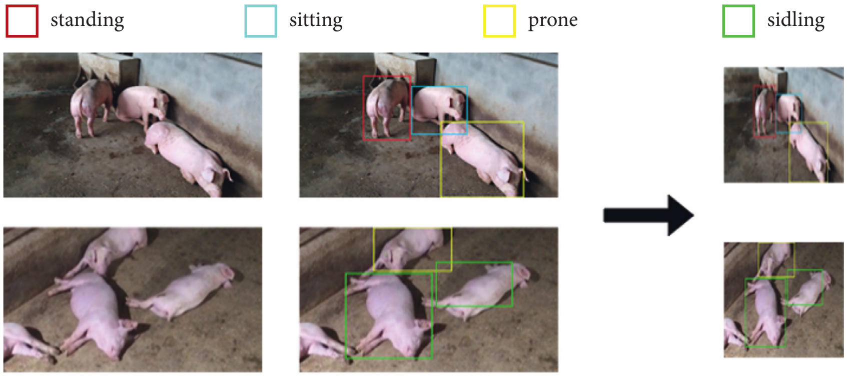


Figure 2: The process of labeled images.

outputs to predict the targets of different scales, and its multi-scale feature extraction module shows strong recognition ability. The multi-scale feature map has multiple recognition results for the same target, so Yolo v3 introduces nonmaximum suppression [21] to eliminate the redundant detection frames, thus each target has a unique detection frame, and each area in the input feature map is assigned to an equal weight. The pig-trunk, the pig-abdomen, the pig-limb, the pig-feces, and the pig-pen in the images have different contributions to recognize the pig-posture behaviors. The weights of the feature information of the pig-feces, the pig-pen, and the other interfering feature information should be reduced, and the weights of feature information of the g-trunk, the pig-abdomen, and the pig-limb should be strengthened to improve the recognition accuracy of the pig-posture behaviors.

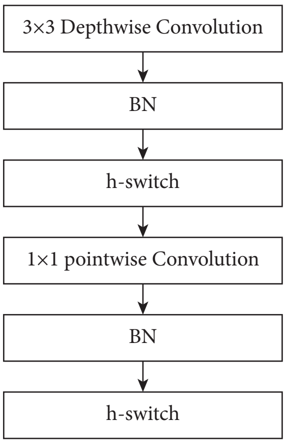


Figure 3: The structure of DPH unit.

2.2.1. Darknet-A Network. The DSC [22] and the h-switch activation function are introduced into the Darknet-53, and the C-Res unit is designed to build the Darknet-A feature extraction network.

(1) DBH Unit. The four pig-posture behaviors of standing, sitting, prone and sidling are closely related. Considering that the large sizes of pigs make it difficult to classify the behaviors, DSC and h-switch activation functions are introduced into Yolo v3 to reconstruct the structure of DBL of Darknet-53, that is, to form the DPH unit, so as to improve the recognition ability of network and reduce the amount of calculation. The structure of the DPH unit is shown in Figure 3.

The DSC decomposes the ordinary convolution in DBL into 3×3 depthwise convolution and 1×1 pointwise convolution, as shown in Figure 3. Of which, 3×3 depthwise convolution used different filters to convolve different channels of the input feature map, and then the feature maps outputted by 3×3 depthwise convolution are further weighted and combined by 1×1 pointwise convolution. Assume that the input feature map has M channels, the output feature map has N channels, and the size of convolution kernel is $D_k\times D_k$ then the amount of calculation of the depthwise convolution and the standard convolution is compared as follows:

$$\frac{H\times W\times M\times D_k\times D_k + H\times W\times M\times N}{H\times W\times M\times N\times D_k\times D_k} = \frac{1}{N}+\frac{1}{D_k}, \qquad (1)$$

where $N \gg D_k$. When $D_k = 3$, the parameter quantity of DSC is about 1/9 of the standard convolution, which means that the DSC can greatly reduce the amount of calculation of Darknet-53.

The Leakly Relu activation function is used in DPH, and its gradient value is zero when the input is negative. As a result, the corresponding parameters of the input feature map will not be updated, and the data expansion will become more significant with the deepening number of network layers, thus it affects the recognition result of Yolo v3. The above problem can be solved by replacing the Leakly Relu activation function with the h-switch activation function [23]. The *h*-switch activation function is as follows:

$$f(x) = x\frac{\text{Relu6}(x+3)}{6}, \tag{2}$$

where Relu6 () is the Relu activation function, but its maximum output is limited to 6.

The curve of the h-switch activation function is shown in Figure 4. It can be seen from Figure 4 that when $x \longrightarrow -\infty$, $f(x) \longrightarrow 0$.That is, the change trend of the *h*-switch activation function is similar to that of the Leakly Relu function, but when x is near 0, the gradient of the *h*-switch function will not completely disappear, which ensures the integrity of the feature information of the pig behaviors, and the *h*-switch activation function also reduces the computation complexity of the network.

(2) C-Res Unit. In the training process of a network based on the stochastic gradient descent algorithm, with the deepening of the network layers, the error is easy to cause gradient dispersion and gradient explosion through multilayer back propagation. Therefore, the C-Res unit is designed based on the residual structure of Resnet.

Assuming there is a convolutional neural network, the input feature map needs to undergo N times of feature extraction, and each feature extraction corresponds to a nonlinear transformation Z_i. Z_i is a set of multiple functions, such as the depthwise separable convolution, the activation function, the down sampling or pooling, etc. So there is as follows:

$$y_i = Z_i(y_{i-1}), \tag{3}$$

where y_i and y_{i-1} represent the feature maps outputted by the *i*th layer and the (i-1) th layer, respectively.

The C-Res unit is expressed as follows:

$$y_i = Z_i(y_{i-1}) + y_{i-1}. \tag{4}$$

The structure of the C-Res unit is shown in Figure 5.

Compared with the ordinary residual structure, the C-Res unit first raises the input feature map to a higher dimension and extracts its features by 3 × 3 depth convolution, and further reduces the dimension of the features, so as to reduce the computational load of Darknet-A and avoid gradient explosion.

2.2.2. DAB

(1) The Spatial Attention Block. Spatial attention mechanism (SAM) [24] can enhance the target features in the input feature map. SAM assigns different weights to the pixels in the input feature map, and simultaneously performs global max pooling and global average pooling on all features of different channels in the same position, and then obtains two feature maps, which are further concatenated in the channel dimension to gain the initial spatial attention feature map of $H \times W \times 2$. Subsequently, $7 \times 7 \times 1$ convolutional layer is used to reduce the initial spatial attention feature map to one channel, and generate the spatial weight coefficients through the sigmoid activation function. Then the spatial weight coefficients are multiplied with the input feature map through the residual structure to obtain the output feature map of SAM. The structure of the SAM module is shown in Figure 6.

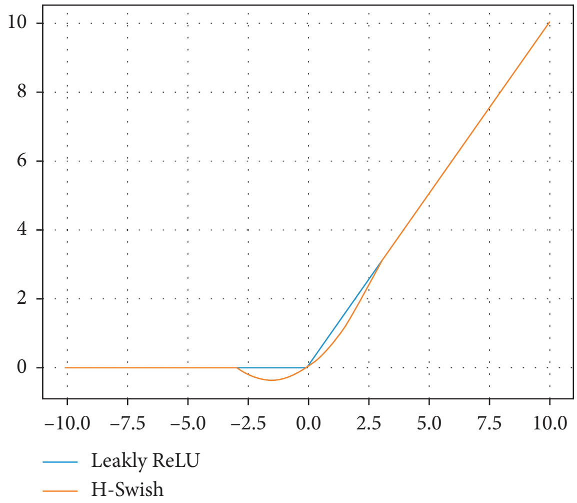


FIGURE 4: The *h*-switch activation function.

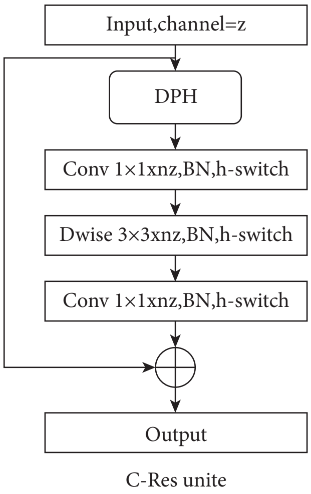


FIGURE 5: The structure of C-Res unit.

(2) The Channel Attention Block. Channel attention mechanism (CAM) [25] weights the features of all channels of the input feature map to strengthen the effective features and weaken the irrelevant information. Among them, the global average pooling is used to aggregate the feature information contained in each spatial channel [26, 27], and the input feature map of $H \times W \times C$ is pooled by global average pooling to obtain two-dimensional compressed feature of $1 \times 1 \times C$. Global average pooling extracts the features of all areas of the

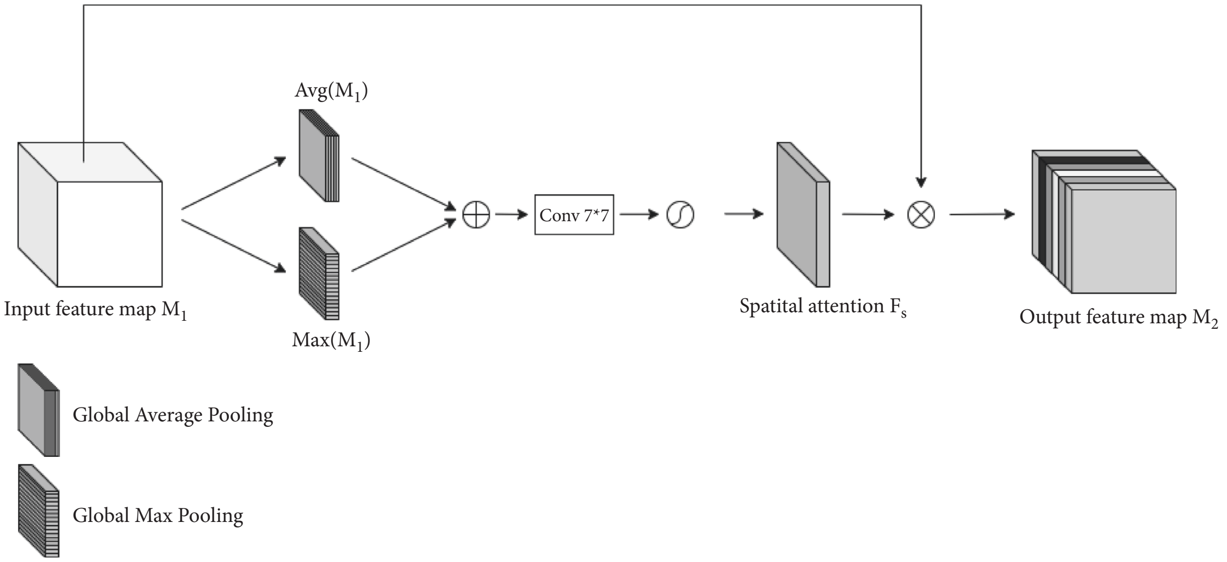


Figure 6: The structure of SAM module.

feature map with equal weight, while global max pooling only calculates the area with the largest feature response, so it can make up for the deficiency of global average pooling in extracting channel attention information in the effective feature region. The structure of CAM module is shown in Figure 7.

(3) The Channel Attention Block. SAM and CAM are connected in series to build a dual attention mechanism (DAM). First, SAM is used to model the feature elements of the input feature map, so as to obtain the positional relationship between the feature elements and filter the spatial information, and then connect them through the residual structure and multiply by space to obtain the spatial attention feature map. Then, CAM is introduced to model the correlation between the channels of the spatial attention feature map, so that the feature elements in each position of the spatial attention feature map can obtain the corresponding spatial information, and they are also connected by the residual structure and multiplied by space, that is, the output feature map of DAM can be obtained. The structure of DAM module is shown in Figure 8.

M and M_0 represent the input and the output feature maps of the DAM module, respectively, and their dimensions $\in R^{H\times W\times C}$. F_S represents the spatial weight coefficients generated by SAM, and F_C represents the channel weight coefficients generated by CAM, $F_S, F_C \in (0,1)$. M_0 is calculated as follows:

$$M_0 = \left(1 + F_C\left(\left(1 + F_S(M)\right) \times M\right)\right) \times \left(1 + F_S(M)\right) \times M. \tag{5}$$

Based on the principle of thermodynamics, the output feature map M_0 obtained by DAM is converted into a heat map, and the visualization results of the feature information of key areas such as the pig-trunk, the pig-abdomen, and the pig-limb are shown in Figure 9.

DAM increases the computational complexity of the network. Since the Inception-v4 [28] module builds deeper and wider residual connections, the Incep-abate module is combined with DAM to form DAB. The average pooling, the max pooling, the asymmetric convolution, and 1×1 and 3×3 convolution layers are stacked together by using a parallel structure, and then the feature maps of each branch are fused by concat to ensure the integrity of the output feature map. Among them, 1×1 convolution changes the number of the channel layers of the input feature map, and the pooling layer changes the dimension of the input feature map. The Incep-abate module is an asymmetric convolution structure, and $N\times N\times C$ convolution is replaced with $1\times N\times C$ convolution and $N\times 1\times C$ convolution to reduce the hardware cost and improve the computing efficiency. The structure of the Incep-abate module is shown in Figure 10.

2.2.3. HE-Yolo.

HE-Yolo (high effection-Yolo) is built by Darknet-A and DAB, as shown in Figure 11. Darknet-A (i.e., the dotted box 1 in Figure 11) is used to compress the calculation of the feature map to reduce the occupation of hardware resources and ensure that feature information is not distorted. DAB (i.e., the dotted box 2 in Figure 11) obtains the coefficient matrix of the effective features through the operations such as the global max pooling, the global average pooling, and the sigmoid activation function to filter the features of the input feature map. HE-Yolo constructs a feature pyramid to achieve multi-scale feature fusion and improves the context semantics of the feature map through the intersection of multi-layer feature information. At the same time, nonmaximum suppression is applied to remove the detection frames with large deviations, so that each target corresponds to a unique detection frame.

The anchor box size of Yolo v3 is obtained by clustering on the coco data set, which is different from the posture behavior size of pigs. Therefore, a new anchor box size is set

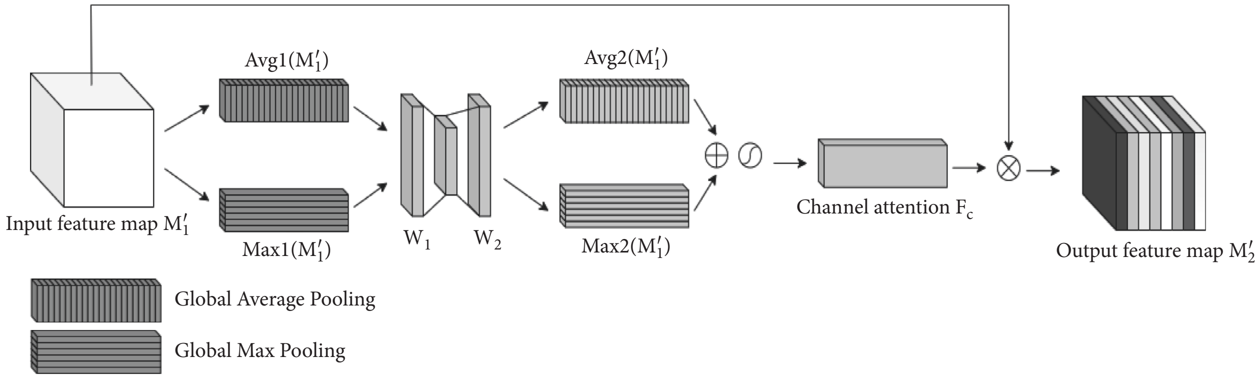


FIGURE 7: The structure of CAM module.

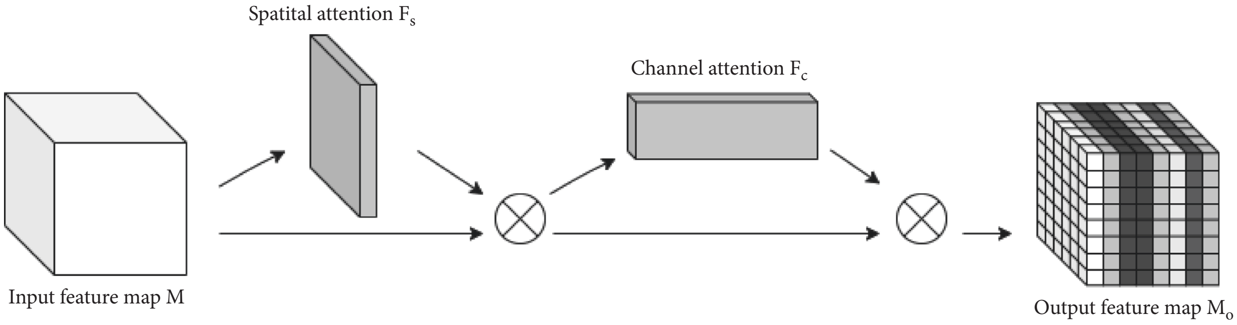


FIGURE 8: The structure of DAM module.

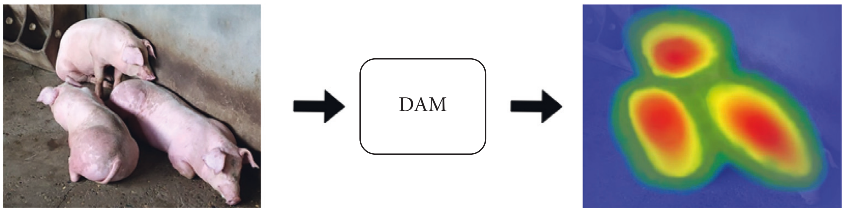


FIGURE 9: Visualization of DAM feature extraction of pigs.

to improve the training effect of HE-Yolo. The K-means algorithm is used to cluster and optimize the labeled images to obtain the anchor box sets of three sizes of the pig image data set, as listed in Table 1.

3. Results and Analysis

3.1. Experimental Environment. Intel (R) core (TM) i7-6700 server with 32 GB memory and equipped with windows 10 operating system. Pychram integrated development environment and TensorFlow open-source library for parallel computing on GPU. GPU is configured as Nvidia GeForce RTX 5000 and GPU memory is 16 GB. The parallel computing environment is CUDA 9.0, cuDNN v7.4.2 and the programming language is Python 3.6.0. There are 2040 images in the training set. The small batch gradient descent training method is adopted and the number of iterations is set to 100. The initial learning rate is set to 0.001 and the learning rate is adjusted by the epoch-decay strategy. When the Intersection over Union (IoU) of the detection frame is greater than the threshold of 0.6, the recognition result is considered correct. Figure 12 is the change curve of Loss with the number of iteration epochs during the training process of HE-Yolo. It can be seen that the Loss decreases rapidly at the beginning of the iteration, and it is stable between 3 and 4 after 65 iteration epochs.

3.2. The Evaluation Parameters. The precision rate *P*, the recall rate *R*, the average precision AP and the mean average precision mAP are selected as the parameters to evaluate the

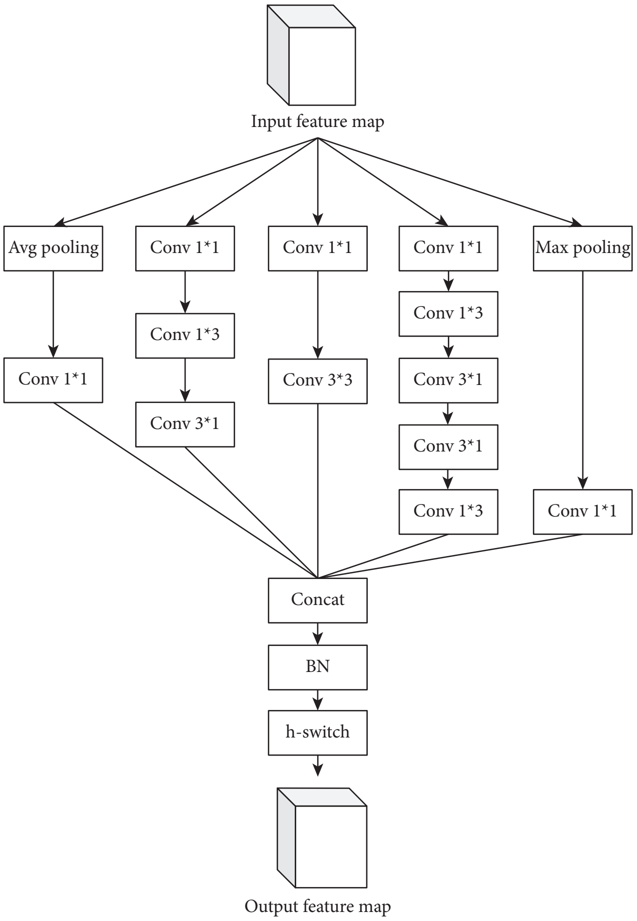


Figure 10: The structure of the Incep-abate module.

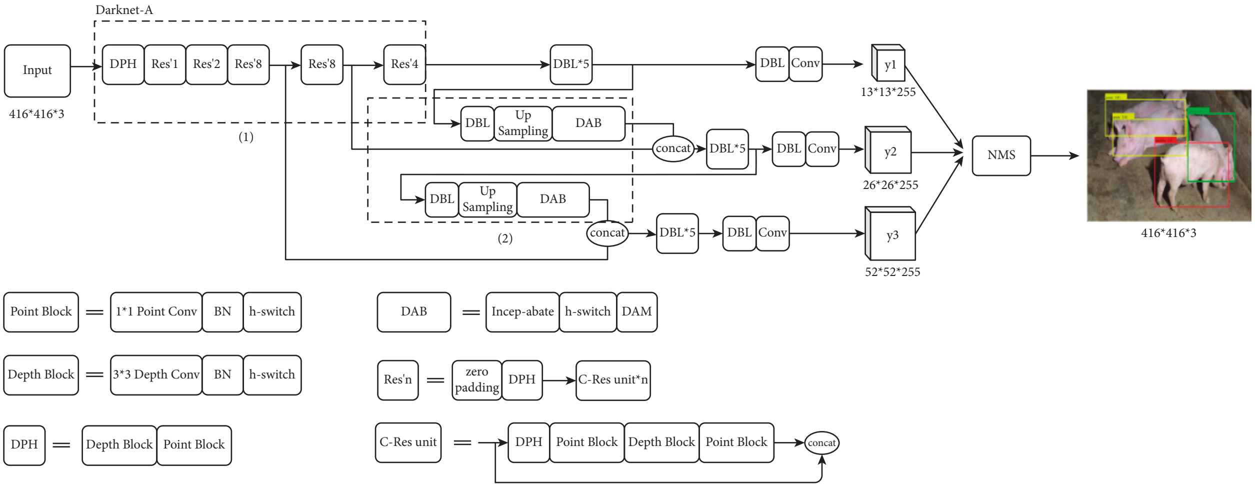


Figure 11: The structure of HE-Yolo.

Table 1: The anchor box size and the corresponding output feature map after K-means clustering.

Size	13 × 13	26 × 26	52 × 52
Anchor box	(171 × 142), (168 × 114), (257 × 283)	(86 × 145), (105 × 70), (122 × 106)	(46 × 50), (72 × 93), (80 × 49)
Receptive field	Big	Middle	Small

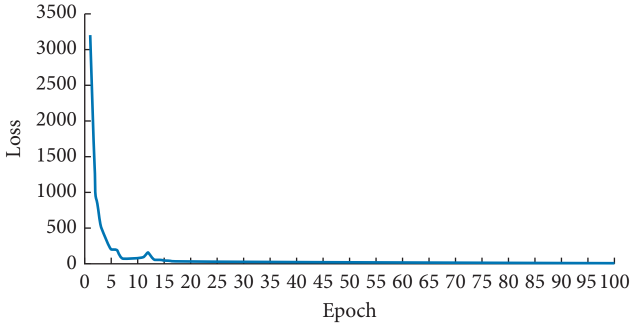


Figure 12: The change curve of Loss with the number of iterations.

recognition performance of HE-Yolo. They are calculated as follows:

$$R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times 100\%,$$
$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \times 100\%,$$
$$\mathrm{AP} = \int_0^1 \mathrm{PR}dr, \tag{6}$$
$$\mathrm{mAP} = \frac{1}{N}\sum_{N_i \in N} \mathrm{AP}_{N_i},$$

where TP is the number of samples correctly recognized by the model, FP is the number of samples that the model misjudges as positive but negative, FN is the number of samples that the model misjudges as negative but positive, N is the category number of the pig-posture behaviors, N_i is the i th category and $i \in [1, 4]$, AP is the area of the P-R curve.

3.3. Selecting the Optimal Model. A total of 436 verification set images were used in 100 test stages, and the optimal model for all stages was selected. First of all, it is necessary to determine the priority of all evaluation parameters. The priority of evaluation parameters from small to large is R, P, and mAP. During the training process, the change curves of evaluation parameters with the number of iteration epochs are shown in Figure 13.

It can be seen from Figure 13 that the evaluation parameters grow fast in the first 70 epochs, and they are stable in the last 30 epochs. The mAP reaches the maximum value of 97.43% when the epoch is 96, and P reaches the maximum value of 96.79% when the epoch is 94. R reaches the maximum value of 97.64% when the epoch is 89. Therefore, HE-Yolo after 96 iteration epochs is selected as the optimal model to recognize the pig-posture behaviors.

3.4. Experimental Results and Analysis

3.4.1. Analysis on the Recognition Results of HE-Yolo. Remove the SAM block in DAB to build the CA-Yolo, and remove the CAM block in DAB to build the SA-Yolo. The SSD [29] uses VGG16 as the backbone network, and the faster R–CNN uses the region generation network to replace the selective search in the fast R–CNN. Six models including HE-Yolo, CA-Yolo, SA-Yolo, Yolo v3, SSD and faster R–CNN are used to recognize four kinds of pig-posture behaviors, and the recognition results are listed in Table 2.

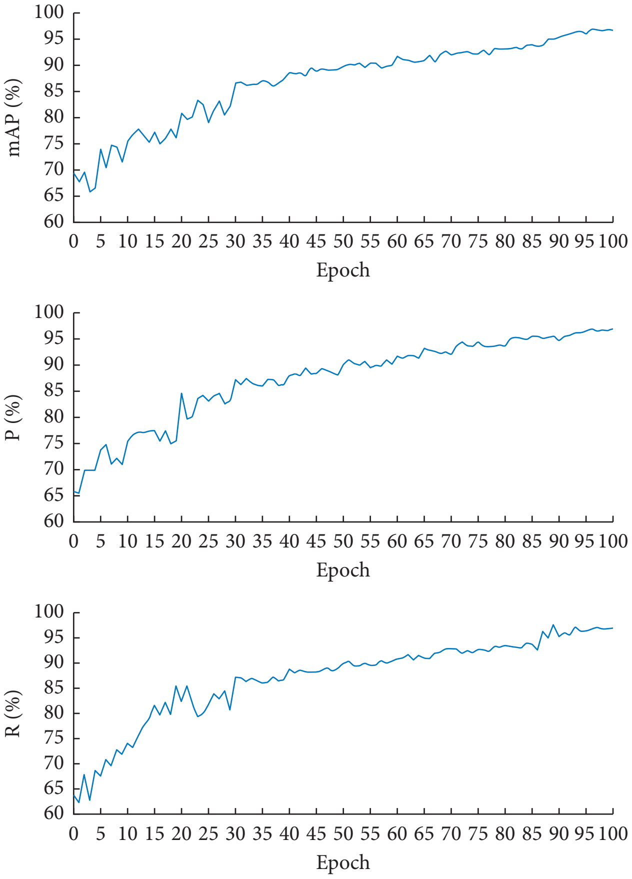


Figure 13: The change curves of P, R, and mAP.

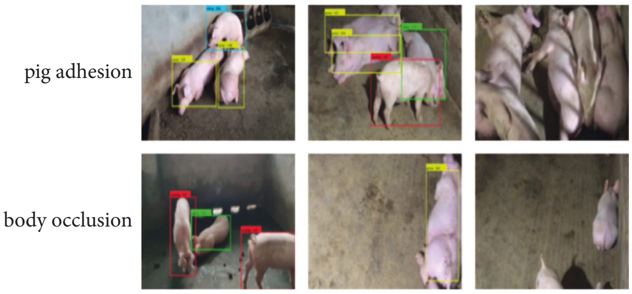
From Table 2 it can be known that:

(1) Compared with Yolo v3, the AP values of HE-Yolo, CA-Yolo, and SA-Yolo for four postures are increased by 0.63%~1.39%, 0.24%~2.33%, 0.07%~5.68%, and 0.48%~4.04%, respectively, and the mAP values of three models are increased by 1.73%~5.61%. That is, the introduction of an attention mechanism can improve the AP value of the model. The mAP value of SA-Yolo is higher than that of CA-Yolo, which shows that the spatial attention mechanism has an important influence on the recognition performance of the detection model. The AP values of a single category and the overall mAP value of HE-Yolo are higher than that of CA-Yolo and SA-Yolo, which shows that DAM can weaken the influence of noise on the recognition result of the detection model.

(2) The AP values of HE-Yolo are higher than that of SSD and faster R–CNN, and its map value is also increased by 4.64% and 0.57%, respectively.

Table 2: The recognition results of the pig-posture behaviors of the test set.

Model	AP (%)				mAP (%)
	Standing	Sitting	Prone	Sidling	
HE-Yolo	99.25	98.41	94.43	97.63	97.43
Yolo v3	97.86	96.08	88.75	93.59	91.82
CA-Yolo	98.49	96.32	88.82	94.07	93.55
SA-Yolo	98.96	96.68	91.01	95.43	95.52
SSD	97.55	94.39	86.32	92.86	92.78
Faster R–CNN	98.63	97.74	93.82	97.25	96.86



(a) mild (b) medium (c) severe

Figure 14: The recognition results of HE-Yolo for different degrees of foreign body occlusion and pig adhesion. (a) Mild. (b) Medium. (c) Severe.

Table 3: Comparison of the single-frame recognition time.

Model	The single-frame recognition time (s)
HE-Yolo	0.045
Yolo v3	0.038
SSD	0.051
Faster R–CNN	0.175

Compared with Yolo v3, the single-frame recognition time of HE-Yolo increases by 0.007 s, but its map value increases by 5.61%. It can be seen from Table 2 that the mAP value of HE-Yolo is slightly higher than that of faster R–CNN, and from Table 3 it can be seen that the single-frame recognition time of HE-Yolo is reduced by 0.13 s.

Therefore, HE-Yolo has high recognition accuracy and fast recognition speed, which shows that HE-Yolo has strong mobility, and it can fully meet the real-time recognition requirement for the pig-posture behaviors.

3.4.2. Recognizing the Images with Foreign Body Occlusion and Pig Adhesion. When the foreign body occlusion and pig adhesion are serious, the model is difficult to recognize the pig-posture behaviors. Figure 14 shows the recognition results of HE-Yolo on the pig images with different degrees of foreign body occlusion and pig adhesion. Among them, there is partial foreign body occlusion and pig adhesion in Figures 14(a)~14(b), and HE-Yolo can successfully recognize the pig-posture behaviors. There is serious foreign body occlusion and pig adhesion in Figure 14(c), and HE-Yolo cannot successfully recognize the pig-posture behaviors.

Table 4: Recognition results of different models for foreign body occlusion and pig adhesion.

Model	HE-Yolo	Yolo v3	SSD	Faster R–CNN
mAP (%)	85.72	81.68	80.78	83.96

Table 5: The mAP values (%) of each model under different lighting conditions.

Type	Bright	Dark
Model		
HE-Yolo	98.12	96.88
Yolo v3	93.76	90.36
SSD	93.41	91.45
Faster R–CNN	97.21	95.53

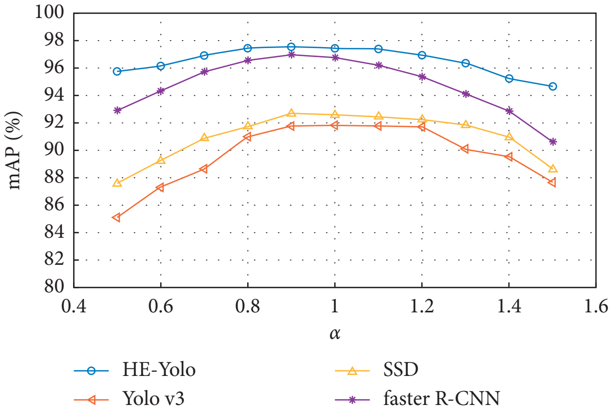


Figure 15: The recognition results of different models.

A total of 280 images with foreign body occlusion and pig adhesion are selected as the testing set, and the mAP value is used as the evaluation parameter. Four models including HE-Yolo, Yolo v3, SSD, and faster R–CNN are tested, and the experimental results are listed in Table 4.

It can be seen from Table 4 that the mAP value of HE-Yolo is 4.04%, 4.94%, and 1.76% higher than that of Yolo v3, SSD, and fast R–CNN, respectively. According to the actual observation and the recognition results, when HE-Yolo simultaneously extracts the feature information of three areas

Table 6: The comparison results of different methods.

Model	HE-Yolo	Literature [3]	Literature [7]	Literature [12]
mAP (%)	97.43	97.17	96.33	96.86
The training convergence epoch	65	74	86	58
The single-frame recognition time (s)	0.045	0.075	0.028	0.175

of the pig-limb, pig-abdomen, and pig-trunk, and the area covered by foreign body and pig adhesion is less than 25%, then HE-Yolo can effectively recognize the pig-posture behaviors.

3.4.3. Analyzing the Influence of Lighting Condition on the Recognition Results. The images in the test set are divided into bright and dark according to the lighting conditions. Four models including HE-Yolo, Yolo v3, SSD, and faster R–CNN are used for the experiments, and the experimental results are listed in Table 5. It can be seen from Table 5 that the mAP values of HE-Yolo are the highest and its mAP values is only 1.24% different in two lighting conditions, which shows that HE-Yolo has good recognition performance under different lighting conditions.

The light varies greatly in the enclosure environment. To further test the robustness of HE-Yolo to the luminance of light, the RGB image is converted into an HSV image [30], and the luminance coefficient α of 10 gradients is set and $\alpha \in \{0.5, 0.6, 0.7 \ldots 1.5\}$. 10 luminance coefficients are multiplied by the luminance V, respectively, and then converted into RGB images to obtain the testing set with 10 different luminances. The experimental results of four models are shown in Figure 15.

It can be seen from Figure 15 that the stability of the mAP value of HE-Yolo is significantly better than that of other models, and the mAP values under 10 luminance conditions are higher than those of other models, which shows that HE-Yolo has stronger generalization ability and better robustness to the luminance of light, so it can meet the real-time recognition requirement of day and night alternation during the breeding process.

HE-Yolo is compared with other methods [3, 7, 12], and the mAP value, the training convergence epoch, and the single-frame recognition time are tested, respectively. The comparison results are listed in Table 6.

It can be seen from Table 6 that the mAP value of HE-Yolo is higher than that of other methods, and it converges after 65 iteration epochs, which shows that HE-Yolo needs lower hardware and less computation time, and its recognition speed is slightly lower than that of reference [7], but it can fully meet the real-time recognition.

4. Conclusion

In this study, HE-Yolo is built for real-time recognition of the pig-posture behaviors in the enclosure environment. HE-Yolo further optimizes the Yolo v3 model in structure, ensures the recognition effect and further reduces the computational complexity, and makes the new model more capable of focusing on effective features by constructing a multiple attention mechanism. The experimental results show that HE-Yolo has better accuracy and anti-interference ability than other models in a complex environment and provides a reliable research method for pig detection. The experiment results are summarized as follows:

The anchor boxes of Yolo v3 are optimized and clustered by the K-means algorithm. The depthwise separable convolution and h-switch activation function are introduced into Darknet-53 to build DBH, and C-Res unit with reverse inverse residual structure is designed to build Darknet-A feature extraction network to avoid network gradient explosion and ensure the integrity of image feature information.

According to the contribution of the feature information of different areas in the feature image, the spatial attention mechanism, and the channel attention mechanism are integrated to construct DAM, and the average pooling, the maximum pooling, the asymmetric convolution, 1×1 pointwise convolution and 3×3 depthwise convolutions together to form the Incep-abate module. DAM and the Incep-abate module are further combined to form DAB to reduce the computational complexity. DAB is spliced with the feature pyramid structure of Yolo v3 to make the model focus more on effective feature information, and HE-Yolo is finally built by Darknet-A and DAB.

Compared with Yolo v3, the mAP values of HE-Yolo, CA-Yolo, and SA-Yolo are increased by 5.61%, 1.73%, and 3.70%, respectively, which shows that the introduction of the attention mechanism can improve the extraction effect of effective features of the pig images. Among them, the mAP value of SA-Yolo is 1.97% higher than that of CA-Yolo, which shows that the spatial attention mechanism has an important influence on the recognition performance of the model.

The single-frame recognition time of HE-Yolo is 0.045 s, and its image recognition performance for foreign body occlusion and pig adhesion is good. The mAP value of HE-Yolo is 4.04%, 5.04% and 1.76% higher than that of Yolo v3, SSD, and faster R–CNN, and the mAP value of HE-Yolo is also higher than that of other models under different lighting conditions. Compared with other methods [3, 5, 7], HE-Yolo has high recognition accuracy and fast recognition speed under the condition of low hardware.

To sum up, the Yolo model designed in this paper has stronger generalization ability, can meet the real-time detection requirements of pig posture and behavior in the actual captive environment, has higher recognition accuracy and speed than other models, and has good robustness. Of course, HE-Yolo will be further optimized to improve its recognition performance and accurately track the pig-posture behaviors of individuals. In the next study, it cannot be limited to the abovementioned four posture behaviors, and the image data can be expanded for other types of pig-posture behavior or even be extended to other livestock.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

The authors Li Huang and Lijia Xu contributed equally to this work.

Acknowledgments

This research was funded by the Sichuan Province Department of Science and Technology, grant number: 2022058; National Innovation Training Program, grant number: 202110626044.

References

[1] J.-H. Su, Y.-C. Piao, Z. Luo, and B.-P. Yan, "Modeling habitat Suitability of Migratory Birds from Remote Sensing images using convolutional neural networks," *Animals*, vol. 8, p. 66, 2018.

[2] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, "Blockchain-based Cross-domain Authentication for intelligent 5G-enabled Internet of Drones," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6224–6238, 2022.

[3] A. K. Schütz, V. Schöler, E. T. Krause et al., "Application of YOLOv4 for detection and Motion monitoring of red Foxes," *Animals*, vol. 11, no. 6, p. 1723, 2021.

[4] C. Liu, J. Su, L. Wang, S. Lu, and L. Li, "LA-DeepLab V3+: a Novel Counting network for pigs," *Agriculture*, vol. 12, no. 2, p. 284, 2022.

[5] L. Du, Y. Sun, S. Chen et al., "A Novel object detection model based on faster R-CNN for Spodoptera frugiperda according to Feeding Trace of Corn Leaves," *Agriculture*, vol. 12, no. 2, p. 248, 2022.

[6] Y. Gao, B. Chen, H. Liao et al., "Recognition method for aggressive behavior of group pigs based on deep learning [J]," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 35, no. 23, pp. 192–200, 2019.

[7] Z. Zhou, X. Dong, Z. Li, K. Yu, C. Ding, and Y. Yang, "Spatio-temporal feature Encoding for Traffic Accident detection in VANET environment," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2022.

[8] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A Lightweight Attention-based convolutional neural networks for Tomato Leaf Disease Classification," *Agriculture*, vol. 12, no. 2, p. 228, 2022.

[9] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[10] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network:learning of attention mechanism for visual Explanation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10697–10706, IEEE, Long Beach, CA, USA, June 2019.

[11] Q. Yang, D. Xiao, and G. Zhang, "Automatic pig drinking behavior recognition with machine vision[J]," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 49, no. 6, pp. 232–238, 2018.

[12] C. Zheng, X. Zhu, X. Yang, L. Wang, S. Tu, and Y. Xue, "Automatic recognition of lactating sow postures from depth images by deep learning detector," *Computers and Electronics in Agriculture*, vol. 147, pp. 51–63, 2018.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature Hierarchies for accurate object detection and semantic Segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, IEEE, Columbus, OH, USA, June 2014.

[14] R. Girshick, "Fast r-cnn[C]," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, IEEE, Santiago, Chile, December 2015.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks[J]," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only Look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, IEEE, Las Vegas, NV, USA, June 2016.

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, IEEE, Honolulu, HI, USA, July 2017.

[18] J. Redmon and A. Farhadi, "YOLOv3: an Incremental Improvement," 2018, http://10.48550/arXiv.1804.02767.

[19] D. Tzutalin, *Label Img*, GitHub Repository, 2015, https://github.com/tzutalin/labelImg.

[20] V. C. Raykar and A. Saha, "Data Split Strategies for Evolving predictive models[C]," *Machine Learning and Knowledge Discovery in Databases*, vol. 3, p. 19, 2015.

[21] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression[C]," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, pp. 850–855, IEEE, Hong Kong, China, August 2006.

[22] A. G. Howard, M.-long Zhu, and Bo. Chen, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, arXiv preprint https://arxiv.org/abs/1704.04861v1.

[23] Y. Peng, A. Jolfaei, and K. Yu, "A Novel real-time Deterministic Scheduling mechanism in Industrial Cyber-Physical Systems for Energy Internet," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5670–5680, 2022.

[24] F. Wang, M. Jiang, C. Qian et al., "Residual attention network for image Classification," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458, IEEE, Honolulu, HI, USA, July 2017.

[25] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered Digital Forensics for Edge consumer Electronics in 5G HetNets," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 42–50, 2022.

[26] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014, http://arxiv.org/abs/1312.4400.

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the Computer Vision and Pattern Recognition*

(CVPR), pp. 2921–2929, IEEE Computer Society, Piscataway, NJ, June 2016.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, pp. 4278–4284, AAAI Press, 2017, http://10.48550/arXiv.1602.07261.

[29] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision – ECCV 2016*, vol. 10, pp. 21–37, 2016.

[30] K. Yu, L. Tan, C. Yang et al., "A Blockchain-based Shamir's threshold Cryptography Scheme for data Protection in Industrial Internet of Things Settings," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8154–8167, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4451792, 9 pages
https://doi.org/10.1155/2022/4451792

Research Article

Blended Ensemble Learning Prediction Model for Strengthening Diagnosis and Treatment of Chronic Diabetes Disease

T. R. Mahesh,[1] **Dhilip Kumar**,[2] **V. Vinoth Kumar**,[1] **Junaid Asghar**,[3] **Banchigize Mekcha Bazezew**,[4] **Rajesh Natarajan**,[5] **and V. Vivek**[1]

[1]*Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bangalore, India*
[2]*Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India*
[3]*Faculty of Pharmacy, Gomal University, Dera Ismail Khan 29050, Khyber Pakhtunkhwa, Pakistan*
[4]*Department of Electrical and Computer Engineering, Wollo University, Ethiopia*
[5]*Department of Information Technology, University of Technology and Applied Science, Shinas. Sultanate of Oman, Oman*

Correspondence should be addressed to Banchigize Mekcha Bazezew; banchigizeb@kiot.edu.et

Received 22 May 2022; Accepted 24 June 2022; Published 14 July 2022

Academic Editor: Muhammad Ahmad

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

Diabetes mellitus (DM), commonly known as diabetes, is a collection of metabolic illnesses characterized by persistently high blood sugar levels. The signs of elevated blood sugar include increased hunger, frequent urination, and increased thirst. If DM is not treated properly, it may lead to several complications. Diabetes is caused by either insufficient insulin production by the pancreas or an insufficient insulin response by the body's cells. Every year, 1.6 million individuals die from this disease. The objective of this research work is to use relevant features to construct a blended ensemble learning (EL)-based forecasting system and find the optimal classifier for comparing clinical outputs. The EL based on Bayesian networks and radial basis functions has been proposed in this article. The performances of five machine learning (ML) techniques, namely, logistic regression (LR), decision tree (DT) classifier, support vector machine (SVM), *K*-nearest neighbors (KNN), and random forest (RF), are compared with the proposed EL technique. Experiments reveal that the EL method performs better than the existing ML approaches in predicting diabetic illness, with the remarkable accuracy of 97.11%. The proposed ensemble learning could be useful in assisting specialists in accurately diagnosing diabetes and assisting patients in receiving the appropriate therapy.

1. Introduction

Diabetes mellitus (DM) is a collection of metabolic disorders indicated by abnormal insulin production as well as its action. This insulin deficiency leads to elevated blood levels of glucose and decreased carbohydrate and protein metabolism [1]. DM is a prevalent endocrine disease that affects roughly 200 million people across the world. Diabetes affected 30.3 million persons in the United States in 2015, accounting for 9.4% of the population. More than one-fourth of them were unaware that they had this condition. One out of four people over the age of 65 has diabetes [2]. Gradually, the human living environment changes dramatically due to increased demand on the health of human beings, human lifestyle, changes in climate, and other environmental changes. Infections and diseases cause millions of individuals to fall unwell and face other harmful health issues, and these problems are quickly increasing every day.

DM is a chronic condition that happens when the pancreas stops producing insulin or when the body's insulin is not used properly. Insulin is a hormone produced by the pancreas that functions as a key to allowing glucose from our diet to move from the bloodstream into our cells for energy production. In the blood, all carbohydrate foods are broken down into glucose. Insulin aids glucose absorption into cells. When the body is unable to make or utilize insulin properly, blood glucose levels rise (known as hyperglycaemia). High glucose levels are linked to long-term harm to the body and the failure of many organs and tissues. Diabetes that is not well controlled can have catastrophic repercussions, including damage to a variety of organs and tissues in the body, including the heart, kidneys, eyes, and nerves. The different symptoms of the DM include frequent hunger, frequent thirst, frequent urination, urinary and genital infections, extreme tiredness, wounds that will not heal, and unexplained weight loss especially in case of children. Because many people do not experience any of these symptoms, diabetes is frequently diagnosed by chance. Those who are at risk of acquiring diabetes should have regular medical examinations to avoid late discovery of the disease. The different symptoms of diabetes are shown in Figure 1.

There are three main kinds of diabetes. Due to the complexity of diabetes, diagnosing and distinguishing among these different kinds are difficult [3–5].

1.1. Type-1. An autoimmune reaction occurs when the body's defense system assaults the cells that make insulin, resulting in type-1 diabetes. As a result, the body produces very little insulin. The specific causes are unknown although they are thought to be connected to a combination of genetic and environmental factors. Type-1 diabetes can strike anyone at any age although it strikes children and adolescents the most commonly. When a person has type-1 diabetes, his or her body generates very little or no insulin, necessitating daily insulin injections to keep blood glucose levels in check.

1.2. Type-2. Type-2 diabetes is the most common form of diabetes, accounting for approximately 90% of all cases. Insulin resistance is a condition in which the body's response to insulin is inadequate. Because insulin is unable to function effectively, blood glucose levels rise, causing more insulin to be released. This can gradually fatigue the pancreas, resulting in the body producing less and less insulin, further raising blood sugar levels (hyperglycaemia). Type-2 diabetes is most typically diagnosed in older persons although it is becoming more common in children, adolescents, and younger adults as obesity, physical inactivity, and poor diet become more prevalent. The major complications that might be caused by type-2 diabetes are depicted in Figure 2.

Diabetes has a dangerous effect on the eyes' small blood vessels. As a result, it may cause problems such as glaucoma (a condition in which fluid builds up in the eyes), cataracts (clouding of the lens of the eyes), and diabetes retinopathy (a condition in which the retina at the back of the eye is damaged). Diabetes can cause visual loss or blur over time. As a result, regular eye examinations by a qualified ophthalmologist are required. Early detection can help to avoid serious consequences. According to polls, early identification can prevent blindness in patients with diabetes by 90 percent. Figure 3 shows the normal eye and eye with retinopathy.

1.3. Gestational Diabetes Mellitus (GDM). GDM is a serious and underappreciated threat to the health of mothers and children. High blood pressure, large birth weight babies, and obstructed labour are common pregnancy problems for women with GDM. Within five to ten years following delivery, around half of women with a history of GDM acquire type-2 diabetes. The prevalence of high blood glucose during pregnancy rises dramatically with age, peaking in women over 45. However, diabetes mellitus, most usually type 2, is discovered in roughly 5–10% of women with GDM after pregnancy. GDM is completely curable, but it necessitates close medical monitoring during pregnancy.

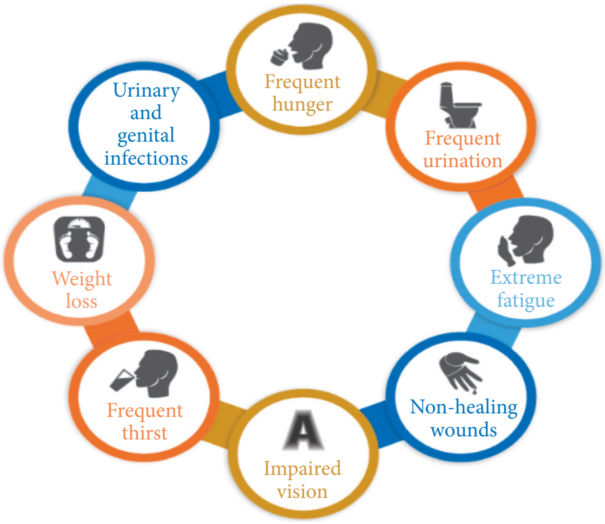


Figure 1: Diabetes' symptoms.

2. Related Work

The utilization of ML approaches to improve the precision of DM risk prediction has been studied in a variety of ways [6]. The authors [7] have done a comparison of the accuracy of numerous algorithms on the dataset of diabetes. They discovered that the J48 algorithm has 73.82% accuracy before data preprocessing, which is superior to others. Following preprocessing, both KNN and RF showed better performance. In this work [8], J48, KNN, and LR were compared on the diabetes dataset in this work. J48 was found to be the most accurate, with a classification accuracy of 78.27%. Based on its accuracy of 80%, this work [9] created web-based software for DM detection. In this work, the authors compared prediction techniques such as decision trees (DT), neural networks, Naive Bayes (NB), LR, and RF, and ensemble techniques. The authors found that RF performed the best in terms of accuracy and ROC score of 75.558% and 0.912, respectively.

The authors in this [10] looked at different ML algorithms on a medical dataset. They compared different performance indicators. In this research, ML and data mining approaches used in DM research were identified and reviewed in a systematic manner. Diabetes mellitus is quickly becoming a very important and pressing worldwide health issue of the twenty-first century. At present, significant research has been conducted in practically every facet of DM research. When compared to other ML algorithms, authors [11] employed

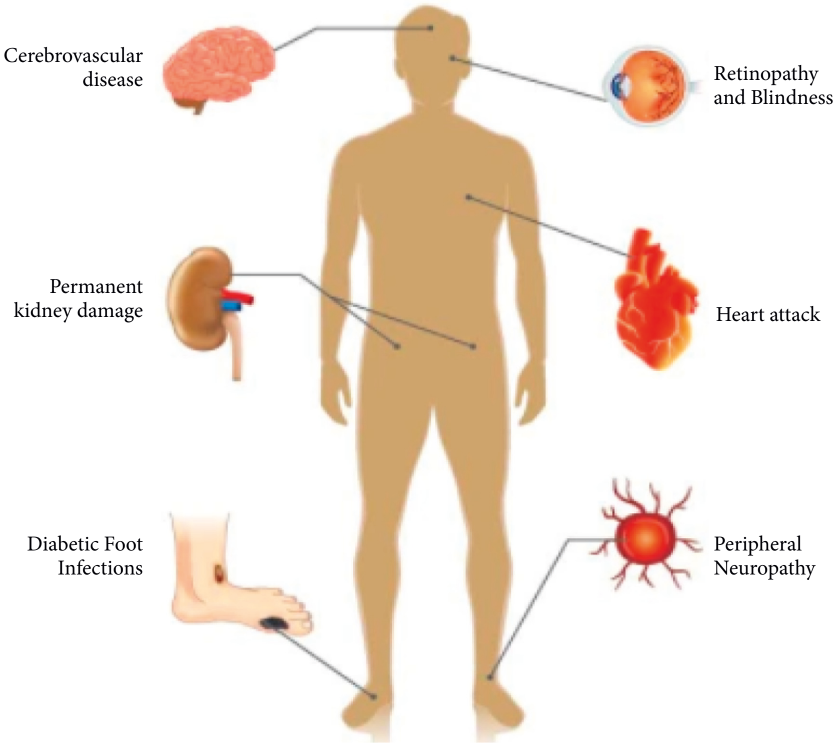


Figure 2: Type-2 diabetes complications.

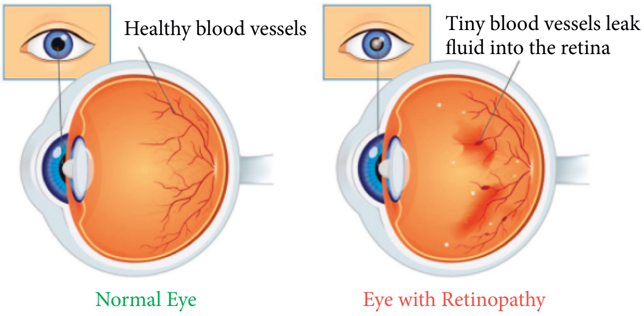


Figure 3: Diabetes retinopathy.

C4.5 with 78.25% accuracy. According to them, J48 is famous for its correctness.

The authors of this article [12] present an overview of how DM is linked to cardiovascular autonomic neuropathy that impairs HRV. As a result, diabetes was discovered using HRV spectral analysis. Using DWT features retrieved from HRV signals, the authors demonstrated an automated DM detection technique. Using the DT classifier, they were able to get sensitivity, accuracy, and specificity of 92%, 92.59%, and 91.46%, respectively, using their proposed method. In their diabetes forecasting experiments, the authors looked into NB, ANN, as well as SVM classifiers [7]. They conducted a research study that was weighted. The majority vote method was used in this investigation. The findings show that a combination of classifiers outperformed any individual classifier.

The authors of this study [13] examined different ML algorithms on the Pima Indian dataset. Feature selection was applied to the dataset. According to their findings, attribute selection improved the performance of diabetes prediction. This study's retrospective cross-sectional methodology prevents the authors from establishing a cause-and-effect link. This strategy may simply be utilized in studies to find the optimal type-2 diabetes phenotype or prediction in different nations. However, because the study group was limited to Korean women and men, the findings could not be extended to other populations. There can be disparities in socioeconomic class, race, gender, and nationality within a single patient cohort.

Using the J48 DT, the authors [6] emphasized the application of Adaboost ensemble methods for diagnosing the disease. According to the findings of the trial, Adaboost ensemble approaches outperform J48 and even bagging. In this study, the scientists employed NB, J48, and RF for the prediction of DM. In terms of accuracy, they discovered that RF outperforms naive Bayes and J48 techniques. According to the authors [14], the accuracy achieved by implementing PCA was not good, but the outcome of using all features along with MRMR was superior. The results showed that by using fasting glucose made the performance better, particularly in the Luzhou dataset. This implies that, while fasting glucose is the most significant predictor, it cannot produce the best results on its own, so multiple indices were required to effectively predict. Furthermore, they discovered that there is little difference between RT, DT, and neural network results when comparing the three classifications; however, RFs are definitely better compared to the other three classifiers. The highest performance for the Luzhou dataset was 80.84%, while that of the Pima Indians dataset was 77.21%, indicating that ML can be used to predict diabetes, but selecting appropriate features, classifiers, and data mining methods is crucial. According to the results, the Adaboost ensemble technique outperforms bagging and the standalone J48 DT in this study [15].

Several ML algorithms for diabetes mellitus prediction were utilized in the studies listed above, and they have been evaluated for getting good results. In this article, the ML techniques DT, RF, LR, SVM, and KNN classifiers are utilized as base learners, and their performance is compared to that of the ensemble classifier using various performance measures, namely, accuracy, precision, sensitivity, and f1-measure in the prediction of diabetes.

3. Methodology

The likelihood of survival and the likelihood of diabetes recurrence are highly dependent on medical treatment and the accuracy of the diagnosis. The various phases of the prediction process of diabetes are depicted in Figure 4.

The dataset taken from Kaggle consists of 693 diabetes-affected people's records. The different symptoms along with their descriptions are shown in Table 1. Arbitrarily extracted information was used in this investigation, with a ratio of 75 : 25 split between training and testing data. The model was trained to utilize training sets (520 records), and its effectiveness was tested via test data. The dataset consists of 16 features or attributes whose values will determine whether the person is likely to be affected by diabetes or not. The target or output variable has a value of 1 for the presence of diabetes and 0 for its absence.

3.1. Logistic Regression (LR). This approach has been modified for binary classification problems. The fundamental goal of LR is to figure out what the coefficients are worth. The LR reduces the value to a number between 0 and 1. The LR model chooses a probability of 0 or 1 for the provided data instance of the class to predict. This method can be used to solve problems where we have various reasons to forecast. Equation (1) is the definition of the LR standard function:

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}. \qquad (1)$$

The constants are represented in β_1 and β_0 by the data label X.

3.2. K-Nearest Neighbor (KNN). KNN is also a supervised ML algorithm [11]. It is mostly used to solve categorization challenges. The object is categorized using K-neighbors. Before running the procedure, the positive number K must be defined. Euclidean distance is frequently used to determine various object measurements [16]. The Euclidean distance and Manhattan distance are calculated using the following equations:

$$\text{Euclidean} = \sqrt{\sum_{i=1}^{k} (a_i - b_i)^2}, \qquad (2)$$

$$\text{Manhattan} = \sum_{i=1}^{k} |a_i - b_i. \qquad (3)$$

From the above equations, Euclidean as well as Manhattan of KNN is computed with a and b data upto I variables.

3.3. Support Vector Machine (SVM). The SVM algorithm [17] is a supervised ML approach. For a small data set with a few outliers, this model is ideal. The goal is to determine the hyperplane that will be used to split the data points. The hyperplane will be used to divide two spaces into different domains. Similar forms of data will be found in such a domain. The support vector machine's decision state is represented by the following equation:

$$||Y|| = \sqrt{y_1^2 + y_1^2 + \cdots + y_n^2}. \qquad (4)$$

3.4. Decision Tree (DT). DT is one technique to show an algorithm that is made up of completely conditional control statements. The DTs are a collection of well-known supervised classification algorithms. They do well on classification problems; the decisional path is straightforward to interpret, and the algorithm for building (training) them is quick and very simple. A single DT is depicted in Figure 5.

3.5. Random Forest (RF). RF is an ML technique that is part of the supervised ML model. The RF classifier is made up of numerous decision trees representing various subjects [18]. It takes the average of each tree's subset to improve predictive accuracy. RF, rather than depending on a single decision tree, uses the majority vote prediction technique from all the trees and then predicts the result [19]. Every node in the decision tree answers a query about the situation. RF example is depicted in Figure 6.

3.6. Proposed Ensemble Learning (EL) Technique. For the categorization of diabetes mellitus, an ensemble classifier based on Bayesian Networks and Radial Basis Functions is proposed in this article [20]. Ensemble classifiers are modelled using Bayesian networks (BNs) as well as Radial Basis Function (RBF). Two individual classifiers and the suggested ensemble classifier will be discussed in the subsections. Figure 7 depicts the suggested approach's process flow diagram.

3.6.1. Bayesian Network (BN). Traditional statistical models do not allow for the incorporation of past knowledge into the model. The probabilistic associations between items or objects are represented using a BN [21]. BNs are directed acyclic graphs with nodes indicating variables and missing edges representing conditional independence between

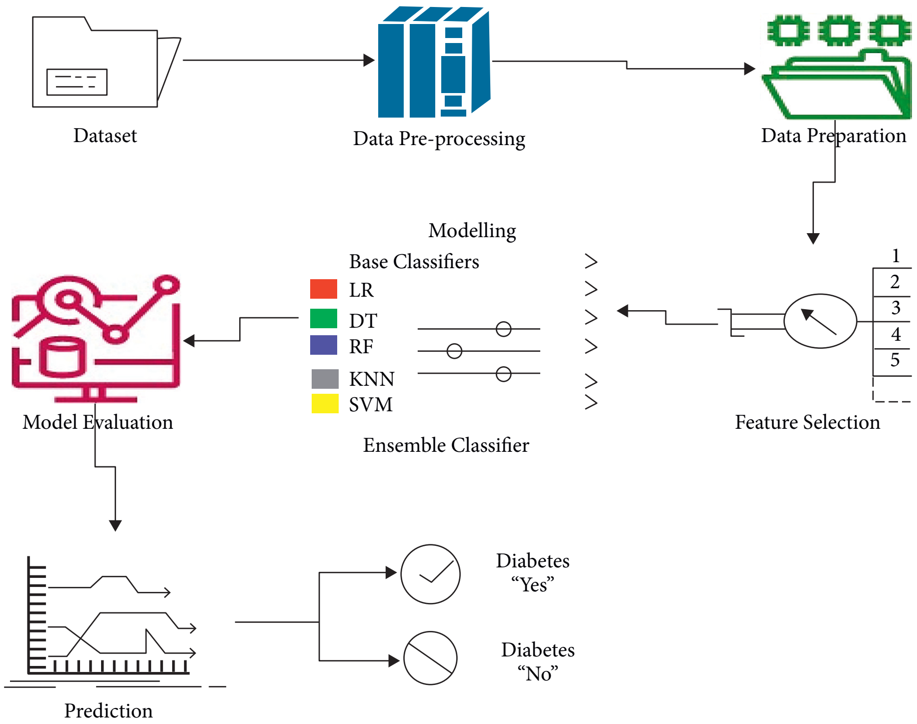


Figure 4: Different phases of the prediction process.

Table 1: Feature descriptions.

Attribute/feature	Description
Age	Between 20 years and 90 years
Gender	1 denotes male, 0 denotes female
Polyuria	1 denotes male, 0 denotes female
Polydipsia	1 denotes male, 0 denotes female
Sudden weight loss	1 denotes male, 0 denotes female
Weakness	1 denotes male, 0 denotes female
Polyphagia	1 denotes male, 0 denotes female
Genital thrush	1 denotes male, 0 denotes female
Visual blurring	1 denotes male, 0 denotes female
Itching	1 denotes male, 0 denotes female
Irritability	1 denotes male, 0 denotes female
Delayed healing	1 denotes male, 0 denotes female
Partial paresis	1 denotes male, 0 denotes female
Muscle stiffness	1 denotes male, 0 denotes female
Alopecia	1 denotes male, 0 denotes female
Obesity	1 denotes male, 0 denotes female
Class	1 denotes positive, 0 denotes negative

variables. The probability function is assigned to each node. Bayesian networks are used to represent knowledge in an area that is uncertain. Statistics, probability theory, graph theory, and computer science are all combined in Bayesian Networks [22].

BN depicts a joint probability distribution over a set of random variables "X." The BN is explained as $Y = \langle G, \theta \rangle$ where G is DAG. It contains the nodes $N_{1,}$ N_2, . . .,N_n, and edges depict dependencies among the variables. Joint distribution for a BN is defined as shown in equation (5).

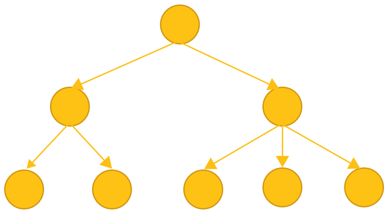


Figure 5: A single DT.

P (nodes|parents (node)) for all nodes

$$\begin{aligned} P\left(X_1, \ldots\ldots, X_n\right) &= \prod_{i=1}^{n} P\left(X_i | X_1, \ldots\ldots, X_{i-1}\right) \\ &= \prod_{i=1}^{n} P\left(X_i | \text{Parents}\left(X_{1i}\right)\right). \end{aligned} \tag{5}$$

When huge networks are used, this combined distribution will reduce computation. The use of a Bayesian network is motivated by the fact that they record high accuracy for complicated and uncertain domains.

3.6.2. Radial Basis Function (RBF). The supervised algorithm RBF is derived from function approximation. RBFs have three layers: an I/P layer, an O/P layer, and a hidden layer [23]. Each hidden unit in the input space has its own receptive field. These units are known as radial centers, and they are denoted by the letters $V_1, V_2, \ldots, V_n$. The transformation from hidden units to output is linear but not from the input layer to the hidden unit. RBF will create a local

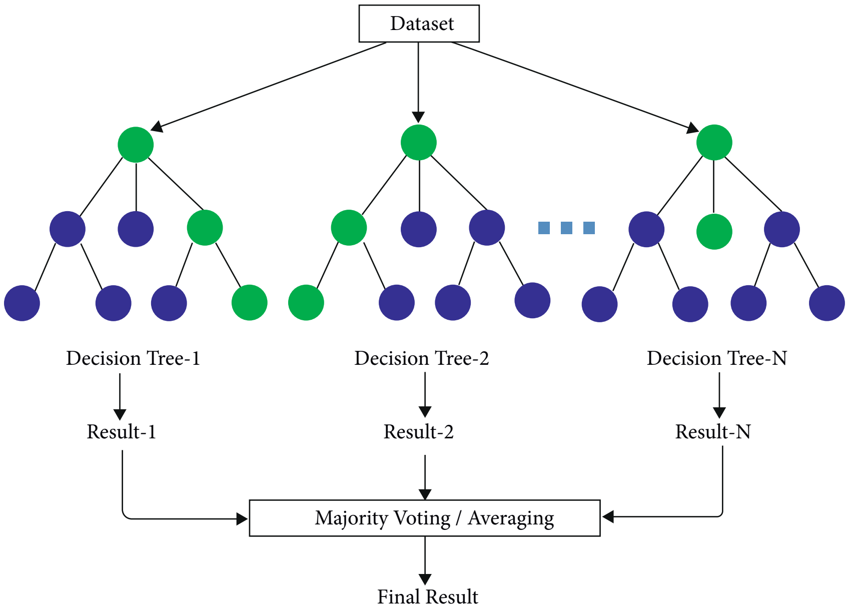


FIGURE 6: RF example.

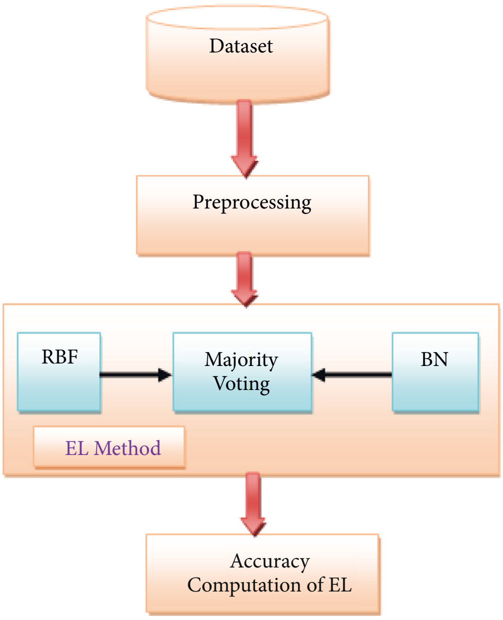


FIGURE 7: Proposed process diagram.

map, as a result of which it will learn quickly. RBF employs two-step learning, requiring the acquisition of both weight and centers. By calculating the input's similarity to instances from the training set, RBF achieves supervised learning. Each neuron computes the distance between the input and its prototype to classify a fresh sample. The Gaussian equation is shown as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left((x-u)^2/2\sigma^2\right)}, \tag{6}$$

where x is the input, u is the mean, and $\sigma\sigma$ is the standard deviation.

The neuron activation function of RBF is defined as shown in the following equation:

$$\varnothing(x) = e^{-\beta\|x-\mu\|^2}. \tag{7}$$

The RBF is motivated by the fact that it learns faster than a simple feed-forward network because of the following advantages: training is quick, MLP output nodes implement linear summation, they are good at interpolation, and radial basis functions are implemented by hidden nodes.

3.6.3. Ensemble Learning (EL) Method. Ensemble learning is an effective machine learning technique for improving model performance. The model's prediction capability is improved by combining the varied sets of learners (Base Learners). The importance of selecting the appropriate ensemble for diabetes prediction cannot be overstated.

RBF and Bayesian Network have been employed as base learners in the proposed methodology, which was then integrated with the EL method. Majority voting can be represented as shown in the following equation:

$$\sum_{t=1}^{T} d_{t,J} = \frac{C}{\max_{j=1}} \sum_{t=1}^{T} d_{t,j}, \tag{8}$$

where T is the dataset and C is the class label.

4. Results and Discussion

A confusion matrix is a matrix that comprises data on actual and expected classifications and is used to assess the algorithm's performance using the matrix data. We get both correct and incorrect predictions from each classifier. False Positive and False Negative predictions are two types of incorrect predictions. When we forecast

```
(i) Input required: Diabetes Dataset
(ii) Output expected: Prediction of Ensemble Technique
    Step_1: Preprocessing on the Pima dataset of diabetes
    Step_2: Separate the dataset as test and training data
    Step_3: Construct EL (Majority voting) using BN and RBF with $k$-Fold-cross-Validation ($k = 10$)
    Step_4: Let us call each new sample "$S$" Test data
    Step_5: EL (test sample) = majority voting (class (RBF), class (BN))
    Step_6: Keep track of the suggested approach's accuracy and evaluate it using a variety of measures.
```

Algorithm 1: Stages in the proposed ensemble learning (EL) model.

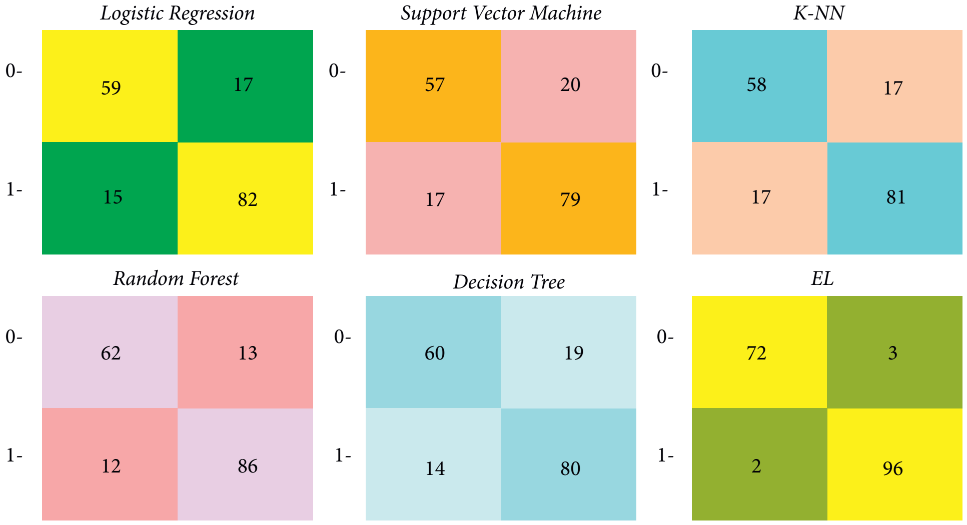


Figure 8: Confusion matrix.

Table 2: Performance metrics of different ML techniques in %.

ML techniques	Precision	Recall	$F1$-score
LR	82.82	84.53	83.51
KNN	82.65	82.65	82.65
RF	86.86	87.75	87.30
DT	80.80	85.10	82.89
SVM	79.79	82.29	81.02
EL (BN + RBF)	**96.96**	**97.96**	**97.46**

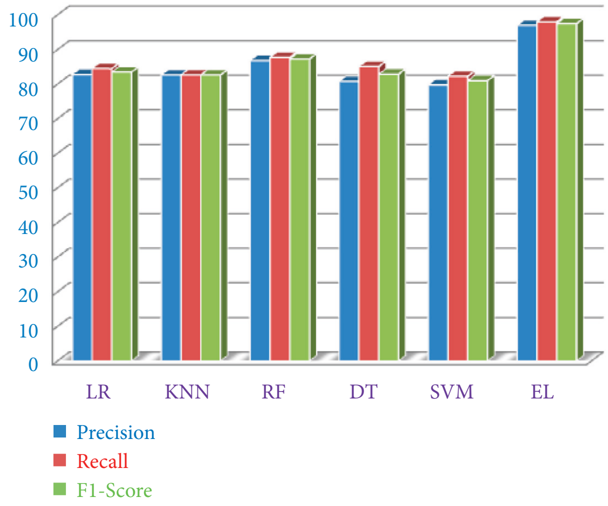


Figure 9: Performance of various ML techniques.

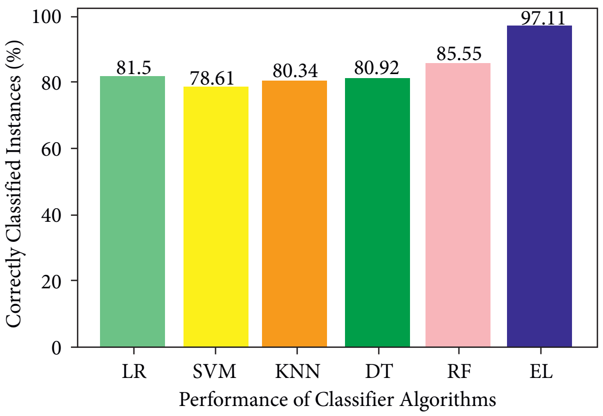


Figure 10: Accuracy of algorithms.

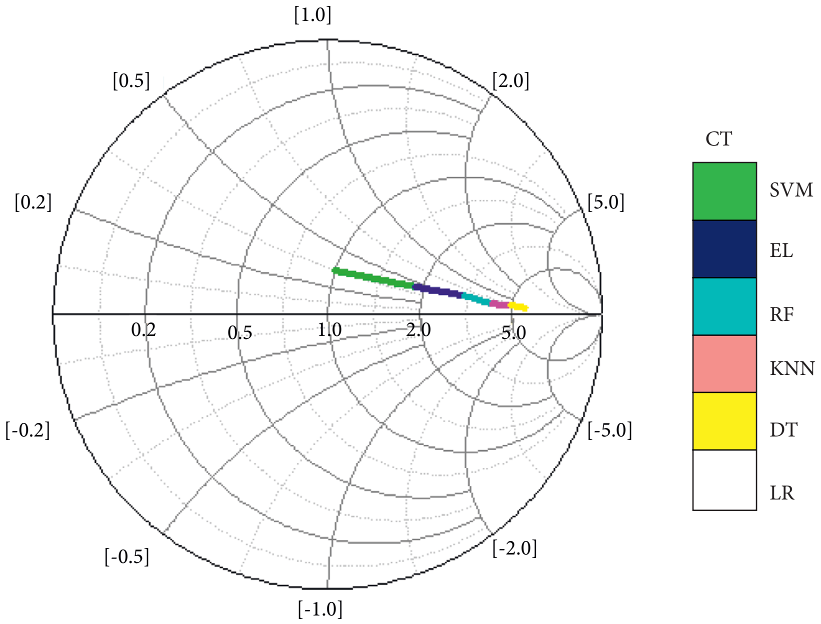


Figure 11: Computation analyses of ML techniques.

that something will happen/occur but it does not, we call it a False Positive also called as type I error (a valid null hypothesis is rejected). For example, we predicted that diabetes would develop, but it did not. When we forecast that something will not happen/occur, but it does, we call it a False Negative also called as type II error (the failure to reject a false null hypothesis). For example, we might anticipate that there will be no diabetes, but diabetes does exist. Type I errors are typically thought to be less dangerous than type II faults. In disciplines, such as medicine, both errors may appear to be fatal. The confusion matrix of all the ML algorithms is depicted in Figure 8.

The entire aforementioned models have been evaluated against precision, $F1$-measure, and recall, and the values obtained are depicted in Table 2.

The precision is computed using the following equation:

$$\text{precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs}}. \tag{9}$$

The precision of LR, KNN, RF, DT, SVM, and EL classifiers is 82.82%, 82.65%, 86.86%, 80.80%, 79.79%, and 96.96%, respectively.

Recall is the proportion of TPs out of TPs and FNs. The recall is computed using the following equation:

$$\text{recall} = \frac{\text{TPs}}{\text{TPs} + \text{FNs}}. \tag{10}$$

The recall of LR, KNN, RF, DT, SVM, and EL classifiers is 84.53%, 82.65%, 87.75%, 85.10%, 82.29%, and 97.96%, respectively.

The $F1$-score which is also called as harmonic mean that considers both precision as well as recall is computed using the following equation:

$$F1 - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{11}$$

The $F1$-score of LR, KNN, RF, DT, SVM, and EL classifiers is 83.51%, 82.65%, 87.30%, 82.89%, 81.02%, and 97.46%, respectively. The recall, precision, and $F1$-scores of all the techniques are shown in Figure 9.

The number of correctly classified data instances divided by the total number of data instances is known as accuracy. The accuracy is computed using the following equation:

$$\text{accuracy} = \frac{\text{TNs} + \text{TPs}}{\text{TNs} + \text{TPs} + \text{FPs} + \text{FNs}}. \tag{12}$$

The accuracy of the different ML algorithms is also computed. The accuracy of LR, SVM, KNN, DT, RF, and EL classifiers is 81.50%, 78.61%, 80.34%, 80.92%, 85.55%, and 97.11%, respectively. The EL method has outperformed with an accuracy of 97.11% in comparison to other ML algorithms. Figure 10 shows the accuracy performance of the aforementioned ML algorithms.

Figure 11 illustrates the processing time required for several classification techniques. In comparison to other methods, EL techniques take longer to process data due to the blending process of base learners. CG (conjugate gradient) optimization was utilized for primary LR with compliance. $L2$ regularization and Alpha = 0.0001 were being used by SVM to get the best learning rate. Using Gini impurity, a DT was divided into smaller trees, which consumed optimal time for processing. It was enlarged until all leaves were pure or included fewer than two nodes, whichever came first.

5. Conclusion

If identified early enough, diabetes has the potential to save thousands of lives. The main purpose of this article is to summarize all of the recent and ongoing investigations using machine learning algorithms for diabetic illness prediction. In addition, this study provides all of the relevant and required information for researchers who are new to deep learning and wish to examine ML methods. In conclusion, EL demonstrates its power in terms of efficacy and efficiency, as well as precision, recall, accuracy, and *F*1-score. The blended EL method provides the best accuracy of 97.11% compared to other models. More study is needed in this field to increase the performance of categorization systems so that they can forecast a larger number of variables. The goal is to figure out how to parameterize our categorization algorithms in order to attain high accuracy. A number of datasets are continuously being investigated to explore how ML techniques might be used to better predict diabetes.

Data Availability

The Irvine ML Repository data used to support the findings of this study are available at https://archive.ics.uci.edu/ml/datasets/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. Rish, “An empirical study of the naive Bayes classifier,” *IJCAI*, vol. 3, 2001.

[2] T. R. Mahesh, V. Vinoth Kumar, and V. Vivek, “Early predictive model for breast cancer classification using blended ensemble learning,” *International Journal of System Assurance Engineering and Managament*, 2022.

[3] U. Fayyad, G. P. Piatetsky-Shapiro, and P. Smyth, “The KDD process for extracting useful knowledge from volumes of data,” *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996.

[5] W. S. Mcculloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biology*, vol. 52, no. 1-2, pp. 99–115, 1990.

[6] M. Abadi, P. Barham, J. Chen et al., “Tensorflow:A system for large-scale machine learning,” *OSDI*, vol. 16, pp. 265–283, 2016.

[7] U. R. Acharya, K. S. Vidya, D. N. Ghista, W. J. E. Lim, F. Molinari, and M. Sankaranarayanan, “Computer-aided diagnosis of di-abetic subjects by heart rate variability signals using discretewavelet transform method,” *Knowledge-Based Systems*, vol. 81, pp. 56–64, 2015.

[8] X. X. Chen, H. Tang, W. C. Li, H. Wu, W. Chen, and H. Ding, H Lin, Identification of bacterial cell wall lyases via pseudo amino acid composition,” *BioMed Research International*, vol. 20168 pages, 2016.

[9] S. Habibi, M. Ahmadi, and S. Alizadeh, “Type 2 diabetes mellitus screening and risk factors using decision tree: results of data mining,” *Global Journal of Health Science*, vol. 7, no. 5, pp. 304–310, 2015.

[10] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, “Machine learning and data mining methods in diabetes research,” *Computational and Structural Biotechnology Journal*, vol. 15, pp. 104–116, 2017.

[11] B. J. Lee and J. Y. Kim, “Identification of type 2 diabetes risk factors using phenotypes consisting of anthropometry and triglycerides based on machine learning,” *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 39–46, 2016.

[12] G. S. Gowramma, T. R. Mahesh, and G. Gowda, “An automatic system for IVF data classification by utilizing multilayer perceptron algorithm,” *ICCTEST*, vol. 2, pp. 667–672, 2017.

[13] K. Sharma, T. R Mahesh, and J Bhuvana, “Big data technology for developing learning resources,” *Journal of Physics: Conference Series*, vol. 1979, no. 1, Article ID 012019, 2021.

[14] A. A. Deshmukh, M. Dubal, T. R Mahesh, and C. R. Chauhan, “Data security analysis and security extension for smart cards using java card,” *International Journal of Advanced Information Technology (IJAIT)*, vol. 2, no. 2, 2012.

[15] T. R. Mahesh, S. Prabhanjan, and M. Vinayababu, “Noise reduction by using fuzzy image filtering,” *Journal of Theoretical and Applied Information Technology*, vol. 15, 2010.

[16] H. K. Shashikala, T. R. Mahesh, V. Vivek, M. G. Sindhu, C. Baig, and T. Z. Baig, “Early detection of spondylosis using point-based image processing techniques,” in *Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 655–659, Bangalore, India, August 2021.

[17] M. J. Dubai, T. R. Ghosh, and P. A. Ghosh, “Design of new security algorithm: using hybrid Cryptography architecture,” in *Proceedings of the 2011 3rd International Conference on Electronics Computer Technology*, pp. 99–101, Kanyakumari, India, April 2011.

[18] S. Perveen, M. Shahbaz, A. Guergachi, and K. Keshavjee, “Performance analysis of data mining classification techniques to predict diabetes,” *Procedia Computer Science*, vol. 82, pp. 115–121, 2016.

[19] S. Mahesh, T. R. Mahesh, and M. Vinayababu, “Using data mining techniques for detecting terror-related activities on the web,” *Journal of Theoretical and Applied Information Technology*, vol. 16, 2010.

[20] P. Ghosh and T. R. Mahesh, “A privacy preserving mutual authentication protocol for RFID based automated toll collection system,” in *Proceedings of the 2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, pp. 1–5, Indore, India, November 2016.

[21] P. K. Shukla, J. Kaur Sandhu, A. Ahirwar, D. Ghai, P. Maheshwary, and P. K. Shukla, “Multiobjective genetic algorithm and convolutional neural network based COVID-19 identification in chest X-ray images,” *Mathematical Problems in Engineering*, vol. 2021, Article ID 7804540, 2021.

[22] G. Pinaki and T. R. Mahesh, “Smart city: concept and challenges,” *International Journal on Advances in Engineering Technology and Science*, vol. 1, p. 1, 2015.

[23] T. R. Mahesh, V. Dhilip Kumar, V. Vinoth Kumar et al., “AdaBoost ensemble methods using K-fold cross validation for survivability with the early detection of heart disease,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–11, Article ID 9005278, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9870153, 1 page
https://doi.org/10.1155/2023/9870153

Retraction

Retracted: Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] M. Dong, H. Yu, Z. Sun et al., "Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 8327006, 13 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8327006, 13 pages
https://doi.org/10.1155/2022/8327006

RETRACTED

Research Article

Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology

Mo Dong,[1,2] Haiye Yu,[1] Zhipeng Sun,[3] Mingzhi Wu,[3] Lei Zhang,[1] Yuanyuan Sui,[1] Guanghao Yu,[2] Ting Han,[2] and Ruohan Zhao[2]

[1]College of Biological and Agricultural Engineering, Jilin University, Changchun 130022, Jilin, China
[2]Mudanjiang Medical University, Mudanjiang 157000, Heilongjiang, China
[3]Nanchang Automotive Institution of Intelligence & New Energy, Nanchang 330052, Jiangxi, China

Correspondence should be addressed to Ruohan Zhao; zhaoruohan@mdjmu.edu.cn

Received 24 April 2022; Revised 10 June 2022; Accepted 3 July 2022; Published 14 July 2022

Academic Editor: Muhammad Ahmad

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

With the extensive use of the Internet of Things (IoT) in agriculture, the number of terminals are also grow rapidly. This will increase the network traffic and computing pressure of the centralized server. The centralized data processing mode used in traditional agriculture cannot meet the needs of the Internet of everything era. This paper designs a gateway based on edge-computing technology for monitoring crop growth environment. It uses virtualized container technology to package long-range wide-area network (LoRaWAN) server, pest identification, and environmental information data fusion functions into images. It forms integrated operation mode of multiple function in agriculture. The gateway applies message-oriented middleware to standardize and customize the data transmission among functional modules, clouds, and edges. Through simulation and field test, the designed gateway can achieve the functions of each module at the same time, the resource utilization, and the transmission quality are stable. The edge-computing gateway has the advantages of low cost, low latency, and low power consumption which has practical significance.

1. Introduction

With the rapid development of Internet of Things (IoT) technology in agriculture, the monitoring of the growth environment of crops can obtain real-time information such as water and fertilizer using, diseases, and pests. These data can be fed back and analyzed by expert to guide production. However, due to the extensive use of the IoT in agriculture, the number of terminals are also grow rapidly which will increase the network traffic and computing pressure of the centralized server. The centralized data processing mode used in traditional agriculture cannot meet the needs of the Internet of everything era. The emergence of edge computing means that many tasks such as identification of pests and analysis of a large amount of agricultural field data that used to require cloud processing can be implemented locally. Edge-computing technology can greatly increase the response speed of agricultural information, improve the monitoring accuracy, and formulate production strategies for monitoring areas more effectively.

In edge-computing technology, downstream data represents cloud center, upstream data represents interconnected services, and the edge refers to computing and network resources from the data source to the cloud center [1]. Edge computing can calculate, store, and apply the data uploaded by nodes near the source and provide real-time processing services. It can meet the requirements of online real-time data processing, safe transmission, and decision-making in intelligent agriculture [2, 3].

Although the edge-computing technology has made great progress in the application of IoT, still this opens up research questions:(1). Edge-computing technology is rarely used in the agricultural IoT, especially when the gateway accesses different types of agricultural data at the same time. (2) Edge-computing technology combined with container technology in the field of agricultural IoT is also basically in

RETRACTED

Table 1: Literature review of edge-computing applications.

Literature	Application scenarios	Advantage			
		Security	Low cost	Low latency	Low power consumption
[5]	Finding nodes of Internet of things				✓
[6]	Traceability of shipyard pipeline processing			✓	
[7]	Detection of industrial thermal anomaly			✓	
[8]	Manufacturing architecture of decision-making		✓	✓	
[9]	Encryption of smart grid data	✓			
[10]	Networked control of industrial robots		✓	✓	
[11]	Application of machine learning in industry	✓		✓	
[12]	A smart IoT monitoring based on UAV		✓	✓	

the initial stage. (3) The research on cloud-edge architecture and online algorithm of IoT applied to crop growth environmental monitoring is still in the blank stage.

To underpin the relevance of the research questions, the following are the contributions of this paper:

(i) We design an IoT gateway for crop growth environmental monitoring based on edge-computing technology which includes both function and communication architecture.

(ii) We use virtualized container technology to package LoRaWAN server, pest identification, and environmental information data fusion functions into images which form integrated operation mode in agriculture.

(iii) We apply message-oriented middleware to standardize and customize the data transmission among functional modules, clouds, and edges in the gateway which reduces the pressure of cloud computing, lowers the transmission delay between cloud and edge.

This paper is outlined as follows. Section 2 gives related work of applications of edge-computing and docker technology. Section 3 describes the design of edge-computing gateway framework. Section 4 provides the experimental results and performance evaluation include transmission stability of LoRaWAN server, stress testing of Message Queuing Telemetry Transport (MQTT) message-oriented middleware, stability test of edge-computing architecture, and field test. Section 5 summarizes our contributions and prospects, concludes this article.

2. Related Works

The research on edge computing began in 1998, Akamai Company designed a content delivery network (CDN) system based on distributed cache server. The system accomplished load balancing, computing scheduling, and data forwarding operations for each distributed cache server through the master server. This improved the response speed and connection success rate of user accessing to the server [4]. In 2009, Satyanarayanan et al. introduced the concept of Cloudlet. It used fixed computing resources closer to clients to increase computing power of traditional cloud servers. The scheme focused on the downlink capability of edge computing, moved some cloud server functions to the edge and further reduced bandwidth consumption and latency. Since then, with the development of the IoT and mobile communication technology, more attention has been paid to the data processing, transmission, and storage of edge computing. This paper summarizes applications of the edge-computing technology in recent years and analyzes their advantages as shown in Table 1.

How to dynamically match computing resources in heterogeneous computing environment to optimize system performance is a challenge for edge-computing field.

In the case of applying edge-computing technology in IoT, application of small board equipment with low cost and limited resources are becoming more and more popular. The virtualization technology applied to small board devices can ensure the implementation of some cloud computing tasks on devices of edge computing, and it has good flexibility and scalability [13–16]. Virtualization technology enables users to make full use of resources by transforming physical resources such as networks, computing, storage so that services are no longer limited by hardware [17]. As a virtualization technology, container has been widely used in resource management, system maintenance, and software deployment of cloud computing platforms in recent years. It is also gradually applied to edge computing. Docker is a container engine based on Linux kernel virtualization technology. It was developed using GO language and apache2.0 protocol and was released on GitHub in 2013. Docker uses Cgroups technology to manage process resources and uses namespace to isolate system resources. While providing isolation, the container can also achieve resource sharing [18]. Compared with traditional virtual machines, Docker starts quickly, deploys easily, and uses less resources [19]. However, the application of real-time container technology is often not considered in edge computing at this stage.

3. Proposed Edge-Computing Gateway Framework

3.1. Design of Edge-Computing Gateway. Gateway is a computer system or device that converts between networks with different communication protocols and data structures. IoT gateway is the core component, and it is the bridge between the terminal nodes and the cloud. This paper designs an edge-computing gateway, and it is the regional network information center of crop growth environmental

RETRACTED

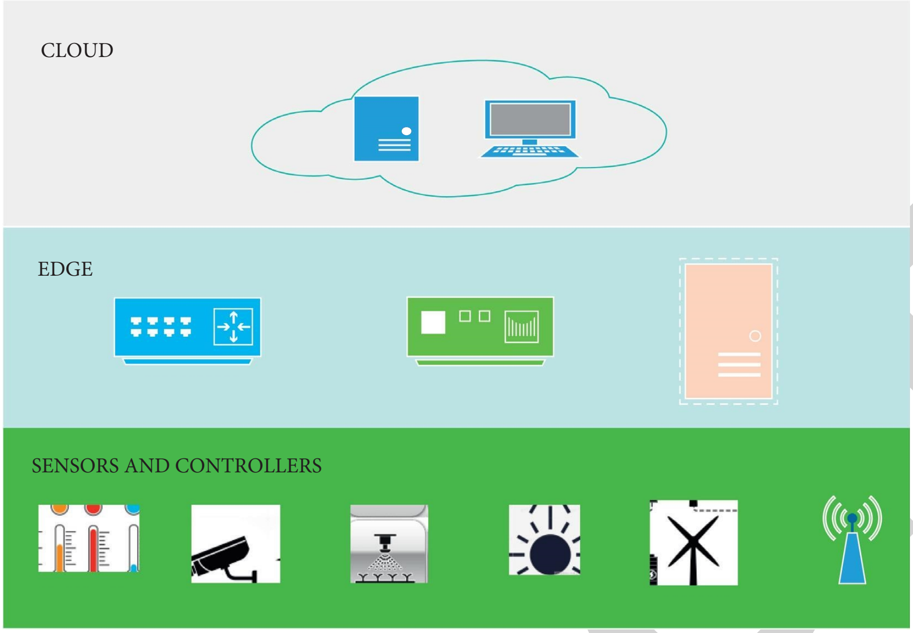


Figure 1: Relationship between cloud, edge-computing layer, and terminal.

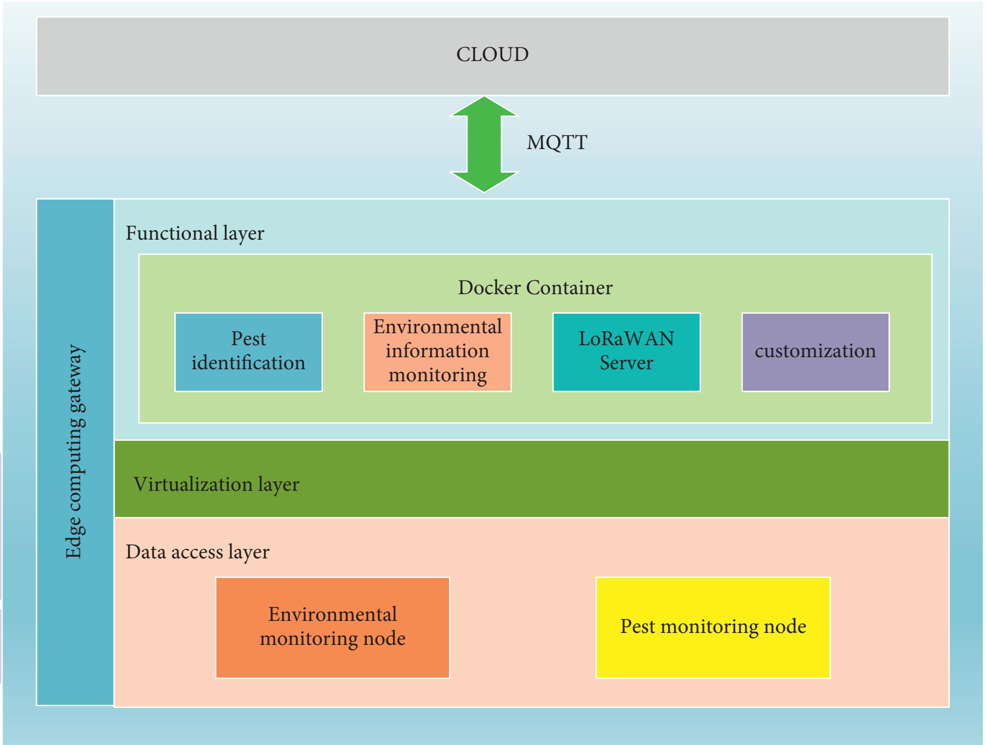


Figure 2: Architecture of edge-computing gateway.

monitoring IoT. The edge-computing gateway provides a layer with communication and data processing capabilities between the terminal devices and the cloud server. This structure has good performance in transmission delay and stability, as shown in Figure 1.

In this paper, we combine the situation of environmental monitoring with the functional view of edge computing and design the architecture of edge-computing gateway. The architecture is divided into data access layer, virtualization layer, and function layer. This architecture enables all kinds

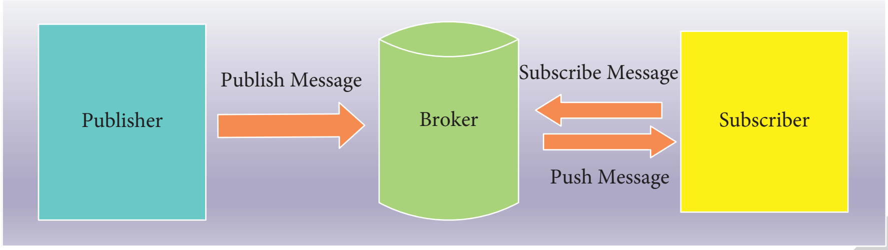


Figure 3: Diagram of message-oriented middleware.

RETRACTED

of situations in agricultural applications to be analyzed and processed quickly. The architecture is shown in Figure 2.

3.1.1. Data Access Layer. The data access layer of the edge-computing gateway supports a variety of hardware interfaces and bus protocols. We customize LoRaWAN and Wi-Fi protocols and equip with communication module. Through the multithreading technology of Linux system, the drivers can be called concurrently. This ensures the real-time data transmission and provides data support for edge computing.

3.1.2. Virtualization Layer. We use Docker as the virtualization platform. The containerized architecture can split the original services that need to be performed on cloud servers. It also provides a clear boundary between different services through isolation technology. This can make better use of the hardware computing resources. Base on Docker virtualization technology, services that need to run independently are packaged into images. In this way, each service is isolated from each other. The virtual network provides data access services for the container, and it improves the resource utilization efficiency and compatibility of the edge-computing gateway.

3.1.3. Functional Layer. The function layer defines the input and output of each function container by managing the images. This layer processes data uploaded by terminal nodes and achieves cloud functions at the edge. The main functions of our design include pest identification, crop growth environmental information monitoring and processing, and integration of LoRaWAN server. In addition, we also retain the functional interfaces, and this will provide resources for the customized services according to the requirements of the agricultural IoT.

3.2. Main Functional Modules of Edge-Computing Gateway

3.2.1. MQTT Service Module. Docker container completes data transmission through virtual network interface. The virtual network communication between containers uses the data exchange method of Linux kernel. Virtual network interface reduces the time loss caused by the transmission of physical network devices. Data packets are directly transmitted and exchanged within the processor. Although the containers can be accessed through Docker bridge, the traditional interaction mode between containers is generally using IP address, which has the problems of complex interaction and poor expansibility.

In this paper, we use the MQTT message-oriented middleware and apply the publish/subscribe mode to complete message transmission. The principle is shown in Figure 3. In this design, message-oriented middleware is also applied to data interaction between edge-computing gateway and cloud. At present, most of Internet architectures support MQTT protocol. The MQTT Broker has become a bridge for information exchange in the cloud-edge architecture.

In this design, RabbitMQ developed by Rabbit Technology Co., Ltd. is used as the software carrier of MQTT message-oriented middleware. RabbitMQ is written in Erlang language. It completes asynchronous message processing through advanced message queuing protocol. RabbitMQ supports cluster operation and has good flexibility and scalability. The RabbitMQ structure is shown in Figure 4.

3.2.2. LoRaWAN Server Module. Under the traditional LoRaWAN architecture, LoRaWAN server is generally installed in the cloud. The cloud communicates with the field gateway through the wireless Internet, and it is greatly affected by the quality of the field network. Once the gateway loses the Internet, the entire IoT system will not work. Moreover, the frequent communication between the gateway and the cloud will also lead to excessive consumption of bandwidth resources and transmission delay. In this design, LoRaWAN server is installed in the edge-computing gateway. However, if the LoRaWAN server is directly installed in the Linux system, due to the complex structure, it will occupy more memory, processes, and interfaces. This will reduce the data processing capacity of the entire edge-computing gateway. Through container technology, we isolate the LoRaWAN server into a separate container. Based on Docker's excellent isolation technology, it can achieve complete LoRaWAN service support at the gateway.

3.2.3. Pest Identification Module. With the increase of high-quality images transmission on the Internet, the delay of cloud computing is significantly higher than that of edge computing. The main function of this module is to monitor, collect, transmit, and process the pests image in farmland. The module uses Python language to write pest identification and counting algorithm based on deep learning, achieves the identification, counting, and other processing of the pest images collected by the monitoring nodes.

RETRACTED

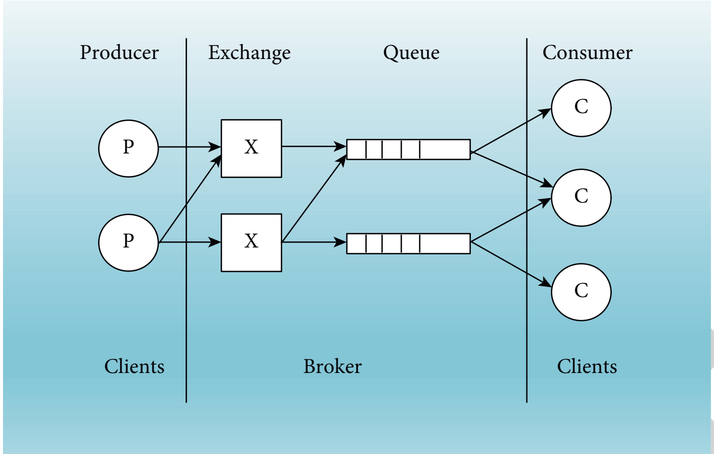


Figure 4: Structure of RabbitMQ.

3.2.4. Crop Growth Environmental Information Monitoring Module. The main function of this module is to monitor, collect, transmit and process the environmental information of farmland. The information includes seven different environmental factors: light intensity, air temperature, air humidity, soil humidity, soil temperature, CO_2 concentration, and soil conductivity. This module collects and transmits the information of various sensors arranged in the farmland to the gateway. Data processing and fusion algorithms are designed and used at the edge to improve the measurement accuracy. The module is written in Python language, using Python scientific computing library Numpy, and data analysis package Pandas. The data fusion algorithm of this module is as follows:

In the process of sensors acquiring data, the measured value X_i, the true value Z_i and the noise value Y_i of the sensor have the relationship of equation:

$$X_i = Z_i + Y_i (i = 1, 2, \ldots, n). \tag{1}$$

The following steps are taken to reduce noise error:

(1) Sort the data collected by all sensors and select the maximum value $X_{\max}$ and the minimum value $X_{\min.}$

(2) Take the average A_0 of the maximum and minimum values obtained.

$$A_0 = \frac{X_{\max} + X_{\min}}{2}. \tag{2}$$

(3) Add the obtained average A_0 to the original data and rearrange them. At this time, the part greater than or equal to A_0 is classified as P_1, and the part less than A_0 is classified as $P_{2.}$

$$\begin{Bmatrix} P_1 = [X(n)|X(n) \geq A_0] \\ P_2 = [X(n)|X(n) \geq A_0] \end{Bmatrix}. \tag{3}$$

(4) Continue to calculate the average value of P_1 and P_2. After obtaining P_{1a} and P_{2a}, average them again

$$A_1 = \frac{P_{1a} + P_{2a}}{2}. \tag{4}$$

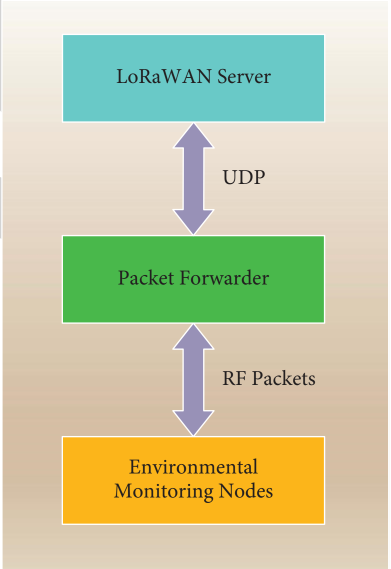


Figure 5: Data access process of environmental monitoring nodes.

(5) Add the obtained average value A_1 to the original data again, repeat the above steps until equation (5) is satisfied, and stop the iteration:

$$A_k = A_{k-1}. \tag{5}$$

At this time, the average value obtained is no longer changed, and it is considered that the value is infinitely close to the real data.

3.3. Workflow of Edge-Computing Gateway

3.3.1. Data Access of Terminal Nodes. (1) Data Access of Environmental Monitoring Nodes.

The environmental information monitoring data from the terminal sensor communicates with the LoRaWAN server in the following ways: Packet Forwarder (PF) is a

RETRACTED

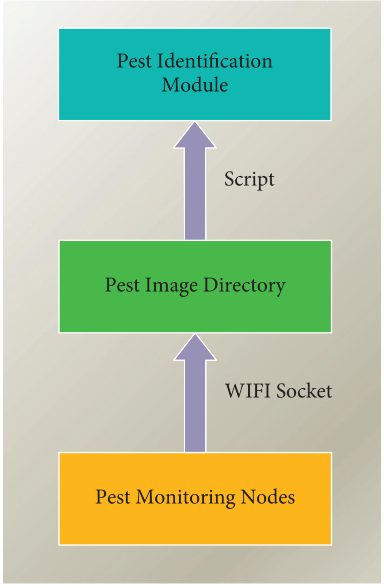


Figure 6: Data access process of pest identification nodes.

Table 2: Topic description of message-oriented middleware.

Topic name	Producer	Consumer	Description
LoRAWAN	LoRaWAN server	Data fusion processing module	Data uploaded by environmental monitoring sensors
Pest identification	Pest identification module	Cloud server	Data uploaded by pest identification module nodes
Data fusion	Data fusion processing module	Cloud server	Environmental information data processed by data fusion processing module

program running on the host of LoRa gateway, which is mainly responsible for data message forwarding between the terminal and LoRaWAN server. PF forwards the data packets uploaded by the environmental monitoring nodes to the LoRaWAN server module in the form of UDP packets. LoRaWAN server performs packets decryption, forwarding, and other operations. Similarly, PF converts the downlink medium access control (MAC) commands and other data of LoRaWAN server into Radio Frequency (RF) packets and sends them to the environmental monitoring nodes, as shown in Figure 5.

(2) Data Access of Pest Identification Nodes.

The edge-computing gateway uses Wi-Fi to communicate with the pest monitoring nodes and obtains the images by establishing a socket connection. The images uploaded through Wi-Fi are converted into image coding by OpenCV and temporarily stored in the specific path of the module. We use the script to regularly check whether there are updates under the path. If there is a new image, the pest recognition processing will be performed. After image preprocessing and identification, the current number and species of pests will be transmitted to cloud through MQTT protocol. The data access of pest identification nodes is shown in Figure 6.

3.3.2. Message-Oriented Middleware Service. In this design, we build message-oriented middleware transmission architecture with MQTT Broker as the hub. The communication between each module uses the push mode with unique "topic" subscription. The topics used in this design are shown in Table 2. The transmission architecture is shown in Figure 7.

The interface of RabbitMQ is shown in Figure 8. After building an image through Docker, we can view the user's status on the page after the device is connected. RabbitMQ as an MQTT server is responsible for message forwarding through topics and saving logs.

3.3.3. Management of Docker. In this design, Docker container is used to isolate the functional modules of edge-computing gateway. The internal communications between containers are carried out in the form of independent services, and each container provides necessary interfaces. The LoRaWAN server, environmental monitoring, pest identification, and other functional modules in the edge-computing gateway are all running in the mode of virtual container. From the perspective of hardware resources, each virtualization service container allocates resources through Docker. This enables more rational use of CPU computing power, data storage, and other resources. Docker is convenient for function splitting and reorganization, and we can manage a container by using commands alone, or use Docker compose (a tool for batch container management) for batch processing. The process is shown in Figure 9.

3.4. Hardware of Edge-Computing Gateway. We choose Raspberry Pi 4b as the hardware of the edge-computing gateway. Raspberry Pi is a single-board microcomputer with ARM processor architecture, low-power consumption, and high performance. Its CPU uses ARM Cortex-A72 1.5 GHz (quad core) processor and supports 64-bit operation. Raspberry Pi has 8 GB of running memory and Wi-Fi module, supports 5 GHz and 2.4 GHz under 802.11 g. It also supports microSD card, serial port, and GPIO communication. Raspberry Pi carries 4G transmission module through USB interface to exchange data with cloud. Although the size is similar to a hand, it can complete most of the processing task of a computer. With the support of hardware interface and external network module, it has the ability to achieve edge-computing tasks. Photo of edge-computing gateway based on Raspberry Pi is shown in Figure 10.

4. Experimental Results and Analysis

4.1. Transmission Stability of LoRaWAN Server. In order to better verify the data transmission stability of LoRaWAN server in an edge-computing gateway, this paper builds a simulation system of LoRaWAN transmission quality based

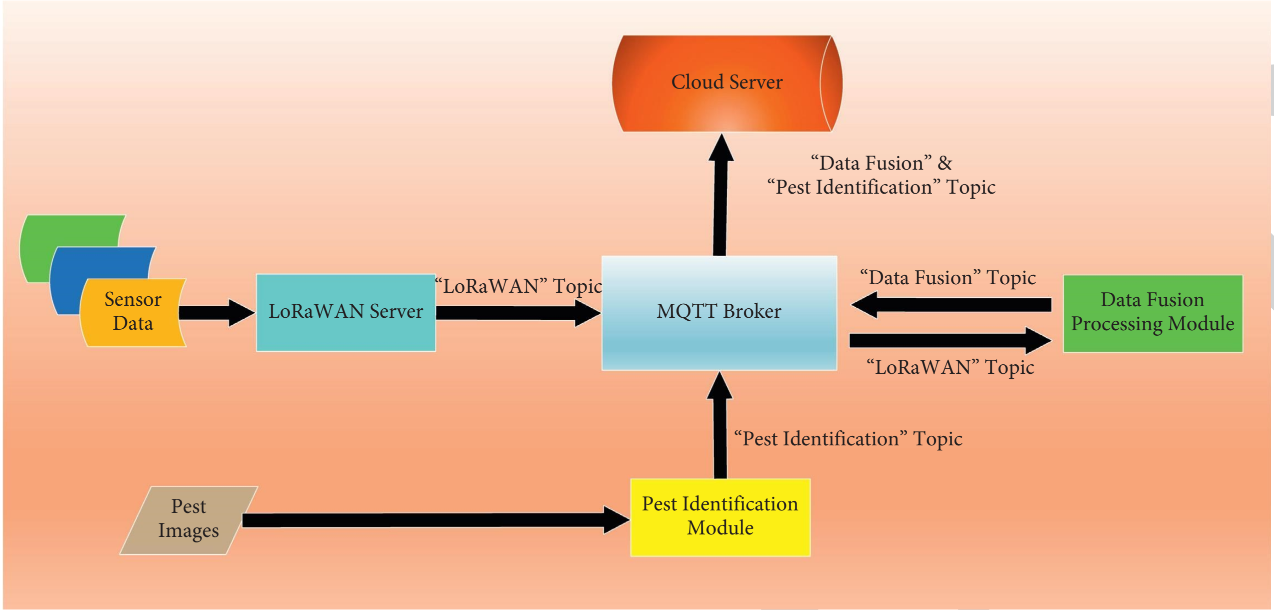
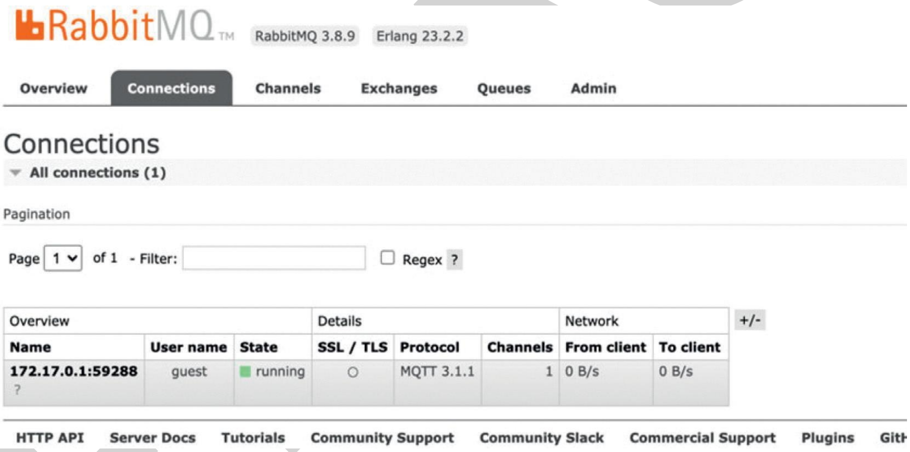


Figure 7: Transmission architecture of edge-computing gateway based on MQTT.


RabbitMQ™ RabbitMQ 3.8.9 Erlang 23.2.2
Overview Connections Channels Exchanges Queues Admin

Connections

All connections (1)

Pagination

Page 1 of 1 - Filter: □ Regex ?

Overview			Details			Network	
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client
172.17.0.1:59288 ?	guest	running	○	MQTT 3.1.1	1	0 B/s	0 B/s

+/-
HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitH

Figure 8: Interface of RabbitMQ.

on ns-3 software using Ubuntu 16.04 OS. ns-3 simulation software, developed in 2006, it is an open-source network simulation software driven by discrete events. ns-3 is mainly used to simulate wireless transmission networks. In this simulation, we choose eight common data transmission intervals of agricultural IoT. In addition, in order to test the stability of concurrent data transmission of multiple monitoring nodes and gateways, we set the number of environmental monitoring nodes to 0–1000 and the number of edge-computing gateways to 1, 2, and 4. We divide the sending data into two types: with acknowledgment (ACK) and without ACK. For the transmission stability between the crop growth environmental monitoring nodes and LoRaWAN server, we conduct the following tests.

4.1.1. Single Gateway. (1) Relationship between Transmission Interval and Packet Loss Rate.

In this simulation, we choose a single edge-computing gateway, set the transmission interval as 60, 120, 200, 400, 600, 800, 1000, and 1200 s, and fix the number of environmental monitoring nodes at 200. We divide the sending data into two types: with ACK and without ACK. The relationship between packet loss rate and transmission interval is shown in Figure 11.

(2) Relationship between the Number of Nodes and Packet Loss Rate.

In this simulation, we choose a single edge-computing gateway, and transmission interval is fixed at 600 s. The relationship between the number of environmental monitoring nodes (0–1000) and the packet loss rate is shown in Figure 12.

4.1.2. Multi Gateway. (1) Relationship between the Number of Nodes and Packet Loss Rate.

In this simulation, we choose 1, 2, and 4 gateways separately, and the transmission interval is fixed at 600 s. We also divide the sending data into two types: with ACK and

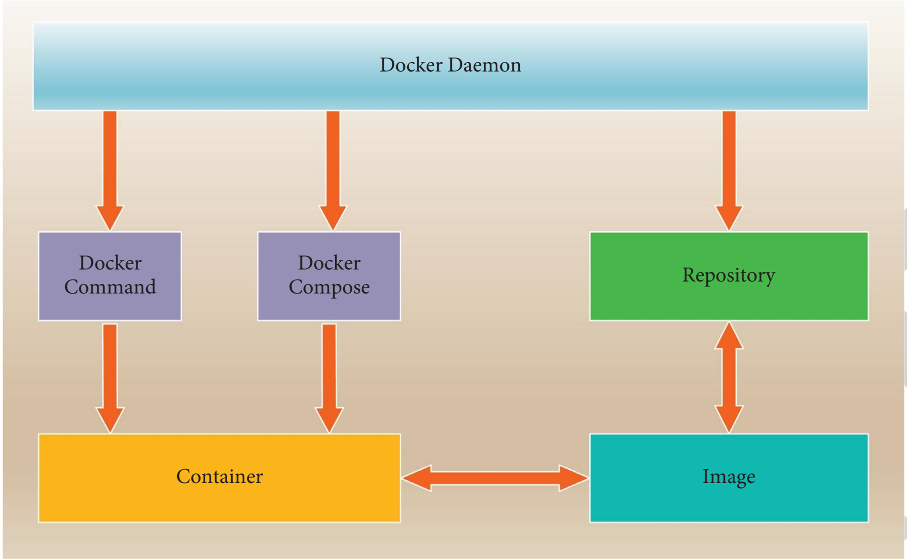


Figure 9: Management of docker container.

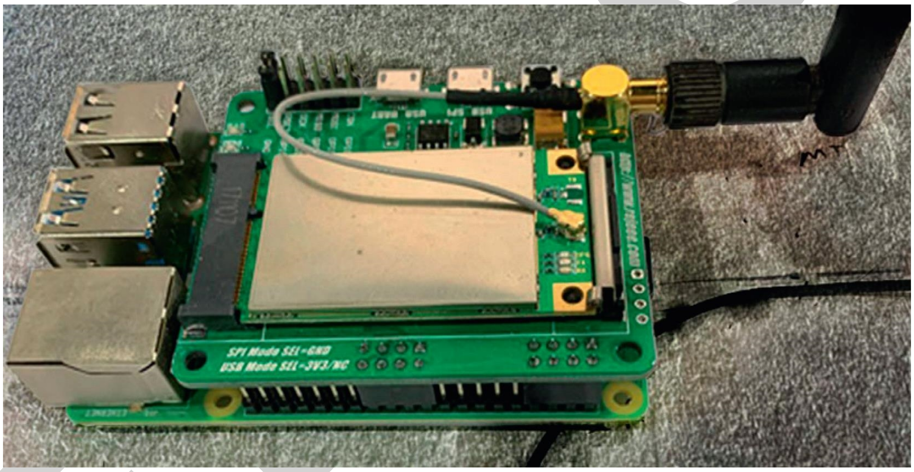


Figure 10: Photo of edge-computing gateway based on raspberry Pi.

RETRACTED

without ACK. Figures 13 and 14 show the relationship between the number of environmental monitoring nodes (0–1000) and the packet loss rate.

4.1.3. Result Analysis.

(1) It can be seen from Figure 11 that with the gradual increase of transmission interval, whether the transmission has ACK or not, the packet loss rate will gradually decrease. When the data transmission interval is once a minute, the packet loss rate still be controlled less than 5%. In the field of communication, it is generally believed that when the packet loss rate is less than 5%, the communication quality is good. Therefore, it meets the requirements of transmission quality.

(2) From Figure 12, we can see that with the increase of the number of monitoring nodes, the packet loss rate of both transmission modes increase, and it is the same as expected. However, even when the number of monitoring nodes reaches 1000, the packet loss rate remains at a low level (less than 5%). Therefore, it also meets the requirements of transmission quality.

(3) From Figures 13 and 14, we can see that using multigateways at the same time will not affect the quality of transmission. This further shows that using multiple gateways with LoRaWAN server can expand the monitoring area of farmland without reducing the transmission quality.

(4) By analyzing Figures 11–14, we can know that whether the transmission with ACK is used or not, the packet loss rate is less than 5%. There is little difference between the two transmission modes. The specific advantages of the two transmission modes need further simulation analysis, which will not be discussed in this paper.

RETRACTED

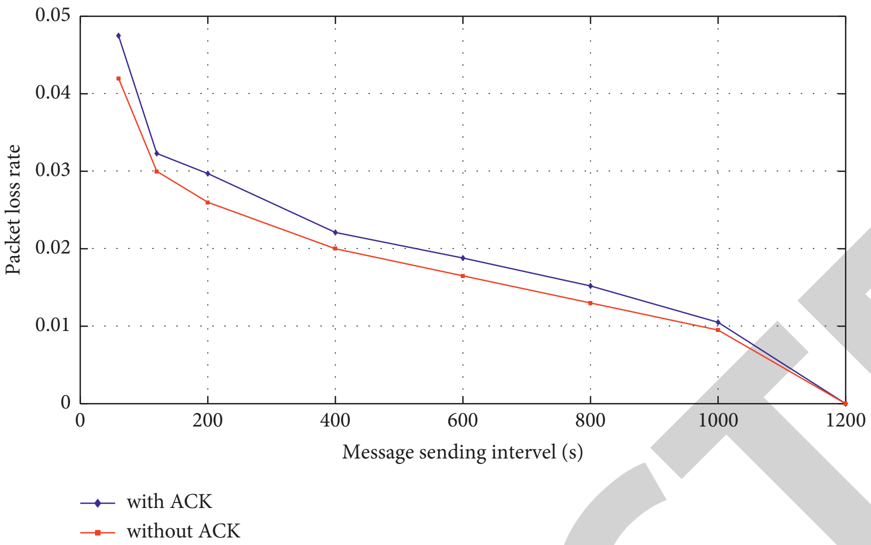


Figure 11: Relationship between transmission interval and packet loss rate under single gateway.

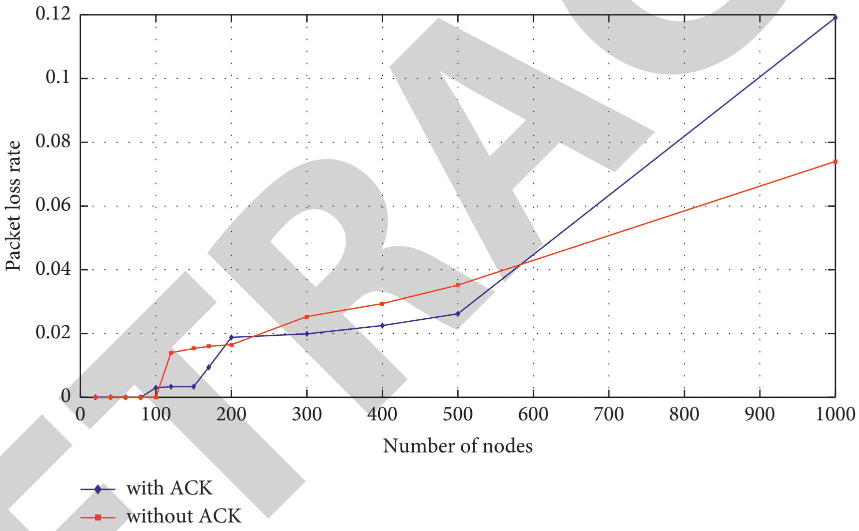


Figure 12: Relationship between the number of nodes and packet loss rate under single gateway.

Through the above analysis, we can conclude that installing LoRaWAN server on the edge-computing device can effectively increase the transmission distance and the monitoring range of farmland. At the same time, the data transmission is stable and meets the transmission quality requirements. This provides an effective method for future research on long-distance transmission and larger area of farmland environmental monitoring.

4.2. Stress Testing of MQTT Message Oriented Middleware

4.2.1. Research Methods. Information forwarding capability is an important performance of edge-computing gateway. Information forwarding capabilities are mainly reflected in the success rate of sending messages, delaying situation, and other information forwarding statistics. We use a concurrent stress testing that sends 1–1000 data per second to test the success rate and delaying situation of message-oriented middleware when processing data. Figure 15 shows the packet loss rate of message forwarding at the frequency of 1–1000 messages per second.

4.2.2. Result Analysis. The test results show that the gateway's information forwarding capability is stable. Under the stress testing of 1000 concurrent messages, the packet loss rate is about 2%. It meets the requirements of transmission quality.

In farmland environmental monitoring, the frequency of data transmission usually does not reach the state of stress testing. This shows the transmission mechanism based on MQTT message-oriented middleware can be well applied to the edge-computing gateway.

4.3. Stability Test of Edge-Computing Architecture

4.3.1. Research Methods. In operation and maintenance, we often use the average load to represent the average number

RETRACTED

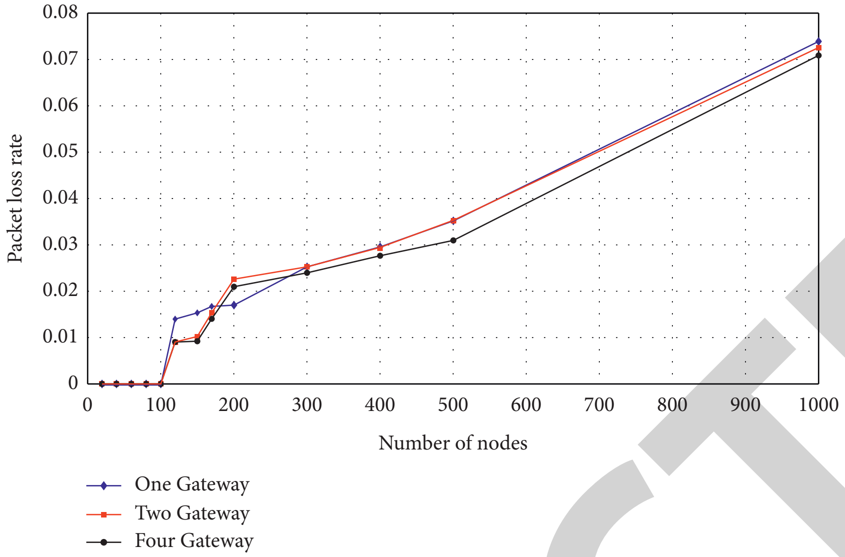


Figure 13: Relationship between the number of nodes and packet loss rate under multigateway (without ACK).

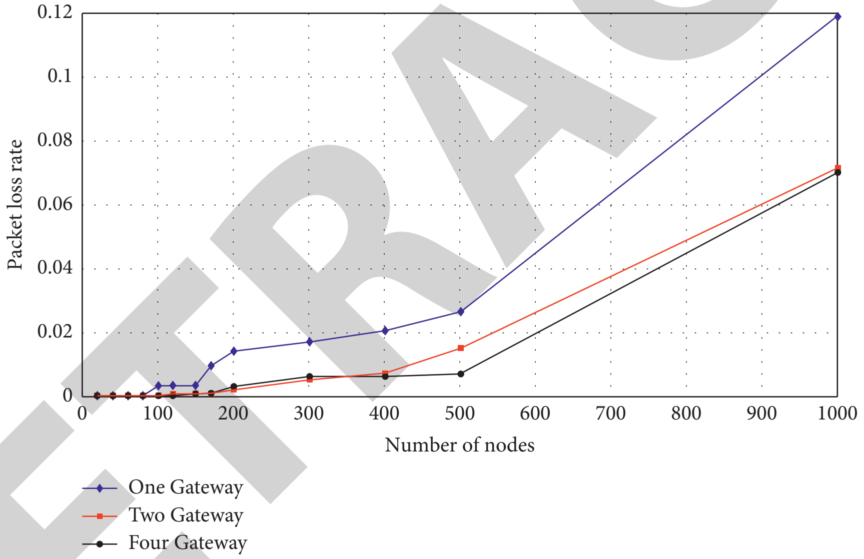


Figure 14: Relationship between the number of nodes and packet loss rate under multigateway (with ACK).

of processes in the running state and noninterruptible state of the system. This is also a measurement parameter to characterize the overall operation state of the system. Average load is closely related to the number of CPU cores in the hardware. If the average load is more than the number of CPU cores, the system is considered to be in an overloaded state, which will affect all processes of the system, causing the system to run slowly or even crash.

In this design, the Raspberry Pi adopts a quad core processor that supports 64-bit operation. By monitoring the average gateway load during the stress testing, we can see the gateway's average load and memory usage under 500/1000 concurrent messages per second as shown in Figures 16 and 17.

4.3.2. Result Analysis. Under the stress testing of 500 concurrent messages per second, the average load of the gateway is 0.22, while with 1000 concurrent messages per second, the average load of the gateway is 2.99, which are both less than the number of CPU cores. The above experiments show that the gateway works stably and reliably in stress testing.

Through the test, we can see that the edge-computing gateway architecture based on Raspberry Pi fully meets the complex state requirements of high-speed data transmission, which provides a new method for the application of mobile edge-computing technology.

4.4. Field Test of Edge-Computing Gateway. We also carried out field tests in the Siping area of Jilin Province in China. The nodes of the pest identification module and the environmental monitoring module both uploaded a set of data per minute. The results showed that all the modules are

RETRACTED

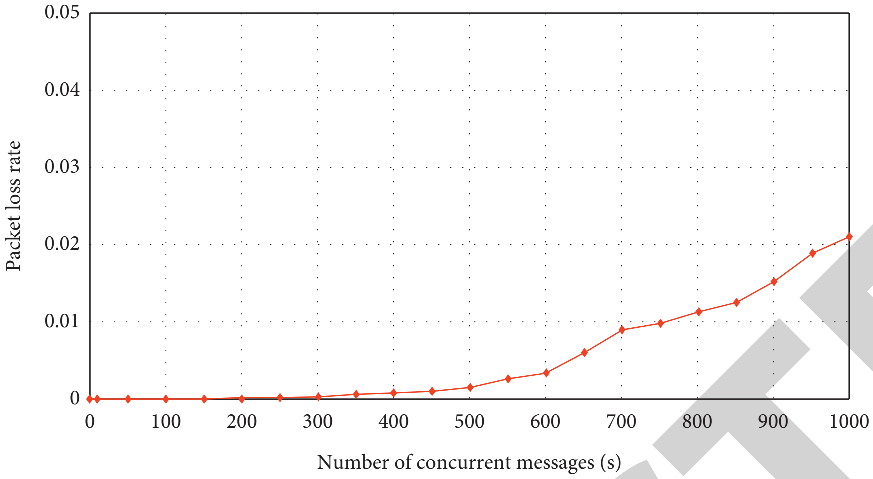
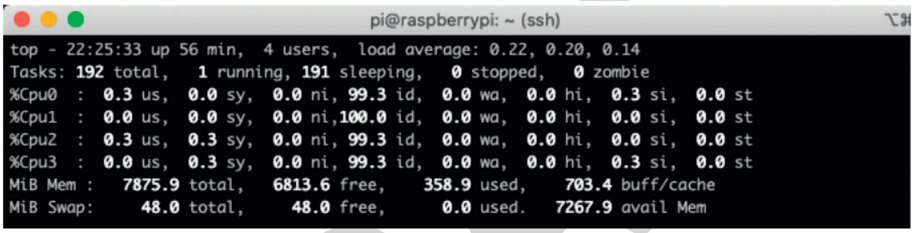
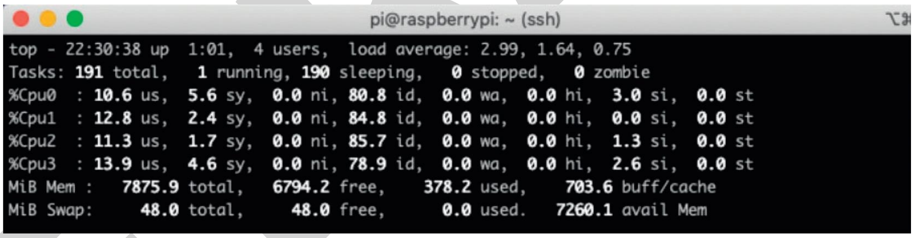


Figure 15: Packet loss rate under the condition of 1–1000 concurrent data per second.



```
top - 22:25:33 up 56 min,  4 users,  load average: 0.22, 0.20, 0.14
Tasks: 192 total,   1 running, 191 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.3 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
MiB Mem :   7875.9 total,   6813.6 free,    358.9 used,    703.4 buff/cache
MiB Swap:     48.0 total,     48.0 free,      0.0 used.   7267.9 avail Mem
```

Figure 16: Average load under 500 concurrent messages per second.



```
top - 22:30:38 up  1:01,  4 users,  load average: 2.99, 1.64, 0.75
Tasks: 191 total,   1 running, 190 sleeping,   0 stopped,   0 zombie
%Cpu0  : 10.6 us,  5.6 sy,  0.0 ni, 80.8 id,  0.0 wa,  0.0 hi,  3.0 si,  0.0 st
%Cpu1  : 12.8 us,  2.4 sy,  0.0 ni, 84.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  : 11.3 us,  1.7 sy,  0.0 ni, 85.7 id,  0.0 wa,  0.0 hi,  1.3 si,  0.0 st
%Cpu3  : 13.9 us,  4.6 sy,  0.0 ni, 78.9 id,  0.0 wa,  0.0 hi,  2.6 si,  0.0 st
MiB Mem :   7875.9 total,   6794.2 free,    378.2 used,    703.6 buff/cache
MiB Swap:     48.0 total,     48.0 free,      0.0 used.   7260.1 avail Mem
```

Figure 17: Average load under 1000 concurrent messages per second.

Table 3: Success rate of sending and receiving data within 48 hours.

Data source	Number of sent messages	Number of received messages	Success rate (%)
Environmental monitoring nodes	8640	8572	99.21
Pest identification nodes	2880	2845	98.78
Total data	11520	11463	99.10

working stably. A through checking of information sent and received by the cloud network within 48 hours, the environmental monitoring nodes sent 8640 messages and the pest identification nodes sent 2880 messages. The cloud received 8572 and 2845 messages, and the success rates were 99.21% and 98.78%, respectively. The average success rate of was 99.1%, as shown in Table 3.

5. Conclusion and Prospects

5.1. Conclusion. This paper designs an IoT gateway for crop growth environmental monitoring based on edge-computing technology. According to the characteristics of online monitoring data transmission, the Gateway uses virtualized container technology to package LoRaWAN server, pest identification, and environmental information data fusion functions into images. And it forms integrated operation mode of multiple function in agricultural field. The gateway applies message-oriented middleware to standardize and customize the data transmission among function modules, clouds, and edges.

The simulation result shows that the data transmission based on LoRaWAN protocol is stable. The packet loss rate is less than 5% and meets the transmission quality

RETRACTED

requirements. The gateway can achieve the functions of each module at the same time. In the stress testing of 500/1000 concurrent data per second, the average load is 0.22/2.99, and the system resource utilization is stable. In the field test, the average success rate of data transmission is 99.1%.

In this paper, the heterogeneous integration of two transmission protocols of LoRaWAN and Wi-Fi is used in farmland environmental monitoring for the first time. From the aspect of computing power distribution, functional modules such as pest identification and data fusion are equipped at the edge which can process the information uploaded by the terminal nodes in real-time. The edge-computing gateway reduces the pressure of cloud computing, lowers the transmission delay between cloud and edge. Also, it reduces the occupation rate of network bandwidth resources and further improves the transmission efficiency. In summary, the edge-computing gateway we designed has the advantages of low cost, low latency, and low power consumption compared with similar devices which has practical significance.

5.2. Prospects. Although this paper has done some actual work on the establishment of gateway based on edge-computing technology and conducted tests, there are still some problems to be solved. For example, when the monitoring range of farmland is very large, Wi-Fi transmission will be limited. How to combine 5G technology will be the research object in future. In addition, there are other two research axes we wish to develop: (1) Further improve the designed edge-computing gateway and add more module functions, such as automatic control module, so as to study the automatic control system of water and fertilizer. (2) Continue to do research on agricultural IoT, we are going to apply the edge-computing gateway to the large-scale farmland environmental monitoring system and form a regional grid. At the same time, combined with the cloud platform, make it become an important part in intelligent agriculture.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by Key R&D Program of Department of Science and Technology in Jilin Province (Grant no. 20200504004YY) and National Natural Science Foundation of China (NSFC) (Grant no. 32001418).

References

[1] W. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, “Edge computing-an emerging computing model for the Internet of everything era,” *Journal of Computer Research and Development*, vol. 54, no. 5, pp. 907–924, 2017.

[2] J. Pan and J. Mcelhannon, “Future Edge Cloud and Edge Computing for Internet of Things Applications,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–499, 2018.

[3] W. Shi, C. Cao, Z. Zhang, Y. Li, and L. Xu, “Edge computing: vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, and F. Lebeau, “A new Edge Architecture for AI-IoT services deployment,” *Procedia Computer Science*, vol. 175, pp. 10–19, 2020.

[5] X. Li, Z. Zhou, J. Guo, S. Wang, and J. Zhang, “Aggregated multi-attribute query processing in edge computing for industrial IoT applications,” *Computer Networks*, vol. 151, pp. 114–123, 2019.

[6] P. Fraga-Lamas, P. Lopez-Iturri, M. Celaya-Echarri et al., “Design and empirical validation of a bluetooth 5 fog computing based industrial CPS architecture for intelligent industry 4.0 shipyard workshops,” *IEEE Access*, vol. 8, pp. 45496–45511, 2020.

[7] M. Ghazal, T. Basmaji, M. Yaghi, M. Alkhedher, M. Mahmoud, and A. S. El-Baz, “Cloud-based monitoring of thermal anomalies in industrial environments using AI and the Internet of robotic Things,” *Sensors*, vol. 20, no. 21, p. 6348, 2020.

[8] H. Tang, D. Li, J. Wan, M. Imran, and M. Shoaib, “A reconfigurable method for intelligent manufacturing based on industrial cloud and edge intelligence,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4248–4259, 2020.

[9] S. Zhao, F. Li, H. Li, and R. Lu, “Smart and practical privacy-preserving data aggregation for fog-based smart grids,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 521–536, 2020.

[10] M. Peng, T. Q. Quek, G. Mao, Z. Ding, and C. Wang, “Artificial-intelligence-driven fog radio access networks: recent advances and future trends,” *IEEE Wireless Communications*, vol. 27, no. 2, pp. 12-13, 2020.

[11] P. O’Donovan, C. Gallagher, K. Leahy, and D. T. O’Sullivan, “A comparison of fog and cloud computing cyber-physical interfaces for Industry 4.0 real-time embedded machine learning engineering applications,” *Computers in Industry*, vol. 110, pp. 12–35, 2019.

[12] M. Salhaoui, A. Guerrero-González, M. Arioua, F. J. Ortiz, A. El Oualkadi, and C. L Torregrosa, “Smart industrial IoT monitoring and control system based on UAV and cloud computing applied to a concrete plant,” *Sensors*, vol. 19, no. 15, p. 3316, 2019.

[13] M. Ali, A. Anjum, O. Rana, A. R. Zamani, D. Balouek-Thomert, and M. Parashar, “RES: real-time video stream analytics using edge enhanced clouds,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 792–804, 2020.

[14] R. Morabito and N. Beijar, “Enabling Data Processing at the Network Edge through Lightweight Virtualization Technologies,” in *Proceedings of the IEEE International Conference on Sensing*, pp. 1–6, IEEE, London, UK, June 2016.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8238375, 13 pages
https://doi.org/10.1155/2022/8238375

Research Article

Human Gait Analysis: A Sequential Framework of Lightweight Deep Learning and Improved Moth-Flame Optimization Algorithm

Muhammad Attique Khan,[1] **Habiba Arshad,**[2] **Robertas Damaševičius,**[3] **Abdullah Alqahtani,**[4] **Shtwai Alsubai,**[4] **Adel Binbusayyis,**[4] **Yunyoung Nam**,[5] **and Byeong-Gwon Kang**[5]

[1] *Department of Computer Science, HITEC University, Taxila, Pakistan*
[2] *Department of Computer Science, University of Wah, Wah Cantt, Pakistan*
[3] *Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania*
[4] *College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia*
[5] *Department of ICT Convergence, Soonchunhyang University, Asan 31538, Republic of Korea*

Correspondence should be addressed to Byeong-Gwon Kang; bgkang@sch.ac.kr

Received 16 April 2022; Revised 23 May 2022; Accepted 3 June 2022; Published 14 July 2022

Academic Editor: Muhammad Ahmad

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

Human gait recognition has emerged as a branch of biometric identification in the last decade, focusing on individuals based on several characteristics such as movement, time, and clothing. It is also great for video surveillance applications. The main issue with these techniques is the loss of accuracy and time caused by traditional feature extraction and classification. With advances in deep learning for a variety of applications, particularly video surveillance and biometrics, we proposed a lightweight deep learning method for human gait recognition in this work. The proposed method includes sequential steps–pretrained deep models selection of features classification. Two lightweight pretrained models are initially considered and fine-tuned in terms of additional layers and freezing some middle layers. Following that, models were trained using deep transfer learning, and features were engineered on fully connected and average pooling layers. The fusion is performed using discriminant correlation analysis, which is then optimized using an improved moth-flame optimization algorithm. For final classification, the final optimum features are classified using an extreme learning machine (ELM). The experiments were carried out on two publicly available datasets, CASIA B and TUM GAID, and yielded an average accuracy of 91.20 and 98.60%, respectively. When compared to recent state-of-the-art techniques, the proposed method is found to be more accurate.

1. Introduction

Person recognition and identification using gait have great importance in the field of machine learning and computer vision [1]. Gait is the walking behavior of a person but to recognize a person by gait from distance and in less illuminated environment it becomes very complicated and difficult [2]. Moreover, as compared to other traditional biometric techniques such as fingerprint, face detection, and iris detection, it does not require direct contact of a person [3]. Due to these discriminative factors, it has taken a lot of attention from researchers and it is used to apply in various applications like security surveillance, dubious person detection, and forensics [4, 5].

In early research, gait recognition was categorized in to two main categories such as model-based and appearance-based [6]. The prior categories are more costly to implement the human model using high-resolution videos and give low average results as compared to modern categories; hence, researchers focus on using modern categories for gait feature detection [7]. In the model-based method, prior information is used to extract the moving features of the human body [8]. Furthermore, in this method, the movement of human body is examined using changing factors like gait path, position of joints, and the torso [9]. This is a challenging method, due to its high computational complications. In the model-free method, gait cycle is used to extract the features from a silhouette, and it is simple to implement due to less computational cost [10].

There are various machine learning and computer vision techniques that are used to overcome the covariate factors like angular deviations [11], lightning conditions [12], and clothing and carrying bags [6, 13], but there exist various challenges in extracting useful features that affect the optimal accuracy results. Feature extraction is considered as the most important step in recognizing gait traits [14], such as if the extracted features are related to the problem, then the system will be able to correctly recognize the human gait patterns. In contrast, if irrelevant features are evaluated, then the system performance will go down and it will not give optimal recognition results [10]. In past, various types of features are used like shape-based features [15], geometrical features [16], and statistical features [17]. Deep features are extracted using deep convolutional neural network techniques to overcome these challenges. Deep learning techniques, rather than manual feature extraction, extract automated features from raw images [18, 19]. In this work, we proposed a sequential lightweight deep learning architecture for human gait recognition. Our major contributions are listed as follows:

(i) Two pretrained deep learning models are modified namely VGG-19 and MobileNet-V2 based on the target dataset classes and adjusted their weights. Then, both models are trained using transfer learning without freezing any layer and obtained newly trained models.

(ii) Feature engineering is performed on fully connected layer 7 (VGG-19) and global pooling layer (MobileNet-V2) and fused by employing discriminant correlation analysis (DCA)

(iii) A modified moth-flame optimization algorithm is developed for the selection of optimum features that are finally classified using extreme learning machine (ELM)

The rest of the article is organized as follows: Section 2 describes the manuscript's related work. Section 3 discusses the specifics of selected datasets. The proposed methodology is presented in Section 4. Section 5 discusses and explains the experimental results. Finally, Section 6 brings the entire manuscript that followed the references to a close.

2. Related Works

Identification of human through gait is the most biometric application, and researchers have made extensive studies for it by extracting feature values [20]. In literature, various machine learning and computer vision-based techniques are implemented for human gait recognition [21]. Liao et al. [22] presented Pose-gait model-based gait recognition method. In this approach, the human 3D pose is estimated using CNN, and then spatiotemporal features extracted from the 3D pose are used for the improvement in recognition. Two publicly available datasets CASIA B and CASIA E are used for experimentation, and it gives auspicious results in the presence of covariate factors. Sanjay et al. [23] introduced an automated approach for human gait recognition in the presence of a covariate situation. In the first step, basic distinct stances in the gait cycle are detected which are used to compute the gait features related to these detected stances and it is termed as dynamic gait energy image (DGEI). Then generative adversarial network (GAN) is used to detect the corresponding dynamic gait energy image. These extracted features are then used to compare with the gallery sequences. Finally, GAN-created DGEI is used for final recognition. Publicly available datasets such as CASIA B, TUM Gait, and OU-ISIR TreadMill B are used to validate the presented approach, and it gives considerably improved results as compared to existing methods. Chen et al. [24] introduced a method for cross-view gait recognition using deep learning. Multiview gait generative adversarial network is introduced for creating fake gait data samples for extension in existing data. The method is then used to train each instance of each view involved in single or multiple datasets. Domain alignment using projected maximum mean dependency (PMMD) is utilized to minimize the effect of distribution divergence. CASIA B and OUMVLP are used for experimentation, and the achieved results show that the introduced method gives better results than existing methods. Hou et al. [25] presented a set residual network-based gait recognition model to detect more discriminative features from the silhouettes. Set residual block is introduced to extract the silhouette level and two-level features in a parallel sequence, and then the residual connection is applied to join the two-level features. Moreover, an efficient method is applied to utilize features from the deep layers. Two datasets CASIA B and OUMVLP are used for experimentation. The applied approach gives consistent results as compared to existing methods. Gul et al. [13] introduced a machine vision method to extract the distinct gait features from covariate factors. Spatiotemporal gait features are extracted, and these features are used to train the 3D CNN model to overcome these challenges. Then, the holistic method is used by the model to implement the distinct gait features in the form of gait energy images. Two publicly available datasets OULP and CASIA B are used to test the validity of the introduced method with large gender and age differences. The presented approach gives promising results using CASIA B dataset as compared to existing methods. The methods presented above concentrated on both spatial and temporal data. None of them focused on feature fusion or optimization of extracted features to achieve better results in the shortest amount of time. As a result, in this article, we proposed a lightweight deep learning framework for human gait recognition that not only improves accuracy but also reduces a system's computational time.

3. Datasets

3.1. TUM GAID Dataset. TUM Gait from Audio, Image and Depth (GAID) [26] dataset consists of RGB audio, video, and depth. It consists of 305 subjects carried out in two indoor walking sequences in which four distinct situations are captured without any view variation through Microsoft Kinect like six normal walk videos ($n1$–$n6$), two videos with carrying a bag ($b1$–$b2$), and two walking videos wearing coating shoes ($s1$-$s2$), and there is an elapsed time instance in which 32 subjects were recorded wearing distinct cloths. A few sample images are shown in Figure 1.

3.2. CASIA B Dataset. CASIA B [27] is a multiview and indoor gait dataset in which 124 subjects are included in recording session of which 93 are male participants and 31 female participants. This dataset considers major factors for gait recognition, that is variation in view angle, clothing, and carrying situations separately. For all, subject videos are captured through a USB camera from 11 different views that include six normal walking videos (NM), two walking videos with wearing a coat (CL), and two walking videos with carrying a bag (BG). A few sample images are shown in Figure 2.

4. Methodology

The proposed lightweight (LW) human gait recognition framework has been presented in this section with detailed mathematical formulations and flow diagrams. The main architecture is shown in Figure 3. In this figure, it is illustrated that the proposed method consists of some important steps such as modification of pretrained CNN models, training of modified models using transfer learning (TL), feature engineering on global average pooling layers, fusion of extracted deep features using discriminant correlation analysis (DCA), selection of best features using the modified moth-flame optimization algorithm, and finally classification using the extreme learning machine (ELM). The details of each step are given.

4.1. Convolutional Neural Network (CNN). The convolutional neural network (CNN) has become an important recognition task in the domain of computer vision. The CNN architecture is employed for feature extraction of an image based on several hidden layers. CNNs have many layers, including convolutional, pooling, fully connected, and others. The convolution layer (CL) is the most important layer of a CNN that performed a 2D convolution process on the input and the kernels through a forward pass. The kernel weights in every CL are assigned randomly and their values are changed at each step by applying the loss function through network training. In the end, the resultant learned kernels may identify some types of shapes within the input images. In CL, three different types of steps are performed like convolution, stack, and nonlinear activation function.

Suppose, we have an input matrix M and an output Z of the CL, and there are some set of kernels $K_l, \forall l \in [1, \ldots, L]$, then the output of the convolution process $O(l)$ after step 1 is represented as

$$O(l) = M \otimes K_l, \quad \forall l \in [1, \ldots, L], \tag{1}$$

where $\otimes$ refers to the convolution process, which is the product of filter and inputs. Second, all $O(l)$ activation maps are combined to create a novel 3D activation map.

$$R = Q(O(1), \ldots, O(L)), \tag{2}$$

where Q represents the combination of operations with channel direction, and L is the total number of filters. Third, the 3D activation map R is given as input into the activation function and gives the resultant activation map as

$$Z = \mathrm{NLAF}(R). \tag{3}$$

The size Q of three main matrices (input, filters, and result) is taken as

$$Q(i) = \begin{cases} T_A \times U_A \times V_A, & i = I, \\ T_P \times U_P \times V_P, & i = K_l, \forall l \in [1, \ldots, L], \\ T_N \times U_N \times V_N, & i = Z, \end{cases} \tag{4}$$

where the variables (T, U, V) represent the size of height, width, and channels of the activation map, and the subscripts A, P, and N represent input, filter, and output, respectively. It contains two equalities. First, $V_A = V_P$ refers the channel of input V_A equals to the channels of filter V_P. Second, $V_N = L$ refers the channels of output V_N equals the number of filters L. Suppose Y represents padding, S represents stride, so the result of T_N, U_N, V_N can be evaluated as

$$\begin{aligned} T_N &= 1 + d_{de}\left[\frac{(2 \times Y + T_A - T_P)}{S}\right], \\ U_N &= 1 + d_{de}\left[\frac{(2 \times Y + U_A - U_P)}{S}\right], \end{aligned} \tag{5}$$

where d_{de} is the floor function. The nonlinear activation Υ generally selects the rectified linear unit (ReLU) function [28].

$$\begin{aligned} \gamma_{\mathrm{ReLU}}(r_{mn}) &= \mathrm{ReLU}(r_{mn}) \\ &= \max(0, r_{mn}), \end{aligned} \tag{6}$$

where $r_{mn} \in R$ is the component of the activation map R. At present, ReLU is the mostly used NLAF as compared to the traditional hyperbolic tangent (HT) and sigmoid function (SM) function, that are computed as

$$\begin{aligned} \gamma_{\mathrm{HT}}(r_{mn}) &= \tanh(r_{mn}) = \frac{(f^{r_{mn}} - f^{-r_{mn}})}{(f^{r_{mn}} + f^{-r_{mn}})}, \\ \gamma_{\mathrm{SM}}(r_{mn}) &= (1 + f^{-r_{mn}})^{-1}. \end{aligned} \tag{7}$$

4.2. Transfer Learning. Transfer learning (TL) is the branch of machine learning that transfer the knowledge of one domain to a different domain within less computational time. Given a

Figure 1: Example images from the TUM GAID gait dataset [26].

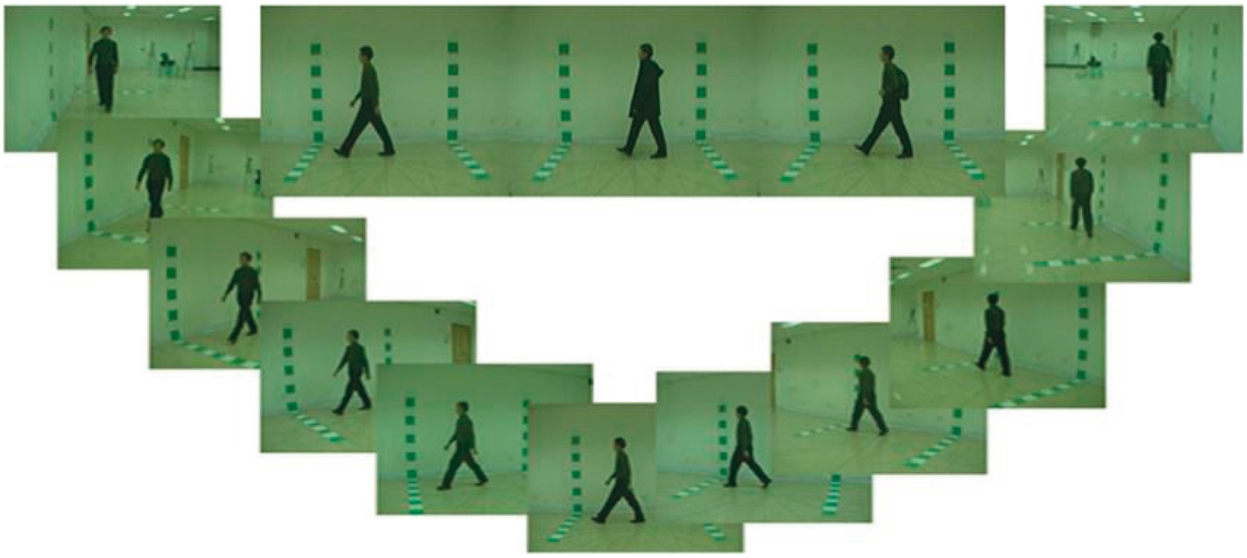


Figure 2: Sample images of the CASIA B gait dataset [27].

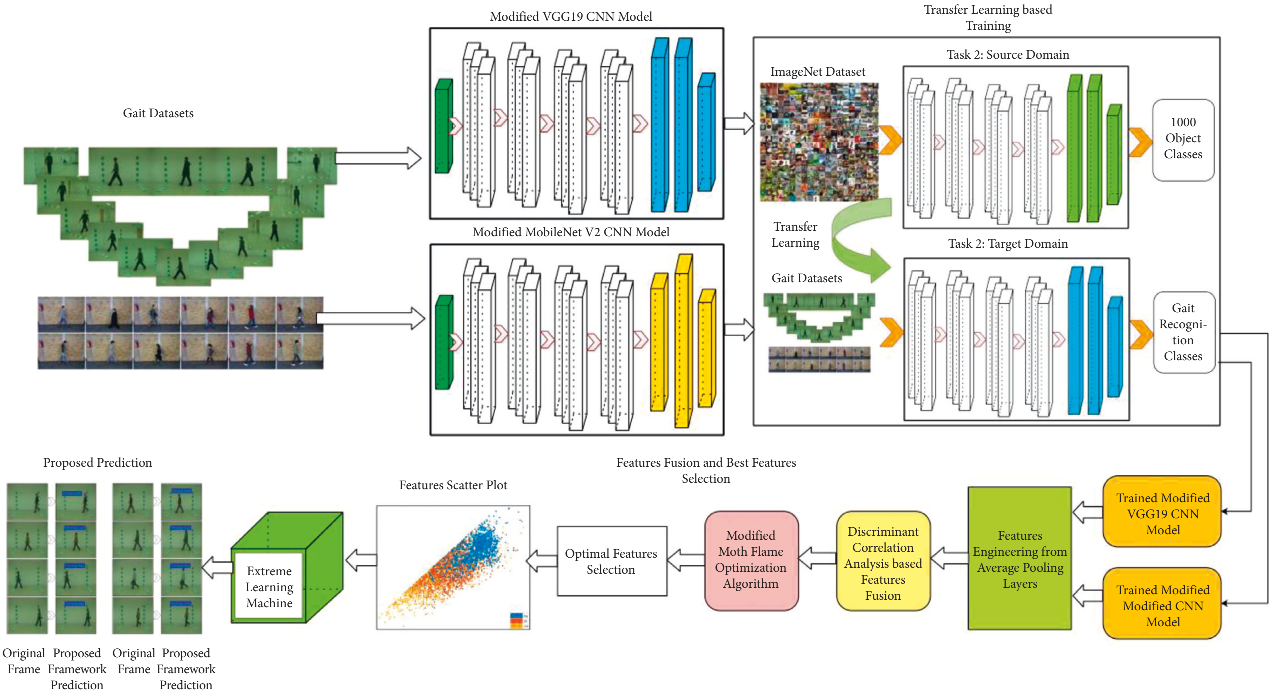

Gait Datasets
Modified VGG19 CNN Model
Modified MobileNet V2 CNN Model
Transfer Learning based Training
Task 2: Source Domain
ImageNet Dataset
1000 Object Classes
Transfer Learning
Task 2: Target Domain
Gait Datasets
Gait Recogni- tion Classes
Trained Modified VGG19 CNN Model
Trained Modified Modified CNN Model
Features Engineering from Average Pooling Layers
Features Fusion and Best Features Selection
Discriminant Correlation Analysis based Features Fusion
Modified Moth Flame Optimization Algorithm
Optimal Features Selection
Features Scatter Plot
Extreme Learning Machine
Proposed Prediction
Original Frame
Proposed Framework Prediction
Original Frame
Proposed Framework Prediction

Figure 3: Proposed main flow of human gait recognition using lightweight deep learning architecture.

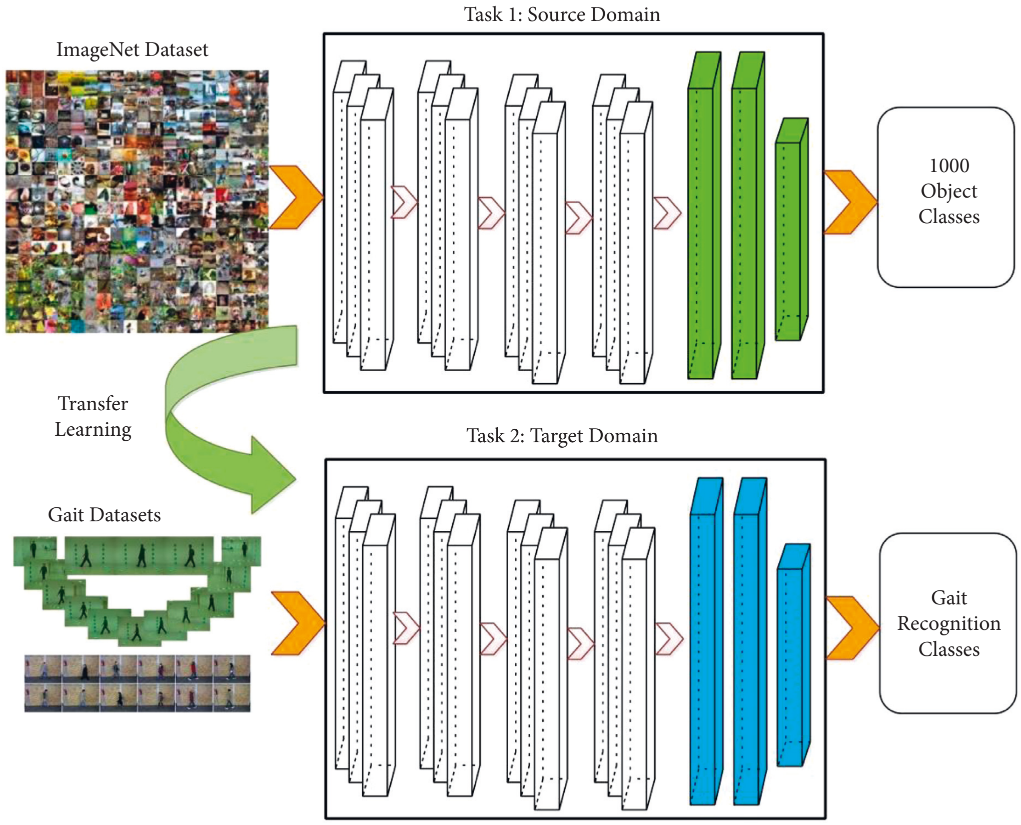


Figure 4: Transfer learning-based training of modified CNN models for human gait recognition.

source domain $\xi^S = \left\{F^S, P^S(\phi^S)\right\}$ with the source task $\lambda^S = \left\{\widetilde{F}^S, f^S(\cdot)\right\}$ and a target domain $\xi^T = \left\{F^T, P^T(\phi^T)\right\}$ with the target task, $\lambda^T = \left\{\widetilde{F}^T, f^T(.)\right\}$ aims to learn a better mapping function $f^T(.)$ for the target task λ^T for the knowledge transfer from the source domain ξ^S and task λ^S. Hence, the TL is expressed as follows:

$$\xi^S \neq \xi^T \text{ or } \lambda^S \neq \lambda^T, \text{ where } \phi = v|v_i \in F, \quad i = 1, 2, 3, \ldots N. \tag{8}$$

Hence, DTL is defined as follows: given a TL task $f^{S\longrightarrow T}(.)$: $F^T \longrightarrow \widetilde{F}^T$ based on $[\xi^S, \xi^T, \lambda^S, \lambda^T]$, DTL objectives to acquire the $f^{S\longrightarrow T}(.)$ by leveraging powerful DL process. Visually, the DTL process is shown in Figure 4.

4.3. Modified VGG-19 Model. Visual geometry group (VGG)-19 [29] is the modified version of VGG, and it consists of 19 layers with 16 convolutional layers of 64, 128, and 256, and 512 filter sizes with stride length and padding is 1 pixel on each side. The convolutional layer consists of 5 sets, two of them contain 64 filters, the next set contains 2 convolutional layers with 128 filters, the next set contains 4 convolutional layers with 256 filters, and the last two sets contain 4 convolutional layers with 512 filters each. Then, a max pooling layer with a 2 × 2 filter size and stride rate of 2 pixels are used after each set of convolutional layers. The output is then passed to the fully connected layer. Three fully connected (FC) layers and one softmax layer are used for the classification. In this work, we removed the last fully connected (FC) layer and added a new FC layer. Then, several hyperparameters are employed such as training rate, epochs, mini batch size, optimizer, and training loss. Based on these hyperparameters, we trained the modified model from scratch through TL and obtained a new model for only gait recognition task. Later, this modified model is used for the feature engineering task.

4.4. Modified MobileNet-V2 Model. MobileNet-V2 [30] is a lightweight CNN-based model specially designed for mobile devices. This architecture can perform well on small datasets as it can overcome the effect of overfitting and it also optimized the memory consumption. In this network, 17 inverted residuals are used between two convolutional layers and one FC layer. So, the depth of the network consists of 53 convolutional layers and one FC layer. The working of this architecture is based on two concepts that include depth-wise separable convolution and the inverted residual methods. In this architecture, a full convolutional layer is replaced with a factorized version that divides the convolution into two separate groups. The first layer is named as depth-wise convolution; its function is to do lightweight filtering by using one convolutional filter on each input channel. The second layer is named as point-wise convolutional layer, which is used for creating new features by computing linear models of the input channels. The depthwise convolutional layers contains 2 convolutional layers: the first layer contains a 3 × 3 filter size, while the other one has a 1 × 1 filter size. The other two regular convolutional layers have filter sizes of 3 × 3 and 1 × 1. Moreover, in this architecture, ReLU6 is used instead of ReLU as it is more efficient for less accurate computation. Dropout and batch normalization are then applied, where layer activators are standardized to mean zero and unit variance, and then, linear transformation is applied [31].

In this work, we removed the last layer and added a new FC layer. Then, several hyperparameters are employed such as training rate (0.005), epochs (100), mini batch size (32), optimizer (Adam), and training loss. Based on these hyperparameters, we trained the modified model from scratch through TL and obtained a new model for only gait recognition task.

4.5. Feature Engineering. Feature engineering is applied on global average pooling layers of both models and obtained two feature vectors of dimension $\times K_1$ and $N \times K_2$. Mathematically, this process is defined as follows:

$$\begin{aligned} K_1 &= \text{activation}\,(\text{layer}), \\ K_2 &= \text{activation}\,(\text{layer}), \end{aligned} \tag{9}$$

where K_1 and K_2 represent the length of feature vectors, and layer defines the selected one like global average pooling. Thereafter, the fusion process is performed using discriminant canonical correlation analysis approach.

4.6. Discriminative Canonical Correlation Analysis-Based Fusion. In this work, the DCCA fusion approach is employed for feature fusion. By applying canonical correlation analysis (CCA), the correlated features $l_m^v m_t$ and $l_n^v n_t$, $t = 1, \ldots, z$, are extracted and merged for identification. Though the features extracted from related class samples are not utilized, resultantly it becomes the constraint of the recognition capabilities of CCA. Moreover, the basic concept to introduce CCA is for modeling instead of recognition, and correlation β refers to the certainty among $l_m^v m_t$ and $l_n^v n_t$, $t = 1, \ldots, z$. CCA was more often utilized for modeling and estimation, for instance image extraction and parameter prediction. If the extracted features are for recognition, then the class description of the instances should be utilized to get more discriminatory features. Finally, the class description was fused with the CCA framework for cooperated feature extraction and presented an innovative approach of fused feature extraction for multimodal recognition, termed as discriminative canonical correlation analysis (DCCA). Mathematically, this approach is defined as follows.

Suppose z pairs of mean-normalized pairwise instances $\{(m_t, n_t)\}_{t=1}^{z} \in \xi^a \times \xi^b$ access from p classes, DCCA can be systematically represented in the below optimization problem:

$$\max_{l_m, l_n} \left(l_m^v S_l l_n - \mu . l_m^v S_h l_n\right), \tag{10}$$

$$\begin{aligned} \text{s.t.}\, l_m^v M M^v l_m &= 1, \\ l_n^v N N^v l_n &= 1, \end{aligned} \tag{11}$$

where the parameters S_l and S_h are used to compute the inside-class association and between-class association, respectively (detailed description is given below), $\mu > 0$ adjustable metric that shows the comparative significance of the inside-class association $l_m^v S_l l_n$ contrasted with the between-class association $l_m^v S_h l_n$. Moreover, the limitation value represents the scale limitation on l_m, l_n.

$$\begin{aligned} M &= \left[m_1^{(1)}, \ldots, m_{z_1}^{(1)}, \ldots, m_1^{(p)}, \ldots, m_{z_c}^{(p)}\right], \\ N &= \left[n_1^{(1)}, \ldots, n_{z_1}^{(1)}, \ldots, n_1^{(p)}, \ldots, n_{z_c}^{(p)}\right], \\ f_{z_t} &= \left[\underbrace{0, \ldots 0,}_{\sum} k^{t-1} z_k \underbrace{1, \ldots 1,}_{z_t} \underbrace{0, \ldots 0}_{z - \sum_{k \leq 1}^{t-1} z_k}\right]^v \in Q^z, \\ 1_z &\leq [1, \ldots 1]^V \in Q^z, \end{aligned} \tag{12}$$

where $m_k^{(t)}$ refers the kth instance in the tth class, similarly $n_k^{(t)}$, and z_t shows the number of instances of $m_k^{(t)}$ or $n_k^{(t)}$ in the tth class. The matrix S_l is represented as

$$\begin{aligned} S_l &= \sum_{t=1}^{p} \sum_{d=1}^{z_t} \sum_{y=1}^{z_t} m_d^{(t)} n_y^{(t)V} \\ &= \sum_{t=1}^{p} \left(M f_{z_t}\right)\left(N f_{z_t}\right) \\ &= M G N^V, \end{aligned} \tag{13}$$

$$G = \begin{bmatrix} 1_{z_1 \times z_1} & & & & \\ & \ddots & & & \\ & & 1_{z_t \times z_t} & & \\ & & & \ddots & \\ & & & & 1_{z_p \times z_p} \end{bmatrix} \in \xi^{z \times z}, \tag{14}$$

where G represents a proportionate, positively defined, block crosswise matrix, and Matrix $(G) = p$. In contrast, the matrix S_h is represented as

$$\begin{aligned} S_h &= \sum_{t=1}^{p} \sum_{\substack{k=1 \\ k \neq t}}^{p} \sum_{d=1}^{z_t} \sum_{y=1}^{z_k} m_d^{(t)} n_y^{(t)V} \\ &= \sum_{t=1}^{p} \sum_{k=1}^{p} \sum_{d=1}^{z_t} \sum_{y=1}^{z_k} m_d^{(t)} n_y^{(t)V} - \sum_{t=1}^{p} \sum_{k=1}^{z_t} \sum_{d=1}^{z_t} m_d^{(t)} n_y^{(t)V} \\ &= \left(M 1_z\right)\left(N 1_z\right)^V - \mathrm{MGN}^V \\ &= -\mathrm{MGN}^V. \end{aligned} \tag{15}$$

The "=" in the end holds because mean normalization is applied on the instances, hence both $M1_z = 0$ and $N1_z = 0$ holds. In contrast between equations (13) and (15), the only difference among S_l and S_h is a single negative sign, thus the objective of equation (10) will be $(1+\mu)l_m^v S_l l_n$, and this enhancement issue is free from parameter μ, consequently μ can be excluded. Hence, DCCA can be represented as

$$\begin{aligned} &\max_{l_m,l_n}\left(l_m^v \mathrm{MGN}^V l_n\right),\\ \text{s.t. } &l_m^v MM^v l_m = 1, l_n^v NN^v l_n = 1. \end{aligned} \tag{16}$$

By applying the Lagrangian multiplier technique, it becomes very simple to get main equation of DCCA, which is represented as

$$\begin{pmatrix} - & \mathrm{MGN}^V \\ \mathrm{NGM}^v & - \end{pmatrix}\begin{pmatrix} l_m \\ l_n \end{pmatrix} = \vartheta\begin{pmatrix} MM^V & \\ & YY^V \end{pmatrix}\begin{pmatrix} l_m \\ l_n \end{pmatrix}. \tag{17}$$

When the vector pairs (l_{mt}, l_{nt}), $i = 1, \ldots, g$, adjacent to the first g largest generalized eigenvalues and attained, let $L_m = [l_{m1}, \ldots, l_{mg}]$, $L_n = [l_{n1}, \ldots, l_{ng}]$, then both feature extraction and the feature fusion can be performed using FFS-I and II, respectively, where g fulfills the limitations $g \leq \min(a, b)$ and $g \leq p$. The formulation returned a fused vector of dimension $N \times K_3$, where $K_3 \in (K_1, K_2)$. Later on, this resultant vector is further improved using a modified moth-flame optimization algorithm.

4.7. Moth-Flame Optimization Algorithm (MFO). Several nature-inspired optimization algorithms have been introduced in the literature for best feature selection such as genetic algorithm, particle swarm optimization, and moth-flame optimization [32]. The improved moth-flame optimization algorithm is utilized in this work for the best feature selection. Originally, the MFO algorithm was presented by Mirjalili [33]. It is under the populace-based metaheuristics algorithm. In this procedure, first the data flow of MFO begins by randomly generating moths within the resultant space. Then it calculates the positional (i.e., fitness) value of each moth and label the best position by flame. Afterwards, changing the moth place depends on a whole movement function used to attain a better position labeled by a flame. Moreover, it updates the new best positions of the individual. The previous process (i.e., updating of moths' location and generating the new location) until it meets the resultant criteria. The MFO algorithm consists of three major steps that are as follows.

4.7.1. Creating the Initial Population of Moths. As stated in [33], it is supposed that an individual moth can fly in 1D, 2D, 3D, or in hyper-dimensional position. The matrix of moths can be represented as

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & \cdots & h_{1,a} \\ h_{2,1} & h_{2,2} & \cdots & \cdots & h_{2,a} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{m,1} & h_{m,1} & \cdots & \cdots & h_{m,a} \end{bmatrix}, \tag{18}$$

where m represents the number of moths' and a represents the number of dimensions in the resultant region. Moreover, the fitness values for entire moths' stored in an array are represented as

$$VH = \begin{matrix} VH_1 \\ VH_2 \\ \vdots \\ VH_m \end{matrix}. \tag{19}$$

The remaining elements in the algorithm are flames that are represented using D-dimensional space with their fitness/position value function in the following matrix set as

$$\begin{aligned} P &= \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & \cdots & P_{1,a} \\ P_{2,1} & P_{2,2} & \cdots & \cdots & P_{2,a} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{m,1} & P_{m,1} & \cdots & \cdots & P_{m,a} \end{bmatrix},\\ VP &= \begin{bmatrix} VP_1 \\ VP_2 \\ \vdots \\ VP_m \end{bmatrix}. \end{aligned} \tag{20}$$

It is important to note that moths and flames both are solutions. The moths are the real search agents that revolve around the search area, while flames are the moth's best position that is obtained yet. Hence, an individual moth hunts around a flame and updates it when it finds the best solution. Following this procedure, a moth never misses its best solution.

4.7.2. Updating Moths' Location/Positions. MFO utilizes three distinct functions to convergent the global optimum of the optimization issues. Mathematically, it is defined as follows:

$$\mathrm{MFO} = (L, M, E), \tag{21}$$

where L represents the first random positions of the moths ($\pi l\colon \varnothing \longrightarrow \{H, VH\}$), M represents that the motion of the moths in the search is ($M\colon H \longrightarrow H$), and E represents end of the search process ($E\colon H \longrightarrow \text{true}, \text{false}$). The equation given below represents L function, which is used for the implementation of random distribution:

$$H(x, y) = (UB(x) - LB(y) \times \text{rand} + LB(x), \tag{22}$$

where UB and LB refers to the upper and lower bound variables, respectively. As discussed before, the moths fly in the search area by means of transverse direction. There are

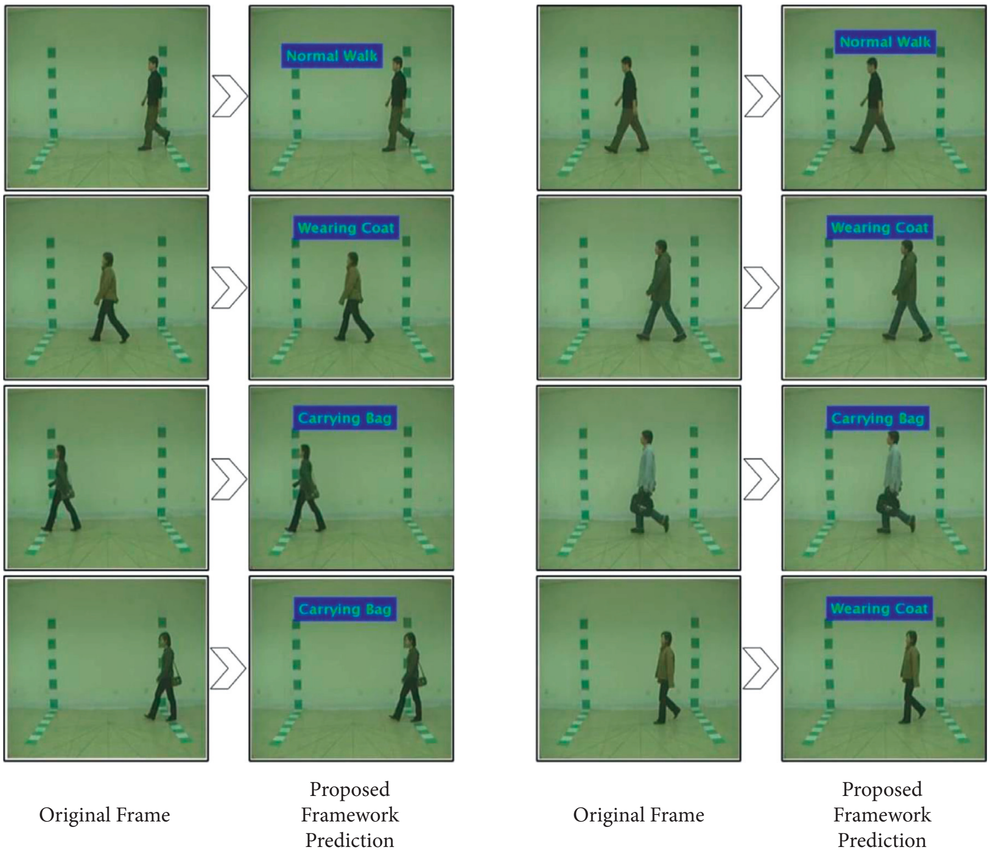


Figure 5: Proposed architecture labeled results on the CASIA B dataset.

three conditions that should be followed when applying a logarithmic spiral: (i) The spiral starting point should start from the moth; (ii) the spiral endpoint should be the location of the flame, and (iii) variation in the range of spiral should not extend from the search area. Thus, in the MFO algorithm, the logarithmic spiral can be defined as

$$S\left(H_x, P_y\right) = R_x.z^{qb}.\cos(2\pi b) + P_y, \tag{23}$$

where R_x represents space between the xth moth and yth flame (computed by equation 24), q represents a solution to define the shape of the logarithmic spiral, b represents a random range between [−1, 1].

$$R_x = \left|P_y - H_x\right|. \tag{24}$$

In MFO, the equalization among exploitation and examination is affirmed by the spiral motion of the moth near the flame in the search area. Moreover, to escape from falling in the trap of the local goal, the best solution has been kept in each step, and the moths fly around the flames by means of *VP* and *VH* matrices. Then, the update criteria are defined as follows:

4.7.3. Updating the Size of Flames. This part highlights to augment the manipulation of the MFO algorithm (i.e., updating the moths' location in m various positions in the search area may minimize the chance of exploitation of the best optimal solutions). However, minimizing the extent of flames helps to overcome this problem using the following equation:

$$\text{FLAME NO} = \text{ROUND}\left(O - c \times \frac{O - c}{I}\right), \tag{25}$$

where O refers to the maximum number of flames, c refers the current number of iterations, and I represents the maximum number of iterations. This equation returns the best features; however, during the analysis stage, it is observed that the best selected features contain some redundant information; therefore, we tried to overcome this problem and speedup the selection process based on Newton Raphson (NR) formulation. Mathematically, the NR method is defined as follows:

$$\delta_n = \delta_{n-1} - \frac{M\left(\delta_{n-1}\right)}{M'\left(\delta_{n-1}\right)}, \tag{26}$$

where $\delta_n \in Sf$ and Sf represent the selected features of moth-flame. Through the above formulation, a stop value is obtained that added in equation 25 for final selection.

$$\text{FLAME NO} = \text{ROUND}\left(\delta - cf \times \frac{\delta - cf}{I}\right). \tag{27}$$

The final selected features are passed to the extreme learning machine (ELM) for classification. A few visual predicted frames are shown in Figure 5.

Table 1: Proposed classification results of human gait recognition on the CASIA B dataset.

Method	Class	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Mean
LightweightDeep-ELM	NM	97.1	98.2	95	93.8	98.1	97.5	98.3	97.2	98	94	98.6	**96.89**
	BG	94.2	95.3	91	92.7	93	89.7	94.8	93.1	92.8	91.8	95.4	**93.07**
	CL	78.8	83.5	82.1	86.3	78.5	90.2	85.9	82	81.3	88.3	83.4	**83.66**
LightweightDeep-SVM	NM	96.2	97.5	96.1	92.8	97.9	96.2	98	97.5	97.1	93.2	98	96.40
	BG	92.5	93.6	93	92.2	94.1	87.6	94	93.5	92.1	92.4	93.4	92.58
	CL	79	82.8	81.5	86	79.1	87.4	83.9	82.3	80.7	87.5	82.1	82.93
LightweightDeep-FKNN	NM	93.2	94.5	92.9	93.1	94.6	91.8	93.5	94.8	94.5	93	94.1	93.63
	BG	87	89.1	90.5	89.6	91.4	82.6	90.8	89.4	87	89.3	90.7	88.85
	CL	73.5	78.4	77.2	81.6	75.3	82.2	80	77.5	76.1	82.4	78.5	78.42
LightweightDeep-EBT	NM	92.8	93.3	90.4	91.5	95	92.6	92.4	93.1	94	93.7	92.9	92.88
	BG	88.5	87.4	90.1	90.5	91.8	82	90.2	90.4	86.1	89.8	91.9	88.97
	CL	72.6	80.1	77	80.4	75	81.6	80.6	76.2	78.5	82	78.1	78.37
LightweightDeep-DT	NM	87.4	88.9	90.1	91.6	93	90.5	88	91.2	91.3	87.4	90.5	89.99
	BG	81.5	82.6	87.5	83.8	90.4	80.6	90	87.9	85.3	84	90.1	85.79
	CL	69.8	72.1	77	78.5	72.7	80	78.3	72.5	72.9	80.1	80.2	75.82

Bold values indicate the best values.

5. Experimental Results and Analysis

In this section of the proposed method, the detailed experimental process is presented in the form of tables and graphs. The proposed method is tested using two publicly available datasets, CASIA B and TUM GAID. Section 2 contains more information on both datasets. Instead of 70 : 30, the selected datasets are divided 50 : 50 for training and testing. The main reason for this portioning is to make the validation process more equitable. All of the results are based on 10-fold cross-validation. For the classification results, several classifiers are used, including the extreme learning machine (ELM), support vector machine (SVM), KNN, ensemble tree (EBT), and decision trees (DTs). The entire proposed framework is implemented on MATLAB 2021b using Personal Desktop Computer Corei7, 32 GB RAM, and 8 GB graphics card.

5.1. Results

5.1.1. CASIA B Dataset Results. The proposed method results for the CASIA B dataset are presented in the form of numerical values and a time plot in this section. Table 1 provides the classification results of the CASIA B dataset from all perspectives. Normally, researchers choose only a few angles, but in this work, we chose all 11 angles to test the capability of the proposed algorithm. Each angle has three classes: normal walk (NM), walk with a bag (BG), and walk while wearing a coat (WC) (CL). On this dataset, ELM performed better, with average accuracies of 96.89, 93.07, and 83.66% for NM, BG, and CL, respectively. For each angle, the obtained accuracy is above 90% that shows the proposed method effectiveness. A comparison of ELM with other classifiers such as SVM, FKNN, EBT, and DT shows that ELM performed better than all of them. Moreover, the time is also noted of each classifier as shown in Figure 6. From this figure, it is observed that the ELM and DT classifiers executed fast than the other listed methods.

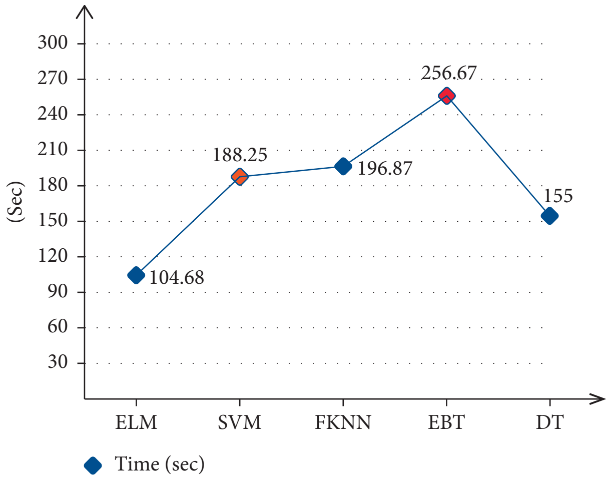


Figure 6: Average classification time of selected classifiers on the CASIA B dataset using the proposed method.

5.1.2. TUM GAID Dataset Results. The results of the proposed method on the TUM GAID dataset are given in Table 2. In this table, accuracy is computed of each class of the selected dataset such as normal walk, walk with a bag, and walk with shoes. Moreover, the average accuracy of each classifier is also computed. Many classifies are selected and ELM shows the better average accuracy of 98.60%. The rest of the classifiers obtained average accuracies of 97.25, 96.73, 96.91, and 96.26%, respectively. The computational time of each classifier is also computed and plotted in Figure 7. It can be seen from this figure that the ELM has a minimum computation time of 86.43 (sec) compared to the rest of the classifiers. Hence, overall, ELM classifier performed better using the proposed method on the TUM GAID dataset.

5.2. Discussion and Comparison. A detailed analysis of the proposed framework has been conducted in this section based on confidence interval and standard error means (SEM). As given in Tables 3 and 4, the proposed

Table 2: Proposed classification results of human gait recognition on the TUM GAID dataset.

Classifier	Class-based accuracy (%)			Mean accuracy (%)
	Normal walk	Walk with a bag	Walk with shoes	
LightweightDeep-ELM	**99.64**	**98.52**	**97.65**	**98.60**
LightweightDeep-SVM	98.58	96.92	96.25	97.25
LightweightDeep-FKNN	98.63	96.21	95.37	96.73
LightweightDeep-EBT	98.12	96.82	95.80	96.91
LightweightDeep-DT	97.52	96.03	95.24	96.26

Bold values indicate the best values.

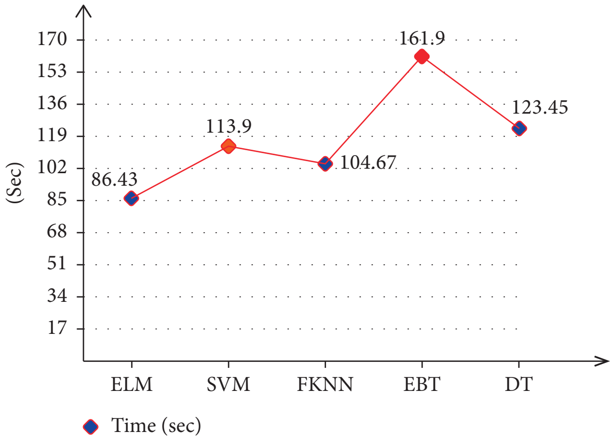


Figure 7: Average classification time of selected classifiers on the TUM GAID dataset using the proposed method.

Table 3: Confidence interval-based analysis of proposed framework on the CASIA B dataset.

Confidence level	Margin of error
Normal walk	
68.3%, $\sigma_{\overline{x}}$	96.005 ± 0.626 (±0.65%)
90%, 1.645 $\sigma_{\overline{x}}$	96.005 ± 1.029 (±1.07%)
95%, $1.960\sigma_{\overline{x}}$	96.005 ± 1.227 (±1.28%)
99%, $2.576\sigma_{\overline{x}}$	96.005 ± 1.612 (±1.68%)
Walk with a bag	
68.3%, $\sigma_{\overline{x}}$	92.59 ± 0.346 (±0.37%)
90%, $1.645\sigma_{\overline{x}}$	92.59 ± 0.57 (±0.62%)
95%, $1.960\sigma_{\overline{x}}$	92.59 ± 0.679 (±0.73%)
99%, $2.576\sigma_{\overline{x}}$	92.59 ± 0.893 (±0.96%)
Walk with a coat	
68.3%, $\sigma_{\overline{x}}$	82.53 ± 0.799 (±0.97%)
90%, $1.645\sigma_{\overline{x}}$	82.53 ± 1.314 (±1.59%)
95%, $1.960\sigma_{\overline{x}}$	82.53 ± 1.566 (±1.90%)
99%, $2.576\sigma_{\overline{x}}$	82.53 ± 2.058 (±2.49%)

LightweightDeep-ELM framework gives the better accuracy than other combinations on the CASIA B dataset. Similarly, the proposed framework (LightweightDeep-ELM) also obtained better results on the TUM GAID dataset. Moreover, the average computational time of each classifier for both datasets is also shown in Figures 6 and 7. The ELM execution time is minimum than the rest of the selected classifiers. To further analyze the performance of the ELM classifier, the proposed framework is executed 500 times and computed two values—minimum accuracy and maximum accuracy. Based on the minimum and maximum accuracy, the standard error mean is computed. Through SEM, a confidence error is obtained that shows the consistency of proposed framework.

Table 4: Confidence interval-based analysis of proposed framework on the TUM GAID dataset.

Confidence level	Margin of error
Normal walk	
68.3%, $\sigma_{\overline{x}}$	98.767 ± 0.62 (±0.63%)
90%, $1.645\sigma_{\overline{x}}$	98.767 ± 1.02 (±1.03%)
95%, $1.960\sigma_{\overline{x}}$	98.767 ± 1.215 (±1.23%)
99%, $2.576\sigma_{\overline{x}}$	98.767 ± 1.597 (±1.62%)
Walk with a bag	
68.3%, $\sigma_{\overline{x}}$	97.71 ± 0.573 (±0.59%)
90%, $1.645\sigma_{\overline{x}}$	97.71 ± 0.942 (±0.96%)
95%, $1.960\sigma_{\overline{x}}$	97.71 ± 1.123 (±1.15%)
99%, $2.576\sigma_{\overline{x}}$	97.71 ± 1.475 (±1.51%)
Walk with a coat	
68.3%, $\sigma_{\overline{x}}$	96.925 ± 0.513 (±0.53%)
90%, $1.645\sigma_{\overline{x}}$	96.925 ± 0.843 (±0.87%)
95%, $1.960\sigma_{\overline{x}}$	96.925 ± 1.005 (±1.04%)
99%, $2.576\sigma_{\overline{x}}$	96.925 ± 1.321 (±1.36%)

Table 3 provides the confidence interval-based analysis of the CASIA B dataset. Confidence level and margin of error (MoE) are calculated for each class such as walk, bag, and coat. We selected several confidence levels such as 68.3%, $\sigma_{\overline{x}}$; 90%, $1.645\sigma_{\overline{x}}$; 95%, $1.960\sigma_{\overline{x}}$; and 99%, $2.576\sigma_{\overline{x}}$ and obtained MOE for each is noted as given below. Based on the MoE, it is observed that the proposed framework showed consistent performance on the CASIA B dataset after 500 iterations. Similarly, Table 4 provides the proposed confidence interval-based analysis of the TUM GAID dataset. From this table, it is also confirmed that the proposed method's accuracy is consistent after the numbers of iterations.

At the end, a detailed comparison is conducted with recent techniques for both selected datasets such as CASIA B and TUM GAID. Table 5 provides the comparison of the proposed method accuracy with recent techniques on the CASIA B dataset. In this table, the authors of [34] obtained an average accuracy of 51.4% on the CASIA B dataset. The authors in [35] improved the average accuracy and reached to 84.2% that was later further improved by [36] of 87.5%. Recently, the authors of [37] obtained an average accuracy of 89.66% on the CASIA B dataset that is improved then the previous noted techniques. Our method achieved an accuracy of 91.20% on the CASIA B dataset that is improved

Table 5: Comparison of proposed method results on the CASIA B dataset with recent techniques.

Reference	Year	Datasets	NM	BG	CL	Mean (%)
[34]	2018	CASIA B	68.1	54.7	31.5	51.4
[35]	2019	CASIA B	95.0	87.2	70.4	84.2
[36]	2022	CASIA B	96.0	91.6	74.8	87.5
[37]	2022	CASIA B	96	92	81	89.66
Proposed		CASIA B	**96.89**	**93.07**	**83.66**	**91.20**

Bold values indicate the best values.

Table 6: Comparison of proposed method results with recent techniques on TUM GAID.

Reference	Year	Datasets	*N*	*B*	*S*	Mean (%)
[26]	2014	TUM GAID	99.4	59.4	94.5	84.4
[38]	2017	TUM GAID	98.7	91.1	94.5	96.7
[39]	2017	TUM GAID	99.7	98.1	95.8	97.9
[40]	2021	TUM GAID	99.4	97.4	96.4	97.73
Proposed		TUM GAID	**99.64**	**98.52**	**97.65**	**98.60**

Bold values indicate the best values.

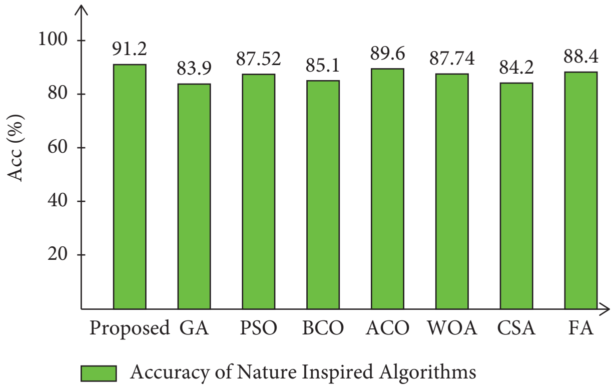


Figure 8: Comparison of proposed optimization accuracy with several other nature-inspired algorithms.

than the existing techniques. Similarly, the comparison of the TUM GAID dataset is given in Table 6. In this table, it is noted that the recently achieved accuracies were 84.4%, 96.7%, 97.9%, and 97.73%. Our proposed method obtained an accuracy of 98.60% that is improved than the recent state-of-the-art (SOTA) techniques.

Finally, the improved moth-flame optimization algorithm is compared to several other nature-inspired algorithms (Figure 8) such as the genetic algorithm, particle swarm optimization, bee colony optimization, ant colony optimization, whale optimization, crow search, and firefly algorithm. This graph shows that the proposed optimization algorithm outperforms the other compared algorithms in terms of accuracy. Moreover, the gait is important for several purposes such as assisting those suffering from Parkinson's disease [41, 42]. In this work, we used Adam as an optimizer [18] during the training of deep learning models instead of stochastic gradient descent (SGD). For the gait recognition task, SGD is not performed better than Adam due to a high number of video frames. As we know, Adam is known to be computationally fast, requires less memory, and needs little tuning.

6. Conclusion

Human gait recognition using lightweight deep learning models and improved moth-flame optimization algorithm has been presented in this work. Two lightweight pretrained CNN models were fine-tuned and deep transfer learning based trained. Features are extracted from the global average pooling layer and fused using a new approach named DCCA. Furthermore, an optimization algorithm is developed for the selection of the best features. The proposed method was compared based on several classifiers such as ELM and SVM and found ELM is more suitable based on accuracy and time. Two publicly available datasets were employed for the validation process and achieved an improved average accuracy of 91.20 and 98.60%. The key findings of this work are as follows: (i) freezing few middle layers can train a model with less time but it is a chance to sacrifice the better accuracy; (ii) fusion of lightweight models features using DCCA approach is time-consuming but at the end, better information in the form of features is obtained; and (iii) improved optimization algorithm provides the better accuracy and reduces the computational time. In the future, a new scratch-based CNN model will be developed for human gait recognition.

Data Availability

The datasets used in this work are publicly available (http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally in this work and have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

References

[1] M. Chaa, Z. Akhtar, and A. Lati, "Contactless person recognition using 2D and 3D finger knuckle patterns," *Multimedia Tools and Applications*, vol. 81, pp. 8671–8689, 2022.

[2] F. Gao, T. Tian, T. Yao, and Q. Zhang, "Human gait recognition based on multiple feature combination and parameter optimization algorithms," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 6693206, 14 pages, 2021.

[3] S. K, F. F. Wahid, and R. G, "Statistical features from frame aggregation and differences for human gait recognition," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18345–18364, 2021.

[4] H. Arshad, M. A. Khan, M. I. Sharif et al., “A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition,” *Expert Systems*, vol. 2020, Article ID e12541, 18 pages, 2020.

[5] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, “Gait recognition via semi-supervised disentangled representation learning to identity and covariate features,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13309–13319, IEEE, Seattle, WA, USA, June 2020.

[6] M. Kumar, N. Singh, R. Kumar, S. Goel, and K. Kumar, “Gait recognition based on vision systems: a systematic survey,” *Journal of Visual Communication and Image Representation*, vol. 75, Article ID 103052,, 2021.

[7] A. Liaqat, M. A. Khan, M. Sharif et al., “Gastric tract infections detection and classification from wireless capsule endoscopy using computer vision techniques: a review,” *Current medical imaging*, vol. 16, no. 10, pp. 1229–1242, 2020.

[8] T. Teepe, A. Khan, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll, “GaitGraph: graph convolutional network for skeleton-based gait recognition,” in *Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2314–2318, IEEE, Anchorage, AK, USA, September 2021.

[9] L. Yao, W. Kusakunniran, Q. Wu, J. Zhang, Z. Tang, and W. Yang, “Robust gait recognition using hybrid descriptors based on skeleton gait energy image,” *Pattern Recognition Letters*, vol. 150, pp. 289–296, 2021.

[10] A. Mehmood, M. A. Khan, M. Sharif et al., “Prosperous human gait recognition: an end-to-end system based on pre-trained CNN features selection,” *Multimedia Tools and Applications*, vol. 202021 pages, 2020.

[11] A. Sepas-Moghaddam and A. Etemad, “Deep gait recognition: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2022.

[12] S. Hou, X. Liu, C. Cao, and Y. Huang, “Gait quality aware network: toward the interpretability of silhouette-based gait recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2022.

[13] S. Gul, M. I. Malik, G. M. Khan, and F. Shafait, “Multi-view gait recognition system using spatio-temporal features and deep learning,” *Expert Systems with Applications*, vol. 179, Article ID 115057, 2021.

[14] F. Saleem, M. A. Khan, M. Alhaisoni et al., “Human gait recognition: a single stream optimal deep learning features fusion,” *Sensors*, vol. 21, no. 22, p. 7584, 2021.

[15] G. Liu, S. Zhong, and T. Li, “Gait recognition method of temporal-spatial HOG features in critical separation of Fourier correction points,” *Future Generation Computer Systems*, vol. 94, pp. 11–15, 2019.

[16] X. Wang, S. Feng, and W. Q. Yan, “Human gait recognition based on self-adaptive hidden markov model,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 963–972, 2021.

[17] S. Das Choudhury and T. Tjahjadi, “Robust view-invariant multiscale gait recognition,” *Pattern Recognition*, vol. 48, no. 3, pp. 798–811, 2015.

[18] M. N. Y. Ali, M. G. Sarowar, M. L. Rahman, J. Chaki, N. Dey, and J. M. R. S. Tavares, “Adam deep learning with SOM for human sentiment classification,” *International Journal of Ambient Computing and Intelligence*, vol. 10, no. 3, pp. 92–116, 2019.

[19] M. I. Sharif, M. A. Khan, A. Alqahtani et al., “Deep learning and kurtosis-controlled, entropy-based framework for human gait recognition using video sequences,” *Electronics*, vol. 11, no. 3, p. 334, 2022.

[20] V. B. Semwal, A. Mazumdar, A. Jha, N. Gaud, and V. Bijalwan, “Speed, cloth and pose invariant gait recognition-based person identification,” *Machine Learning: Theoretical Foundations and Practical Applications*, vol. 87, pp. 39–56, 2021.

[21] A. Zhao, J. Dong, J. Li, L. Qi, and H. Zhou, “Associated spatio-temporal capsule network for gait recognition,” *IEEE Transactions on Multimedia*, vol. 24, pp. 846–860, 2022.

[22] R. Liao, S. Yu, W. An, and Y. Huang, “A model-based gait recognition method with body pose and human prior knowledge,” *Pattern Recognition*, vol. 98, Article ID 107069, 2020.

[23] S. K. Gupta and P. Chattopadhyay, “Gait recognition in the presence of co-variate conditions,” *Neurocomputing*, vol. 454, pp. 76–87, 2021.

[24] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, and Q. Tian, “Multi-view gait image generation for cross-view gait recognition,” *IEEE Transactions on Image Processing*, vol. 30, pp. 3041–3055, 2021.

[25] S. Hou, X. Liu, C. Cao, and Y. Huang, “Set residual network for silhouette-based gait recognition,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 384–393, 2021.

[26] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, “The tum gait from audio, image and depth (gaid) database: multimodal recognition of subjects and traits,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 195–206, 2014.

[27] S. Yu, D. Tan, and T. Tan, “A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition,” in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR’06)*, pp. 441–444, IEEE, Hong Kong, China, 20-24 August 2006.

[28] D. R. Nayak, D. Das, R. Dash, S. Majhi, and B. Majhi, “Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images,” *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15381–15396, 2020.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, https://arxiv.org/abs/1409.1556.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, IEEE, Salt Lake City, UT, USA, June 2018.

[31] C. Buiu, V.-R. Dănăilă, and C. N. Răduţă, “MobileNetV2 ensemble for cervical precancerous lesions classification,” *Processes*, vol. 8, no. 5, p. 595, 2020.

[32] N. Dey, A. S. Ashour, and S. Bhattacharyya, *Applied Nature-Inspired Computing: Algorithms and Case Studies*, Springer, Berlin, Germany, 2020.

[33] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[34] Y. He, J. Zhang, H. Shan, and L. Wang, “Multi-task GANs for view-specific feature learning in gait recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2019.

[35] H. Chao, Y. He, J. Zhang, and J. Feng, “Gaitset: regarding gait as a set for cross-view gait recognition,” in *Proceedings of the*

AAAI conference on artificial intelligence, pp. 8126–8133, Hawaii, HI, USA, January 2019.

[36] F. Han, X. Li, J. Zhao, and F. Shen, "A unified perspective of classification-based loss and distance-based loss for cross-view gait recognition," *Pattern Recognition*, vol. 125, Article ID 108519, 2022.

[37] H. Li, Y. Qiu, H. Zhao et al., "Gaitslice: a gait recognition model based on spatio-temporal slice features," *Pattern Recognition*, vol. 124, Article ID 108453, 2022.

[38] F. M. Castro, M. J. Marín-Jiménez, N. Guil, S. López-Tapia, and N. P. de la Blanca, "Evaluation of CNN architectures for gait recognition based on optical flow maps," in *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, Darmstadt, Germany, September 2017.

[39] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. Pérez de la Blanca, "Automatic learning of gait signatures for people identification," in *Proceedings of the International Work-Conference On Artificial Neural Networks*, pp. 257–270, Salamanca, Spain, June 2017.

[40] R. Delgado-Escano, F. M. Castro, N. Guil, and M. J. Marin-Jimenez, "Gaitcopy: disentangling appearance for gait recognition by signature copy," *IEEE Access*, vol. 9, pp. 164339–164347, 2021.

[41] A. El-Attar, A. S. Ashour, N. Dey, H. Abdelkader, M. M. Abd El-Naby, and R. S. Sherratt, "Discrete wavelet transform-based freezing of gait detection in parkinson's disease," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 4, pp. 543–559, 2021.

[42] A. El-Attar, A. S. Ashour, N. Dey, H. A. El-Kader, M. M. A. El-Naby, and F. Shi, "Hybrid DWT-FFT features for detecting freezing of gait in parkinson's disease," in *Information Technology and Intelligent Transportation Systems*, pp. 117–126, IOS Press, Amsterdam, Netherlands, 2019.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3170244, 12 pages
https://doi.org/10.1155/2022/3170244

Research Article

Early Detection of Forest Fire Using Mixed Learning Techniques and UAV

Varanasi LVSKB Kasyap,[1] D. Sumathi,[1] Kumarraju Alluri,[1] Pradeep Reddy CH,[1] Navod Thilakarathne ,[2] and R. Mahammad Shafi [3]

[1]*VIT-AP University, Amaravati, Andhra Pradesh 522237, India*
[2]*Faculty of Technology, University of Colombo, Colombo, Sri Lanka*
[3]*College of Engineering and Technology, Tepi Campus, Mizan-Tepi University, Mizan Teferi, Ethiopia*

Correspondence should be addressed to R. Mahammad Shafi; mahammadshafi.r@mtu.edu.et

Received 30 March 2022; Revised 2 June 2022; Accepted 21 June 2022; Published 9 July 2022

Academic Editor: Muhammad Ahmad

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

Over the last few decades, forest fires are increased due to deforestation and global warming. Many trees and animals in the forest are affected by forest fires. Technology can be efficiently utilized to solve this problem. Forest fire detection is inevitable for forest fire management. The purpose of this work is to propose deep learning techniques to predict forest fires, which would be cost-effective. The mixed learning technique is composed of YOLOv4 tiny and LiDAR techniques. Unmanned aerial vehicles (UAVs) are promising options to patrol the forest by making them fly over the region. The proposed model deployed on an onboard UAV has achieved 1.24 seconds of classification time with an accuracy of 91% and an F1 score of 0.91. The onboard CPU is able to make a 3D model of the forest fire region and can transmit the data in real time to the ground station. The proposed model is trained on both dense and rainforests in detecting and predicting the chances of fire. The proposed model outperforms the traditional methods such as Bayesian classifiers, random forest, and support vector machines.

1. Introduction

Recent advancements in technology have overwhelmingly shaped society, the economy, and the environment. With the help of the various state-of-art technologies such as IoT, blockchain, AI, geospatial mapping, and so on, leading to the fifth industrial revolution, which focuses more on solving climate goals in line with the revolution [1]. New requirements in the ecological environment arise due to the expeditious development of society. Among the various natural disasters, fire hazard seems to own the characteristics of spreading, and also, it becomes very challenging to control, and thus, it results in heavy destruction that might be irrevocable [2–4]. Over the past few years, there is a tremendous increase in the count, occurrence, and severity of wildfires across the world that has created a great impact on the economy and ecosystem of the country. There are various techniques such as watchtowers, spotter planes, infrared, aerial patrols, and automatic detection systems to detect fire events [1]. There is no need for the exposure of humans to perilous activities when remote sensing is deployed [5]. Various techniques are as follows:

(i) Usage of the satellite images to observe, detect, and report fire events

(ii) Implementation of the wireless sensor networks to observe the fire events exist in all areas.

Yet there are certain limitations associated with the satellite images [6–8]. It has an inadequate resolution, and hence, the data pertinent to the corresponding area would be taken as an average, and it is restricted to a particular pixel that results in the detection of small fires. The predominant limitation is that the satellites cover only a limited area and require a preprocessing time before the resurvey of the same region. The other limitations such as the shortage of real-time data and inadequate precision are inapt for persistent monitoring. There is a need for the infrastructure in advance if WSNs are deployed [4]. There is more chance for the destruction of the sensors during the fire, and this might lead to more expensive restoration of the sensors [9]. Several factors such as the static nature of the sensors, their coverage, difficulty in maintenance, the deficit in power independence, and nonscalability are the reasons for the sensor networks to limit their efficiency. Therefore, unmanned aerial vehicles (UAVs) are proposed to overcome the limitations. The sovereignty, less cost, autonomous, and flexibility make the UAV technology the best choice for fire management efforts in the wildland. There are researchers who put more effort into focusing on the development of frameworks and techniques that could be associated with UAVs. The motive of the implementation of UAV is to detect the fire and its coverage in an optimal manner [3, 10]. The aim of this work is to develop a model to detect the fire and its coverage area, and in addition, it also observes the fire in the low region. Section 2 describes the related works associated with fire detection. Section 3 elaborates on the proposed model and architecture. Finally, the results and discussion to prove the proposed model are covered in Section 4. The last section concludes with a summary and future scope.

2. Related Works

Detection of forest fire and smoke in wildland areas is done through remote sensing-based methods such as satellites, high-resolution static cameras fixed on the ground, and unmanned aerial vehicles (UAVs).

The limitations of the satellites [11] are described as follows:

(i) Images that are captured through the satellites have poor resolution, and hence, it becomes difficult to detect the particular area

(ii) Continuous information about the status of the forest could not be obtained due to the restrictions in the monitoring of forests

(iii) Weather might not be stable in all situations as it might vary, and thus, it results in the collection of noisy images

Optical/thermal cameras deployed on the observation towers together with the other sensors such as smoke, temperature, and humidity sensors might detect the hazards in the closed environment rather than in the open environment as these sensors need vicinity to the fire or smoke. The information obtained through these sensors is not appropriate. Distance covered by these methods could be limited, and to cover a large area, more sensors have to be deployed that might incur expenses. Through the deployment of UAV, large areas could be covered, and the images with high spatial and temporal resolutions could be captured properly. The operational cost is very low when compared with the other methods.

In [12], detection of forest fire is done through the deployment of YOLOv4 to UAV-based aerial images. The initial phase of the process is that the authors developed the hardware platform and proposed the YOLOv4 algorithm. Frame detection rate through this method obtained is 3.2 fps, and the recognition rate achieved is 83%. This works when the intensity of the fire is huge. The limitation of this algorithm is that the detection rate is very less in the small fire-spot areas. The authors have made use of the NetImage classifier that has the combination of Yolov5 and EfficientDet. The data set used comprises 10,581 images of which 2,976 images are categorized as forest fire and 7,605 as nonfire images. The model undergoes an adequate training process, and an accuracy of 99.6% has been obtained with the 476 fire images, and for 676 images that looked similar to images that display fire, the accuracy achieved was 99.7%. Yet the limitation is that it does not detect the smoke since it is needed in the initial stage of the detection process.

In this work [2, 13], the detection of forest fire is done automatically with the help the image processing methods. The principle behind the proposed work is that the image brightness and motion clues are used with the image processing techniques that depend on histogram-based segmentation. Initially, the hot objects are identified, and they are recognized as the candidate regions. Next, the motion vectors of the candidate regions are computed based on the optical flow. Furthermore, the vectors are used to isolate the fires from the other systems that might look similar to the fire. Tracking of fire from IR images is done through the Blob counter technique and morphological operations. Results have shown that the proposed method does the extraction and detection of the fire pixels effectively. Extraction of the background from the video and determination of the proper motion regions by analyzing the difference between the subsequent frames is done through the ViBe method. Several functions such as median filtering, color space conversion, Otsu threshold segmentation, morphological operations, and Blob counter are used [3, 14]. The fire and smoke areas are identified through the extraction of both static and dynamic features. Caffemodel that works based on deep learning is the principle that is used to detect fire and smoke areas. Apart from these, the degree of irregularity of the smoke and fire is also analyzed.

The false alarm rate is reduced, and the original position of the fire is also identified in this work by considering that each and every frame image of the video is partitioned into 16×16 grids and the occurrences of each part of the fire and smoke are recorded. The evidence is collected so as to decide on the final detection. From the experimental results, it is shown that the loss has been reduced, and fire and smoke are detected. Various researchers have worked in exploring forest fire detection in a diversified manner. In [4, 15–17], fuzzy logic-based smoke detection and segmentation schemes are introduced along with the principle of the extended version of the Kalman filter. Segmentation of smoke is done with fuzzy logic, and thus, the prospects of occurrence of smoke are identified. This is done by observing the difference in the background images and the intensity. The extended version of the Kalman filter is used to eradicate the effects that might be due to the disparities in the environmental conditions by reshaping the inputs of the fuzzy smoke detection rule. The authors [18] have worked on the technique that works based on the principle of color and motion. In this work, the UAV is equipped with obscure and an optical camera which is used to detect the forest fire.

Cameras mounted are used to procure free flame infrared and visual pictures. Images could be combined with the landscape information and meteorological data to observe the forest fire. With this technique, the false alert rates of forest fire detection are decreased. A novel method based on the new color index named Forest Fire Detection Index (FFDI) has been proposed in [1, 19–21]. Index computation is done based on vegetation categorization. The tones of the flame and smoke are also detected that in addition is used to form the regions of interest (RoI). The precision of the detection is found to be 96.82% for the image sizes of 960 × 540 pixels that have been processed in 0.447 seconds. The frame considered during the experimental investigations was 22 frames/second for smaller images, and it has been extended to 54 frames/second. The precision rate at the early detection stage has been observed as 96.62%. A deep learning framework named Fire-Net has been devised in [9]. Here, the model is trained on Landsat-8 imagery so that the detection of the active fires and burning biomass is done. The images that are obtained have been represented effectively with the help of optical fusion and thermal modalities. Extraction of a deep set of features is facilitated by providing more attention to the residual convolution and discrete convolution blocks. From the experiments, the overall accuracy is 97.35%, and the model is robust in the detection of small fires. Continuous monitoring of the potential areas that might be prone to fire should be monitored. In this work [22–25], the design of UAVs has been done based on the advantages of AI. Furthermore, the onboard processing abilities have also been equipped. The inputs to the model are the still images or the video input that are captured through the cameras mounted on the drone [14, 26, 27]. The drones are supported by both fixed and the rotary-wing. The fixed-wing drone is used to monitor the area frequently. It covers an altitude from 350 m to 5,500 m, and there might be a chance of reporting a false alarm. As soon as the fixed-wing drone detects the fire, an alarm is triggered that in turn stimulates the rotary-wing drone. With the help of GPS coordinates, it then examines the area. The second drone decides whether to trigger the alarm based on its observations. The main advantage of the second drone is to decrease the false alarm rate. Many research works have been carried out for detecting fire through IR images, whereas a few works are done on UAV platforms [28, 29].

From the various inferences, it has been understood that most of the researchers have worked to increase the accuracy, and the area coverage was until 1,500 meters to the maximum. To provide more accuracy and precision, the 3D modeling of data is required, and further visualization of forest fire images could be made very easy for interpretation. The objective of this work is to deploy an efficient and robust detection fire in the early stage. Hence, a deep learning model is required so that the boundary region could be extended, and the 3D modeling images must be considered for the prediction process to augment the accuracy. The contributions in this work are as follows:

(i) An efficient and robust 3D modeling is used to augment the accuracy of the detection.

(ii) A deep learning technique YOLOv4 is combined with the Otsu method along with LiDAR. The key objective of utilizing the Otsu method is to repeat all the values of the threshold and evaluate the extent of the background and foreground pixels. The objective is to determine the threshold by examining the region of the spread, and it should be minimum.

(iii) Traditional methodologies are found to be difficult for performing the sampling since the constraints are bound to the relative position. Hence, the orientation of the images is required, and that is obtained by computing the distance between the tree and other entities with the help of LiDAR. Section 3 describes the proposed methodology.

3. Proposed Methodology

The flow of the proposed architecture is shown in Figure 1. The video input is captured from the camera, and the other inputs such as wind speed, wind directions, and IR image sensing are calculated using the sensors mounted on the UAV for navigation. These images are provided as input to the deep learning models, and it checks for the existence of the fire. The region is predicted clearly since there is a possibility of more projections of the images provided to the model due to the 3D modeling. Further detection is made, and the details are stored in the database for further.

3.1. Autonomous Drone Routing

3.1.1. Drone Moment to the Target. In this whole operation, navigation of UAVs is significant to patrol the risk-prone areas and fire-detected areas. This work monitors the forest area with the help of the navigational analysis technique [27, 30]. To facilitate this, the drone makes the navigation. UAVs have the following three navigational features [31]:

(i) Awareness: This provides details about UAV's neighborhood obstacles. The data is collected using internal sensors

(ii) Basic Navigation: Collisions are avoided, and the obstacles such as birds, trees, poles, and so on in the forest farms are detected

(iii) Expanded Navigation: Advanced features such as pathway planning and depth deployment are included and play a crucial role in autonomous navigation

This categorization of features is shown in Figure 2 that could provide a better understanding of the navigation of UAVs.

On detection of fire with YOLOv4 tiny, the autonomous patrol in the affected area is found to influence the decisions that must be considered for stopping the fire. The visual servoing algorithm [8] operates for the autonomous drift of UAV towards the fire; it works as follows:

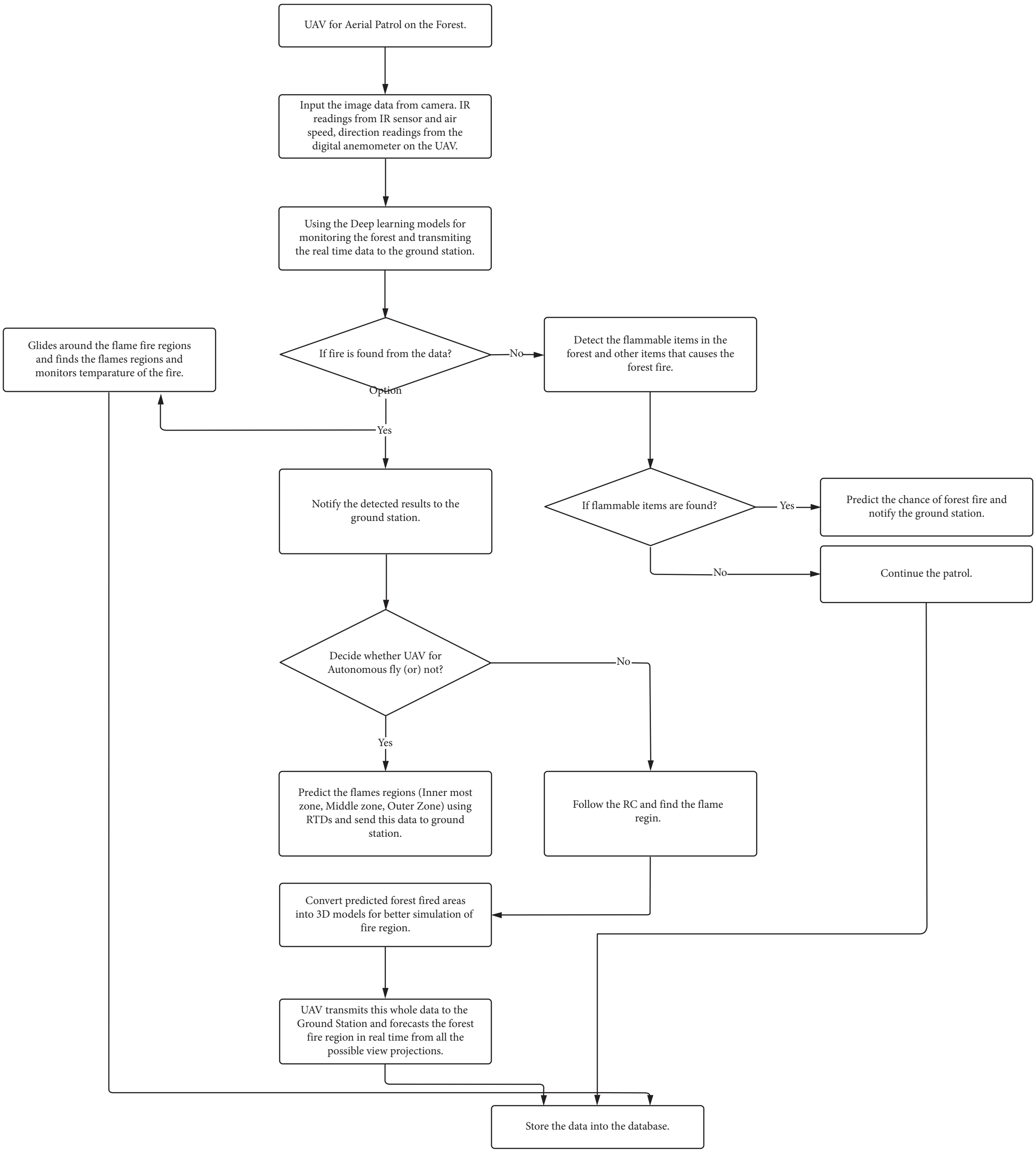

UAV for Aerial Patrol on the Forest.
Input the image data from camera. IR readings from IR sensor and air speed, direction readings from the digital anemometer on the UAV.
Using the Deep learning models for monitoring the forest and transmiting the real time data to the ground station.
If fire is found from the data?
No
Detect the flammable items in the forest and other items that causes the forest fire.
Glides around the flame fire regions and finds the flames regions and monitors temparature of the fire.
Option
Yes
Notify the detected results to the ground station.
If flammable items are found?
Yes
Predict the chance of forest fire and notify the ground station.
No
Continue the patrol.
Decide whether UAV for Autonomous fly (or) not?
No
Yes
Predict the flames regions (Inner most zone, Middle zone, Outer Zone) using RTDs and send this data to ground station.
Follow the RC and find the flame regin.
Convert predicted forest fired areas into 3D models for better simulation of fire region.
UAV transmits this whole data to the Ground Station and forecasts the forest fire region in real time from all the possible view projections.
Store the data into the database.

Figure 1: Proposed architecture flow.

Step 1. The location of the fire is requested, and it is captured through the (YOLOv4 tiny-bounding box of fire). If the fire is detected, go for step 2.

Step 2. Calculate the step size depending on the location of the bounding box of fire that is relative to the midpoint of the frame, along with the direction.

Step 3. Normalize the drone for drift by changing the roll and angle of pitch of the current state.

Step 4. Set the next set point to control the flight and iterate the process.

When the forest is very thick and it is in the GPS-denied area, UAV makes use of hybrid-localization methods to achieve the maximum performance of the task. Many research works [31, 32] suggest the simultaneous localization and mapping (SLAM) algorithm to use in GPS-denied areas. The solitary idea of SLAM is to process mapping and localization concurrently and recursively. Using the Kalman filter, SLAM overcomes the problem of feature evaluation. SLAM algorithms develop the landmark-based navigation

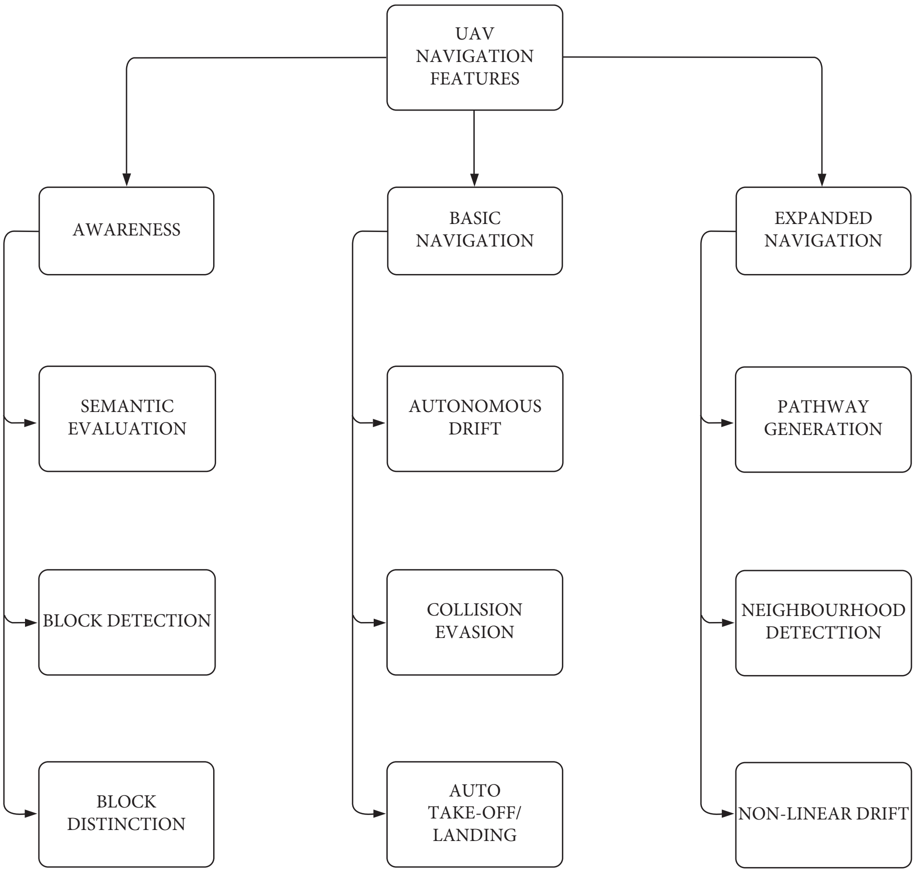


Figure 2: Categorization of navigation features of UAV.

system with the capability of virtual-map navigation; this became a typical technique for the drone-based navigation application. Figure 3 given below depicts the structure of the proposed localization solution based on vision. This structure operates the space and features on two different levels. It has three main modules: (1) hybrid feature extraction, (2) generating map, and (3) hybrid localization. The feature extraction component combines two different methods to find a location at two scales. The map generation component performs feature compression and feature evaluation. The filters extract the most appropriate features using an info-theoretic method, then compress the features, and evaluate them.

3.1.2. Technical Information of UAV. Flight planning is considered a salient feature in designing the architecture of UAVs [33]. This planning illustrates the division of mass on UAVs and provides a better understanding of the performance analysis of UAVs. Specifically, maximum take-off weights (MTOW) assess the UAV payload capacity at different heights above the ground. The payload of the UAV and the mass of onboard equipment are given in Table 1, and the components of the UAV are depicted in Figure 3. The battery used on the UAV reserves the UAV in GPS-enabled environments for 107 minutes of duration, whereas on the GPS-disabled environment, maximum flight time is 87 minutes.

Figure 4 displays the digital anemometer, manifold CPU, IR sensors, and 12K camera.

The total payload of the UAV is 6,825 grams. The thrust of the UAV is capable to fly with a 7,000-gram payload and is calculated as follows:

$$\tau(T) = \frac{\pi}{4} d^2 \rho v \Delta v, \tag{1}$$

where τ = thrust of UAV, d = distance of propeller = 0.6556 m, ρ = air density (1.225 m kg/cubic meter), and v = aerial velocity at propeller (m/s) = 1/2Δv.

The power of the UAV is as follows:

$$\begin{gathered} P(W) = \frac{\tau(\Delta v)}{2}, \\ \mathrm{P}(W) = \text{PropellerConstant} * \left(\frac{rpm}{1000}\right)^{\text{powerfactor}}, \\ \mathrm{P}(W) = \mathit{PropellerConstant} * d^4 * p * rpm^3, \end{gathered} \tag{2}$$

where Propeller Constant = 1.11(for APC propeller) and P = pitch of propeller in air.

The total mass lifted by the UAV is as follows:

$$\begin{gathered} m = \frac{\tau}{g}, \\ m = \frac{\left(\pi/2d^2\rho v \mathrm{P}\right)^{1/3}}{g}, \end{gathered} \tag{3}$$

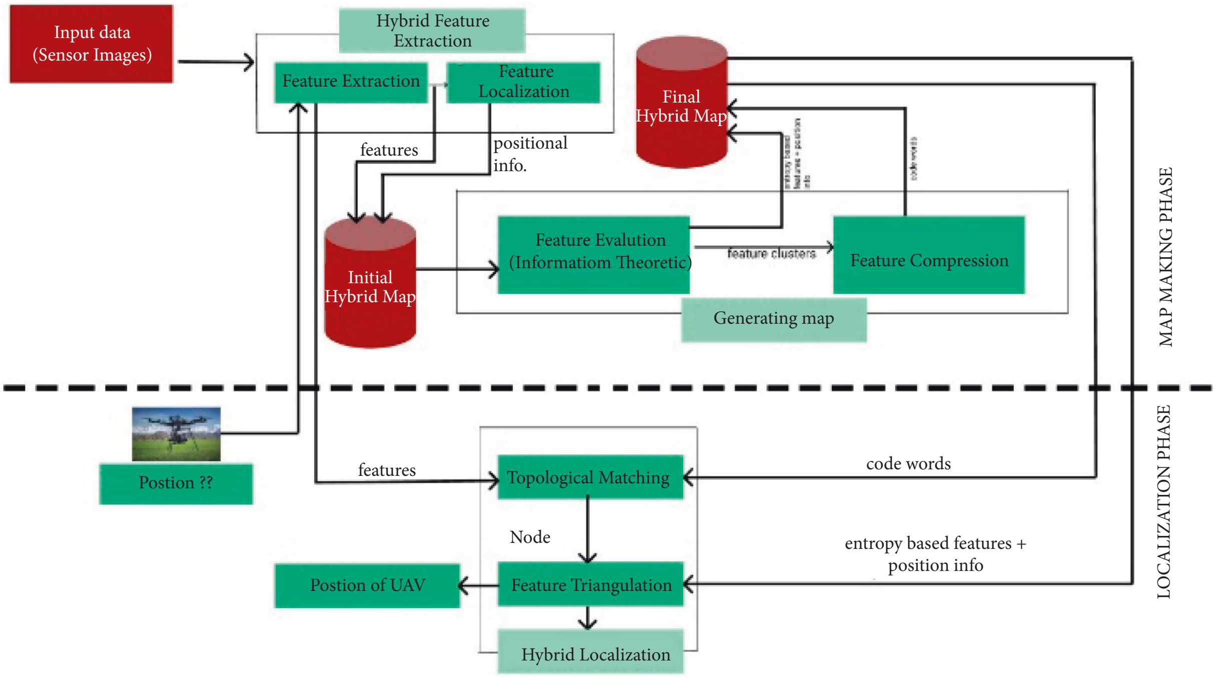


Figure 3: Structure of main modules.

Table 1: Specifications of the onboard equipment and masses.

S. No.	Component	Specification	Mass in grams
1.	Motor (x4)	T motor *u* power u5 400 kv	815
2.	Battery	6s-hv (lihv) 25,000 ma (22.8v)	1,568
3.	12 K Camera (with IR)	Blackmagic ursa mini pro 12 k	2,742
4.	Anemometer	Breezesonic 345P-4 R	300
5.	Sensors	—	600
6.	Structure	DJI	100
7.	Others (with CPU)	—	700
8.	Total payload		6,825

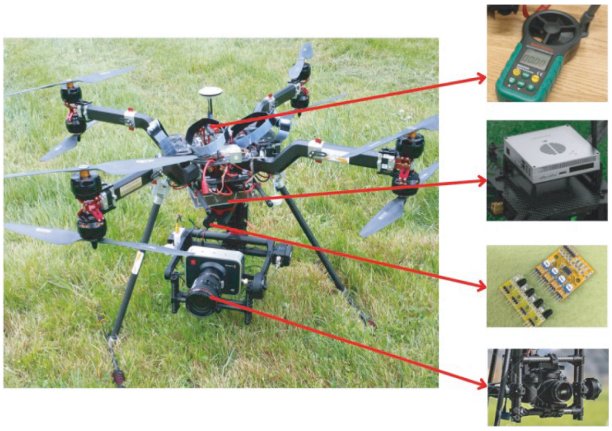


Figure 4: Digital anemometer manifold with CPU, IR sensors, and 12 k camera.

where g = acceleration due to gravity.

Therefore, the maximum payload of the UAV is 8,000 grams.

Due to this high Power of UAV, it can take steep turns with a very low pitch angle in any direction and is capable to fly 350–400 m above the ground with a range of 3,000–4,000 m.

The time of flight of a UAV is calculated as follows:

$$T_f = c_b * \left(\frac{d_b}{AAD}\right), \tag{4}$$

where T_f = time of flight in hours, c_b = capacity of battery (mAh), d_b = battery discharge, and AAD = average ampere draw (ampere) of the UAV = AUW * p/v.

3.2. Fire Detection and Fire Region Prediction

3.2.1. Fire Detection in the Forest Region. A classic UAV can autonomously fly over the forest area and detect the forest fire and yaw around the burning forest fire area. The UAV is well equipped with IR sensors, a 12K camera for image accretion, and the onboard CPU, which can broadcast the real-time video of the forest fire to the ground station using the signals that are used for remote navigation. The ground station would diagnose and take necessary measures to stop the forest fire. In parallel, the ground station can also control the UAV by sending the operational commands.

The onboard CPU has good computation power to perform the forest fire detection using YOLOv4 tiny, which has good detection speed with well-grounded accuracy [11]. The YOLOv4 tiny model is divided into two layers, that is, the feature extraction layer and the processing layer. The feature extraction layer is the combination of the DarkNet and ResNet, similar to the feature-like pyramid network that has the convolutional layer, batch-normalization layer, and leaky ReLU layer. The problem of overfitting is shut out using batch normalization. The combination of the convolutional layer, batch-normalization layer, and leaky ReLU layer is called CBL. The combination of the convolutional layer, batch-normalization layer, and mish activation function is called CBM. The structure of CBL and CBM is shown in Figure 5.

The five max-pooling layers of the network of 2 ∗ 2 size with stride = 2 give the reduced feature map of 1/32th size of the original image. Since there is no fixed shape for fire and smoke, size is varied. YOLO layers are operated for the detection of fire using logistic regression. YOLOv4 tiny has an advantage of high speed and accuracy, due to its speed-up algorithm. The bounding boxes of YOLOv4 tiny can be achieved using region proposal network (RPN) [11]. Intersection over union (IOU) score is considered the metric and is used to obtain the bounding box. The distance measure for the clustering is given in

$$D(\text{boundingbox}, \text{centre}) = 1 - IOU(\text{boundingbox}, \text{centre}). \tag{5}$$

The algorithm finds the bounding box as follows: image is captured by the camera, and the target is drawn to fix the center. The coordinate calculation of the image is given in Equation 2.

$$\begin{cases} BB_x = \sigma(p_x) + c_x \\ BB_y = \sigma(p_y) + c_y \\ BB_{\text{width}} = C_w e^{p_x} b_h = C_h e^{th} \end{cases}, \tag{6}$$

where (p_x, p_y, p_w, p_h) are the image center points and (C_w, C_h) are the bounding box center points.

Onboard IR sensors are cast to find the heat distribution in the forest farm and then generate unichannel 2D images. The forest fire is classified into three different regions based on the temperature of the fire; it can be detected from the UAV by using highly accurate IR sensors that measure the intensity of forest fire pixels and analyze the region they fall under. The brightness of the pixels is converted into a graph, and the local maxima are considered the high-intensity region [12]. This histogram-based segmentation of pixels helps in finding the regions of fire, by taking the advantage of Otsu method. This method is defined to find a threshold that can minimize the intraclass variance as a weighted sum as follows:

$$\sigma_w^2(k) = \omega_a(k)\sigma_a^2(k) + \omega_b(k)\sigma_b^2(k), \tag{7}$$

where the class probabilities are ω_a and ω_b (foreground and background) given by a threshold value and k, σ_a^2, and σ_b^2 are variances of both the classes.

Probabilities of classes a and b are calculated through h histograms are as follows:

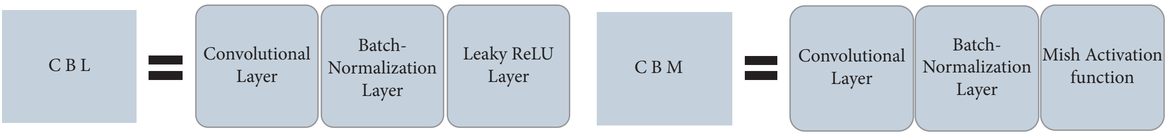
$$\begin{aligned} \omega_a(k) &= \sum_{j=0}^{k-1} p(j), \\ \omega_b(k) &= \sum_{j=k}^{h-1} p(j). \end{aligned} \tag{8}$$

Intraclass variance is minimized to maximize interclass variance. The architecture of YOLOv4 tiny is shown in Figure 6.

3.2.2. Prediction of the Possibility of Forest Fire. When UAV is patrolling over the forest region, it observes for the forest fire; if the fire is found, it drifts to that affected area and broadcasts all the data to the ground station and then helps the people extinguish the fire. If there is no fire in the forest, then UAV tries to find the possibilities of forest fire in that region. In general, forest fire is caused either by man-made errors or natural errors. The man-made errors that lead to forest fire are campfires that are not completely turned off, used and thrown mosquito coils, the smoked cigarettes remain, and tribal traditions related to fire. The natural causes that lead to forest fire are lightning [6, 7, 34], combustion of dry vegetation, and volcanic activities. UAV predicts the occurrence of forest fire based on any of the above-stated situations [35]. UAV finds the possibilities of fire such as oxygen, fuel, and heat (shown in Figure 7), while it is patrolling and transmits the results to the ground station.

3.3. 3D Modeling of Forest Fire. 3D modeling of the forest-fire-affected area helps the ground station to diagnose and analyzes the situation for extinguishing the fire and helps know the direction of the forest fire; this information is very crucial and reduces the time of extinguishing. Existing techniques for forest fire modeling are empirical and enhancement of the modeling is needed [36]. The motivation for generating a 3D forest fire model is from photogrammetric research, which enables us to generate 3D models from images with high accuracy. In this work, much effort is spent on the creation of 3D images and LiDAR.

3.3.1. Construction of 3D Forest Fire Modeling. The spatial resection technique of photogrammetry is used to estimate the position of the trees in the forest by measuring evenly



(a)

(b)

Figure 5: (a) CBL – combination of convolutional layer, batch-normalization layer, and leaky ReLU layer and (b) CBM – combination of convolutional layer, batch-normalization layer, and mish activation function.

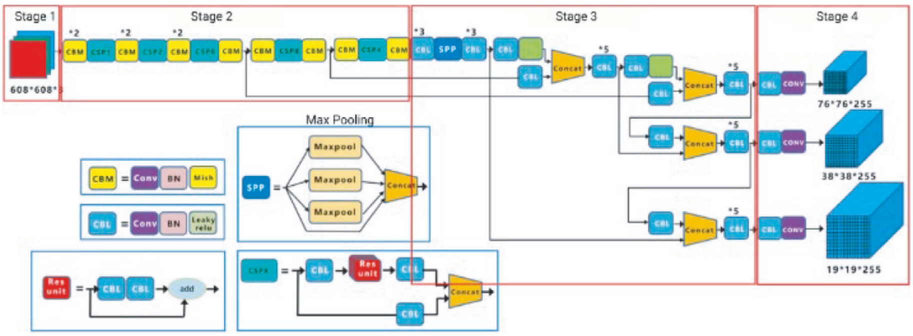


Figure 6: The architecture of YOLOv4 tiny.

distributed feature points across the 2D images of the forest. The recovery of positions of tress from various directions is called “relative orientation.” Many works have been done using pixel correspondence from scaled tree positions. Recent developments have been made to generate 3D models using the 2D images using LiDAR for outdoor 3D modeling. In this work, we have adopted some of the techniques from [33]. Unlike well-designed interior images, outdoor areas such as forest, farms, and parks contain many objects. Due to relative positioning constraints for data acquisition, sampling of the surfaces is difficult using the traditional methods. Orientation of the images and distance between the tress and other objects can be easily done using LiDAR. To deal with the inconsistency of the data bottom-up approach is used. LiDAR-generated 2D images are collected perpetually from the LASER while flying in forest areas at a high speed.

3.3.2. Terrestrial Image-Based 3D Modeling. Along with accurate orientations methods, tie point measure, and adjusting bundles allow sensor calibrations. Once the images are aligned, the surface measure is performed using automated procedures. Automatic photogrammetric matching algorithms are advanced and use multiple image inputs. Dense point clouds are developed using these methods and often ignore geometric constraints using smoothing. The results are shown in Figure 8.

4. Experiments and Results

This model is first trained on a desktop and later loaded into UAV-CPU for testing. It is trained over 100 images for 50,000 steps each. The performance of the UAV is shown in Table 2. The comparison table of existing works and the proposed model is given in Table 3.

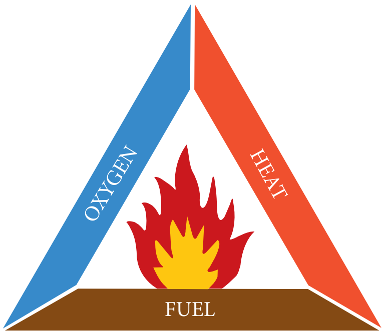


Figure 7: The fire triangle represents the three major components (oxygen, heat, and fuel) that are necessary to generate a fire.

YOLOv4 tiny has acceptable FPS on the UAV CPU and achieves real-time detection and analysis. Few of the augmentation techniques such Mix-up and Mosaic are performed to generate the random image pairs from training data. The convolution layer in the network has good performance in detecting smaller parts of images as shown below. Figure 9 shows that the UAV is able to detect the fire from low light intensity regions also using the Otsu method, but the high smoke zones are misclassified as the high fog zones, and fire is not detected in such cases. The model is able to detect fire in ale environmental conditions such as rainy, sunny, snow, and so on, and the training data of the model contains all three examples in equal proportion to avoid variance and bias. The other parameters such as the wind speed and temperature of the regions are analyzed, and fire is detected in such cases. Therefore, in order to improve accuracy in the high smoke zones, we used IR sensors; by utilizing the optical flow, fire is easily detected. The experimental results obtained demonstrate that the model is capable of detecting forest fires and flame regions, 3D

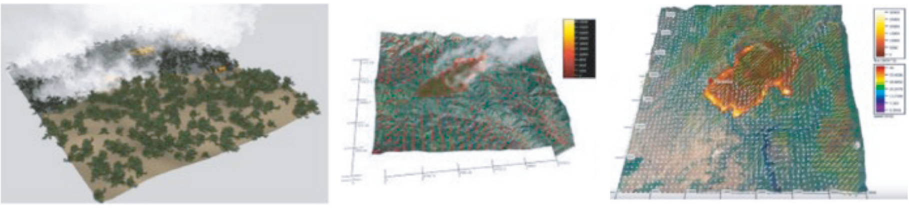


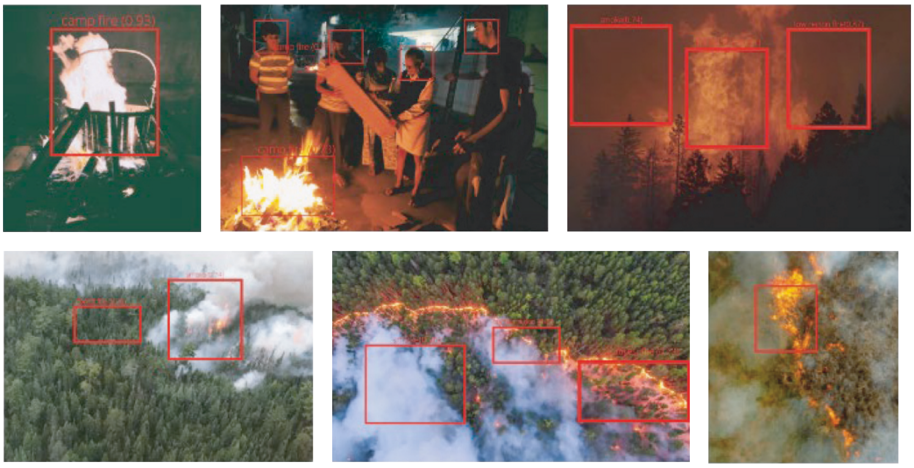
Figure 8: Generated 3D model of the forest fire.

Table 2: Performance metrics of the proposed model.

S. No.	Parameter	Result
1	mAP@0.5	85.36%
2	mAP@0.75	82.45%
3	Average_IOU	83.17%
4	Precision	0.93
5	Recall	0.89
6	F1-score	0.91

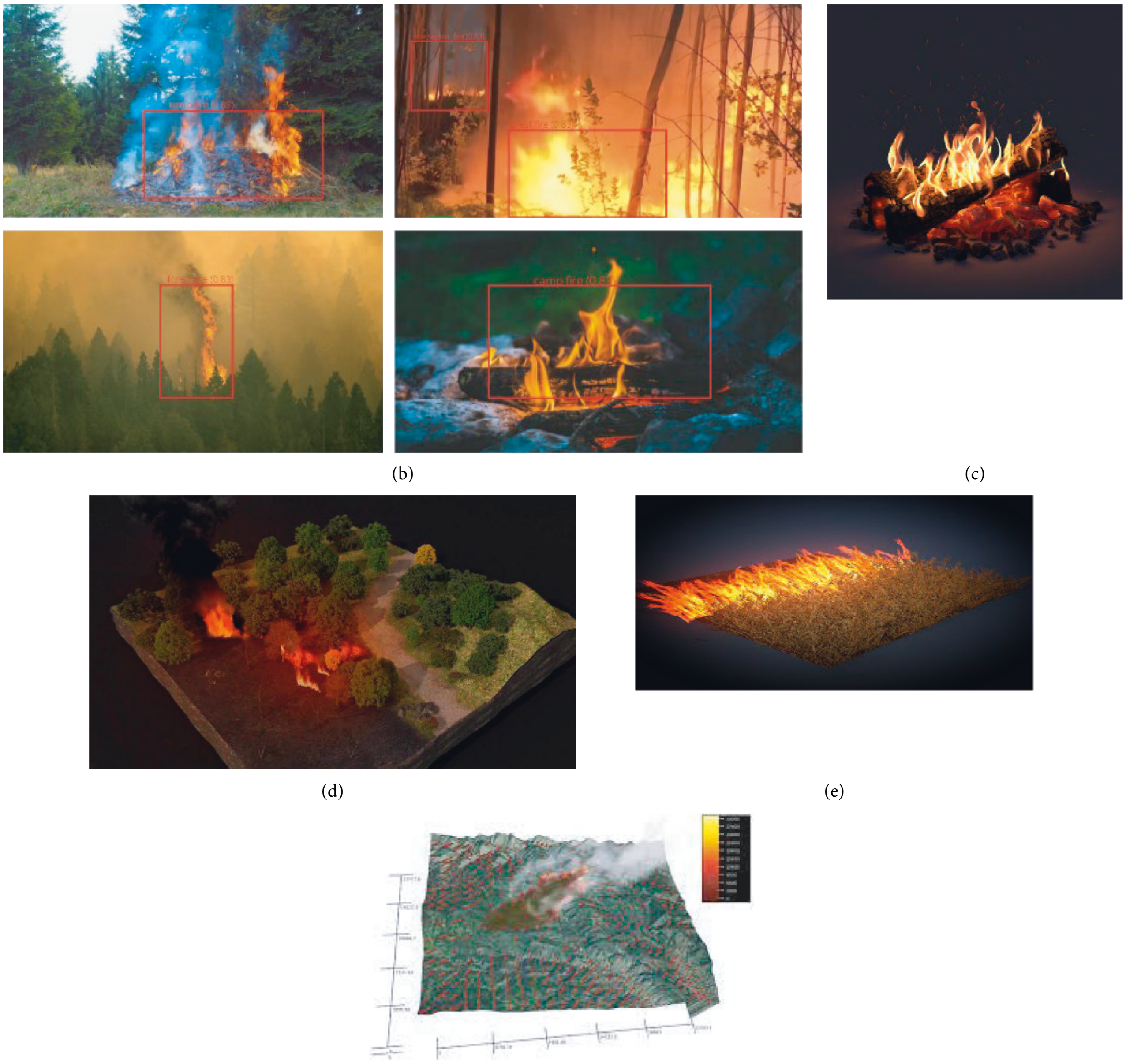
Table 3: Performance comparison of proposed model.

S. No.	Metric	[11]	[12]	[2]	Proposed
1	Accuracy	87%	86.7%	89%	**93.3%**
2	Precision	0.87	0.86	0.89	**0.93**
3	Recall	0.78	0.81	0.83	**0.89**
4	F1-score	0.86	0.87	0.89	**0.91**



(a)

Figure 9: Continued.



(b)

(c)

(d)

(e)

(f)

Figure 9: (a) Sample 3D models of a forest fire, (b) sample 3D models of a forest fire, and (c)–(f) testing results with the developed UAV model.

modeling of the affected, area and forest fire tracking with satisfactory results, while the problem of using satellite imagery and low-level performance is significantly reduced as well.

The 3D models of forest fire as shown in Figures 9(a) and 9(b) are made from the predicted models. Testing results are shown in Figures 9(c)–9(f). The obtained results are worth noting and help in the analysis of the affected area.

5. Conclusion and Future Scope

Evolution emerges in the processing, computation, and algorithms. This strives many researchers to pay attention in many domains where they work in the processing of surveillance video streams so that abnormal or unusual actions could be detected. The usage of UAVs is recommended in the detection of forest fire due to the high mobility and ensures the coverage areas at various altitudes and locations at a low cost. Hence, an efficient and scalable UAV is used for detection. This work aims in developing the 3D model for the captured scene. YOLOv4 tiny network is deployed to detect the fire. The accuracy of the detection rate achieved through this model is 91%. The proposed model outperforms the other existing techniques in terms of detecting in the early stage. However, this model is sensitive to the forest with dense fogs and clouds. This is because smoke appears as the same as fog, and the model may misclassify the fog as smoke. As our future works, focus to meet practical detection and meet the necessity of early detection including the generation of the mixed reality model of the forest fire area that gives more information, and prevention analysis will be made easy. The 3D modeling techniques presented in this paper can also be extended to various natural disaster prediction models.

Data Availability

The data that support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

References

[1] A. Ollero, B. C. Arrue, and J. R. Martinez, "For reducing false alarms in forest-fires," *Computer Communications S0140366419308655*–, 2019.

[2] C. Yuan, Z. Liu, and Y. Zhang, "Fire Detection Using Infrared Images for UAV-Based Forest Fire Surveillance," in *Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, June, 2017.

[3] X. Wu, X. Leung, and H. Leung, "An adaptive threshold deep learning method for fire and smoke detection," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1954–1959, Banff, AB, Canada, October, 2017.

[4] C. Yuan, Z. Liu, and Y. Zhang, "Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1-2, pp. 337–349, 2018.

[5] K. Amer, M. Samy, R. ElHakim, M. Shaker, and M. ElHelw, "Convolutional neural network-based deep urban signatures with application to drone localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, p. 2138, October, 2017, –2145.

[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December, 2015.

[7] R. Ramakrishna, M. Rajeevan, and S. Ramakrishna, "Prediction of severe thunderstorms over Sriharikota Island by using the WRF-ARW operational model," *SPIE Proceedings*, vol. 9882, Article ID 988214, 2016.

[8] R. HussinM and R. Juhari, "Detection using image processing based techniques," *Indian Journal of Computer Science and Engineering*, vol. 41, 2012.

[9] F. Guede-Fernández, L. Martins, R. V. de Almeida, H. Gamboa, and P. Vieira, "A deep learning based object identification system for forest fire detection," *Fire*, vol. 4, no. 4, p. 75, 2021.

[10] M. Mueller, P. Kolesov, I. Tannenbaum, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2786–2797, 2013.

[11] P. Chen, T. Zhang, K. Xin, N. Yi, D. Liu, and H. Liu, "A UAV-based forest fire detection algorithm using convolutional neural network," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 10305–10310, Wuhan, China, June, 2018.

[12] Z. Jiao, Y. Zhang, X. Xin et al., "A deep learning based forest fire detection approach using UAV and YOLOv3," in *Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pp. 1–5, Shenyang, China, July, 2019.

[13] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, "Early forest fire detection using drones and artificial intelligence," in *Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1060–1065, Opatija, Croatia, May, 2019.

[14] P. M. Wyder, Y. S. Chen, A. J. Lasrado et al., "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments," *PLoS ONE*, vol. 14, no. 11, Article ID e0225092, 2019.

[15] P. Cortez and A. Morais, "A Data Mining Approach to Predict Forest Fires Using Meteorological Data," 2017.

[16] G. Yadav, V. Gupta, V. Gaur, and Bhattacharya, "Mahua, OPTIMIZED FLAME," *Indian Journal of Computer Science and Engineering*, vol. 3, no. 2, 2012.

[17] K. Lakshmanna, N. Khare, and N. Khare, "Constraint-based measures for DNA sequence mining using group search optimization algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 3, pp. 91–100, 2016.

[18] S. Sudhakar, V. Vijayakumar, C. S. Kumar, V. Priya, L. Ravi, and V. Subramaniyaswamy, "Unmanned Aerial Vehicle (UAV) Based Forest Fire Detection and Monitoring," *Computer Communications*, vol. 149, 2019.

[19] K. Poobalan and S. C. Liew, "Fire Detection Algorithm Using Image Processing Techniques," in *Proceedings of the International Conference on Artificial Intelligence and Computer Science*, Penang, Malaysia, October, 2017.

[20] M. Haque, M. Muhammad, D. Swarnaker, and M. Arifuzzaman, "Autonomous quadcopter for product home delivery," in *Proceedings of the 2014 International Conference on Electrical Engineering and Information & Communication Technology*, pp. 1–5, Dhaka, Bangladesh, April, 2014.

[21] R. Kaluri, D. S. Rajput, Q. Xin et al., "Roughsets-based Approach for Predicting Battery Life in IoT," 2021, https://arxiv.org/abs/2102.06026.

[22] S. T. Seydi, V. Saeidi, B. Kalantar, N. Ueda, and A. A. Halin, "Fire-net: a deep learning framework for active forest fire detection," *Journal of Sensors*, vol. 2022, Article ID 8044390, 14 pages, 2022.

[23] S. Rady, A. A. Kandil, and E. Badreddin, "A hybrid localization approach for UAV in GPS denied areas," *SII*, in *Proceedings of the 2011 IEEE/SICE International Symposium on System Integration*, Kyoto, Japan, December, 2011.

[24] R. Wang, "3D building modeling using images and LiDAR: a review," *International Journal of Image and Data Fusion*, vol. 4, no. 4, pp. 273–292, 2013.

[25] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, Article ID 54776, 2020.

[26] H. Pan, D. Badawi, and A. E. Cetin, "Computationally Efficient Wildfire Detection Method Using a Deep Convolutional Network Pruned via Fourier Analysis," *Sensors (Basel)*, vol. 20, 2022.

[27] C. Iwendi, P. K. R. Maddikunta, T. R. Gadekallu, K. Lakshmanna, A. K. Bashir, and M. J. Piran, "A metaheuristic optimization approach for energy efficiency in the IoT networks," *Software: Practice and Experience*, vol. 51, no. 12, pp. 2558–2571, 2021.

[28] F. Remondino and S. F. El-Hakim, "Image-based 3d modelling: a review," *The Photogrammetric Record*, vol. 21, no. 115, pp. 269–291, 2006.

[29] M. Pollefeys and L. V. Gool, "From images to 3D models," *Communications of the ACM*, vol. 45, no. 7, pp. 50–55, 2002.

[30] H. Qays, B. Jumaa, and A. Salman, "Design and implementation of autonomous quadcopter using SITL simulator," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 1–16, 2020.

[31] K. Nonami, "Research and Development of Drone and Roadmap to Evolution," *Journal of Robotics and Mechatronics*, vol. 30, pp. 2–6, 2018.

[32] T. Celik, H. Demirel, and H. Ozkaramanli, "Fire detection in video sequences using statistical color model," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May, 2006.

[33] L. Castillo and Dzul, *Modelling and Control of Mini-Flying Machines*, Springer, Berlin/Heidelberg, Germany, 2005.

[34] L. Aj, S. Mary Idicula, C. Naveen Francis, and C. Francis, "Artificial neural network model for the prediction of thunderstorms over Kolkata," *International Journal of Computer Applications*, vol. 50, no. 11, pp. 50–55, 2012.

[35] C. Alex and A. Vijay Chandra, "Autonomous cloud-based drone system for disaster response and mitigation," in *Proceedings of the 2016 International Conference onRobotics and Automation for Humanitarian Applications (RAHA)*, pp. 1–4, IEEE, Amritapuri, India, December, 2016.

[36] T. Lee, S. Mckeever, and J. Courtney, "Flying free: a research overview of deep learning in drone navigation autonomy," *Drones*, vol. 5, no. 2, p. 52, 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6781740, 14 pages
https://doi.org/10.1155/2022/6781740

Research Article

Hyperspectral Image Classification with Optimized Compressed Synergic Deep Convolution Neural Network with Aquila Optimization

Tatireddy Subba Reddy,[1] Jonnadula Harikiran,[2] Murali Krishna Enduri,[3] Koduru Hajarathaiah,[3] Sultan Almakdi,[4] Mohammed Alshehri,[4] Quadri Noorulhasan Naveed,[5] and Md Habibur Rahman[6]

[1]*Computer Science and Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India Pin: 502313*
[2]*School of CSE, VIT-AP University, Vijayawada, Pin: 522237, Andhrapradesh, India*
[3]*Computer Science and Engineering, SRM University-AP, Amaravati, India*
[4]*Department of Computer Science, College of Computer Science and Information System, Najran University, Najran, Saudi Arabia*
[5]*Department of Computer Science, College of Computer Science, King Khalid University, Abha, Saudi Arabia*
[6]*Dept. of Computer Science and Engineering, Faculty of Engineering and Technology, Islamic University, Kushtia-7003, Bangladesh*

Correspondence should be addressed to Md Habibur Rahman; habib@iu.ac.bd

Received 12 May 2022; Accepted 21 June 2022; Published 7 July 2022

Academic Editor: Muhammad Ahmad

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

The classification technology of hyperspectral images (HSI) consists of many contiguous spectral bands that are often utilized for a various Earth observation activities, such as surveillance, detection, and identification. The incorporation of both spectral and spatial characteristics is necessary for improved classification accuracy. In the classification of hyperspectral images, deep learning has gained significant traction. This research analyzes how to accurately classify new HSI from limited samples with labels. A novel deep-learning-based categorization based on feature extraction and classification is designed for this purpose. Initial extraction of spectral and spatial information is followed by spectral and spatial information integration to generate fused features. The classification challenge is completed using a compressed synergic deep convolution neural network with Aquila optimization (CSDCNN-AO) model constructed by utilising a novel optimization technique known as the Aquila Optimizer (AO). The HSI, the Kennedy Space Center (KSC), the Indian Pines (IP) dataset, the Houston U (HU) dataset, and the Salinas Scene (SS) dataset are used for experiment assessment. The sequence testing on these four HSI-classified datasets demonstrate that our innovative framework outperforms the conventional technique on common evaluation measures such as average accuracy (AA), overall accuracy (OA), and Kappa coefficient (k). In addition, it significantly reduces training time and computational cost, resulting in enhanced training stability, maximum performance, and remarkable training accuracy.

1. Introduction

Due to the fast growth of photonics with optics, sensors in hyperspectral (HS) are needed to install in several satellites. HSI classification is an essential and challenging task that is targeted towards labelling each pixel contained in a hyperspectral image. HSI images contained spatial-spectral information which is useful for detecting scene objects [1]. This had been used in many fields like environmental surveillance, astronomy, and precise agriculture [2].

In the earlier days, HSI classification was done by the machine learning methods such as support vector machines (SVM) [3, 4], k-nearest neighbor (KNN) [5, 6], multinomial logistic regression (MLR) [7, 8], and decision tree [9, 10]. Within the similar data which exists, spectral changes in various materials and various spaces might have the same features, so the attained details were still corrupt because of inadequate spatial structure feature extraction. To solve these issues, it is hard to perfect the classification of HSI. So, numerous spectral and spatial feature extraction methods are proposed.

These techniques have validated major classification performance, which is not in effect for classifying HSI in difficult situations. In recent times, deep learning techniques had achieved maximum success for this kind of task [11–13]. So, this method had reached admirable performance for different analysis-oriented tasks, e.g., object recognition and image classification. To classify HSI, entire spatial and spectral perspectives must be considered for the processing. Intuitively, HSI consists of a higher number of images and every image signifies electromagnetic spectrum classification. Temporarily, the spatial perspective denotes 2D spatial data of objects consistent in the HSI. Thus, HSI is typically denoted as the 3D spectral-spatial data. Therefore, many methods had been proposed in the literature [14, 15].

Towards concurrently modelling spectral-spatial data, certain developer attempts were made. This method performed operations in a stacked manner along with convolution over spectral and spatial feature space in a stacked manner, named CNN model [16]. Apparently, the benefit of this CNN model may create rich feature maps. Moreover, the major drawback of this method is threefold. Initially, It is hard to generate a deeper CNN structure. An intention in the resultant area increasingly improves through cumulative amount in the convoluted function that confines the interpretation ability and depth of the model. Next to that, the cost of the memory is too expensive while maximum convolution operations were performed [17–20]. To reduce the abovementioned challenges, we introduced the new CNN model namely compressed synergic deep convolution neural network with Aquila optimization (CSDCNN-AO).

The significant goals to achieve the above-said objectives are listed below:

(i) to determine the suitable deep learning method which provides huge support for HSI image classification.

(ii) To reduce the complexity and loss function in classification.

(iii) To develop the future outcome based on both present and traditional output.

The major contribution of this technique is given below.

This combination will reduce the learning complexity of the wavelet concept and reduce the loss function with the Aquila optimization. This Aquila optimization method could reduce the enormous amount of data features by maintaining its unique possessions and using less time for computation and less memory space. Furthermore, a synergic deep convolutional neural network (CNN) is useful and intended for getting an initial result, similarly, the CNN weights are optimized by Aquila optimization for reducing an error rate. Here, the key role is the compression of data with the Aquila optimization technique with CNN for increasing accuracy with maximum steadiness among both exploitation and exploration of optimization.

The organization of the work is given below:

the literature survey is given in Section 2. In Section 3, the proposed methodology is given. In Section 4, the experimental results and discussions are explained. At last, in Section 5, the conclusion is given.

2. Literature Review

Yang et al. [21] present a novel synergistic CNN for an accurate HSI classification. The SyCNN contains the hybrid structure of 2D and 3D CNNs with a data interaction module with feature learning that fuses both spatial and spectral HSI data. Moreover, it presents a three-dimensional process earlier to a fully connected layer that supports and extracts features effectively. But still, they could not handle high-dimensional data.

Li et al. [22] suggested an HSI model called local and hybrid dilated convolution fusion network (LDFN) that combines both the local and rich spatial features through expanding the perception field. Initially, several functions were considered, such as dropout, standard convolution, batch normalization, and average pooling. After that, both local and dilated convolution operations were involved in efficient spatial-spectral feature extraction. On the other hand, parameters were manually selected in the suggested paper.

Patel et al. [23] suggested HSI categorization by an autoencoder through CNN (AECNN). Pre-processed by autoencoder-enhanced HSI features that helped towards obtaining optimized weights in CNN initial layers. Thus, here, CNN with a shallow model could be applied towards extracted features from the HSI data. But still, they need to cover more contextual information and advanced strategies for robustification of the spatial information.

Wang et al. [24] suggested a semi-supervised HSI classification model which improved deep learning. Here, the suggested model namely the arbitrary multiple graphs method, and then replaced skilled learning with the anchor graph method that could be labelled a significant unlabelled data automatically and precisely. In this, the number of training samples is limited.

Shi et al. [25] presented a model namely the 3D coordination attention mechanism (3DCAM). This attention process could not attain the HIS's spatial position in both vertical and horizontal ways. Also, HSIs spatial and spectral data were extracted, using CNN. The drawback is that the implementation complexity is not considered.

Zhao et al. [26] suggested combining stacked autoencoder (SAE) with 3D deep residual network (3DDRN) to classify HSI. An SAE neural network was designed to reduce HSI size. 3DCNN and residual network module were used to develop 3DDRN. The 3DDRN extracted spectral-spatial features from dimension-reduced 3D HSI cubes. 3DDRN continuously identified deep features, which were passed into SoftMax to complete classification. Batch normalization (BN) and dropout were used to avoid overfitting training data.

Yin et al [27] developed a spatial-spectral mixed network for HSI categorization. The network collects spatial-spectral information from HSI using three layers of 3-D convolution and one layer of 2-D convolution. This network employs Bi-LSTM to boost spectral band interactions and extract spectral features as a series of images. Combining two FC layers and utilising SoftMax for classification creates a unified neural network. However, the model misclassified samples in the dataset.

Paul et al. [28] developed SSNET, which blends 3D and 2D convolutions of HSI spectral-spatial information with SPP for creating spatial features at various scales. SPP is employed in two-dimensional local convolutional filters for HSI classification because it resists object distortions. SPP layer's fixed feature vector output reduces trainable parameters and improves classification performance. They do, however, have a complicated structure.

Zhang et al. [29] introduced an SSAF-DCR for hyperspectral image classification. Three components were linked to extract features in the recommended network. First, a dense spectral block reuses spectral characteristics as much as possible. Then, a spectral attention block refines and optimises the spectral features. In the second segment, a dense spatial block and an attention block pick spatial features. But in this, the selection of the number of features is not considered.

Yan et al. [30] offer a 3D cascaded spectral-spatial element attention network (3D-CSSEAN) for picture classification. Using the spectral element attention module and the spatial element attention module, the network may concentrate on key spectral and spatial aspects. Two-element attention modules were built using activation functions and element-wise multiplication. The model can extract classification-helping properties and is computationally efficient. The network structure is also suitable for small sample learning since the attention module has few training parameters. On the other hand, obtaining labelled samples are expensive and difficult.

To overcome existing challenges, our proposed work introduces novel techniques which are discussed in the following section.

3. Proposed Synergic Deep Learning Model

Let us assign the hyperspectral image $x = [X_1, X_2, X_3, \ldots, X_s]^t \in r^{s\times(c\times d)}$, where s represented entire bands with $c \times d$ band samples. Additionally, t is the sample in which $x = (X_i, Y_i) \in (r^{s\times(c\times d)}, r^y)$ with Y_i labels. Usually, HSI classification is affected due to inter-class similarity and high intra-class variability. To compensate for these issues, we introduce the proposed technique namely, the synergic deep learning model with the feature reduction principle. This method minimizes complexities for computation by reducing spectral and spatial feature dimensions. Here, we evaluate the efficiency of the subsequent feature suppression methods using a hybrid synergic deep CNN model. The proposed synergic deep learning model consists of synergic deep learning (SDL)-based feature extraction, feature reduction, classification, and loss function optimization. The schematic representation of the proposed method is represented in Figure 1, which is given in the following sections.

3.1. Synergic Deep Convolutional Neural Network Feature Extraction.

In this proposed model as shown in Figure 2, we extract the HSI useful features which are normally represented by the input layer, n DCNN components and synergic network (c_n^2). Recently, DCNN yields more attention for the classification which is proposed to reduce the number of input variables and develop the neural network architecture. DCNN is a combination of layers where each layer performs different functions. Pre-processing, convolution, pooling, and final classification operations are sequentially performed in synergic DCNN [31]. The forward process is a convolution operation on the inputs. The multiplication between weights and inputs is combined across layers. The filter has the same number of layers as input volume channels, and output volume has the same depth as the number of filters. In the convolution process, several computations are carried out. Every layer is composed of neurons that take input values, perform calculations, and produces output values, which are forwarded to the next layer. Under CNN, there are four important operations performed in feature learning: the convolution, the activation, the pooling, and the normalization. Before convolution operation, pre-processing is worked out.

3.1.1. Pair Input Layer.

Synergic pair input layers are trained randomly, and here, each 200-data group with corresponding class labels is given to the DCNN units. Here, the image is in the size of $224 \times 224 \times 3$. Before applying the data to the next layer, we have to apply the feature reduction principle.

3.1.2. Feature Reduction by Wavelet Transform.

In this feature reduction concept, we used wavelet transform with the Haar basis model so that they can handle the high-dimensional data efficiently. Here, two filters h and g are applied for effective feature reduction. These filters are incorporated with the transforms to yield deducted input coefficients. The following equation is for the feature reduction which is given in equation (1).

$$\begin{aligned} x' &= \begin{bmatrix} h \\ g \end{bmatrix} x \left[\frac{h}{g}\right]^T = \begin{bmatrix} hx \\ gx \end{bmatrix} \left[h^T g^T\right] \\ &= \begin{bmatrix} hxh^T hxg^T \\ gxh^T gxg^T \end{bmatrix}. \end{aligned} \tag{1}$$

As a result of this transformation into the DCNN, learning complexity and learning time can be reduced. In this process, it reduces CNN architecture with the number of features.

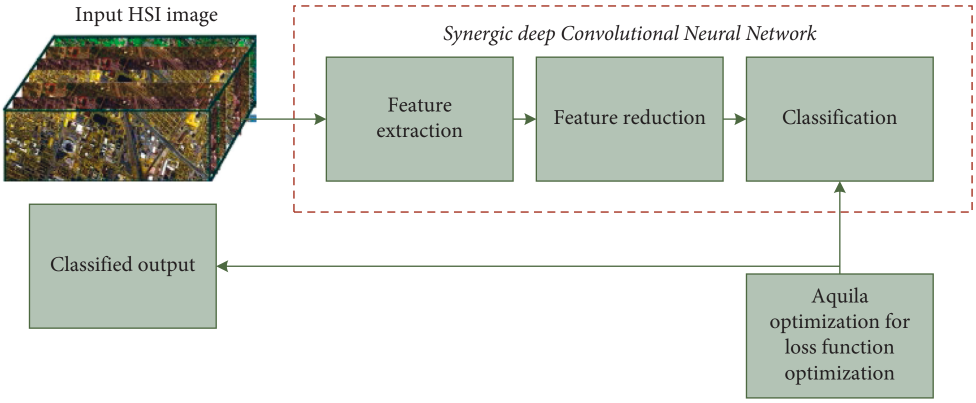

Input HSI image
Synergic deep Convolutional Neural Network
Feature extraction
Feature reduction
Classification
Classified output
Aquila optimization for loss function optimization

Figure 1: Schematic representation of the proposed methodology.

3.1.3. DCNN Component. In every DCNN component, we initiate with ResNet-101 architecture which is denoted as DCNN-n ($n = 1$, 2, …, N). This type of architecture is suitable for synergic deep learning (SDL) method. Here, we consider the data sequence with compressed features $x' = \left\{X'^{(1)}, X'^{(2)}, \ldots, X'^{(n)}\right\}$ and output class label series $y' = \left\{Y'^{(1)}, Y'^{(2)}, \ldots Y'^{(n)}\right\}$. This has to be intended with the θ variable which undertakes cross-entropy loss expressed in equation (2).

$$\log(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} 1\left\{Y^{(1)} = j\right\}\log\frac{e^{z_j^{(i)}}}{\sum_{L=1}^{k} z_L^{(i)}}\right]. \tag{2}$$

The above equation (2), $z^{(i)} = f(X^{(A)}, \theta)$ means the forward computing process. In the same way, the variable used in DCNN-n is mentioned as θ^{i}, and these components will not share enormous DCNN components.

In this SDN model, synergic labels in DCNN are applied to input layers, embedding, and learning layers. In SDN, the consequence data pair is denoted as (z_I, z_J), and this pair of input is given to (DCNNi, DCNNj). Output from the FC layer is given in the following equations (3) and (4).

$$f_I = \psi\left(z_I, \theta^{(i)}\right), \tag{3}$$

$$f_J = \psi\left(z_J, \theta^{(j)}\right). \tag{4}$$

In the next stage, all the deep features are embedded $f_{I\circ J}$ and the resultant outcome is expressed in the following equation (5).

$$Y_{\mathrm{SDL}}\left(z_I, z_J\right) = \begin{cases} 1, & \text{if } Y_I = Y_J, \\ , & \text{if } Y_I \neq Y_J. \end{cases} \tag{5}$$

Loss in binary cross-entropy is given as below:

$$L^{\mathrm{SDL}}\left(\theta^{\mathrm{SDL}}\right) = Y_{\mathrm{SDL}}\log \widehat{Y}_{\mathrm{SDL}} + \left(1 - Y_{\mathrm{SDL}}\right)\log\left(1 - \widehat{Y}_{\mathrm{SDL}}\right). \tag{6}$$

The above expression θ^{SDL} represents the synergic attributes, and Y_{SDL} represents the synergic forward computation. This process validates data pair classes and yields a recovery response belonging to the synergic (SN) errors.

3.1.4. Training and Testing. In this stage, we do the SN maximization process

$$\begin{cases} \theta^{(i)}(z+1) = \theta^{(a)}(z) - \gamma(z).\Delta^{(i)}, \\ \theta^{\mathrm{SDL}(i)}(z+1) = \theta^{\mathrm{SDL}(a)}(z) - \gamma(z).\Delta^{\mathrm{SDL}(i,j)}, \end{cases} \tag{7}$$

where, SDL(i, j) and $\gamma(z)$ represents the learning rate

$$\begin{aligned} \Delta^{(i)} &= \frac{\partial L^{(a)}\left(\theta^{(i,j)}\right)}{\partial \theta^{(i,j)}} + \vartheta \sum_{j=1, j\neq a} \frac{\partial L^{SDL(i)}\left(\theta^{SDL(i,j)}\right)}{\partial \theta^{SDL(i,j)}}, \\ \Delta^{SDL(i)} &= \frac{\partial L^{SDL(a)}\left(\theta^{SDL(i,j)}\right)}{\partial \theta^{SDL(i,j)}}, \end{aligned} \tag{8}$$

where, ϑ refers to the trade-off among synergic error and classification sub-model. Additionally, test data classification belonging to the SN DCNN component is processed under some of the prediction vectors which are represented as $p^{(i)} = (P_1^{(i)}, P_2^{(i)}, \ldots, P_k^{(i)})$. Further, the test data class label is deliberated as below:

$$y'(z) = \arg\max_v \left\{\sum_{U-1}^{K} P_1^{(U)}, \ldots, \sum_{U-1}^{K} P_v^{(U)}, \ldots, \sum_{U-1}^{K} P_k^{(U)}\right. \tag{9}$$

3.2. Image Classification. This is the final stage to classify the HSI images concerning the different class labels. This classification is performed under the SoftMax layer which has more attention for the multi-label classification. It leads to a mapping function on behalf of the C input vector as of space n to class k labels, which is given in equation (10).

$$v_Q = \frac{e\left(\theta_{zQ}C\right)}{\sum_{k=1}^{k} e\left(\theta_K^z C\right)}, \tag{10}$$

where, $Q = 1, 2, \ldots, k$ and $\theta_k = [\theta_{K1}, \theta_{K2} \ldots \theta_{Kn}]^z$ refers the weights, and this has to be tuned using the optimization

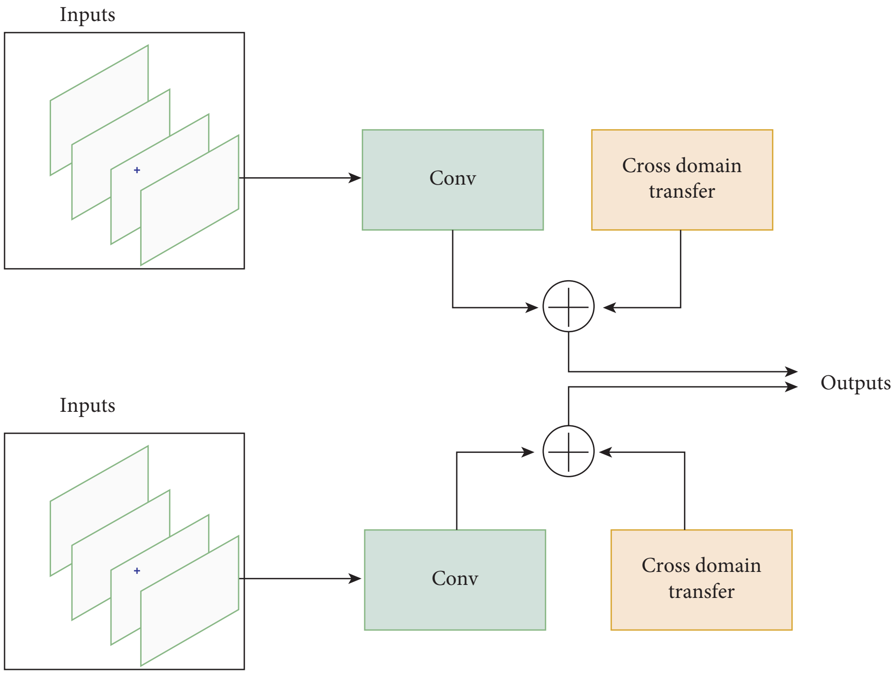


Figure 2: Synergic deep learning model.

process. As a result, we can reduce the loss function in this architecture.

3.3. Loss Reduction by Aquila Optimization Algorithm.

Losses in this SDL are reduced by the Aquila optimization algorithm with the weight tuning process. This Aquila optimization algorithm yields the best solution despite the definite limitations.

The mathematical model of Aquila optimization (AO) [32] consists following stages: expanded exploration, narrowed exploration, expanded exploitation, and narrowed exploitation.

3.3.1. Expanded Exploration.

In this work, Aquila recognizes the best weight θ_k based on the best hunting area. Here, the best hunting area refers to the minimum losses. In this process, the AO (weight optimization) extensively explores extraordinary soar to conclude the search space area.

$$\theta_1(T+1) = \theta_{\mathrm{BEST}}(T) \times \left(1 - \frac{T}{t}\right) + \left(\theta_m(T) - \theta_{\mathrm{BEST}}(T) * RAND\right), \tag{11}$$

where, $\theta_1(T+1)$ refers to the next iteration solution, and this is estimated by the initial search method θ_1. $\theta_{\mathrm{BEST}}(T)$ is considered as the best until iteration T. Expanded search (exploration) is controlled by the $(1 - T/t)$ iteration. In addition to that, $\theta_m(T)$ represented the current location mean value which is calculated in the following equation. t and T are the maxima and current iterations.

$$\theta_m(T) = \frac{1}{n}\sum_{a=1}^{n} \theta_a(T), \tag{12}$$

where, n is the population size.

3.3.2. Narrowed Exploration.

In this stage, AO barely discovers (explores) the certain space of the targeted prey for the solution.

$$\begin{aligned} \theta_2(T+1) = {} & \theta_{\mathrm{BEST}}(T) \times \mathrm{LEVY}(d) \\ & + \theta_r(T) + (v - u) * \mathrm{RAND}, \end{aligned} \tag{13}$$

where, $\theta_2(T+1)$ is the next iteration solution. LEVY (d) and d is the levy flight distribution function and dimension space, respectively. Additionally, $\theta_r(T)$ is the random solution which is taken from the range of $(1, \ldots, n)$.

$$\mathrm{LEVY}(\mathrm{d}) = S \times \frac{U \times \rho}{|V|^{1/\beta}}, \tag{14}$$

where, S refers to the constant which has the value of 0.01. Moreover, U and V are constant numbers.

$$\rho = \frac{\Gamma(1+\beta) \times \sin e(\pi\beta/2)}{\Gamma(1+\beta/2) \times \beta \times 2(\beta - 1/2)}. \tag{15}$$

In the above equation (15), β is the constant value. Moreover, the value of u and v are calculated as follows, which is used for spiral search in this optimization.

$$\begin{aligned} v &= R\cos(\phi), \\ u &= R\sin(\phi), \\ R &= R_1 + \varepsilon \times d_1, \\ \phi &= -\varpi \times d_1 + \phi_1, \end{aligned} \tag{16}$$

R_1 has the values from 20 to toward fixed search cycles, and ε has the value of 0.00565. d_1 differs based on dimension, then ϖ is a minimum value which is a constant 0.005.

3.3.3. Expanded Exploitation (X3). In this stage, weight optimization exploits the accurate value of the solution for getting nearer to prey and attack.

$$\theta_3(t+1) = \theta_{\text{BEST}}(T) - \theta_m(T) \times \delta - \text{RAND} + \left(\left(B_{\text{upper}} - B_{\text{lower}}\right) \times \text{RAND} + B_{\text{lower}}\right) \times \alpha, \tag{17}$$

where $\theta_3(t+1)$ refers to the next iteration solution, and $\theta_{\text{BEST}}(T)$ represents the estimated prey location. In addition to that, $\theta_m(T)$ represents the current mean value at the Tth iteration, and RAND means the random value which is between 0 and 1. α and δ are the small values (0, 1) which are adjustment parameters for the exploitation process. B_{upper} and B_{lower} represents the upper and lower bound of the problem, respectively.

3.3.4. Narrowed Exploitation. In this phase, attacking is processed in the last location.

$$\theta_4(T+1) = qf \times \theta_{\text{BEST}}(T) - \left(g_1 \times \theta(T) \times \text{RAND}\right) - g_2 \times \text{LEVY}(d) + \text{RAND} \times g_1, \tag{18}$$

where $\theta_4(T+1)$ demonstrates the next iteration solution. qf mentions the quality function which is applied for balancing the search strategies. g_1 specifies several optimization motions that are applied for tracking the prey. g_2 specifies the values that are reduced from two to zero. $\theta(T)$ represents t iteration with the current solution.

$$\begin{aligned} qf(T) &= T^{2\times\text{RAND}-1/(1-t)^2}, \\ g_1 &= 2 \times \text{RAND} - 1, \\ g_2 &= 2 \times \left(1 - \frac{T}{t}\right). \end{aligned} \tag{19}$$

$qf(T)$ refers to the tth iteration's quality function, and RAND means random value between 0 and 1. T and t presents the maximum and current iteration, respectively. Levy(D) is the levy flight distribution function calculated using equation (6). As a result, we can get optimum weights which reduces losses in the architecture.

4. Experimental Results and Discussion

In our work, we have used four HSI datasets which are used for analyzing our proposed CSDCNN-AO technique. Here, we use Houston U (HU) dataset [33], Indiana Pines (IP) [34], Kennedy Space Center (KSC) [35], and Salinas Scene (SS) dataset [17]. In the case of the IP dataset, the size of the dataset is 145 × 145. For the KSC dataset, the size is equivalent to 512 × 614 with 13 classes of ground truths.

4.1. Dataset and Its Description

4.1.1. Houston U (HU) Dataset. The first dataset is GRSS DFC 2013, which measures 349 1905 bytes, and has 144 bands spanning the wavelength range 380–1050 nm. It was obtained by the National Center for Airborne Laser Mapping (NCALM) and has a spatial resolution of 2.5 metres over the University of Houston. The picture is separated into two halves: the bright and dark sections. The bright section has 4143 samples, whereas the dark section contains 824 samples.

4.1.2. Indiana Pines (IP). This agricultural dataset was collected in 1992 from Northwest Indiana utilising the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. It has 145 × 145 pixels and 16 vegetation classifications with 20 m per pixel spatial resolution. After removing 4 zero bands and 20 bands affected by water absorption effects, 200 spectral bands ranging from 400 to 2500 nm with 10-nm intervals were used for analysis.

4.1.3. Kennedy Space Center (KSC). The AVIRIS instrument in Florida collected the Kennedy Space Center dataset in 1996. It has a resolution of 512 by 614 pixels, 176 bands, and 13 categories.

4.1.4. Salinas Scene (SS) Dataset. Experiments on the Salinas Scene collected by the AVIRIS sensor over Salinas Valley, California, USA, with a spatial resolution of 3.7 m per pixel in the wavelength range of 0.4–2.5 m and a spectral resolution of 10 nm, used a second set of AVIRIS data. It measures 512 × 217 × 224 pixels (water absorption bands included).

The model for comparison enactment depending on the IP dataset through different classes are evaluated.

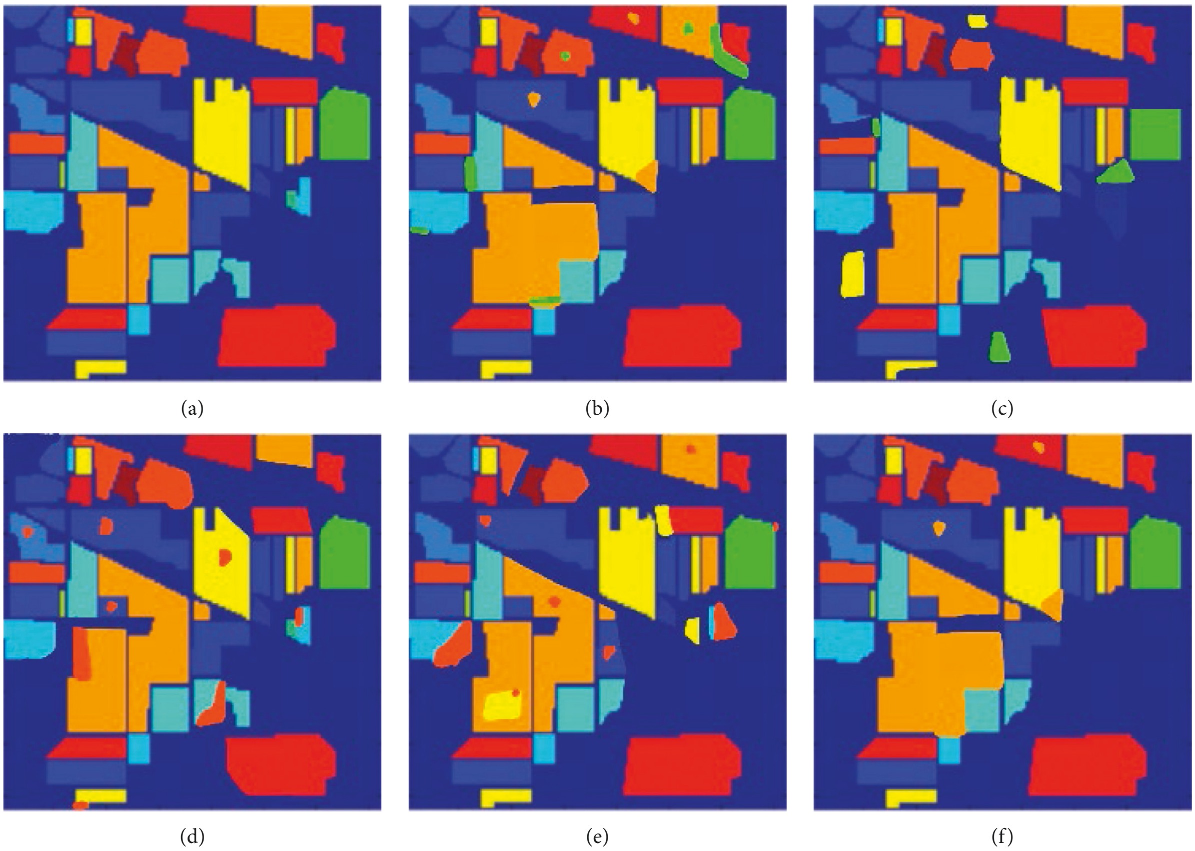
In this Table 1, we evaluated the classification performance for the Indian Pines Scene dataset. Here, overall accuracy, average accuracy, and Kappa coefficients are evaluated. From the results, we can show that our proposed CSDCNN-AO yields maximum performance than other techniques. In Table 1, CSDCNN-AO achieves a better result for the 13th class. In the case of CSDCNN, the 8th class achieves a better performance. For SDCNN, the 16th class has the maximum performance. DCNN also attains maximum performance for the 16th class only. For RNN, it has the maximum performance under the 6th class.

In the above Figure 3, (a) represents the original image and here we evaluated the results of the proposed algorithm with other algorithms like CSDCNN-ALO [36], CSDCNN-PSO [37], CSDCNN-WOA [38], and CSDCNN-GWO [39]. Different application [40–45] were used in different fields for optimization. Among these methods, our proposed work yields the maximum performance since the performance of our proposed work is nearly equivalent to the original ground truth image compared to others.

In this Table 2, we evaluated the classification performance for the KSC dataset. Here, abovementioned performances are evaluated. From the results, we can show that our proposed CSDCNN-AO yields the maximum performance than other techniques. In Table 2, CSDCNN-AO achieves a better result for the 10th class. In the case of CSDCNN, the 11th class achieves a better performance. For SDCNN, the 13th class has the maximum performance. DCNN attains the

Table 1: HSI categorization for Indiana Pines (IP) dataset.

Methods	RNN	DCNN	SDCNN	CSDCNN	CSDCNN-AO
OA	46.33 ± 0.45	48.73 ± 0.89	89.36 ± 1.13	89.57 ± 0.86	93.44 ± 1.08
AA	36.20 ± 1.06	49.60 ± 3.29	88.46 ± 1.17	83.14 ± 1.11	94.44 ± 1.82
K	53.97 ± 0.58	51.04 ± 1.03	89.62 ± 2.54	89.12 ± 0.26	98.33 ± 1.25
1	22.89 ± 1.09	1.33 ± 7.33	90.00 ± 1.03	30.21 ± 30.0	93.77 ± 11.6
2	45.46 ± 5.00	41.53 ± 3.04	87.35 ± 3.80	81.79 ± 0.26	90.38 ± 4.87
3	26.69 ± 2.61	30.91 ± 8.28	87.18 ± 7.24	75.93 ± 1.26	90.06 ± 4.53
4	22.79 ± 9.7	21.17 ± 3.25	83.17 ± 5.52	89.11 ± 1.12	96.84 ± 4.89
5	37.71 ± 6.67	69.79 ± 2.13	86.75 ± 2.55	79.28 ± 1.34	95.65 ± 1.95
6	89.57 ± 1.71	91.78 ± 0.78	89.08 ± 3.06	92.82 ± 0.32	96.95 ± 0.96
7	39.54 ± 11.4	19.85 ± 7.59	69.89 ± 29.7	39.69 ± 2.13	91.48 ± 24.0
8	87.46 ± 2.15	87.84 ± 3.15	85.25 ± 2.40	99.22 ± 0.31	89.11 ± 3.09
9	47.78 ± 19.04	0.00 ± 0.00	49.0 ± 49.0	19.00 ± 2.05	92.72 ± 8.38
10	49.46 ± 1.91	52.53 ± 1.24	86.47 ± 7.69	74.28 ± 0.89	92.40 ± 2.86
11	70.89 ± 2.49	61.88 ± 4.33	91.88 ± 5.03	91.12 ± 0.25	93.97 ± 3.29
12	37.14 ± 5.56	37.46 ± 3.85	77.82 ± 5.16	85.88 ± 2.37	87.56 ± 3.44
13	32.68 ± 7.17	85.02 ± 1.22	96.26 ± 5.29	50.86 ± 3.56	98.89 ± 0.87
14	81.32 ± 8.95	89.94 ± 2.59	89.16 ± 2.22	93.89 ± 1.37	96.89 ± 2.57
15	45.75 ± 5.12	44.64 ± 4.56	94.00 ± 7.49	93.76 ± 1.98	89.74 ± 2.65
16	29.60 ± 34.12	95.38 ± 1.94	99.89 ± 3.86	98.11 ± 2.67	96.89 ± 4.98



(a) (b) (c) (d) (e) (f)

Figure 3: HSI classified image for IP dataset (a) original ground truth image (b) CSDCNN-ALO, (c) CSDCNN-PSO, (d) CSDCNN-WOA, and (e) CSDCNN-GWO (f) CSDCNN-AO.

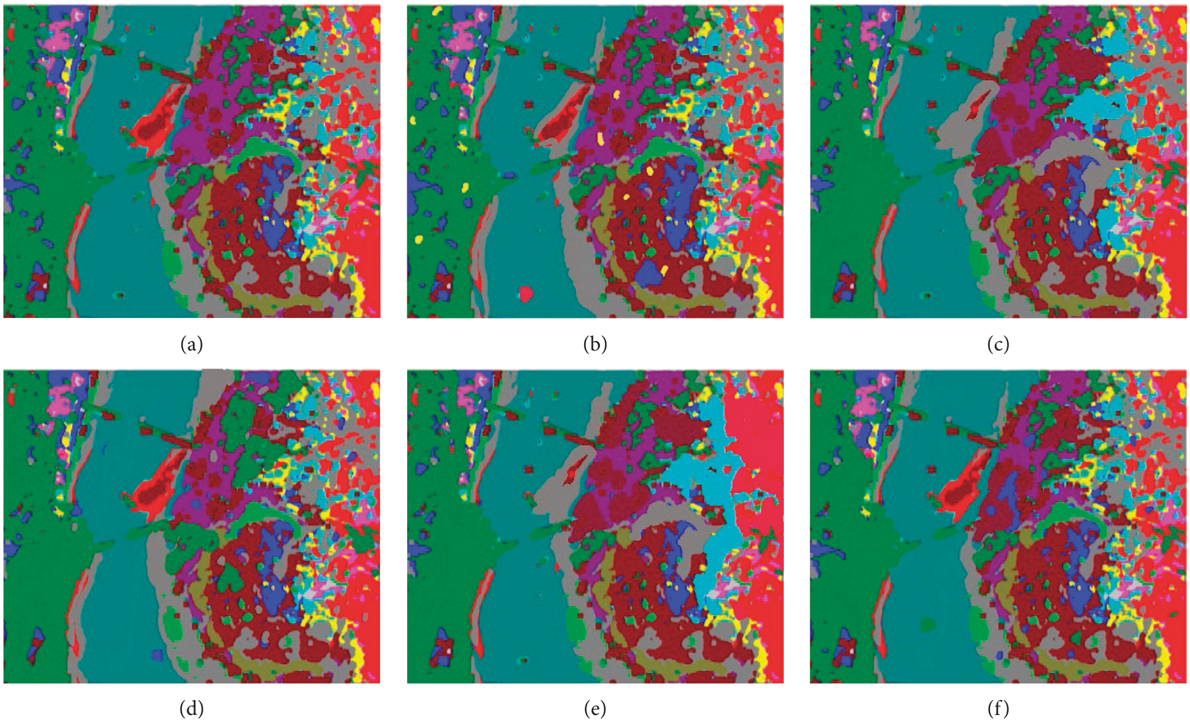
maximum performance for the 8th class. For RNN, it has the maximum performance under the 6th class.

In the above Figure 4, (a) represents the original image and here we evaluated the results of the proposed algorithm with other algorithms like CSDCNN-ALO, CSDCNN-PSO, CSDCNN-WOA, and CSDCNN-GWO. From these methods, our proposed work yields the maximum performance since the obtained proposed image is nearly equivalent to the original ground truth image.

In this Table 3, we evaluated the classification performance for the Salinas Scene (SS) dataset. From the results, we can show that our proposed CSDCNN-AO yields the

Table 2: HSI categorization for KSC dataset.

methods	RNN	DCNN	SDCNN	CSDCNN	CSDCNN-AO
OA	45.22 ± 0.45	47.89 ± 0.89	88.36 ± 1.13	87.48 ± 0.86	92.33 ± 1.08
AA	37.31 ± 1.06	56.60 ± 3.29	85.39 ± 1.17	82.43 ± 1.11	93.44 ± 1.82
K	52.97 ± 0.58	53.12 ± 1.03	91.62 ± 2.54	93.12 ± 0.26	97.22 ± 1.25
1	22.89 ± 1.09	30.33 ± 7.33	89.00 ± 1.03	29.21 ± 30.0	84.77 ± 11.6
2	44.46 ± 5.00	42.53 ± 3.04	86.35 ± 3.80	84.79 ± 0.26	92.38 ± 4.87
3	46.69 ± 2.61	53.91 ± 8.28	86.18 ± 7.24	85.93 ± 1.26	93.06 ± 4.53
4	29.79 ± 9.7	21.17 ± 3.25	84.17 ± 5.52	88.11 ± 1.12	95.84 ± 4.89
5	36.71 ± 6.67	68.79 ± 2.13	85.75 ± 2.55	87.28 ± 1.34	94.65 ± 1.95
6	88.57 ± 1.71	92.78 ± 0.78	88.08 ± 3.06	93.82 ± 0.32	95.95 ± 0.96
7	38.34 ± 11.4	20.85 ± 7.79	71.89 ± 29.45	42.69 ± 2.13	92.48 ± 24.0
8	91.46 ± 2.15	94.84 ± 3.15	91.25 ± 2.40	95.22 ± 0.31	97.11 ± 3.09
9	64.78 ± 19.04	0.00 ± 0.00	54.0 ± 49.0	32.00 ± 2.05	89.72 ± 8.38
10	61.46 ± 1.91	61.53 ± 1.24	89.47 ± 7.69	81.28 ± 0.89	98.40 ± 2.86
11	79.89 ± 2.49	66.88 ± 4.33	93.77 ± 5.03	95.85 ± 0.67	93.97 ± 3.29
12	47.14 ± 5.56	47.57 ± 3.85	77.97 ± 5.16	83.77 ± 4.37	95.89 ± 4.55
13	49.68 ± 7.17	87.02 ± 1.22	97.45 ± 5.29	79.86 ± 4.56	97.89 ± 0.87



(a) (b) (c)

(d) (e) (f)

Figure 4: HSI classified image for KSC dataset: (a) original ground truth image, (b) CSDCNN-ALO, (c) CSDCNN-PSO, (d) CSDCNN-WOA, (e) CSDCNN-GWO, and (f) CSDCNN-AO.

maximum performance than the other techniques. In Table 3, CSDCNN-AO achieves a better result for the 13th class. In the case of CSDCNN, the 16th class achieves a better performance. For SDCNN, the 14th class has the maximum performance. DCNN also attains the maximum performance for the 16th class only. For RNN, it has the maximum performance under 11th class.

In the above Figure 5, (a) represents the original image and here we evaluated the results of the proposed algorithm with other algorithms like CSDCNN-ALO, CSDCNN-PSO, CSDCNN-WOA, and CSDCNN-GWO. From these methods, our proposed work yields the maximum performance since the obtained proposed image is nearly equivalent to the original ground truth image.

In Table 4, we evaluate the classification performance for the Houston U dataset. From the results, we can show that our proposed CSDCNN-AO yields the maximum performance than the other techniques. In Table 4, CSDCNN-AO achieves a better result for the 11th class. In the case of CSDCNN, the 8th class achieves a better performance. For SDCNN, the 15th class has the maximum performance. DCNN also attains the maximum performance for the 14th class only. For RNN, it has the maximum performance under 8th class.

Table 3: HSI categorization for Salinas Scene (SS) dataset.

Methods	RNN	DCNN	SDCNN	CSDCNN	CSDCNN-AO
OA	46.78 ± 1.45	57.49 ± 1.39	89.65 ± 1.13	91.89 ± 0.86	95.77 ± 1.08
AA	78.20 ± 1.06	49.60 ± 3.29	88.46 ± 1.17	83.14 ± 1.11	94.44 ± 1.82
K	53.97 ± 0.58	51.04 ± 1.03	89.62 ± 2.54	89.12 ± 0.26	98.33 ± 1.25
1	22.89 ± 1.09	1.33 ± 7.33	90.00 ± 1.03	30.21 ± 30.0	93.77 ± 11.6
2	45.46 ± 5.00	41.87 ± 3.04	88.79 ± 2.80	82.69 ± 0.26	92.56 ± 4.87
3	26.69 ± 2.61	30.91 ± 8.28	87.18 ± 7.24	75.93 ± 1.26	90.06 ± 4.53
4	22.79 ± 9.7	21.17 ± 3.25	83.17 ± 5.52	89.11 ± 1.12	96.84 ± 4.89
5	37.71 ± 6.67	69.79 ± 2.13	86.75 ± 2.55	79.28 ± 1.34	95.65 ± 1.95
6	89.57 ± 1.71	91.78 ± 0.78	89.08 ± 3.06	92.82 ± 0.32	96.95 ± 0.96
7	39.54 ± 11.4	19.85 ± 7.59	69.89 ± 29.7	39.69 ± 2.13	91.48 ± 24.0
8	87.46 ± 2.15	87.84 ± 3.15	85.25 ± 2.40	96.22 ± 0.31	88.11 ± 3.09
9	51.81 ± 19.04	0.00 ± 0.00	48.0 ± 49.0	22.00 ± 2.05	95.72 ± 8.38
10	67.46 ± 1.91	63.53 ± 1.24	90.47 ± 6.87	87.28 ± 0.89	95.30 ± 2.86
11	90.56 ± 3.49	73.88 ± 4.33	94.57 ± 5.03	95.21 ± 0.23	97.94 ± 3.29
12	54.14 ± 5.56	38.46 ± 3.85	76.47 ± 5.16	92.77 ± 3.37	92.56 ± 3.44
13	34.47 ± 7.17	83.02 ± 1.22	97.26 ± 5.29	49.93 ± 3.56	99.89 ± 0.87
14	84.32 ± 8.95	87.94 ± 2.59	99.16 ± 2.22	92.89 ± 1.37	95.89 ± 2.57
15	48.75 ± 5.12	47.64 ± 4.56	91.00 ± 7.49	96.76 ± 2.98	93.74 ± 2.65
16	34.60 ± 34.12	97.38 ± 1.94	97.89 ± 3.86	99.11 ± 2.67	95.89 ± 4.98

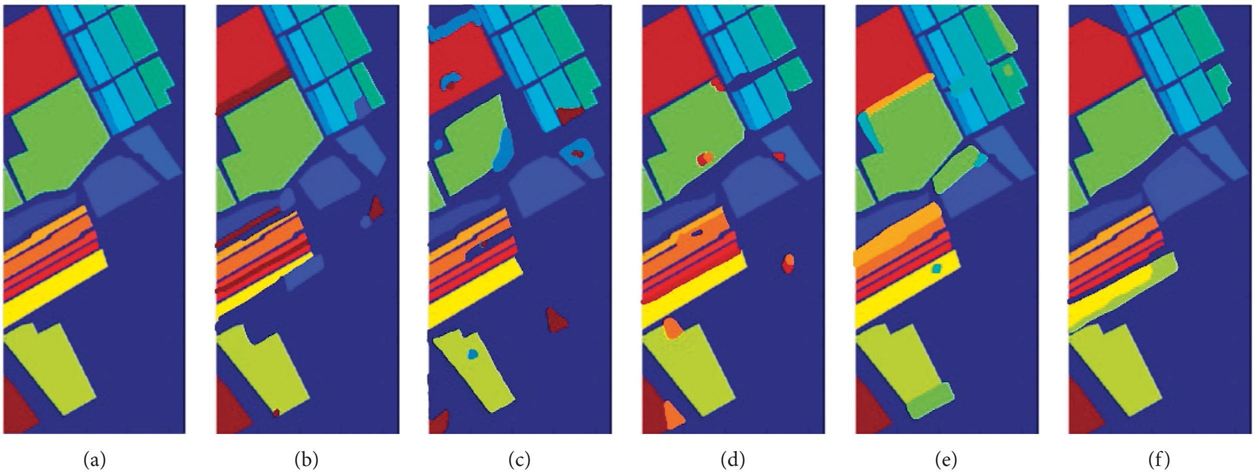
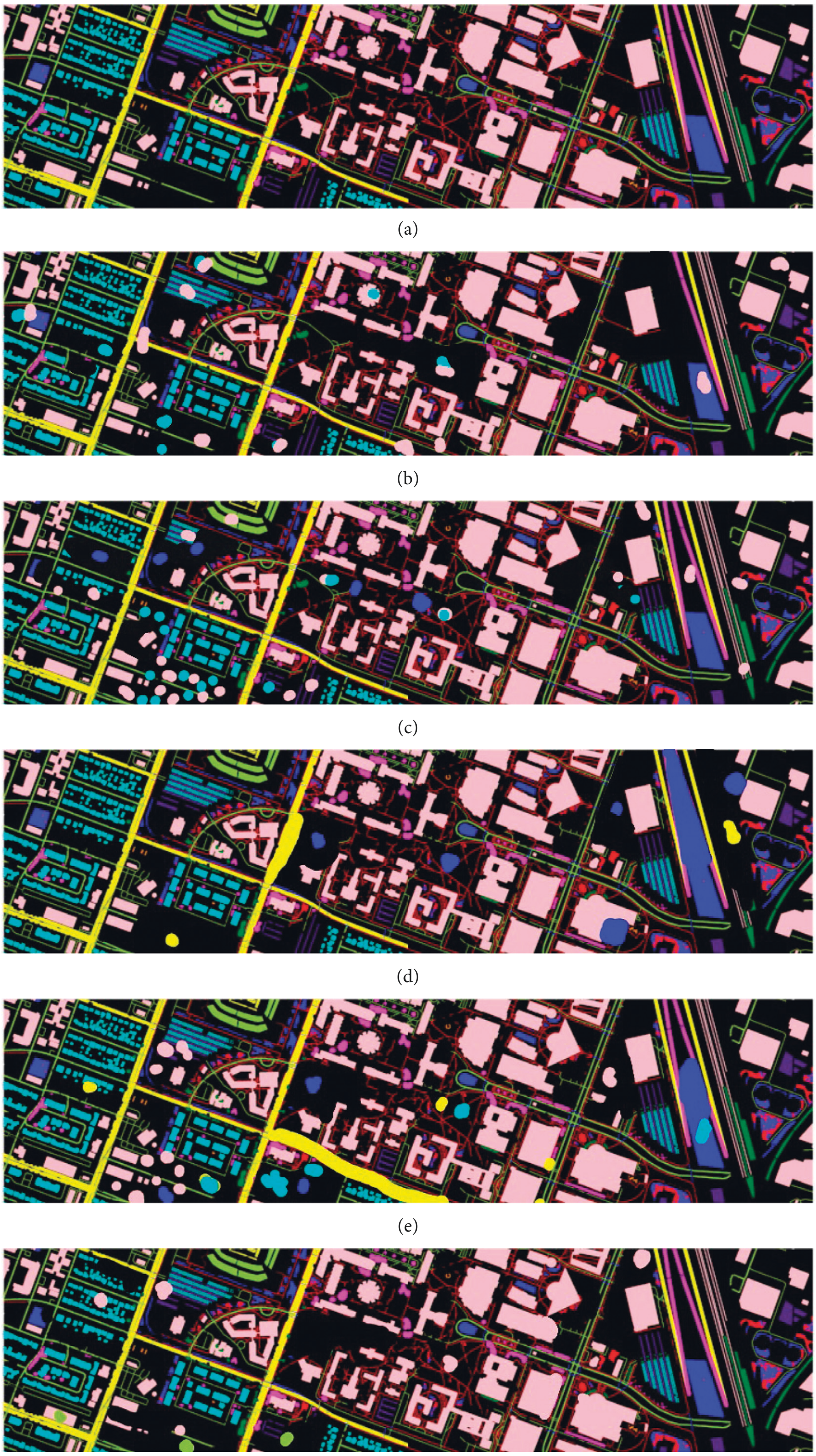


Figure 5: HSI classified image for SS dataset (a) original ground truth image (b) CSDCNN-ALO, (c) CSDCNN-PSO, (d) CSDCNN-WOA, and (e) CSDCNN-GWO (f) CSDCNN-AO.

Table 4: HSI categorization for the Houston U dataset.

Methods	RNN	DCNN	SDCNN	CSDCNN	CSDCNN-AO
OA	49.23 ± 1.45	57.49 ± 1.39	88.73 ± 1.13	93.78 ± 0.86	94.67 ± 1.08
AA	78.20 ± 1.06	49.60 ± 3.29	88.46 ± 1.17	83.14 ± 1.11	94.44 ± 1.82
K	53.97 ± 0.58	51.04 ± 1.03	89.62 ± 2.54	89.13 ± 0.26	97.33 ± 1.25
1	27.89 ± 1.09	1.66 ± 7.33	92.00 ± 1.03	37.21 ± 30.0	94.77 ± 11.6
2	49.46 ± 5.00	42.87 ± 3.04	89.79 ± 2.80	82.69 ± 0.26	92.56 ± 4.87
3	26.69 ± 2.61	30.91 ± 8.28	87.18 ± 7.24	75.93 ± 1.26	90.06 ± 4.53
4	22.79 ± 9.7	21.17 ± 3.25	83.17 ± 5.52	89.11 ± 1.12	96.84 ± 4.89
5	37.71 ± 6.67	69.79 ± 2.13	86.75 ± 2.55	79.28 ± 1.34	95.65 ± 1.95
6	88.79 ± 1.71	92.78 ± 0.78	88.08 ± 3.06	93.82 ± 0.32	98.95 ± 0.96
7	38.54 ± 11.4	20.85 ± 7.59	72.89 ± 29.7	40.69 ± 2.13	92.48 ± 24.0
8	89.96 ± 2.15	88.84 ± 3.15	84.25 ± 2.40	99.33 ± 0.31	86.11 ± 3.09
9	57.81 ± 19.04	0.00 ± 0.00	76.0 ± 49.0	28.00 ± 2.05	99.72 ± 8.38
10	67.46 ± 1.91	67.53 ± 1.24	90.47 ± 6.87	87.28 ± 0.89	95.30 ± 2.86
11	82.56 ± 3.49	73.88 ± 4.33	94.57 ± 5.03	95.21 ± 0.23	99.84 ± 3.29
12	54.14 ± 5.56	48.46 ± 3.85	77.47 ± 5.16	92.77 ± 3.37	92.56 ± 3.44
13	34.47 ± 7.17	83.02 ± 1.22	97.26 ± 5.29	49.93 ± 3.56	97.89 ± 0.87
14	84.32 ± 8.95	93.94 ± 2.59	89.16 ± 2.22	95.89 ± 1.37	95.89 ± 2.57
15	48.75 ± 5.12	47.64 ± 4.56	98.00 ± 7.49	96.77 ± 2.98	93.74 ± 2.65


(a)

(b)

(c)

(d)

(e)

(f)

Figure 6: HSI classified image for Houston U dataset: (a) original ground truth image, (b) CSDCNN-ALO, (c) CSDCNN-PSO, (d) CSDCNN-WOA, (e) CSDCNN-GWO, and (f) CSDCNN-AO.

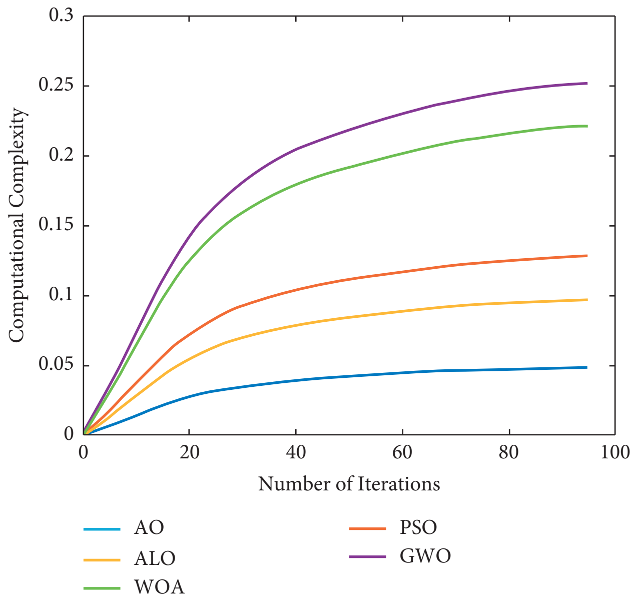


Figure 7: Comparative analysis of computational complexity.

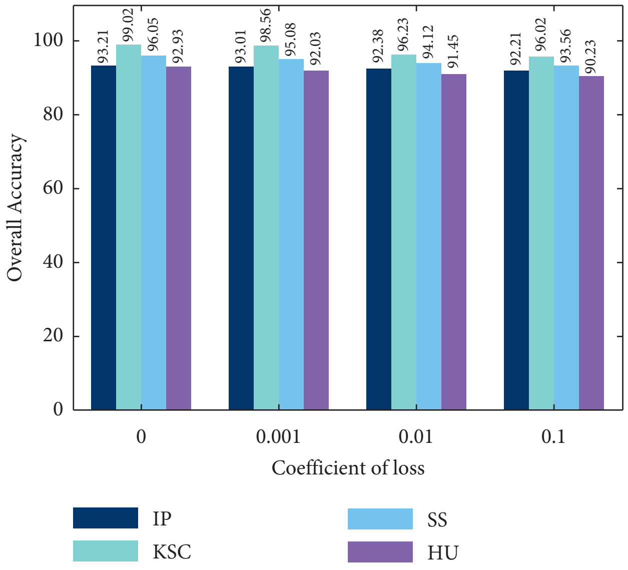


Figure 8: A comparative analysis of overall accuracy.

In the above Figure 6, (a) represents the original image then we evaluated the outcome of the proposed algorithm with other algorithms like CSDCNN-ALO, CSDCNN-PSO, CSDCNN-WOA, and CSDCNN-GWO. From these methods, our proposed work yields the maximum performance since the obtained proposed image is nearly equivalent to the original ground truth image.

The input images were obtained from the four datasets. Results are obtained after feature extraction, feature reduction, classification, and loss function optimization. The four different datasets taken for testing purposes are the HU dataset, IP, KSC, and SS dataset. These four datasets have shown promising results in this classification. The results (i.e., computational complexity, overall accuracy and loss functions) that are obtained by these datasets are given in the following figure.

The computational complexity attained for various iterations are shown in Figure 7. The usage of various optimizations along with the synergic deep CNN has improved the performance of the proposed algorithm. The computational complexity attained by Aquila optimization is much better as it has identified the optimal solution in lesser number of iterations, due to this the computational complexity has to be increased while increasing the iterations. Not like other meta-heuristic algorithms, this optimization algorithm has provided satisfactory results on weight parameter selection compared to ALO, WOA, PSO, and GWO. Therefore, in this proposed process, the Aquila optimization is encouraged.

The overall accuracy comparison for the above-mentioned data sets is shown in Figure 8. Among all the

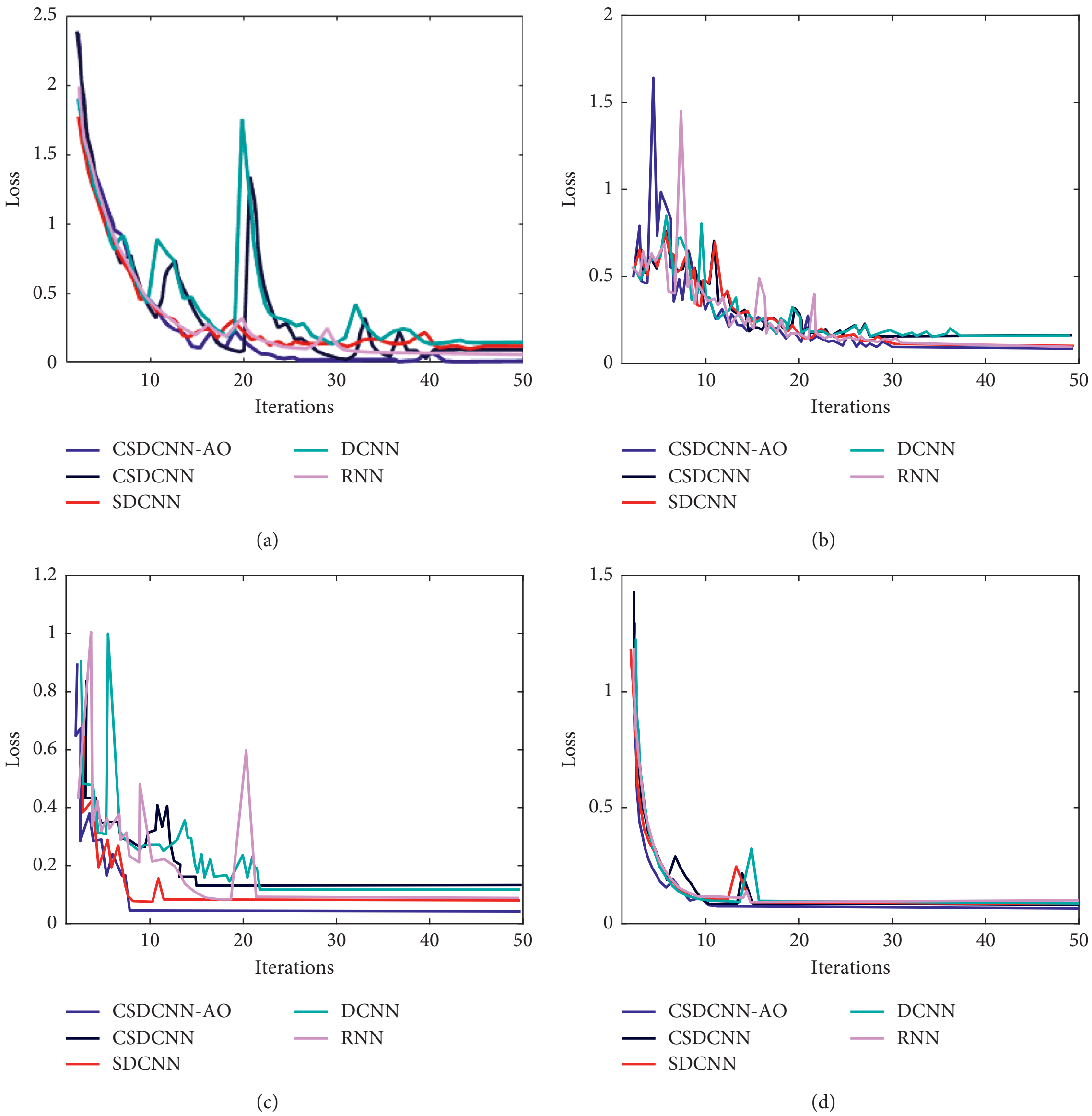


Figure 9: Loss comparison for (a) HU dataset, (b) SS dataset, (c) IP dataset, and (d) KSC dataset.

datasets, the dataset named KSC has shown a higher accuracy value than other algorithms. These four efficient datasets are taken for comparison. However, the overall accuracy is evaluated with the coefficient loss. The comparison analysis in terms of overall accuracy is affected while coefficient loss is increased. Our proposed work yields maximum accuracy of 99.02%, and this is lagged for the increasing coefficient losses.

The loss comparison for proposed and existing algorithms for the four datasets are shown in Figure 9. Among all the techniques, our proposed CSDCNN-AO has shown a lower loss value than other algorithms. The four efficient existing algorithms that are taken for comparison are CSDCNN, SDCNN, DCNN, and RNN. However, the loss shown in all these datasets are found to be much less than that in other existing algorithms. Especially for the KSC dataset, obtained losses are very low compared to another one. This is because the proposed technique has enhanced the effectiveness of the classification process.

5. Conclusion

Compressed spatial and spectral characteristics are employed as the key perception to develop a compressed synergic deep convolution neural network with Aquila optimization (CSDCNN-AO) for efficient HSI classification in this study. This combination will reduce the wavelet concept's learning difficulty and the Aquila optimization's loss function. This Aquila optimization approach may minimize the maximum number of data features without losing their characteristic state, while using less computing time and memory. Our proposed approach is superior to existing deep learning models due to higher learning ability of our synergic deep learning model based on compressed features. While comparing with the other techniques, our proposed approach can reach the maximum level of classification. In addition, the experimental results showed that the loss function does not significantly impact classification accuracy. In addition, the outcome demonstrates that the CSDCNN-AO approach has the highest accuracy among all the four datasets. Furthermore, the performance of average accuracy, total accuracy, and Kappa coefficients is optimal when implemented on all datasets. However, the proposed technique lacks optimal performance with certain samples. In future research, this issue will be resolved using a new model.

Data Availability

The dataset can be obtained from the corresponding author based on the request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Large Groups Project under grant number RGP. 2/252/43.

References

[1] M. Fauvel, J. A. Benediktsson, J. Boardman et al., “Recent advances in techniques for hyperspectral image processing,” *Remote Sensing of Environment*, vol. 113, pp. S110–S120, 2007.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[3] S. Zhong, C.-I. Chang, and Y. Zhang, “Iterative support vector machine for hyperspectral image classification,” in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3309–3312, Athens, Greece, October 2018.

[4] S. Sun, P. Zhong, H. Xiao, F. Liu, and R. Wang, “An active learning method based on SVM classifier for hyperspectral images classification,” in *Proceedings of the 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–4, Tokyo, Japan, June 2015.

[5] B. Tu, J. Wang, X. Kang, G. Zhang, X. Ou, and L. Guo, “KNN-based representation of superpixels for hyperspectral image classification,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4032–4047, 2018.

[6] K. Huang, S. Li, X. Kang, and L. Fang, “Spectral-spatial hyperspectral image classification based on KNN,” *Sensing and Imaging*, vol. 17, no. 1, p. 1, 2016.

[7] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, pp. 809–823, 2012.

[8] T. N. Prabhakar, G. Xavier, P. Geetha, and K. P. Soman, “Spatial preprocessing based multinomial logistic regression for hyperspectral image classification,” *Procedia Computer Science*, vol. 46, pp. 1817–1826, 2015.

[9] L. Velásquez, J. P. Cruz-Tirado, R. Siche, and R. Quevedo, “An application based on the decision tree to classify the marbling of beef by hyperspectral imaging,” *Meat Science*, vol. 133, pp. 43–50, 2017.

[10] A. D. Amirruddin, F. M. Muharam, M. H. Ismail, M. F. Ismail, N. P. Tan, and D. S. Karam, “Hyperspectral remote sensing for assessment of chlorophyll sufficiency levels in mature oil palm (Elaeis guineensis) based on frond numbers: analysis of decision tree and random forest,” *Computers and Electronics in Agriculture*, vol. 169, Article ID 105221, 2020.

[11] E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, “Comparison of regularization methods for imagenet classification with deep convolutional neural networks,” *Aasri Procedia*, vol. 6, pp. 89–94, 2014.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, http://arXiv.org/abs/1409.1556.

[13] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, “Graph convolutional networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.

[14] Z. Pan, G. Healey, and B. Tromberg, “Comparison of spectral-only and spectral/spatial face recognition for personal identity verification,” *EURASIP Journal on Applied Signal Processing*, vol. 2009, Article ID 943602, 6 pages, 2009.

[15] B. Fang, Y. Li, H. Zhang, and J. C. Chan, “Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism,” *Remote Sensing*, vol. 11, no. 2, p. 159, 2019.

[16] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, “Hyperspectral image classification with deep learning models,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, 2018.

[17] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, “HybridSN: exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2020.

[18] Z. Zhong, J. Li, Z. Luo, and M. Chapman, “Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[19] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, “Deep pyramidal residual networks for spectral-spatial hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 740–754, 2019.

[20] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” in *Proceedings of the In2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4959–4962, IEEE, Milan, Italy, July 2015.

[21] X. Yang, X. Zhang, Y. Ye et al., “Synergistic 2D/3D convolutional neural network for hyperspectral image classification,” *Remote Sensing*, vol. 12, no. 12, p. 2033, 2020.

[22] C. Li, Z. Qiu, X. Cao, Z. Chen, H. Gao, and Z. Hua, “Hybrid dilated convolution with multi-scale residual fusion network for hyperspectral image classification,” *Micromachines*, vol. 12, no. 5, p. 545, 2021.

[23] H. Patel and K. P. Upla, “A shallow network for hyperspectral image classification using an autoencoder with convolutional neural network,” *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 695–714, 2022.

[24] Q. Wang, M. Chen, J. Zhang, S. Kang, and Y. Wang, “Improved active deep learning for semi-supervised classification of hyperspectral image,” *Remote Sensing*, vol. 14, no. 1, p. 171, 2021.

[25] C. Shi, D. Liao, T. Zhang, and L. Wang, “Hyperspectral image classification based on 3D coordination attention mechanism network,” *Remote Sensing*, vol. 14, no. 3, p. 608, 2022.

[26] J. Zhao, L. Hu, Y. Dong, L. Huang, S. Weng, and D. Zhang, “A combination method of stacked autoencoder and 3D deep residual network for hyperspectral image classification,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 102, Article ID 102459, 2021.

[27] J. Yin, C. Qi, Q. Chen, and J. Qu, “Spatial-spectral network for hyperspectral image classification: a 3-D CNN and Bi-LSTM framework,” *Remote Sensing*, vol. 13, no. 12, p. 2353, 2021.

[28] A. Paul, S. Bhoumik, and N. Chaki, “SSNET: an improved deep hybrid network for hyperspectral image classification,” *Neural Computing & Applications*, vol. 33, no. 5, pp. 1575–1585, 2021.

[29] T. Zhang, C. Shi, D. Liao, and L. Wang, “A spectral spatial attention fusion with deformable convolutional residual network for hyperspectral image classification,” *Remote Sensing*, vol. 13, no. 18, p. 3590, 2021.

[30] H. Yan, J. Wang, L. Tang et al., “A 3D cascaded spectral-spatial element attention network for hyperspectral image classification,” *Remote Sensing*, vol. 13, no. 13, p. 2451, 2021.

[31] C. S. S. Anupama, M. Sivaram, E. L. Lydia, D. Gupta, and K. Shankar, “Synergic deep learning model-based automated detection and classification of brain intracranial hemorrhage images in wearable networks,” *Personal and Ubiquitous Computing*, vol. 26, no. 1, pp. 1–10, 2020.

[32] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-qaness, and A. H. Gandomi, “Aquila optimizer: a novel meta-heuristic optimization algorithm,” *Computers & Industrial Engineering*, vol. 157, Article ID 107250, 2021.

[33] T. Alipour-Fard and H. Arefi, “Structure aware generative adversarial networks for hyperspectral image classification,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5424–5438, 2020.

[34] N. Laban, B. Abdellatif, H. M. Ebeid, H. A. Shedeed, and M. F. Tolba, “Reduced 3-d deep learning framework for hyperspectral image classification,” in *Proceedings of the InInternational Conference on Advanced Machine Learning Technologies and Applications*, pp. 13–22, Springer, Cham, Cairo, Egypt, March 2019.

[35] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, “Attention-based adaptive spectral-spatial kernel ResNet for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7831–7843, 2021.

[36] L. Abualigah, M. Shehab, M. Alshinwan, S. Mirjalili, and M. A. Elaziz, “Ant lion optimizer: a comprehensive survey of its variants and applications,” *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1397–1416, 2021.

[37] K. E. Parsopoulos and M. N. Vrahatis, “Particle swarm optimization method in multiobjective problems,” *InProceedings of the 2002 ACM symposium on Applied computing*, vol. 11, pp. 603–607, 2002.

[38] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[39] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[40] M. Alazab, K. Lakshmanna, G. Thippa Reddy, Q.-V. Pham, and P. K. Reddy 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 ID 100973, 2021.

[41] T. R. Gadekallu, M. Alazab, R. Kaluri, P. K. R. Maddikunta, S. Bhattacharya, and K. Lakshmanna, “Hand gesture classification using a novel CNN-crow search algorithm,” *Complex Intell. Syst.*vol. 7, pp. 1855–1868, 2021.

[42] S. Aiswarya, S. Sasikumar, and S. Ramesh, “IoT based big data analytics in healthcare: a survey,” in *Proceedings of the Fist International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020*, Chennai, India, May 2020.

[43] A. Priyanka, M. Parimala, K. Sudheer et al., “BIG data based on healthcare analysis using IOT devices,” *IOP Conference Series: Materials Science and Engineering*, vol. 263, no. No. 4, Article ID 042059, 2017.

[44] K. Lakshmanna and N. Khare, “Constraint-based measures for DNA sequence mining using group search optimization algorithm”,” *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 3, pp. 91–100, 2016.

[45] S. P. R.M., S. Bhattacharya, P. K. R. Maddikunta et al., “Load balancing of energy cloud using wind driven and firefly algorithms in internet of everything,” *Journal of Parallel and Distributed Computing*, vol. 142, pp. 16–26, 2020.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9107430, 17 pages
https://doi.org/10.1155/2022/9107430

Research Article

PCA-Based Incremental Extreme Learning Machine (PCA-IELM) for COVID-19 Patient Diagnosis Using Chest X-Ray Images

Vinod Kumar,[1] **Sougatamoy Biswas**,[1] **Dharmendra Singh Rajput**,[2] **Harshita Patel**,[2] **and Basant Tiwari**[3]

[1]*Koneru Lakshmaiah Education Foundation, Vaddeswaram, India*
[2]*Vellore Institute of Technology, Vellore, India*
[3]*Hawassa University, Awasa, Ethiopia*

Correspondence should be addressed to Basant Tiwari; basanttiw@hu.edu.et

Received 25 March 2022; Accepted 29 April 2022; Published 4 July 2022

Academic Editor: Muhammad Ahmad

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

Novel coronavirus 2019 has created a pandemic and was first reported in December 2019. It has had very adverse consequences on people's daily life, healthcare, and the world's economy as well. According to the World Health Organization's most recent statistics, COVID-19 has become a worldwide pandemic, and the number of infected persons and fatalities growing at an alarming rate. It is highly required to have an effective system to early detect the COVID-19 patients to curb the further spreading of the virus from the affected person. Therefore, to early identify positive cases in patients and to support radiologists in the automatic diagnosis of COVID-19 from X-ray images, a novel method PCA-IELM is proposed based on principal component analysis (PCA) and incremental extreme learning machine. The suggested method's key addition is that it considers the benefits of PCA and the incremental extreme learning machine. Further, our strategy PCA-IELM reduces the input dimension by extracting the most important information from an image. Consequently, the technique can effectively increase the COVID-19 patient prediction performance. In addition to these, PCA-IELM has a faster training speed than a multi-layer neural network. The proposed approach was tested on a COVID-19 patient's chest X-ray image dataset. The experimental results indicate that the proposed approach PCA-IELM outperforms PCA-SVM and PCA-ELM in terms of accuracy (98.11%), precision (96.11%), recall (97.50%), F1-score (98.50%), etc., and training speed.

1. Introduction

The World Health Organization (WHO) identified COVID-19 (virus known as SARS-CoV-2) as a worldwide pandemic in February 2020. This triggered never expected countermeasures, such as the closure of cities, districts, and foreign travel. Coronaviruses (CoV) are death-defying viruses that may cause severe acute respiratory syndrome (SARS-CoV). Various researchers and institutions have attempted an effective solution from different possible diminutions in encountering the COVID-19 pandemic. Multimedia dataset (audio, picture, video, etc.) is booming in a massive amount of text information as civilization enters the information era. Image classification has become more essential as the need for real-world vision systems grows [1] and has recently attained a lot of attention from many researchers. It has evolved into one of the most essential operations, serving as a requirement for all other image processing operations. Image classification using learning algorithms is a special open issue in image processing that has sparked a lot of interest due to its promising applications. In general, an image categorization system has two primary processes. The first stage is to create an effective image representation that has enough information about the image to allow for classification further. The second step is to use a good classifier to classify the new image. Thus, there are two major challenges to consider when improving picture classification performance: dimensionality reduction and classifier. Apart from computer vision and image operation, one of the most important stages in image classification is feature extraction which determines the invariant characteristic of images when using computer devices to assess and deal with image data.

In a practical scenario, feature extraction has been applied in many fields like historic structures, medical image processing, remote image sensing, etc. The image's essential lower-level qualities include color, texture, and shape. The color feature has globality, which may be retrieved using tools such as the color histogram, color set, and color moment. It might simply explain the proportions of different colors across the image. The useful characteristic is color for identifying photos that are difficult to distinguish automatically, and the spatial variation should be ignored. However, it is unable to explain the image's local distribution as well as the description of the distinct colors' spatial positions. Image classification with feature extraction using incremental extreme learning machines is proposed in this paper. Firstly, on the COVID-19 dataset of chest X-ray images, features were extracted from an image using PCA. Eventually, the SVM, ELM, and IELM are applied to image classification [2] once the dimension is reduced by PCA method. Different metrics were employed to achieve the robust evaluation: classification accuracy, recall, precision, F-score, true-negative rate (TNR), true-positive rate (TPR), AUC, G-mean, precision-recall curve, and receiver operating characteristics (ROC) curve.

The paper is arranged in the following sequence: several related approaches have been discussed in Section 2. The suggested technique is described and critiqued in Section 3. Section 4 contains a description of PCA and feature extraction techniques. Subsections 4.1–4.6 contain different algorithmic approaches that are compared with the proposed method. In Section 5, the proposed method and algorithm have been discussed. Section 6 describes the different evaluation criteria that are used. Section 7 discusses the experimental setup that has been used. Section 8 describes the dataset. Finally, Section 9 discusses the experimental results, and the research is concluded.

2. Related Works

The content of image features comprises color, texture, and other visual elements. The extracted content from visual features is the main component for analyzing the image. In this segment, some of the earlier work based on PCA and other feature extraction techniques along with different classification techniques has been discussed.

Sun et al. [3] suggested an image classification system based on multi-view depth characteristics and principal component analysis. In this method, depth features are extracted from the image, and from RGB depth, characters are independently extracted and PCA is applied to reduce dimension. The Scene15 dataset, Caltech256 dataset, and MIT Indoor datasets are used in the evaluation process. Eventually, the SVM [4] is used to classify images. The method's performance is demonstrated by the experimental results.

Mustaqeem and Saqib [5] suggested a hybrid method that is based on PCA and SVM. PROMISE (KC1: 2109 observations, CM1: 344 observations) data from NASA's directory have been used for the experiment. The dataset was divided into two parts: training (KC1: 1476 observations, CM1: 240 observations) and testing (KC1: 633 observations, CM1: 104 observations). Principal components of the features are extracted by PCA, and it helps in dimensionality reduction and minimizing time complexity.

In addition to this, SVM is used for further classification, and for hyperparameter tuning, GridSearchCV is used. From this, precision, recall, F-measure, and accuracy for KC1 dataset analysis are 86.8%, 99.6%, 92.8%, and 86.6%, respectively, and for CM1 dataset analysis, precision, recall, F-measure, and accuracy are 96.1%, 99.0%, 97.5%, and 95.2%, respectively. Similarly, Castaño et al. [6] provide a deterministic approach for starting ELM training based on hidden node parameters with activation function. The hidden node parameters are calculated with the help of Moore–Penrose generalized inverse, whereas the output node parameters are recovered through principal component analysis. Experimental validation with fifteen well-known datasets was used to validate the algorithm. The Bonferroni–Dunn, Nemenyi, and Friedman tests were used to compare the results obtained. In comparison with later ELM advancements, this technique significantly reduces computing costs and outperforms them.

Mateen et al. [7] suggested VGG-19 DNN-based DR model with better performance than AlexNet and the spatial invariant feature transform (SIFT) in terms of classification accuracy and processing time. For FC7-SVD, FC7-PCA, FC8-SVD, and FC8-PCA, respectively, classification accuracies are 98.34%, 92.2%, 98.13%, and 97.96% by using SVD and PCA feature selection with fully connected layers.

Zhao et al. [8] suggested extreme learning machines with no iteration along with supervised samples are used for model building as a class incremental extreme learning machine. The algorithm is shown to be stable and has almost equivalent accuracy of batch learning. Similarly, Huang and Chen [9] proposed an algorithm that analytically calculates hidden nodes' output after randomly producing and adding computational nodes to the hidden layer as a convex incremental extreme learning machine. Using a convex optimization, the existing hidden node output is calculated again. This can converge faster while maintaining efficiency and simplicity.

Zhu et al. [10] proposed a principal component analysis (PCA)-based categorization system with kernel-based extreme learning machine (KELM). Based on the resultant output, this model achieves better accuracy than SVM and other traditional classification methods. For the classification of HSIs, Kang et al. [11] developed the PCA-EPF extraction approach. In this research work, they have proposed the combination of PCA and standard edge preserving filtering (EPF)-based feature extraction. The proposed method achieves better classification accuracy with limited training samples. Similarly, Perales-González et al. [12] introduced a new ELM architecture based on the negative correlation learning framework dubbed negative correlation hidden layer ELM (NCHL-ELM). This model shows better accuracy when compared with other classifications by integrating a parameter into each node in the original ELM hidden layer.

Based on fractal dimension technology, Li et al. [13] suggested an enhanced ELM algorithm (F-ELM). By reducing the dimension of the hidden layer, the model improves in training speed. From the experimental results, it can be concluded that as compared to the standard ELM technique, the suggested algorithm significantly reduces computing time while also improving inversion accuracy and algorithm stability.

Because of the complexity of the data models, deep learning is incredibly pricey to train. Furthermore, deep learning necessitates the use of high-priced GPUs and hundreds of computer machines. There is no simple rule that can help you choose the best deep learning tools since it necessitates the understanding of topology, training technique, and other characteristics, whereas the simple ELM is a one-shot computation with a rapid learning pace. But the biggest advantage in IELM is the ability to randomly increase hidden nodes incrementally and analytically fix the output weights. The output error of the IELM rapidly diminishes as the number of hidden neurons increases.

In our method, SVM, ELM, and IELM based on the PCA technique are employed for image classification [14] for COVID-19 patient detection using the COVID-19 chest X-ray dataset. A summary of the most recent and related research works is described in Table 1 [3, 5–13].

3. Proposed Methodology

The back propagation (BP) approach is commonly used to train multi-layer perceptron (MLP). Various algorithms can be used to train this typical architecture. Gradients and heuristics are two types of algorithms that are commonly used. These algorithms have a few things in common: they have a hard time dealing with enormous amounts of data, and they have a slow convergence rate in these situations. Huang et al. (Huang et al.) [15] introduced the extreme learning machine as a solution to this problem.

The typical computing time required to train an SLFN using gradient-based techniques is reduced by this algorithm. The ELM, on the other hand, has several flaws. The randomly generated input weights and bias for ELM [16] result in some network instability. In case if there are outliners in the training data, then the hidden layer's output matrix will have ill-conditioned problems and it results in low generalization performance and lower forecasting accuracy. There are two types of ELM called fixed ELM and IELM [17]. In comparison with the ELM, the output error of the IELM rapidly diminishes and it tends toward zero with the growth in number of hidden neurons (Huang et al.) [15]. In online continuous learning regression and classification problem, this approach is very prominent (Xu and Wang; Zhang et al.) [18, 19].

A trained classifier can be obtained after training the classifiers with a sufficient amount of image data and then fed into the trained classifier for observation and analysis.

4. Feature Extraction

A single feature cannot describe the image feature and quality properly. The image classification will not yield acceptable results unless distinguishing features are described. Three images corresponding to three viewpoints are placed on each RGB color image. Our method uses PCA to extract the image's important information and minimize the input dimension [20–23].

4.1. Classification of Images and PCA Feature Extraction.

Extracting useful features from an image is a prominent task in image classification, and principal component analysis (PCA) is used for this purpose. PCA uses orthogonal transformation and converts variables to fewer independent components than the original variables. The output data with this approach will not lose important data features, and PCA loadings can be used for the identification of important data. A multivariate statistical analysis approach is used by PCA, which can perform linear transformation of numerous variables to pick a few key variables. PCA transforms data using eigenvectors from N-dimension to M-dimension where $M < N$. The new features are a linear mixture of the old ones, allowing them to capture the data's intrinsic unpredictability with little information loss. Figure 1 reveals the steps of the proposed model.

Suppose that the research object has p indexes, these indexes are regarded as p random variables and represented as X_1, X_2, , X_p. With this, new indexes are created by combining p random variable $F_1, F_2, ..., F_p$, which can mirror the data from the original indexes [24]. The independent replacement indexes reflect the original indexes' essential information.

$$\begin{aligned} F_1 &= a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p, \\ F_2 &= a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p, \\ &\qquad\qquad\qquad\qquad \cdots\cdots\cdots, \\ F_p &= a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p. \end{aligned} \tag{1}$$

The following are the PCA stages in detail:

(1) *Data standardization*: The following calculation formula is used to standardize the matrix X:

Table 1: Similar work summarization.

SN	References	Applied method	Problem approached	Resulted outcome	Impediments
1	Sun et al. [3]	PCA of multi-view deep representation	Image classification	Comparison result from different databases	Limited classifiers are compared
2	Mustaqeem and Saqib [5]	Principal component-based support vector machine	Software defect detection	Better accuracy than other methods	No probabilistic explanation for SVM classification
3	Castaño et al. [6]	Pruned ELM approach based on principal component analysis	Classification	ELM model based on PCA	Limited classifiers are compared
4	Mateen et al. [7]	VGG-19 architecture with SVD and PCA	Fundus image classification	Better accuracy than other methods	Limited to nonimbalance data
5	Zhao et al. [8]	IELM	Activity recognition	Stable and similar accuracy to the batch learning method	Limited to batch learning
6	Huang et al. [9]	Convex incremental extreme learning machine	Convergence rate of IELM	Faster convergence rate	Limited classifiers are compared
7	Zhu et al. [10]	PCA and kernel-based ELM	Side-scan sonar image classification	Better classification accuracy with stable model	Classify underwater targets only
8	Kang et al. [11]	PCA-based edge-preserving features (EPF)	Hyperspectral image classification	Better accuracy than SVM	Parameters of EPFs are given manually
9	Perales-González et al. [12]	Negative correlation hidden layer for the ELM	Regression and classification	Better accuracy	Variety in the transformed feature space
10	Li et al. [13]	Improved ELM	Transient electromagnetic nonlinear inversion	Improves the inversion accuracy and stability	Less implementation in other industrial domains

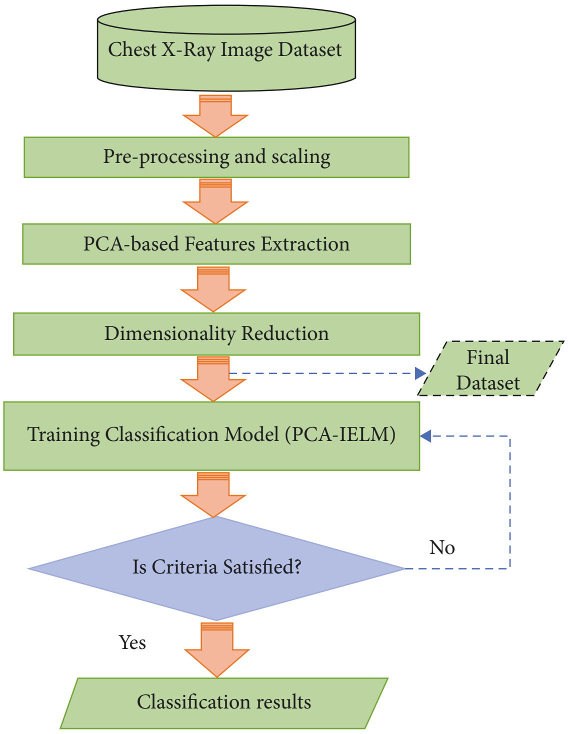

Chest X-Ray Image Dataset
Pre-processing and scaling
PCA-based Features Extraction
Dimensionality Reduction
Final Dataset
Training Classification Model (PCA-IELM)
Is Criteria Satisfied?
No
Yes
Classification results

Figure 1: Flowchart of the proposed model (PCA-IELM).

$$y_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{\mathrm{Var}(x_j)}}, \tag{2}$$

where $X = \{x_{ij}\}$, $Y = \{y_{ij}\}$, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$,

$$\begin{gathered}\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_{ij},\\ \operatorname{var}(xj) = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{ij} - \overline{x}_j\right)^2.\end{gathered} \tag{3}$$

(2) The following formula is used to solve the correlation coefficient matrix R:

$$R = \frac{Y^T Y}{n-1}. \tag{4}$$

(3) The following formula is used to calculate the eigenvalue and eigenvector of the coefficient matrix:

$$\left|R - \lambda I_p\right| = 0. \tag{5}$$

The calculated eigenvector is $a_i = (a_{i1}, a_{i2}, \ldots, a_{ip})$, where $i = 1, 2, 3, 4, \ldots\ldots\ldots\ldots, p$, and the eigenvalue is $\lambda\ i$ $(i = 1, 2, \ldots, p)$. To get a collection of main components Fi, the eigenvalues are sorted in descending order:

$$F_i = a_{i1}Y_1 + a_{i2}Y_2 + \cdots + a_{ip}Y_p. \tag{6}$$

(4) The following are the main factors to consider kth primary component contribution rate and expressed as

$$\lambda_k\left(\sum_{j=1}^{p}\lambda_j\right)^{-1}. \tag{7}$$

The rate of the first k primary components' cumulative contribution is expressed as

$$\sum_{j=1}^{k} \lambda_j \left(\sum_{j=1}^{P} \lambda_j \right)^{-1}. \tag{8}$$

The first principal component, F_1, is the one with the highest variance out of all the combinations of Y_1, Y_2, ..., Y_p; the second principal component F_2 is one with the highest variance among all the combinations of Y_1, Y_2, ..., Y_p, and they have no relation with F_1.

4.2. SVM.

4.2. SVM. Several algorithms have been implemented and suggested in machine learning to solve the classification problem. Among the different classification problems, support vector machine (SVM) is one of the supervised algorithms in machine learning with [5, 25] the advantages as follows:

(i) It employs L2 regularization to overcome overfitting problems.

(ii) Even with minimal data, provide suitable findings.

(iii) Different kernel functions to match the features' complicated functions and interactions.

(iv) Manages the data nonlinearity.

(v) The model is stable thanks to the hyper-plane splitting rule.

(vi) Analyzes the data with a high degree of dimensionality.

Instead of focusing on decreasing prediction error, SVM focuses more on optimizing classification decision boundaries, which is why the hyper-plane is used to separate classes. If the data dimension is n and the hyper-plane is a $(n-1)$ vector function, then it can be represented mathematically as follows:

$$y = w_0 x_0 + w_1 x_1 + \cdots + w_{n-1} x_{n-1} + b. \tag{9}$$

It also signifies, in a broader sense,

$$y = w^T x + b, \tag{10}$$

where x denotes the input feature vector, w is the weight vector, and b is the bias. By adjusting w and b, several hyper-planes can be created, but the hyper-plane with the best margin will be chosen. The largest feasible perpendicular distance between each class and the hyper-plane is defined as ideal margin. The cost function or objective function is minimized to get the best margin. The cost function may be written as follows:

$$J(w) = \frac{1}{2} \|w\|^2 + \frac{1}{n} \sum_{i=0}^{n} \max\left(0, \left(1 - y_i * \left(w^T x + b\right)\right), \tag{11}$$

Even if the predictions are right and the data are correctly categorized by hypothesis, SMV utilized to penalize any y_i that are close to the borders $(0 < y_i < 1)$. The main goal is to figure out optimal w value to minimize $J(w)$, so differentiating Eq. 11 concerning w, we get the gradient of a cost function as follows:

$$\nabla_w J(w) = \frac{\partial J(w)}{\partial w}, \\ = \frac{1}{n} \sum_{i=0}^{n} \begin{cases} w, & \text{if } \max\left(0, \left(1 - y_i * \left(w^T x + b\right)\right), \\ w - y_i x_i, & \text{otherwise.} \end{cases} \tag{12}$$

As far as we have calculated $\nabla_w J(w)$, weights of w can be updated as

$$W_{\text{new}} = W_{\text{old}} - \alpha [J(w)]. \tag{13}$$

We go through the procedure again and again until smallest $J(w)$ discovered. Because data are rarely linearly separable, we must sketch a decision boundary between the classes rather than using a hyper-plane to separate them. We will need to convert (13) into a decision boundary to deal with the dataset's nonlinearity:

$$y = w \cdot \phi(x) + b. \tag{14}$$

$\phi(x)$ is the kernel function in (14). There are various types of kernel functions that may be used to create SVM, such as linear, polynomial, and exponential, but we will use the radial basis function in this model (RBF). Distance parameter that is used is Euclidean distance, and the smoothness of the borders is defined by the parameter σ.

$$\phi(x) = \exp\left(-\frac{x - \overline{x}^2}{2\sigma^2} \right), \tag{15}$$

where $x - \overline{x}^2$ is the square of Euclidean distance between any single observation x and mean of the training sample x.

4.3. PCA-SVM.

4.3. PCA-SVM. The motive of the support vector machine (SVM) [3] is to find the best possible hyper-plane that will separate two planes on the training set. The coefficient of the hyper-plane is w that we have to project. It uses structural risk minimization theory to build the best hyper-plane segmentation in the feature space and a learning editor to achieve global optimization.

Assume the training data, (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n) \in R^n$, $y \in \{-1, 1\}$.

This could be projected into a hyper-plane:

$$(\omega \cdot x) + b = 0, \omega \in R^n b \in R. \tag{16}$$

For the normalization,

$$y_i (\omega \cdot x_i) + b \geq 1, \quad i = 1, 2, \ldots, l. \tag{17}$$

The classification of the interval is equal to $2/\omega$, when the maximum interval is equal to the minimum ω^2.

Before classifying the data through SVM, the necessary features from the image data need to be extracted. The high-dimensional data can be converted to the low-dimensional data with this approach. For this, the PCA method as a feature extraction through convergence matrix and eigenvalue proportion calculation is used. PCA-based SVM is the method that is used for classification and regression. After that, SVM is used to classify low-dimensional data. Figure 2

```
    Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
    Output: SVM classification model.
(1) Procedure SVM.
(2) Initialize weights $W$ and bias $b$ with any arbitrary number.
(3) Feature optimization.
(4) TrainDataSet, TestingDataSets = Data.Split(Ratio).
(5) Define $y$.
(6) Define $f(y)$.
(7) Define $J(W)$.
(8) Calculate $\nabla_w J(w)$.
(9) Repeat for minimum $J(W)$:
        Update $W$.
        Call steps 6–8.
        End.
(10) Calculate accuracy.
(11) Return accuracy.
(12) End procedure.
    Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
    Output: SVM classification model.
```

Algorithm 1: SVM algorithm.

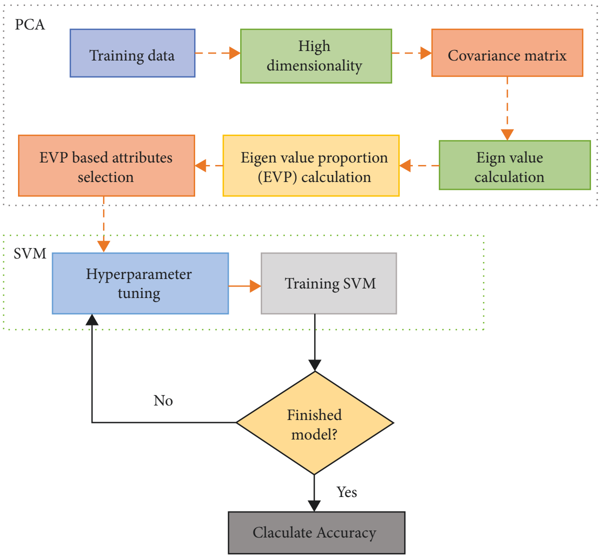


Figure 2: Flowchart of PCA-SVM.

depicts the working flow of PCA-SVM. Once the parameter optimization is done, the model is ready to predict categorization.

4.4. Extreme Learning Machine (ELM).

An extreme learning machine is a single hidden layer feedforward network that can be used for both classification and regression. In ELM [26], weights between the input layer, hidden layer, and biases are randomly generated. The output weights are calculated using the generalized Moore–Penrose pseudo-inverse. ELM performs faster than other feedforward networks [27] and outperforms other iterative methods. Figure 3 shows the basic network architecture of ELM.

Suppose [$\mathbf{x_i}$, $\mathbf{t_i}$] denotes N training samples, wherein training instances $i \in 1, 2, 3, \ldots\ldots\ldots\ldots, N$ and $\mathbf{x_i} = [x_{i1}, x_{i2},\ldots, x_{im}]T \in R^m$ denotes i^{th} training instance and its desired output $\mathbf{t_i} = [t_{i1}, t_{i2},\ldots, t_{iC}\ ]^T \in R^C$.

Let the number of input features and number of neurons be equal and represented by m; similarly, let L be the number of hidden neurons. The number of output neurons and number of classes are equal and denoted by c. Figure 4 [24] shows the flowchart of the principal component analysis [28]. The input weight matrix is represented by $\mathbf{U} = [\mathbf{u_1}, \mathbf{u_2},\ldots, \mathbf{u_j},\ldots\mathbf{u_L}]^T \in R_{L\times m}$, and the hidden neuron bias is represented by $\mathbf{b} = [b_1, b_2,\ldots, b_j,\ldots b_L]^T \in RL$. $\mathbf{u_j} = [u_{j1}, u_{j2},\ldots u_{jm}]$ are the connecting weights between the j^{th} hidden neuron with the input neurons. Bias of the j^{th} hidden neuron is bj, and jth hidden layer output for i^{th} instance is represented by

$$h_{ij} = g\left(\mathbf{u_j}\mathbf{x_i} + b_j\right). \tag{18}$$

Here, activation function is represented by g. For all the training instances hidden layer output is represented by $\mathbf{H}$ and can be represented by

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{1L} \\ \ldots & \ldots & \ldots \\ h_{j1} & h_{j2} & h_{jL} \\ \ldots & \ldots & \ldots \\ h_{N1} & h_{N2} & h_{NL} \end{pmatrix}. \tag{19}$$

Between the hidden layer and the output layer, the output weight $\boldsymbol{\beta}$ can be computed using Eq. (20). Linear activation function is used by the output layer in this computation.

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}. \tag{20}$$

Here,

```
     Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^{m}$, $\in R^{C}$.
     Output: PCA-SVM model for classification.
 (1) Procedure PCA-SVM
 (2) Identify the relationship among features through a covariance matrix.
 (3) Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
 (4) Transform our data using eigenvectors into principal components.
 (5) Quantify the importance of these relationships using eigenvalues and keep the important principal components.
 (6) Data extracted from PCA and will be given as input.
 (7) Initialize weights $W$ and bias $b$ with any arbitrary number.
 (8) Feature optimization.
 (9) TrainDataSet, TestingDataSets = Data.Split(Ratio).
(10) Define $y$.
(11) Define $f(y)$.
(12) Define J($W$).
(13) Calculate $\nabla_{w} J(w)$.
(14) Repeat for minimum J($W$):
         Update $W$.
         Call steps 10–13.
     End.
(15) Calculate accuracy.
(16) Return accuracy.
(17) End procedure.
```

Algorithm 2: PCA-SVM algorithm.

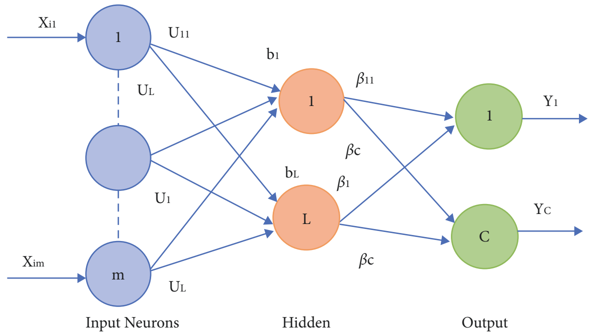


Figure 3: Network architecture of ELM.

$$\beta = \left[\beta_1^T, \ldots, \beta_j^T, \ldots, \beta_L^T\right]_{L\times C}^T. \tag{21}$$

The vector $\boldsymbol{\beta_j} = [\beta_{j1}, \ldots, \beta_{jk}, \ldots, \beta_{jC}]^T$, where $j = (1, 2, 3, \ldots \ldots \ldots, L)$ represents the connecting weights between the j^{th} hidden neuron and the k^{th} output neuron. The predicted outcome of all the output neurons for all training instances is represented as

$$\mathbf{f}(\mathbf{x}) = \mathbf{H}\beta. \tag{22}$$

Here, the output function is $\mathbf{f}(\mathbf{x}) = [f_k(\mathbf{x}), \ldots, f_C(\mathbf{x})]$. From Eq. 23, label for class $\mathbf{x}$ can be predicted.

$$\text{label}(\mathbf{x}) = \arg\max f_k(\mathbf{x}), \quad k = 1, \ldots, C. \tag{23}$$

4.5. PCA-ELM : Classification Method Based on PCA-ELM. In the PCA technique [6], variables are first scaled. The different steps of PCA that has been applied in PCA-ELM are

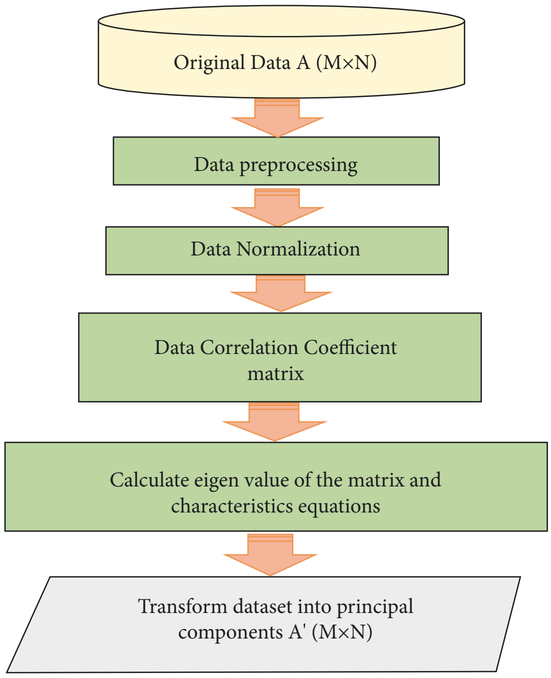

Original Data A (M×N)
Data preprocessing
Data Normalization
Data Correlation Coefficient matrix
Calculate eigen value of the matrix and characteristics equations
Transform dataset into principal components A' (M×N)

Figure 4: Flowchart of the principal component analysis [24].

(1) Scaling of trained data.

(2) Covariance matrix evaluation.

(3) Eigenvalues for the covariance matrix along with eigenvectors are defined.

(4) Evaluating the principal components.

The output from PCA is given as an input to ELM [29]. The process of PCA-ELM [30] is shown in Figure 5.

```
    Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\in R^C$.
    Output: ELM classification model.
(1) ELM procedure.
(2) Hidden biases $\mathbf{b}$ and input weights $\mathbf{U}$ randomly selected.
(3) From (19), $\mathbf{H}$ as hidden layer output is defined.
(4) From (20), $\boldsymbol{\beta}$ is defined as the weights between the hidden layer and the output layer.
(5) Return $\boldsymbol{\beta}$.
```

Algorithm 3: ELM algorithm.

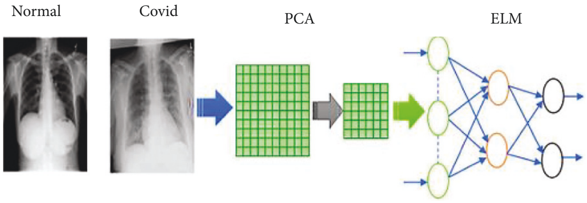


Figure 5: Process of PCA-ELM.

```
     Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\in R^C$.
     Output: PCA-ELM model for classification.
 (1) Procedure PCA-ELM.
 (2) Identify the relationship among features through a covariance matrix.
 (3) Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
 (4) Transform our data using eigenvectors into principal components.
 (5) Quantify the importance of these relationships using eigenvalues and keep the important principal components.
 (6) Extracted principal components given as the input data.
 (7) Hidden biases $\mathbf{b}$ and input weights $\mathbf{U}$ randomly selected.
 (8) From (19), hidden layer output $\mathbf{H}$ can be computed.
 (9) From (20), weights between the hidden and the output layer computed as $\boldsymbol{\beta}$.
(10) Return $\boldsymbol{\beta}$.
(11) End procedure.
```

Algorithm 4: PCA-ELM algorithm.

4.6. ELM. Compared to the other neural networks, the ELM learns faster as there is no need to adjust hidden nodes and provides better generalization capability. But there are various flaws with the ELM. Randomly generated bias and input weights in ELM network [31] are results in some network instability. Training data outliers from the hidden layer's output matrix result in poor network generalization performance. In comparison to the ELM, the output error of the IELM rapidly diminishes and resolves the issue of very small weights of output and validity of hidden layer neurons. In online continuous learning, it is appropriate for regression and classification tasks.

The IELM [32] network model structure is shown in Figure 6. Suppose the size of input, hidden nodes, and outputs are m, l, and n, respectively, and ω_i is the input weight matrix with $l \times m$ dimension of the current hidden layer neuron and uniformly distributed between random numbers $[-1, 1]$. The bias of the ith hidden node b_i is a random number between $[-1, 1]$ uniformly distributed, the activation function for the hidden layer neuron is sigmoid function given by (24), and output weight matrix β is with $l \times n$ dimension.

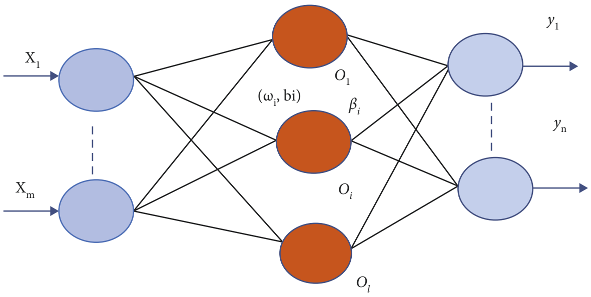


Figure 6: The structure of IELM.

The hidden node activation function (sigmoid) is given by

$$g(x) = \frac{1}{1 + e^{-x}}, \tag{24}$$

where x is the input matrix.

A matrix X is of $m \times N$ dimension, and it represents N dataset input. Y is a $n \times N$ matrix that represents the output where N datasets for a training set $\{(X, Y)\}$. Training steps of IELM algorithm are described as follows:

Step 1. In the initialization phase, suppose $l = 0$ and L is the maximum number of the hidden nodes. Output Y is defined in terms of the initial value of the residuals E (difference between target and actual error) is set to be the and ε is the expected training accuracy.

Step 2. Training phase, while $l < L$ and $E > \varepsilon$

(1) Hidden nodes l will be increased by 1, i.e.,

$$l = l + 1. \tag{25}$$

(2) Hidden layer neuron O_l is evaluated randomly from input weights ω_l and bias b_l.

(3) Output of the activation function $g\ (x')$ is calculated for the node O_l (b_l needs to be extended into a $l \times N$ vector b_l).

$$x' = \omega_l x + b_l. \tag{26}$$

(4) Hidden layer neuron output vector $\overline{H}$ can be calculated from

$$\overline{H} = g(x'). \tag{27}$$

(5) Output weight for O_l can be evaluated from

$$\overline{\beta} = \frac{E \cdot (\overline{H})^T}{\overline{H} \cdot (\overline{H})^T}. \tag{28}$$

(6) After increasing the new hidden node, residual error is calculated:

$$E' = E - \overline{\beta}\overline{H}. \tag{29}$$

The network error rate can be reduced by the output weight O_l. All these steps will iteratively work till the residual error becomes smaller than ε. The training process restarts through the determination of the random input weight ω_l and the bias b_l. Whether the trained network has fulfilled the desirable result or not can be determined from $\{(X', Y')\}$ set.

5. Proposed PCA-Based Incremental ELM (PCA-IELM)

An orthogonal transformation is used to extract meaningful characteristics from data in PCA [33]. PCA may also be used to minimize the dimensions of a large data collection. Principal components from COVID-19 X-ray images are extracted using PCA and given as input to IELM which gradually adds concealed nodes produced at random. A conventional SLFNs function with n hidden nodes can be expressed as

$$f_n(x) = \sum_{i=1}^{n} \beta_i g_i(x), \quad x \in R^{\mathrm{d}}, \beta_i \in R. \tag{30}$$

where $g_i(x) = g(a_i, b_i, x)$ denotes the output of the ith hidden node:$g_i(x) = g(a_i.x + b_i)$ (for additive nodes) or $g_i(x) = g(b_i x - a_i)$.

The ith hidden layer and the output node are linked with output weights β_i. Hidden nodes are randomly added to the existing networks in IELM. The randomly generated hidden node parameters a_i and b_i and fixed output weight are β_i.

Suppose the residual error function for the current network f_n is defined as $e_n \cong f - f_n$. where n is the number of hidden nodes and $f \in L^2(x)$ is the target function. IELM is mathematically represented as

$$f_n(x) = f_{n-1}(x) + \beta_n g_n(x). \tag{31}$$

6. Evaluation Criteria for Effective Measure of Model

For evaluation of the different models, generally, the confusion matrix is prepared. Table 2 defines a simple representation of the confusion matrix [34, 35], and it can classify between predicted and actual values. From the confusion matrix, we can derive different performance metrics, e.g., accuracy, precision, recall, sensitivity, and F-score. To assess the model, nine different metrics are calculated by formula as given in Table 3 [36].

7. Experimental Setup

The whole experiment was performed on a system having a configuration of 10th Generation Intel (R) Core (TM) i7-10750H CPU @ 2.60 GHz processor, 8 GB RAM, and NVIDIA GTX graphics 1650TI. The code is written in *Python* 3.10.0 and uses Jupyter Notebook as a debugger, which can be installed from the link: https://jupyter.org/install.

8. Dataset Description

The COVID-19 chest X-ray images [37] dataset encompasses a total of 13808 images in which 3616 COVID-19 positive cases (26.2%) along with 10,192 (73.8%) normal cases are downloaded from Kaggle. COVID-19 and normal patient chest X-ray images are kept in separate files. Dataset was divided into training and testing images which had been done randomly with a condition that testing images will not be repeated in training images. During the experiment, 80% of the total images were used for training and 20% for testing. All images have the same dimension (299×299) pixels in the PNG file format. Figure 7 demonstrates the X-ray images of normal and COVID-19 cases.

The histogram of an image gives a global description of the image's appearance. It represents the relative frequency of occurrences of various intensity values in an image. In the histogram of the COVID-19 image, the intensity value is highest between bins 14–15, whereas in the normal image the histogram has the highest intensity value at bins 16–17.

```
Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
Output: IELM model for classification.
Procedure IELM.
Initialize: hidden nodes $l = 0$ and maximum L.
For the newly increased hidden layer neuron $O_l$, input weights $\omega_l$ and bias $b_l$ randomly evaluated.
Output of the activation function $g\ (x\prime)$ calculated for the node $O_l$.
Hidden layer output $\overline{H}$ calculated from Eq. (27).
Weights between the hidden and the output layer $\overline{\beta}$ can be evaluated from Eq. (28).
Return $\overline{\beta}$.
End procedure.
```

Algorithm 5: IELM algorithm.

```
Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\in R^C$.
Output: PCA_IELM model for classification.
Procedure PCA_IELM.
Identify the relationship among features through a covariance matrix.
Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
Transform our data using eigenvectors into principal components.
Quantify the importance of these relationships using eigenvalues and keep the important principal components.
Input for the model is given in terms of extracted principal components.
Initialize: hidden nodes $l = 0$ and maximum L.
For the newly increased hidden layer neuron $O_l$, input weights $\omega_l$ and bias $b_l$ randomly evaluated.
g $(x')$ calculated as the output function for the node $O_l$.
Hidden layer output $\overline{H}$ calculated from Eq. (27).
Weights between the hidden and the output layer $\overline{\beta}$ can be evaluated from Eq. (28).
Return $\overline{\beta}$.
End procedure.
```

Algorithm 6: PCA-IELM algorithm.

Table 2: Confusion matrix.

	Predicted		Total
Actual	TP (true positive)	FP (false positive)	TP + FP
	FN (false negative)	TN (true negative)	FN + TN
Total	TP + FN	TN + FP	ALL

Table 3: Performance evaluation measures [36].

SL	Measures	Formula
1.	Accuracy	TP + TN/TP + FP + TN + FN
2.	Specificity (TN_{rate})	TN/TN + FP
3.	FN_{rate}	FN/TP + FN
4.	Sensitivity (TP_{rate})/recall	TP/TP + FN
5.	FP_{rate}	FP/TN + FP
6.	Precision	TP/TP + FP
7.	G-mean	$\sqrt{\text{TP}_{\text{rate}} \times \text{TN}_{\text{rate}}}$
8.	AUC	$1 + \text{TP}_{\text{rate}} - \text{FP}_{\text{rate}}/2$
9.	F_1-score	$2 * \text{TP}/2 * \text{TP} + \text{FP} + \text{FN}$

This difference in the color intensity value assists in making the distinction between COVID-19 and normal images. Figure 8 demonstrates the histogram plot of normal and COVID-19 images. Figure 9 shows the training images for X-ray images of COVID-19 and normal.

Because PCA uses orthogonal transformation to convert all features into a few independent features, all features are considered during the feature selection process. The data to be processed are reduced to a set of features called a "reduced representation set."

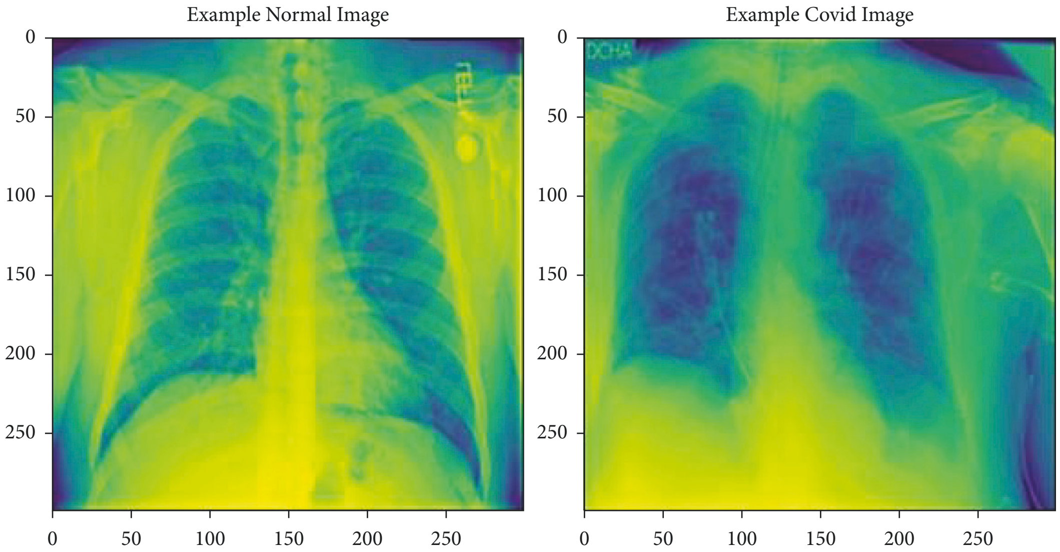


FIGURE 7: Chest X-ray images of COVID-19 and normal.

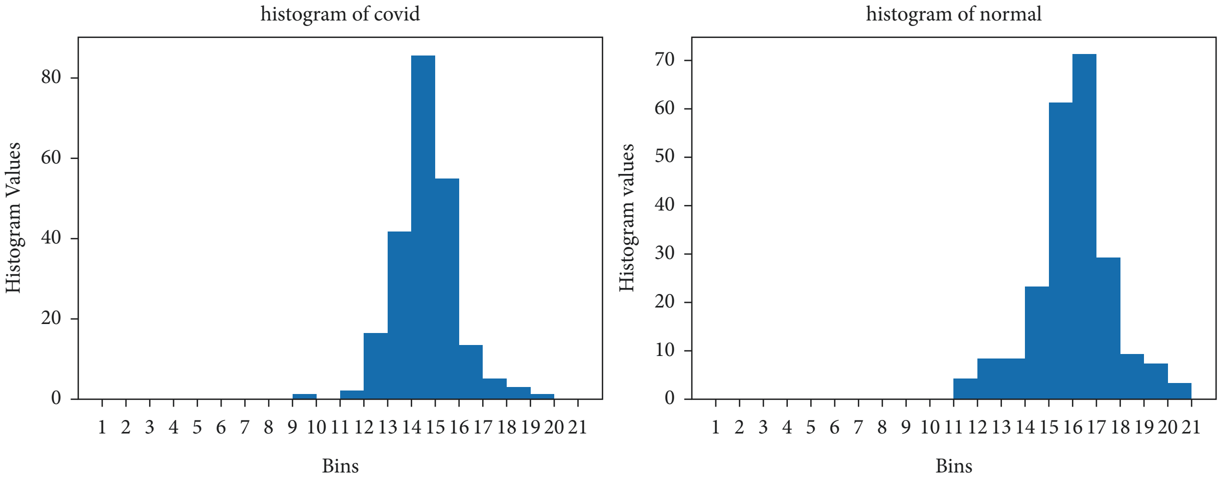


FIGURE 8: Histogram for X-ray images of COVID-19 and normal.

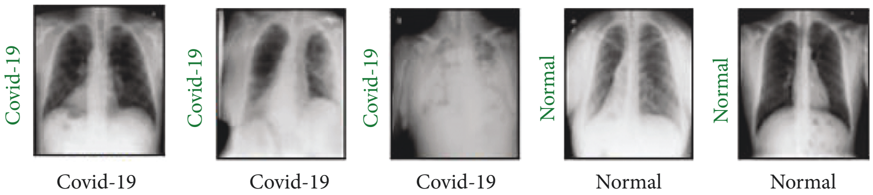


FIGURE 9: Training images for X-ray images of COVID-19 and normal.

9. Results and Discussion

In this segment, we present the outcomes and analysis of the experiments performed in the COVID-19 patient prediction using the chest X-ray dataset. From the experimental results, the proposed method shows better performance in terms of accuracy, precision, recall, F1-score, AUC, G-mean, and other parameters. For each model, PCA-SVM, PCA-ELM, and PCA-IELM, a separate confusion matrix is formed. All the performance metrics values are derived from the confusion matrix (Tables 4–6). Classification accuracy gained by the proposed method PCA-IELM is 98.11% over the chest X-ray dataset, which suggests better results than the other two models, PCA-based SVM (91.8%) and PCA-based ELM (93.80%) in terms of accuracy. Sometimes, performance metrics' accuracy may be misleading and can misclassify instances. So, other metrics are also taken into consideration to confirm the

Table 4: Confusion matrix for PCA-SVM.

	Predicted		Total
Actual	819	152	971
	187	2985	3172
Total	1006	3137	4143

Table 5: Confusion matrix for PCA-ELM.

	Predicted		Total
Actual	828	110	938
	147	3058	3205
Total	975	3168	4143

Table 6: Confusion matrix for PCA-IELM.

	Predicted		Total
Actual	1192	48	1240
	30	2873	2903
Total	1222	2921	4143

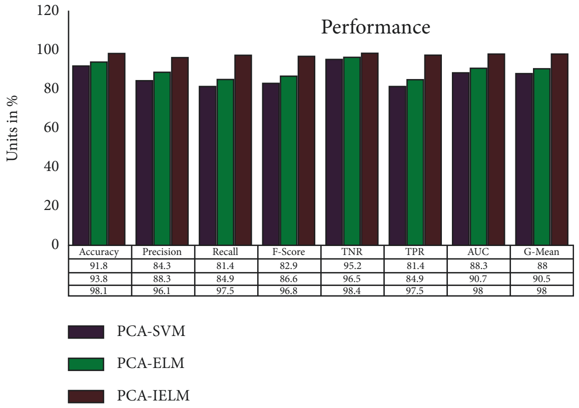


Figure 10: Performance comparison of different classifiers.

claim made by the classifier. PCA-IELM has the highest precision value of 96.11%. That means PCA-IELM is 96.11% reliable in making decisions, whereas models PCA-SVM and PCA-ELM record less precision, 84.3% and 88.3%, respectively. Similarly, for the proposed method PCA-IELM, other metrics (refer to Figure 10) recall, F_1-score, TPR, TNR, and G-mean are considerably higher than the other two methods, PCA-SVM and PCA-ELM.

The geometric mean (G-mean) is a statistic that analyzes categorization performance across majority and minority classes. Even if negative examples are correctly labelled as such, a poor G-mean suggests weak performance in identifying positive occurrences. This statistic is essential for preventing overfitting the negative class while underfitting the positive class, since the COVID-19 dataset understudy is also class imbalanced (IR = 2.81). Even then, the PCA-ELM model indicates good performance by attaining the highest G-mean value of 98%. Similarly, PCA-SVM and PCA-ELM have 88% and 90.5% success rates, respectively.

Table 7 demonstrates the performance variation (sensitivity, specificity, precision, F1-score, accuracy) based on different counts of hidden nodes in the range of 10–150 with an interval of 10 hidden nodes. Training and testing accuracies of PCA-IELM demonstrated almost the same behavior on the COVID-19 dataset (refer to Figure 11). There is moderate variation in the accuracy of PCA-IELM with respect to different numbers of hidden nodes. The accuracy at 10 numbers of hidden nodes was found to be 97.73%, and 98.11% was achieved at 140 numbers of hidden nodes in the PCA-IELM model and beyond (refer to Table 7).

When there is a moderate to large class imbalance, precision-recall curves should be drawn. Here, the COVID-19 dataset is imbalanced with an imbalance ratio (IR) of 2.81. It is worth noticing that precision is also called the positive

Table 7: Performance variation based on different hidden nodes.

Number of hidden nodes	Performance metrics (%)				
	Sensitivity	Specificity	Precision	F_1-score	Accuracy
10	94.13	96.23	93.74	95.19	97.73
20	94.16	96.19	93.36	95.19	97.73
30	94.24	96.35	94.48	95.24	97.74
40	93.98	96.07	93.01	94.98	97.70
50	94.11	96.16	93.09	95.11	97.71
60	94.18	96.28	93.45	95.18	97.73
70	94.73	96.82	94.69	95.64	97.75
80	94.84	96.79	94.86	95.29	97.75
90	94.25	96.77	94.92	95.36	97.75
100	94.03	96.11	93.11	95.02	97.71
110	94.17	96.24	93.22	95.16	97.73
120	94.48	96.46	94.59	95.71	97.75
130	94.10	96.15	93.19	95.11	97.71
140	97.62	98.12	96.33	96.50	98.11
150	97.54	98.35	96.12	96.83	98.11

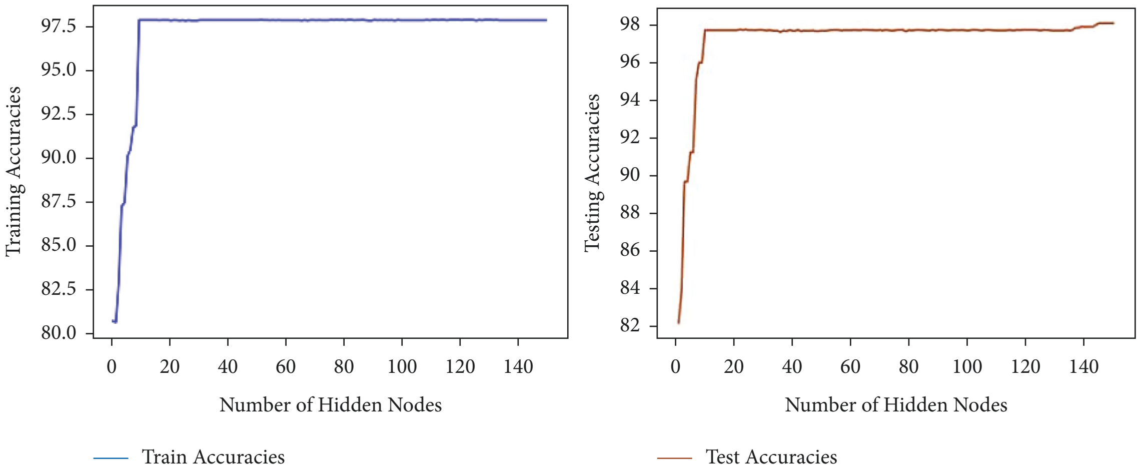
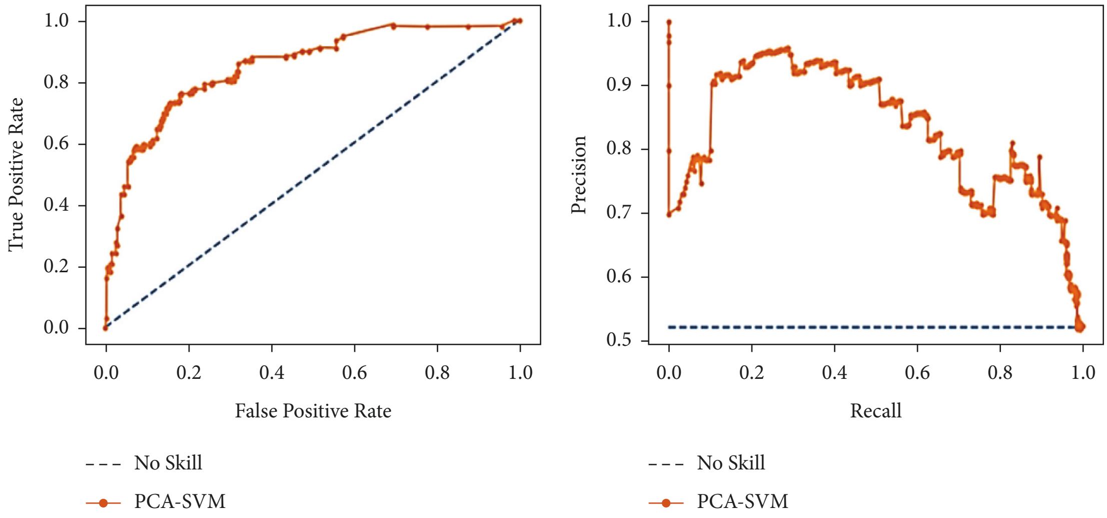


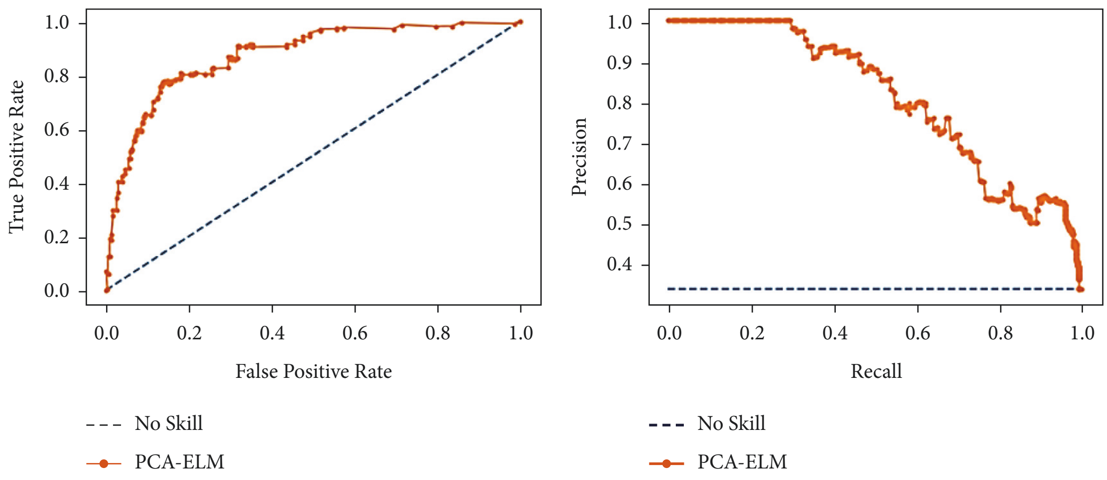
Figure 11: Accuracy variation with number of hidden nodes for PCA-IELM.



(a)

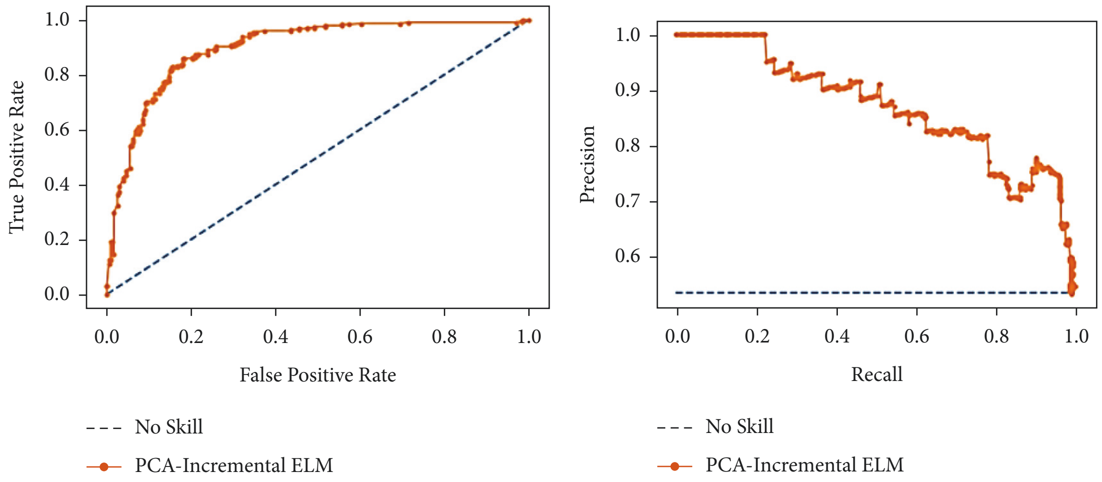
(b)

Figure 12: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-SVM.



(a) (b)

Figure 13: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-ELM.



(a) (b)

Figure 14: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-incremental ELM.

Table 8: Proposed method and other related models' comparative analysis [38].

S.No.	Study	Method used	Number of cases	Type of images	Accuracy (%)
1	Ioannis et al. [39]	VGG-19	700 pneumonia, 504 healthy, 224 COVID-19 (positive)	Chest X-ray	93.48
2	Gunraj, Wang, and Wong [40]	COVID-Net	5526 COVID-19 (negative), 8066 healthy, 53 COVID-19 (positive)	Chest X-ray	92.4
3	Sethy et al. [41]	ResNet50þ SVM	25 COVID-19 (negative), 25 COVID-19 (positive)	Chest X-ray	95.38
4	Hemdan et al. [42]	COVIDX-Net	25 normal, 25 COVID-19 (positive)	Chest X-ray	90.0
5	Narin et al. [43]	Deep CNN ResNet-50	50 COVID-19 (negative), 50 COVID-19 (positive)	Chest X-ray	98
6	Ying et al. [44]	DRE-Net	708 healthy, 777 COVID-19 (positive)	Chest CT	86
7	Wang et al. [45]	M-Inception	258 COVID-19 (negative), 195 COVID-19 (positive)	Chest CT	82.9
8	Zheng et al. [46]	UNetþ3D deep network	229 COVID-19 (negative), 313 COVID-19 (positive)	Chest CT	90.8
9	Xu et al. [47]	ResNetþ location attention	175 healthy, 224 viral pneumonia, 219 COVID-19 (positive)	Chest CT	86.7
10	Tulin et al. [38]	DarkCovidNet	500 Pneumonia, 500 no-findings, 125 COVID-19 (positive	Chest X-ray	98.08
11	Proposed model	PCA-IELM	10,192 normal, 3616 COVID-19 (positive)	Chest X-ray	98.11

Table 9: Time elapsed during training and testing of models.

Dataset	Algorithm	Training time (s)	Testing time (s)
COVID-19 chest X-ray	PCA-SVM	0.027	0.022
	PCA-ELM	0.053	0.049
	PCA-IELM	12.353	9.525

predictive value (PPV). Moreover, recall is also known as sensitivity, hit rate, or true-positive rate (TPR). It means they talk about positive cases and not negative ones. Most machine learning algorithms often involve a trade-off between recall and precision. A good PR curve has a greater AUC (area under curve). Figures 12(b), 13(b), and 14(b) depict PR curves. Figure 13(b) shows the greater AUC, which is an indication of the better performance of PCA-IELM than the other two models. In addition to these, ROC of Figure 14(a) also grabs more AUC than two other Figures 12(a) and 13(a). Therefore, PCA-IELM claims better performance than PCA-SVM and PCA-IELM. The proposed PCA-IELM model outperforms other previously developed models for identification of COVID-19 patients from chest X-ray image (refer Table 8 [38–47]). As far as the training and testing time taken by the proposed model PCA-IELM is concerned, it was higher (refer to Table 9) because the execution of the model happened in an incremental way and not in one go.

10. Conclusions

In this paper, an effective classification model is proposed on the COVID-19 chest X-ray image dataset using principal component analysis (PCA) and incremental extreme learning machine (IELM). This study established the valuable application of the ELM model to classify COVID-19 patients from X-ray images by developing the PCA-IELM model. The proposed PCA-based IELM algorithm is an efficient IELM-based algorithm. The hidden node parameters are measured by the information returned to the PCA in the training dataset, and using the Moore–Penrose generalized inverse output, the node parameters are determined. PCA-IELM utilizes the best feature of IELM, which is to increase hidden nodes incrementally and wisely determine the output weights, whereas ELM requires you to set the appropriate number of hidden nodes manually, and this is similar to the hit and trial method. In comparison with the ELM, the output error of the IELM rapidly reduces and is near to zero as the number of hidden neurons increases. It was observed that as the number of hidden nodes increased, the performance of the PCA-IELM increased and it became stable at 150 hidden nodes. PCA-IELM outperforms PCA-SVM and PCA-ELM in terms of accuracy (98.11%), precision (96.11%), recall (97.50%), F1-score (98.50%), G-mean (98%), etc. The suggested research contributes to the prospect of a low-cost, quick, and automated diagnosis of the COVID-19 patient, and it may be used in clinical scenarios. This effective system can provide early detection of COVID-19 patients. As a result, it is helpful in controlling the further spread of the virus from an affected person. This is an intelligent assistance for radiologists to accurately diagnose COVID-19 in X-ray images.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] F. Cao, B. Liu, and D. Sun Park, “Image classification based on effective extreme learning machine,” *Neurocomputing*, vol. 102, pp. 90–97, 2013.

[2] L. Ying, “Orthogonal incremental extreme learning machine for regression and multiclass classification,” *Neural Computing & Applications*, vol. 27, no. 1, pp. 111–120, 2016.

[3] Y. Sun, L Li, L. Zheng et al., “Image classification base on PCA of multi-view deep representation,” *Journal of Visual Communication and Image Representation*, vol. 62, pp. 253–258, 2019.

[4] V. Kumar, “Evaluation of computationally intelligent techniques for breast cancer diagnosis,” *Neural Computing & Applications*, vol. 33, no. 8, pp. 3195–3208, 2021.

[5] M. Mustaqeem and M. Saqib, “Principal component-based support vector machine (PC-SVM): a hybrid technique for software defect detection,” *Cluster Computing*, vol. 24, no. 3, pp. 2581–2595, 2021.

[6] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, “Pca-Elm PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis,” *Neural Processing Letters*, vol. 37, no. 3, pp. 377–392, 2013.

[7] M. Mateen, J. Wen, N. Nasrullah, S. Song, and Z. Huang, “Fundus image classification using VGG-19 architecture with PCA and SVD,” *Symmetry*, vol. 11, no. 1, 2018.

[8] Z. Zhao, Z. Chen, Y. Chen, S. Wang, and H. Wang, “A class incremental extreme learning machine for activity recognition,” *Cognitive Computation*, vol. 6, no. 3, pp. 423–431, 2014.

[9] G. B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.

[10] M. Zhu, Y. Song, J. Guo et al., “PCA and Kernel-based extreme learning machine for side-scan sonar image classification,” *2017 IEEE Underwater Technology (UT)*, vol. 238, p. 2017, 2017.

[11] X. Kang, S. Member, X. Xiang, J. A. Benediktsson, S. Li, and S. Member, “PCA-based edge-preserving features for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, 2017.

[12] C. Perales-González, F. Fernández-Navarro, J. Pérez-Rodríguez, and M. Carbonero-Ruz, “Negative correlation hidden layer for the extreme learning machine,” *Applied Soft Computing*, vol. 109, Article ID 107482, 2021.

[13] R. Li, H. Zhang, S. Gao, Z. Wu, and C. Guo, “An improved extreme learning machine algorithm for transient

electromagnetic nonlinear inversion," *Computers & Geosciences*, vol. 156, Article ID 104877, 2021.

[14] J. Zhang, S. Ding, N. Zhang, and Z. Shi, "Incremental extreme learning machine based on deep feature embedded," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 1, pp. 111–120, 2016.

[15] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[16] B. S. Raghuwanshi and S. Shukla, "Classifying imbalanced data using SMOTE based class-specific kernelized ELM," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 5, pp. 1255–1280, 2021.

[17] G. B. Huang, M. B. Li, L. Chen, and C. K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, no. 4–6, pp. 576–583, 2008.

[18] S. Xu and J. Wang, "A fast incremental extreme learning machine algorithm for data streams classification," *Expert Systems with Applications*, vol. 65, pp. 332–344, 2016.

[19] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, p. 1223, 2019.

[20] S. M. Basha and D. S. Rajput, "A supervised aspect level sentiment model to predict overall sentiment on tweeter documents," *International Journal of Metadata Semantics and Ontologies*, vol. 13, no. 1, pp. 33–41, 2018.

[21] S. M. Basha and D. S. Rajput, "A roadmap towards implementing parallel aspect level sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29463–29492, 2019.

[22] A. K. Dubey and K. K. Mohbey, "Enabling CT-Scans for covid detection using transfer learning-based neural networks," *Journal of Biomolecular Structure and Dynamics*, pp. 1–12, 2022.

[23] S. M. Basha and D. S. Rajput, "Survey on evaluating the performance of machine learning algorithms past contributions and future roadmap," *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, Academic Press, Cambridge, Ma, pp. 153–164, 2019.

[24] Y. Z. Miao, X. P. Ma, and S. P. Bu, "Research on the learning method based on PCA-ELM," *Intelligent Automation & Soft Computing*, vol. 23, no. 4, pp. 637–642, 2017.

[25] R. Sakthivel, I. S. Thaseen, M. Vanitha et al., "An efficient hardware architecture based on an ensemble of deep learning models for COVID-19 prediction," *Sustainable Cities and Society*, vol. 80, Article ID 103713, 2022.

[26] S. Zhang and W. Tan, "An extreme learning machine based on the mixed kernel function of triangular kernel and generalized hermite dirichlet kernel," *Discrete Dynamics in Nature and Society*, vol. 2016, pp. 1–11, 2016.

[27] B. S. Raghuwanshi and S. Shukla, "UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning," *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 252–270, 2018.

[28] B. S. Raghuwanshi and S. Shukla, "Classifying imbalanced data using ensemble of reduced kernelized weighted extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3071–3097, 2019.

[29] K. Vishnu Vandana and D. S. Rajput, "A Review on the significance of machine leaning for data analysis in big data," *Jordanian Journal of Computers and Information Technology*, vol. 06, no. 01, pp. 41–57, 2020.

[30] J. Qin, T. F. Burks, M. S. Kim, K. Chao, and M. A. Ritenour, "Citrus canker detection using hyperspectral reflectance imaging and PCA-based image classification method," *Sensing and Instrumentation for Food Quality and Safety*, vol. 2, no. 3, pp. 168–177, 2008.

[31] K. Cheng, S. Gao, W. Dong, X. Yang, Q. Wang, and H. Yu, "Boosting label weighted extreme learning machine for classifying multi-label imbalanced data," *Neurocomputing*, vol. 403, pp. 360–370, 2020.

[32] S. Song, M. Wang, and Y. Lin, "An improved algorithm for incremental extreme learning machine," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 308–317, 2020.

[33] G. T. Reddy, M. P. K Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, Article ID 54776, 2020.

[34] P. Kumar and R. Singh Thakur, "An approach using fuzzy sets and boosting techniques to predict liver disease," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3513–3529, 2021.

[35] V. Kumar and G. S. Lalotra, "predictive model based on supervised machine learning for heart disease diagnosis," in *Proceedings of the 2021 IEEE International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES)*, pp. 1–6, IEEE, Raipur, India, 17 December 2021.

[36] P. Kumar and R. S. Thakur, "Liver disorder detection using variable- neighbor weighted fuzzy K nearest neighbor approach," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16515–16535, 2021.

[37] D. P Tuan, "COVID-19 Radiography Database," *COVID-19 Chest X-ray Database*, vol. 9, no. 1, 2021, https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[38] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, Article ID 103792, 2020.

[39] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*vol. 43, no. 2, pp. 635–640, 2020.

[40] H. Gunraj, L. Wang, and A. Wong, "COVIDNet-CT: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest ct images," *Frontiers of Medicine*, vol. 7, pp. 1–12, 2020.

[41] P. K. Sethy, S. K. Behera, P. K. Ratha, and P. Biswas, "Detection of coronavirus disease (COVID-19) based on deep features and support vector machine," *Int. J. Math. Eng. Manag. Sci*, vol. 5, no. 4, pp. 643–651, 2020.

[42] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," *Biomedical Image Analysis*, 2020, http://arxiv.org/abs/2003.11055.

[43] T. Gadekallu Reddy, P Quoc-Viet, T Thien Huynh, B Sweta, M Praveen Kumar Reddy, and L Madhusanka, "Federated learning for big data: a survey on opportunities, applications, and future directions," *Mobile Edge Computing and Wireless AI in 5G and BeyondarXiv preprint*, 2021.

[44] S. Ying, S. Zheng, L. Li et al., "Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, no. –1, p. 1, 2020.

[45] S. Wang, B. Kang, J. Ma et al., "A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19)," *European Radiology*, vol. 31, no. 8, pp. 6096–6104, 2021.

[46] S. Bhattacharya, P. K. Reddy Maddikunta, Q. V. Pham et al., "Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey," *Sustainable Cities and Society*, vol. 65, Article ID 102589, 2021.

[47] X. Xu, X. Jiang, C. Ma et al., "A deep learning system to screen novel coronavirus disease 2019 pneumonia," *Engineering*, vol. 6, no. 10, pp. 1122–1129, 2020.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2218594, 20 pages
https://doi.org/10.1155/2022/2218594

Review Article

Harris Hawk Optimization: A Survey on Variants and Applications

B. K. Tripathy,[1] **Praveen Kumar Reddy Maddikunta**,[1] **Quoc-Viet Pham**,[2] **Thippa Reddy Gadekallu**,[1] **Kapal Dev**,[3] **Sharnil Pandya**,[4] **and Basem M. ElHalawany**[5]

[1]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India*
[2]*Korean Southeast Center for the 4th Industrial Revolution Leader Education, Pusan National University, Busan 46241, Republic of Korea*
[3]*Department of Institute of Intelligent Systems, University of Johannesburg, Johannesburg, South Africa*
[4]*Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune, Maharashtra, India*
[5]*Faculty of Engineering at Shoubra, Benha University, Banha, Egypt*

Correspondence should be addressed to Thippa Reddy Gadekallu; thippareddy.g@vit.ac.in

Received 19 February 2022; Accepted 24 May 2022; Published 27 June 2022

Academic Editor: Shahzad Sarfraz

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

In this review, we intend to present a complete literature survey on the conception and variants of the recent successful optimization algorithm, Harris Hawk optimizer (HHO), along with an updated set of applications in well-established works. For this purpose, we first present an overview of HHO, including its logic of equations and mathematical model. Next, we focus on reviewing different variants of HHO from the available well-established literature. To provide readers a deep vision and foster the application of the HHO, we review the state-of-the-art improvements of HHO, focusing mainly on fuzzy HHO and a new intuitionistic fuzzy HHO algorithm. We also review the applications of HHO in enhancing machine learning operations and in tackling engineering optimization problems. This survey can cover different aspects of HHO and its future applications to provide a basis for future research in the development of swarm intelligence paths and the use of HHO for real-world problems.

1. Introduction

Thanks to recent advances in computing capabilities and big data analytics, artificial intelligence (AI) has been considered in various applications, ranging from natural language processing and computer vision to wireless 6G systems and medicine [1]. As a subset of AI and nature-inspired algorithms, swarm intelligence (SI) has become a hot topic over the last decade. Conceptually, SI studies the complex collective behavior of the systems that are comprised of many simple agents. More particular, these simple agents can interact with others and also with their surrounding environment. According to [2], SI has many advantages compared with the conventional optimization approaches: (1) black-box optimizer, (2) gradient-free operation, (3) ability to obtain high-quality solutions by properly balancing exploratory and exploitative features, and finally (4) ease and simplicity of implementation. These characteristics and applicability are the main reasons behind the wide use of SI approaches. Some well-known SI methods are not limited to particle swarm optimization (PSO) [3], grey wolf optimizer (GWO) [4], genetic programming (GP) [5], biogeography-based optimizer (BBO) [6], and firefly algorithm (FA) [7]. Also, SI methods have found their applications in various applications and real-world problems, such as control engineering, civil engineering, electrical engineering, image processing, wireless communications, and vertical domains (e.g., smart cities and smart grids).

Proposed by Ali Asghar Heidari in 2019, the Harris Hawks optimization (HHO) has received much interest from the research communities [8]. HHO mimics the hunting behavior of the Harris Hawks in nature, namely surprise pounce. Considered as one of the most intelligent birds in nature, Harris Hawks can simulate different chasing styles based on different scenarios and escaping prey behaviors. More specifically, four chasing strategies are developed in HHO [8], including soft besiege, hard besiege, soft besiege with progressive rapid dives, and soft besiege with progressive rapid dives. The results tested over benchmark functions and several engineering optimization problems confirm that HHO outperforms many well-known SI approaches such as PSO, GWO, GP, BBO, and FA. Moreover, the results also show that HHO achieves a good balance of exploration and exploitation, thus improving the scalability of HHO and the ability to obtain high-quality solutions.

HHO has been leveraged for many applications and engineering problems due to its optimization features and competitive performance. For example, HHO is used to solve the problem of unmanned aerial vehicle (UAV) placement and radio resource allocation in visible light communications (VLC) [9]. This work demonstrates that HHO effectively tackles the nonlinearity caused by the VLC channel modeling and solves multiple optimization variables simultaneously. The work in [10] proposes a hybrid SI method based on HHO and WOA for feature selection. To improve the production quality, a convolutional neural network (CNN) jointly with HHO is leveraged in [11], in which CNN is used to classify the control chart patterns and HHO is used to tune the parameters of the CNN model such as the number of kernels and learning rate. All the above studies show that the HHO-based method has superiority over the baseline and state-of-the-art SI methods.

According to the no-free-lunch theorem, no single algorithm can solve all the real-world problems; that is, one algorithm can perform well for a set of problems but perform poorly for the other problems [12]. Therefore, HHO has been improved by different techniques, for example, binary HHO version, evolutionary-updating structures, chaotic operations, multi-objective HHO, and hybrid HHO. As HHO is originally invented for solving continuous optimization problems, a number of studies have been conducted for binary HHO versions. For example, the work in [13] develops a hybrid SI approach by integrating HHO with the Salp swarm algorithm (SSA), which is then applied to the feature selection problem. To exploit the effectiveness of chaotic-based updates in avoiding immature convergence, the work in [14] proposed adding the chaotic local search into the original HHO to improve its performance. Another application of HHO can be found in [15], where HHO is used as a trainer of feed-forward neural networks, which is then used for load forecasting in the Queensland electric market. There is also an effort on review of a few papers on HHO in [16]. However, that paper coverage is very different, and it provides another methodology of research for conducting a review.

1.1. Contributions. To foster the development of the HHO and its applications, this work sets to provide an overview, recent improvements, and applications from the available literature. Motivated by this observation, we set to provide a survey on HHO, including its mathematical model, recent variants, and applications. In a nutshell, the contributions offered by our work can be summarized as follows:

(i) We first present the underlying inspiration and the mathematical model of the HHO optimizer. This part is to help the readers to understand the underlying principle of the HHO and how it can be applied to solve engineering optimization problems.

(ii) We review the state-of-the-art improvements of HHO, focusing mainly on fuzzy HHO and a new intuitionistic fuzzy HHO algorithm.

(iii) We review the applications of HHO in various disciplines such as machine learning (ML), electrical/civil/image engineering, wireless communications, and control engineering.

We note that the references reviewed in this work are obtained from high-reputed publishers such as IEEE, Elsevier, Wiley, Springer Nature, and Taylor & Francis and also well-known archival websites such as arXiv. Moreover, the following queries are used to find the references, including “Harris Hawk optimization,” “HHO,” “swarm intelligence,” “artificial intelligence,” and “metaheuristic.”

1.2. Paper Organization. The remaining parts of this study are organized as follows. In Section 2, we present the inspiration and mathematical models of HHO. In Section 3, we review the state-of-the-art studies on variants of HHO. Next, in Section 4, we discuss the applications of HHO in ML applications and engineering applications. Finally, we conclude this study in Section 5. The list of frequently used acronyms is summarized in Table 1.

2. Harris Hawk Optimization

This section presents an overview of HHO, including its inspiration and mathematical model.

2.1. Inspiration of HHO. HHO was proposed by Heidari et al. in 2019 to simulate the hunting behavior of the Harris Hawks [8]. In 1997, Louis Lefebvre’s survey revealed that Harris Hawks are the most intelligent birds found in southern Arizona, USA [17]. The foraging behavior of Harris Hawks varies significantly from that of other birds, as Harris Hawks continue to forage with other family members of the same species. Harris Hawks use a technique called the “surprise pounce,” known as the “seven kills” approach to ambush the prey. During this attack, a few other hawks used to ambush in a number of directions and converge on the target rabbit, and the attack would be over in a matter of seconds. Harris Hawks use different hunting styles based on the escape behavior of the prey and the dynamic change in instances. For example, Hawks use switching tactics when the leader hawk dives quickly to attack the prey, and the prey is trying to escape from the leader hawk, and then, another hawk in the party team will immediately continue the chase. These switching tactics confuse the targeted prey and seek to exhaust the detected prey and increase its danger. Finally, tired prey

Table 1: Acronyms.

HHO	Harris Hawk optimization
AI	Artificial intelligence
SI	Swarm intelligence
PSO	Particle swarm optimization
GWO	Grey wolf optimizer
CE	Civil engineering
SCC	Soil compression coefficient
GOA	Grasshopper optimization algorithm
WOA	Whale optimization algorithm
ANN	Artificial neural networks
RMSE	Root-mean-square error
CoD	Coefficient of determination
MAE	Mean absolute error
SS	Soil slopes
HHOSA	HHO-simulated annealing
ASI	Acceleration severity index
CC	Correlation coefficient
PSO	Particle swarm optimization
GP	Genetic programming
FORM	First-order reliability method
MCE	Minimum cross-entropy
DEAHHO	Differential evolutionary adaptive HHO
MVO	Multi-verse optimization algorithm
DE	Differential evolution
SSA	Salp swarm algorithm
PSNR	Peak signal-to-noise ratio
FSI	Feature similarity index
SSI	Structural similarity index
WHHO	WAO-HHHO
CNN	Convolutional neural network
PCNN	Pulse coupled neural network
PV	Photovoltaic
TDOV	Three-diode photovoltaic
PS	Partial shading
MPPT	Maximum power point tracking
WSN	Wireless sensor networks
FiWi	Fiber wireless

cannot escape the hawk's team, as one of the mighty hawks slaughters the tired prey and shares it with the party members.

HHO's main inspiration is the collaborative action and hunting style of the Harris Hawk in the wildlife called the surprise pounce. The HHO mathematical model generates dynamic patterns and behaviors for the development of an optimization algorithm. The performance of the HHO optimization algorithm is evaluated by comparing it with other existing metaheuristic techniques, 29 benchmark challenges, and many real-world engineering issues. Experimental findings and comparative results have shown that the HHO algorithm delivers better results than other existing metaheuristic techniques [8, 18].

2.2. Mathematical Models of HHO. This section discusses the mathematical model of the HHO algorithm, which comprises an exploration phase, an exploitation phase, and a number of Harris Hawk attacking approaches. HHO is a nature-inspired algorithm that can be applied to any optimization problem. This section presents all phases of HHO, which are further explained in the following subsections.

2.3. Exploration Phase. In this subsection, the exploration phase of HHO is discussed. The Harris Hawks have powerful eyes that can monitor and identify prey, but sometimes the prey is not visible. During this condition, the hawks have been waiting for long hours and monitoring to identify the prey. In HHO, hawks are considered as candidate solutions, and in each iteration, the prey is considered the optimal solution. Hawks perch in specific locations and constantly monitor the surrounding environment to identify prey using two strategies, which are represented in equation (1). If $p < 0.5$, the hawks perch based on the position of the family members. If $p \geq 0.5$, the hawks perch in a random space within the population area.

$$A(x+1) = \begin{cases} A_{\mathrm{r}}(x) - a_1\left|A_{\mathrm{a}}(x) - 2a_2A(x)\right|, & p \geq 0.5, \\ \left(A_{\mathrm{rabbit}}(x) - A_p(x)\right) - a_3, & \\ \left(L_B + a_4\left(U_B - L_B\right)\right), & p < 0.5. \end{cases} \quad (1)$$

In equation (1), where $A(x+1)$ denotes the position of Hacks at the next iteration. $\mathrm{A}_{\mathrm{rabbit}}(x)$ denotes the position of rabbit. $A(x)$ denotes the current position of hawks. a_1, a_2, a_3, a_4, and p are random variables ranging from 0 to 1. LB and UB are the lower bound and the upper bound of random variables. $A_p(x)$ indicates the average hawk position, which is represented in the following equation, where $A_i(x)$ denotes the location of each hawk at ith iteration and H denotes the number of Hacks in the search space.

$$A_{\mathrm{p}}(x) = \frac{1}{H}\sum_{i=1}^{H} A_i(x). \quad (2)$$

2.4. Transition from Exploration to Exploitation. This subsection explains the transformation from the exploration phase to the exploitation phase, based on the energy level of the prey to escape, which is mathematically defined as follows:

$$P = 2P_0\left(1 - \frac{x}{I}\right), \quad (3)$$

where P denotes the energy of prey to escape at iteration x, I denotes the total number of iterations, and P_0 denotes the initial energy of prey. During each escape, the energy level of the prey drops dramatically. For each iteration, P_0 will change the value from (−1, 1). When P_0 drops down from 0 to −1, the prey is exhausted; similarly, when P_0 value is increasing from 0 to 1, the prey is reinforced. When $|P| \geq 1$, exploration took place, and when $|P| < 1$, exploitation arises. Figure 1 represents the escaping energy behavior [8].

2.5. Exploitation Phase. This subsection explains the exploitation phase in which the hawks attack the targeted prey. Then, however, the prey tries to escape the attack. Based on hawk attacking behavior and escaping prey behavior, four approaches are discussed in the following subsections.

2.6. Soft Besiege. In HHO, soft besiege occurs when $a \geq 0.5$ and $|P| \geq 0.5$, and the prey has sufficient energy to escape from the attack, but cannot escape from the attack as the

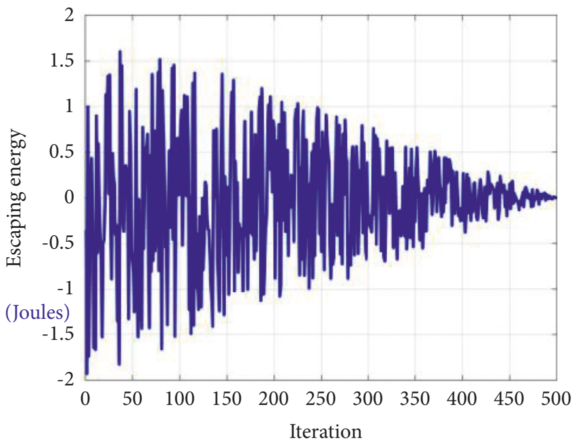


Figure 1: Escaping energy behavioral pattern.

hawks encircle the prey and the energy of the prey gets drained. The following equations explain the mathematical behavioral model.

$$A(x+1) = \Delta A(x) - P\left|UA_{\text{rabbit}}(x) - A(x)\right|, \tag{4}$$

$$\Delta A(x) = A_{\text{rabbit}}(x) - A(x). \tag{5}$$

$\Delta A(x)$ denotes the difference between the position vector of the rabbit and present position at xth iteration. U represents the rabbit's random jump strength during escape, where U is updated as $U = 2\ (1 - a_5)$, and a_5 is a random number ranging from [0, 1].

2.7. Hard Besiege. In HHO, hard besiege occurs when $a \geq 0.5$ and $|P| < 0.5$, and the prey is exhausted and does not have enough energy. The hawks encircle the prey and perform the surprise pounce. The updated position of the hawks is shown in equation (6). Figure 2 depicts the vectors during hard besiege.

$$A(x+1) = A_{\text{rabbit}}(x) - P|\Delta A(x)|. \tag{6}$$

2.8. Soft Besiege with Progressive Rapid Dives. This subsection deals with soft besiege with progressive rapid dives where the prey has enough energy $|P| \geq 0.5$ to escape the attack, but the hawk builds a soft besiege $a < 0.5$. In this step, the hawk must think intelligently and choose the best position to target the prey. The respective steps accomplish the process.

(1) Performing various moves

(2) Analyzing and thinking on a new move using equation (7)

(3) Evaluating the movement with the previous dive to the prey and realizing whether the movement is favorable or not

$$T = A_{\text{rabbit}}(x) - P\left|UA_{\text{rabbit}}(x) - A(x)\right|. \tag{7}$$

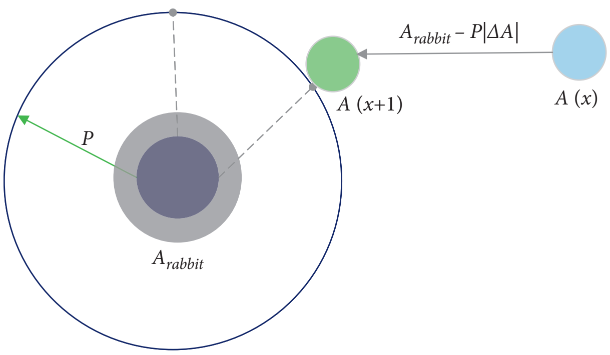


Figure 2: Vectors during hard besiege.

(4) If the movement is not favorable to attack the prey, a dive is selected based on a levy flight (LF) using the following equation:

$$L = T + V \times \text{LF}(z), \tag{8}$$

where Z is represented as the dimension of the problem, V is considered as a random vector of size $1 * Z$, and LF function is calculated using the following equation:

$$\begin{aligned} \text{LF}(f) &= 0.01 \times \frac{m \times \psi}{|n|}, \\ \psi &= \left(\frac{\xi(1+\chi) \times \sin(\pi\chi/2)}{\xi(1+\chi/2) \times \chi \times 2^{(x-1/2)}} \right)^{1/\chi}, \end{aligned} \tag{9}$$

where m and n are random values that range between (0, 1) and χ is a constant that is set to 1.5. Equation (10) is used to update the positions of the hawks during the soft besiege phase.

$$A(x+1) = \begin{cases} T \text{ if } F(T) < F(A(x)), \\ L \text{ if } F(L) < F(A(x)), \end{cases} \tag{10}$$

where T and L are acquired using equations (7) and (8), and F is considered to be a fitness function for the problem. Figure 3 depicts the vectors during soft besiege with progressive rapid dives.

2.9. Hard Besiege with Progressive Rapid Dives. This subsection deals with hard besiege with progressive rapid dives when $|P| < 0.5$ and $a < 0.5$ where the prey does not have enough energy to escape the attack and hawk builds a hard besiege to catch and kill the prey. In this phase, the prey's condition is similar to that of the soft besiege, but the hawks intend to minimize the distance between their locations towards escaping prey. The equation explains the hard besiege condition.

$$\mathbf{A}(\mathbf{x}+1) = \begin{cases} T \text{ if } F(x) < F(A(x)), \\ L \text{ if } F(x) < F(A(x)), \end{cases} \tag{11}$$

where T and L are derived using the following equations.

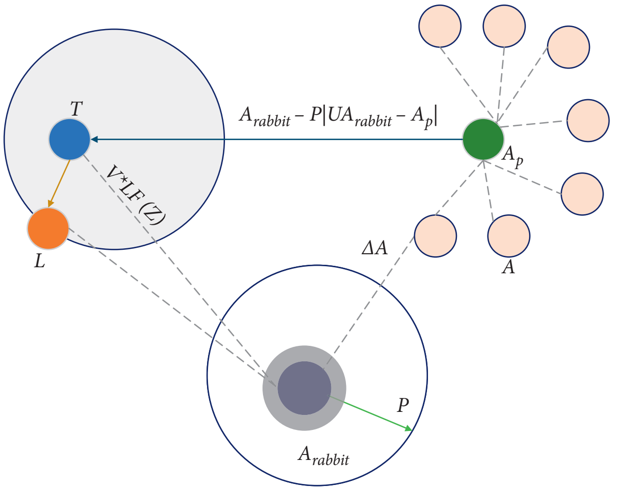


FIGURE 3: Vectors during soft besiege with progressive rapid dives.

$$T = A_{\text{rabbit}}(x) - P\left|UA_{\text{rabbit}}(x) - A_p(x)\right|, \tag{12}$$

$$L = T + V \times LF(z). \tag{13}$$

Figures 4 and 5 depict vectors during hard besiege with progressive rapid dives in 2D and 3D space. Algorithm 1 explains pseudocode for HHO.

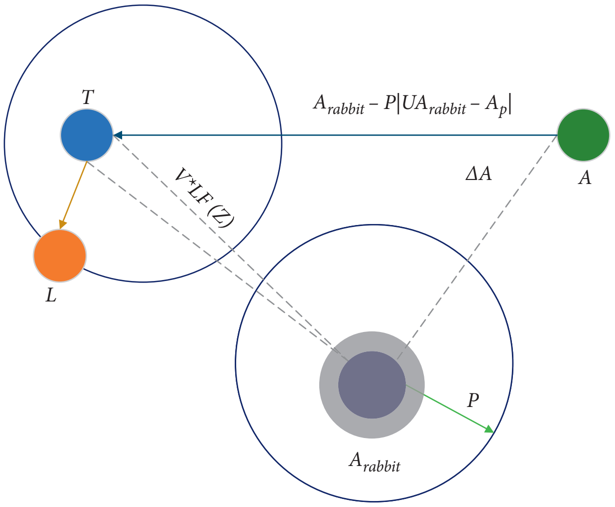


FIGURE 4: Vectors during hard besiege with progressive rapid dives in 2D.

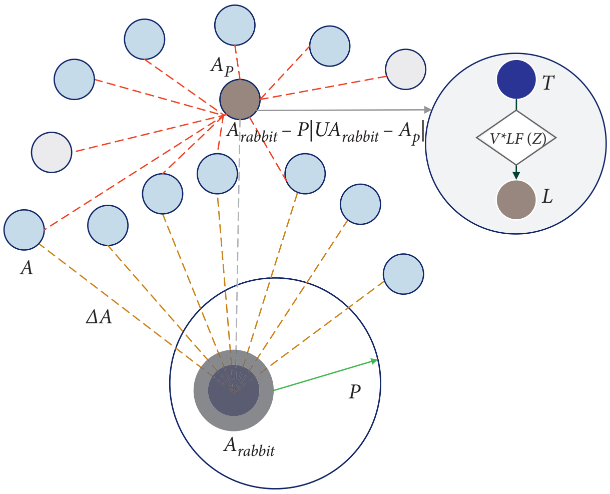


FIGURE 5: Vectors during hard besiege with progressive rapid dives in 3D.

3. Recent Variants of HHO

This section discusses various HHO variants that are used in a variety of engineering and ML applications, as shown in Tables 2–4. We discussed the fuzzy HHO in this section and proposed a new intuitionistic fuzzy HHO algorithm that is significantly different from other existing papers [16].

3.1. Fuzzy Harris Hawk Algorithm. Fuzzy sets [48] were introduced as an extension of crisp sets to take care of graded membership of objects in a class, which are more general and natural. This has led to the extension of crisp concepts to fuzzy concepts. Fuzzy logic (FL), in the generalized sense, is synonymous with fuzzy sets. One of the crucial components of FL is the fuzzy inference system (FIS).

A FIS has five functional blocks, a rule base that contains several fuzzy If... then rules. In addition, this database defines the membership functions of fuzzy sets used in fuzzy rules. This decision-making unit operates on the rules, a fuzzification interface unit that converts crisp quantities into fuzzy quantities and a defuzzification interface unit that converts the fuzzy quantities into crisp quantities. FIS is the most critical tool in fuzzy set theory, and in literature, we find two crucial FIS: the Mamdani FIS proposed in 1975 [49] and the Sugeno FIS proposed in 1985 [50]. Of these, Mamdani FIS is the popular one and finds a greater acceptance. However, both approaches have certain advantages and disadvantages. The significant differences are in the output membership functions and the consequents of the fuzzy rules used. As far as the functionality of a FIS is concerned, if the input is crisp, it is fuzzified in the fuzzification unit using one of the various fuzzification techniques. Fuzzification is the process of decomposing a system input and/or output into one or more fuzzy sets. Many types of curves and tables can be used, but triangular- or trapezoidal-shaped membership functions are the most common, since they are easier to represent in embedded controllers. A set of fuzzy if-then rules are determined. The rule strengths are obtained by combining the fuzzified inputs. The output membership function is combined with rule strength to obtain the consequent. An output distribution is obtained by combining all the consequents. The defuzzification unit is used to get a defuzzified output distribution.

The two parameters involved in the HHO algorithm are the energy of the prey E and the uniform random number parameter q, which determines the hawk's position the next time. The general tendency to fuzzify any algorithm is to fuzzify its parameters. Following the same approach in

```
**Inputs**: initial the population size $H$, number of iterations $I$.
**Outputs**: the Prey's position and fitness value.
Generate an arbitrary population
$A_i (i = 1, 2, \ldots, H)$
**while** stopping requirement is not reached **do**
    Compute the hawk's fitness values
    Set Arabbit as the best position of rabbit
    **for** each hawk ($A_i$) **do**
        Update energy $P_0$ and jump capacity $U$
        $P_0 = 2r() - 1,\ U = 2(1 - r())$
        Update the $P$ with equation (3)
    **end**
    **if** $|P| \geq 1$ **then**
        Update the position vector with equation (1)
    **end**
    **if** $|P| < 1$ **then**
        **if** $a \geq 0.5$ and $|P| \geq 0.5$ **then**
            Update the position vector with equation (4)
            **else if** $a \geq 0.5$ and $|P| < 0.5$ **then**
                Update the position vector with equation (6)
            **else if** $a < 0.5$ and $|P| \geq 0.5$ **then**
                Update the position vector with equation (10)
            **else if** $a < 0.5$ and $|P| < 0.5$ **then**
                Update the position vector with equation (11)
            **end**
        **end**
    **end**
**return** Arabbit
```

Algorithm 1: HHO algorithm [8].

Table 2: A summary of HHO variants.

Ref	Variant	Methodology	Objective
[19]	Multi-objective HHO	HHO is integrated with roulette wheel selection method with probability	To improve the extreme learning machine parameter selection
[20]	Gaussian barebone HHO	HHO is integrated with Gaussian barebone	To optimize kernel extreme learning machines for the prediction of entrepreneurial intention
[21]	Chaotic sequence-guided HHO	HHO is integrated with chaotic sequences	For data clustering
[22]	Dynamic HHO with mutation	HHO used mutation and dynamic control strategy to balance the exploitation and exploration phases in the HHO method	To perform segmentation on satellite and oil pollution images
[23]	Hybrid HHO differential evolution	Making two equal subpopulations from a complete one and training both the subpopulation parallelly using HHO and differential evolution	Multilevel image segmentation
[24]	Adaptive HHO	Mutation is used by HHO to clip the escape energy	Multilevel image segmentation
[25]	Hybrid OBL-HHO	OBL generates a solution for HHO through adversarial learning approach	To select the informative features from the feature space in conjunction with support vector regression
[26]	Elite OBL (EOBL)-HHO	The EOBL stacks upon OBL by selecting the fittest individual that would direct the population towards global minimum	To select informative features from the feature space
[27]	HHOBSA	The bitwise operations help HHO to improve the feature selection process, whereas the simulated annealing helps HHO to find the global minimum	Optimal feature selection
[28]	Chaotic HHO	Simulated annealing to improve HHO and the chaotic maps are used instead of random variables to achieve global optimum	To select most informative features to train using K-nearest neighbor for classification task

Table 2: Continued.

Ref	Variant	Methodology	Objective
[13]	Salp swarm HHO	HHO is improved by adding Salp swarm optimization, which adjusts the populations and uses greedy selection to update the agent	To select the informative features
[29]	Hybrid differential evolution HHO	Nonlinear control formula balances the exploitation and the exploration of HHO throughout the convergence process	To optimize phase space reconstructions and kernel extreme learning machine parameters for wind speed forecasting
[30]	Vibrational HHO	Periodic mutations are added for enhancing swarm diversity in basic HHO method	To optimize SVM parameters for roll bearing fault diagnosis
[31]	Boosted HHO	HHO algorithm is boosted by integrating it with the exploratory phase of flower pollination algorithm and mutation step of differential evolution	To estimate the parameters efficiently for a single diode PV model
[32]	Horizontal and vertical crossover of HHO	Crisscross optimizer and the Nelder–Mead simplex algorithm are used to improve the searching capabilities of individuals for achieving faster convergence rate	Simulating an efficient PV system and extracting the unknown parameters

Table 3: A summary of HHO variants (continued).

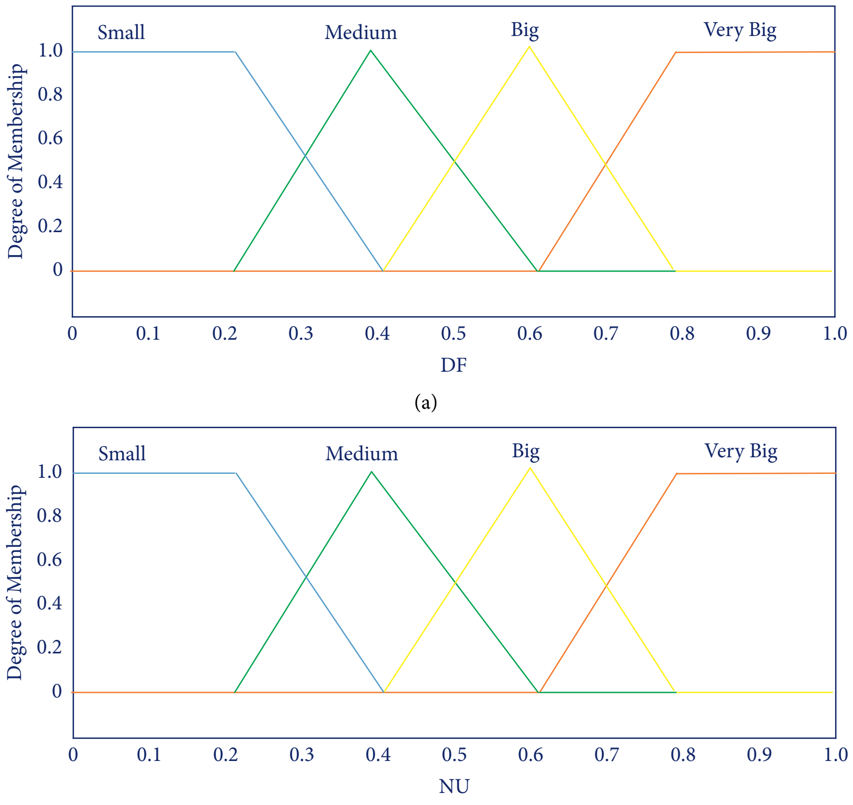
Ref	Variant	Methodology	Objective
[33]	Hybrid GWO-HHO	Mutation-based GWO is used to update the bottom layer in the population, and HHO is used to find global optimal solution in the upper layer	To optimize the parameters of phase space reconstruction and kernel-based extreme ML algorithms to predict the wind speeds accurately
[34]	Modified HHO	To improve the exploration phase's global search, the Levy flight is used to generate the ambiguous zigzag position of the prey once the hawk is deducted	To relieve the PV systems from the issue of mismatch power loss problems resulting due to the phenomenon of partial shading
[35]	Chaotic HHO	HHO is enhanced with ten chaotic functions to avoid local optima trapping of conventional HHO	To accurately estimate the proton exchange membrane fuel cell's operating parameters that mimic and simulate its electrical performance
[36]	Improved HHO	Instead of random location, the rabbit location is used to find the optimal position	To find the location of distribution generation optimally in a radial distribution system to minimize the voltage deviation and total active power loss and also to increase the voltage stability index under several operational constraints
[14]	Diversification enhanced HHO (EHHO)	OBL is used in HHO to do a comprehensive search. The OBL is used to select each agent's opposite position to select the optimal agent from the available pool, and its opposite agent will be treated as the next-generation agent in HHO	To identify the optimal agents and unknown parameters of modules of PV model
[37]	Hybrid of HHO and GOA	The ensemble of GOA-ANN and HHO-ANN is performed, and then, optimal of these two is found by a process known as sensitivity analysis	To optimize the artificial neural network for predicting SCC dataset
[38]	HHOSA	SA is used to optimize HHO and improve its global convergence	To optimize the design parameters of highway guardrail systems
[39]	HHO-FORM	The reliability index is formulated in the HHO-FORM model for a constrained optimization problem. Later, the exterior penalty methodology is used to handle the constraints. HHO determines the optimal reliability index to improve the convergence through the strategy of Levy Flight and population-based mechanism	To reduce the high dimensionality in designing and analyzing risks of structuring in civil engineering
[40]	HHO-minimum cross-entropy (MCE)-MCET-HHO	MCET is used as a fitness function in HHO for determining optimal thresholds to segment an image	To find the optimal configuration of thresholds for image processing
[41]	Hybrid WHHO	WOA is integrated with HHO to improve the convergence rate of HHO in obtaining global optimum	To classify brain tumor using MRI images

Table 4: A summary of HHO variants (continued).

Ref	Variant	Methodology	Objective
[42]	Differential evolutionary adaptive HHO	The HHO is updated by making the Harris Hawk adaptive to decide when it has to move to a random tall tree or when it has to do perching. Also, to improve the exploration ability of HHO, the authors have used the differential evolutionary concept	To improve the exploration ability of HHO for multilevel image thresholding
[43]	Hybrid HHO-SSA	To overcome the HHO's property of stagnating in local optima and prevent immature convergence during exploitation and exploration. The initial solutions generated are divided into two halves in which HHO's exploratory and exploitation are applied to the first half, and SSA's searching stages are utilized to update solutions in the other half. Hence, HHO-SSA chooses the best solution among the two	To address the global optimization problem and find the optimal threshold values
[44]	Hybrid multi-population differential evolution-HHO	The exploitation phase of the HHO is enhanced by chaos. The multi-population strategy is used to improve the ability of global search. Later, differential evolution is used to improve the quality of the solution from the previous stage	To optimize de-noising in satellite images in wavelet domain
[23]	HHO and differential evolution (DE)	Kapur's entropy and Otsu's method are used as fitness functions to find the threshold values of segmentation. The proposed model divides the entire population into two equal parts assigned to DE and HHO algorithms. During the iterative process, both HHO and DE will update each subpopulation position simultaneously	To extract optimal features from images for segmentation of color images, the optimal threshold values of segmentation are found
[22]	Dynamic HHO with mutation	HHO is integrated with a novel dynamic control parameter strategy to avoid the HHO being trapped in the local optimum. A disturbance term is added to update the formulation of the escaping energy formulation. Cosine and sine are integrated to control when the disturbance peak appears. To increase the randomness of the HHO, a Gaussian distribution is adopted	To segment the satellite images
[45]	Hybrid cuckoo search-HHO	To strengthen the HHOs being trapped in local solutions, inaccuracy, inadequate search coverage, and slow convergence, cuckoo search's property of dimension decision strategy, and Gaussian mutation are integrated with the HHO during exploration and exploitation phases	To optimize the parameters in cantilever beam design problem, welded beam design problem, and tension/compression spring design problem
[46]	Hybrid HHO-SSA	SSA is employed to enhance the performance of HHO by acting as a local search for HHO	To enhance the performance of HHO by acting as a local search for HHO
[47]	Hybrid HHO-WOA	HHO is applied to the first half of the population, and WOA is applied to the second half. Hence by integrating WOA with HHO, the exploitation and exploration phases of HHO are enhanced to select the optimal parameters	To predict the values of several parameters such as hydrocarbon, brake thermal efficiency, carbon monoxide, and carbon dioxide based on the data gathered from the experimental setup of the dual-fuel engine by varying injection timings, blends of rice bran biodiesel, engine operating load, and air-fuel ratio

developing the fuzzy HHO (FHHO), the parameters q and E have been fuzzified by the researchers and were introduced in [51]. To achieve this, the Mamdani FIS model is used in this study. In many cases, we find the inputs in the crisp form only. However, the beauty of fuzzy logic is the way it turns common sense, and linguistic descriptions, into a computer-controlled system. Hence, the crisp inputs are transformed into fuzzy forms and then transformed into fuzzy form at the first stage of FIS.

Figure 6(a) shows a system of fuzzy sets for an input with trapezoidal and triangular membership functions. Each fuzzy set spans a region of input (or output) values graphed against membership. Any particular input is interpreted from this fuzzy set, and a degree of membership is obtained. The membership functions should overlap, to allow smooth mapping of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms to allow rules to be applied in a simple manner to



(a)

(b)

Figure 6: Input of the FHHO fuzzy inference system. (a) DF and (b) NU.

express a complex system. In Figure 6(a), the fuzzy variable DF can take four linguistic values, e.g., small, medium, big, and very big. The linguistic values small and very big are represented by trapezoidal-shaped membership functions, whereas the linguistic values medium and big are represented by triangular-shaped membership functions. The inputs provided to Mamdani FIS to generate these two parameters q and E are the input variables df and NU, where df represents the quality of the best solution at the end of a searching iteration and NU represents the number of iterations with the unchanged best solution.

$$\mathrm{df} = \mathrm{CP}(t) - \overline{\mathrm{CP}}. \tag{14}$$

The coverage percentage of the observed area determines the quality of the solution. So, we have the coverage percentage at the iteration number t, which is denoted by CP(t), and the arithmetic mean of the obtained covered percentages is represented by $\overline{\mathrm{CP}}$. NU is also normalized to NU and is given by the following expression:

$$\mathrm{NU} = \frac{\mathrm{NU} - \mathrm{NU}^{\min}}{\mathrm{NU}^{\max} - \mathrm{NU}^{\min}}, \tag{15}$$

where $NU^{\max}$ and $NU^{\min}$ are the maximum and minimum values of NU, respectively. The inputs df and NU are fuzzified using the fuzzification process. df and NU are fuzzy variables, with each one taking the values from the domain of fuzzy granules medium, big, very big, and small (Figure 6). Together, these two variables can take 16 values, and accordingly, 16 rules are framed. The antecedents contain these 16 values, and the consequent provides 16 pairs of values for the fuzzy variables energy and q. Energy takes 8 values (here P denotes positive and N denotes negative) leading to NVB, NB, NM, NS, PS, PM, PB, and PVB and q takes values small, medium, big, and very big. There were 16 fuzzy rules in the rule base; e.g., DF is medium and NU is small and then energy is PM and q is medium.

3D sensors were reallocated in FHHO to supervise cardiomyopathy. The candidate solutions of the n-hawks in the vector form of candidates are $\{x_n, y_n, z_n\}$ in the observed area. So, the dimension of the hawks will be three times the number of sensors required.

Two validation experiments were conducted to evaluate the performance of the FHHO algorithm with different scenarios. Three different scenarios were formulated, and the performance was compared with several other AI algorithms with an iteration of 150 and the number of search agents being 10. It has been observed that FHHO reaches the best coverage value. This is concerning 30 runs. In another experiment, another constraint is added so that the search space reduces in size leading to difficulty in obtaining the best solution. In this case, the number of iterations is fixed at 200. Keeping with the additional constraint, the coverage rate of FHHO was found to be higher. Here also several cases are considered. Wilcoxon signed-rank statistical analysis [52] shows that in comparison with other algorithms, FHHO has a higher coverage percentage. Also, violations of constraint for FHHO are significantly small as the tie's values are minimal.

3.2. Proposed Variants of HHO

3.2.1. Intuitionistic Fuzzy HHO Algorithm. Exact membership functions are used in FIS. In the systems where insufficient information about imprecise concepts is provided, FIS may not be useful. Intuitionistic fuzzy sets (IFSs) provide additional information about imprecise concepts. In the case of fuzzy sets, the membership functions and nonmembership functions are one's complement of each other. Nevertheless, by relaxing this constraint, such as their sum lies in [0, 1], we get the IFS, which has an additional parameter called the hesitation. IFIS has been proposed in many ways. However, they have the problem of overfitting and being unable to deal with complex systems, So, an IFIS that comes over these difficulties has been proposed in [53] by generating the if-then rules through the fuzzy association mining algorithm. They have used the Takagi-Sugeno type of FIS. In fact, the two types of FISs differ in their ways of generating the outputs. The different formulation of outputs leads to different if . . . then rule formulation. The if . . . then rules are automatically extracted from the input data, and logical connectives AND, OR, and NOT are used in these rules in framing implications and aggregations of these rules. The number of if . . . then rules can also be optimized. In the Takagi-Sugeno, FIS has higher computational effectiveness as defuzzification is not required.

Let U be a non-empty set. Then, an IFS A over U is defined as the set of tuples $\{(x, \mu_A(x), \nu_A(x)) | x \in U\}$.

Where $\mu_A, \nu_A\colon U \longrightarrow [0,1]$ such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1, \forall x \in U$. Here, $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ is called the hesitation margin. An IFS A is a fuzzy set if and only if $\pi_A(x) = 0, \forall x \in U$. The $\pi_A(x)$ is called the IF indices. Larger $\pi_A(x)$ values lead to a higher hesitation margin by the decision-maker. Best (or worst) final results are determined by these indices, which are finally in an optimal decision. With the above notations, if y^η is the output of an IF system then with y^μ and y^ν being the outputs of the two FISs F^μ and F^ν of the membership and nonmembership functions, we have the relation as follows:

$$y^n = (1 - \pi_A(x))y^{\mu t} + \pi_A(x)y^\nu. \tag{16}$$

This is the fundamental equation in designing the Takagi-Sugeno type IFIS.

3.2.2. First Version of IFIS. Two FISs FIS$^\mu$ and FIS$^\nu$ are formed using the membership function μ_A and the nonmembership function ν_A. This is called IFIS with composition defuzzification [53, 54]. Here, $x_1, x_2, \ldots x_m$ are input variables and one output variable y^η.

3.3. IFHHO Algorithm (Proposed). We may note that the efficiency of the IFIS over the FIS is established in [55] by taking the real-life application of genetic tuning for predicting financial performance, plant monitoring in [56], and air quality modeling in [57].

4. Applications of HHO

This section discusses various ways in which HHO can be used in ML and engineering applications. Also, the recent state of the art on applications of HHO and its variants in several applications is discussed. The summary of these applications is pictorially summarized in Figure 7.

4.1. Machine Learning Applications. Lefebvre et al. [17] observed the feeding behaviors of different species based on the avian "IQ." In [17, 58, 59], he listed hawks as one of the most intelligent birds. The Parabuteo unicinctus (Harris Hawk) belongs to the same hawk species listed in the intelligent bird category. The Harris Hawk is mainly found in Arizona, USA [60]. The Harris Hawk follows the "surprise pounce" strategy to capture its prey, suggesting that several hawks from different directions attack cooperatively and converge simultaneously to the detected prey (rabbit). The chasing styles of Harris Hawk may adapt to the dynamic nature and prey's escape pattern. The Harris Hawk can also perform a switching strategy, suggesting that if the leader (best hawk) loses its way to the prey while performing a quick dive, a member of the same hawk fleet will continue the chase. This tactic is helpful as it confuses the prey and might exhaust the prey. Inspired by its attack tactics, the Harris Hawk optimization (HHO) method was proposed [8]. The HHO follows mainly two stages to hunt its prey, i.e., exploration and exploitation. The exploration phase is related to discovering its prey, and the exploitation is to decide whether hard or soft besiege should be applied. The hard besiege is applied when the prey is fatigued, and the hawk can perform surprise pounce, while the soft besiege is used when the prey possesses adequate energy to escape. The transition stage models the energy of its prey. We consolidate a brief review of existing works using the HHO method.

4.2. Harris Hawk Optimization for Artificial Neural Networks. HHO method is categorized as one of the metaheuristic approaches, which are extensively used to optimize the hyperparameters of ML algorithms such as artificial neural networks (ANNs), support vector machines (SVMs), and so forth [61, 62]. Some of the works summarizing the use of HHO for ANN parameter optimization are elaborated. Sammen et al. [63] proposed the use of HHO for optimizing the weights in ANN. The method was applied to predict the scour depth for ski jumping. They mainly computed the energy parameter and jumped strength for HHO and weight optimization for ANN. Their analysis showed that the weights optimized through HHO yield the lowest mean absolute percentage error, mean absolute error, and root-mean-square error compared with ANN without metaheuristic, genetic algorithms, and particle swarm optimization. Furthermore, they also showed that HHO achieves the best correlation coefficient and Willmott index values for

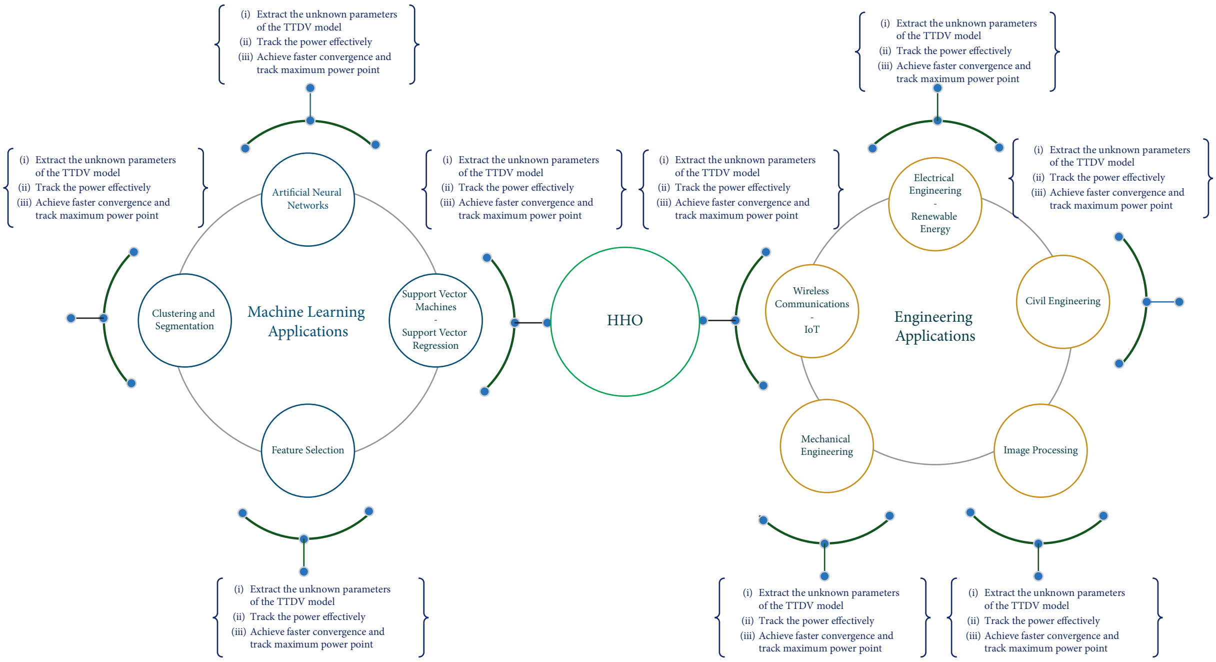

(i) Extract the unknown parameters of the TTDV model
(ii) Track the power effectively
(iii) Achieve faster convergence and track maximum power point
Artificial Neural Networks
Clustering and Segmentation
Machine Learning Applications
Support Vector Machines - Support Vector Regression
Feature Selection
HHO
Electrical Engineering - Renewable Energy
Wireless Communications - IoT
Engineering Applications
Civil Engineering
Mechanical Engineering
Image Processing

Figure 7: Applications of HHO.

scour depth prediction compared with the techniques mentioned above. Similarly, Essa et al. [64] used HHO to optimize ANN weights for the prediction of active and solar stills. Their experimental results revealed that HHO-optimized ANN can provide 53.21% increased productivity concerning active still in comparison with traditional ANN and SVM classifiers. The same conclusion was drawn by Moayedi et al. [37] when using HHO-optimized ANN for soil conditions. They concluded that the HHO-optimized ANN can decrease the mean absolute error by 11.32% and 4.12% for seen and unseen soil conditions compared with traditional ANNs. Moayedi et al. in [29, 65, 66] explored the use of HHO with ANN for predicting landslide vulnerability, bearing ability over soil footing, and slope stability, respectively. The results revealed that the HHO improves the predictive performance in both studies compared with the conventional ANN method. Wei et al. [20] improved the HHO's performance using the Gaussian barebone strategy to optimize kernel extreme learning machines for the prediction of entrepreneurial intention. The Gaussian barebone allows the population to opt for the directions, which leads to faster convergence. Their results showed better and promising results in terms of specificity, sensitivity, accuracy, and Matthews correlation coefficient.

4.3. Clustering and Segmentation. There are a few studies that specifically use the HHO method for the clustering and segmentation process.

Pham et al. [67] proposed the use of HHO for joint power allocation and UAV placement problems. Their study used HHO in conjunction with an efficient user clustering strategy to allocate the power resources for UAV-assisted systems. The comparison was carried out with orthogonal multiple access, non-orthogonal multiple access with visible light communication, non-orthogonal multiple access without pairing, and random user clustering. Experimental results show that the HHO-based clustering outperforms the techniques mentioned earlier for power allocation and UAV placement problems.

Singh [21] used the HHO with chaotic sequences for the application of data clustering. The clustering performance was compared with various ML algorithms on 12 benchmark datasets. It was revealed that the HHO performs better on the majority of the benchmark datasets in terms of statistical tests and performance analysis. Jia et al. [22] employed mutation operator and dynamic control strategy to balance the exploitation and exploration phases in the HHO method. The modified HHO was used to perform segmentation on satellite and oil pollution images. The results were compared with several thresholding methods in terms of Otsu between-class variance, Tsallis entropy, and Kapur's entropy. The results revealed that the HHO-based approach outperforms the thresholding techniques for the specified segmentation task.

Rodrıguez-Esparza et al. [68] used the minimum cross-entropy function to optimize HHO for multilevel image segmentation task. The performance of HHO-based method was compared with fuzzy IterAg and K-means clustering. It was shown that the HHO performed better on the said segmentation task in terms of peak signal-to-noise ratio, feature similarity, and structure similarity. Bao et al. [23] proposed the hybridization of the HHO method by making two equal subpopulations from a complete one and train both the subpopulation using HHO and differential evolution, accordingly. The hybridized HHO was then used to perform multilevel image segmentation on 10 benchmark datasets. Compared with super-pixel segmentation approaches, the results revealed that hybridized HHO performs better in terms of feature similarity and structure similarity, respectively. Wunnava et al. [24] proposed an adaptive HHO method by clipping the range and constraining the escape energy. Furthermore, they allowed the method to decide whether to opt for average fitness value and approach a tall tree or roost and other hawks in the family. The adaptive HHO was used for multilevel image segmentation on the Berkeley segmentation dataset and was compared with well-known segmentation methods. Experimental results showed that the adaptive HHO achieved state-of-the-art segmentation performance.

4.4. Feature Selection. HHO has been used extensively for the feature selection process to optimize the parameters for classification methods [69, 70]. Ismael et al. [25] proposed to improve the HHO method by employing an opposition-based learning approach (OBL). The OBL generates a solution for meta-heuristic algorithms through an adversarial learning approach. The OBL-HHO was applied to select the informative features from the feature space in conjunction with support vector regression. Their experimental results proved that OBL-HHO achieved better results in comparison with cross-validation and grid search methods.

Sihwail et al. [26] proposed an improved HHO using elite opposition-based learning (EOBL). The EOBL stacks upon OBL by selecting the fittest individual that would direct the population towards global minimum. The improved HHO was used to select informative features from the feature space and was compared to well-known optimizers such as slime mould, butterfly optimization, whale optimization, ant lion optimization, and grasshopper optimization. Experimental results revealed that the improved HHO performed better in terms of fitness value, accuracy, and feature selection compared with the techniques mentioned earlier.

Abdel-Basset et al. [27] modified the HHO method using simulated annealing and bitwise operations and termed it as HHOBSA. The bitwise operations help HHO improve the feature selection process, whereas the simulated annealing helps HHO find the global minimum. It was revealed in their experimental results that HHOBSA performed better on 19 artificial and 24 standard datasets in comparison with the classical HHO.

Elgamal et al. [28] also used simulated annealing to improve HHO. In addition, they also introduced chaotic maps instead of random variables to achieve global optimum. The chaotic HHO was used for selecting the most informative features to train using K-nearest neighbor for the classification task. The results reveal that the chaotic HHO achieved the best accuracy against several optimization algorithms.

Zhang et al. [13] improved the HHO algorithm by adding the Salp swarm optimization, which adjusts the populations and uses greedy selection to update the agent. The Salp swarm optimization is also used to maintain a balance between the exploitation and exploration phases. The improved HHO is applied in a binary tree strategy to select the informative features. Their proposed method was used in conjunction with K-nearest neighbor and is compared with classical swarm-based approaches. Experimental results show that their improved HHO performs better on the optimization functions and classification tasks. Thaher and Arman [71] also used binary tree-like structure with HHO to select the informative features. They trained the selected features with three classification algorithms: linear discriminant analysis, decision trees, and K-nearest neighbor on SFP classification datasets. Their results show that the selected features from HHO, when trained with linear discriminant analysis, achieved the best accuracy. Houssein et al. [72] proposed the use of HHO to find the chemical compound activities and descriptor selection followed by the training using SVM and K-nearest neighbor, accordingly. The experiments were carried out to compare the HHO-SVM and HHO-K-nearest neighbor-based methods with the classical optimization algorithms on QSAR biodegradation and monoamine oxidase datasets. The HHO-SVM provides superior results on both datasets in comparison with all other algorithms.

4.5. Support Vector Machines and Support Vector Regression. Like the studies with ANN, the HHO method has been used to perform parameter optimization of SVM and support vector regression techniques. For example, Tikhamarine et al. [65] presented a rainfall-runoff prediction technique using least-squares SVM, ANN, and multiple linear regression optimized through the HHO method. Their study performed a detailed comparative analysis and concluded that the least-squares SVM optimized through HHO achieves the best precision values compared with the other classification algorithms. Fu et al. [29] suggested that the control formula in HHO is linear, suggesting that the optimization at the start of the process focuses more on exploration, whereas the focus diverts to the exploitation at a later stage. They proposed an improved hybrid differential evolution HHO by proposing nonlinear control formula that balances the exploitation and the exploration throughout the convergence process. They used the improved version of HHO to optimize phase space reconstruction and kernel extreme learning machine parameters for wind speed forecasting and showed that the improved version achieves better results than the HHO optimized classifiers.

Malik et al. [66] performed a comparative analysis for water stream flow prediction by employing support vector regression optimized through multiple metaheuristic approaches that include Bayesian optimization, particle swarm optimization, HHO, spotted hyena optimizer, multi-verse optimizer, and ant lion optimization. The methods were evaluated in terms of the Willmott index, correlation coefficient, scatter index, and root-mean-square error. They concluded that the support vector regression optimized through HHO yields the best results among all optimization techniques. Shao et al. [30] added periodic mutations for enhancing swarm diversity in the basic HHO method and termed it as vibrational HHO. They used the vibrational HHO to optimize SVM parameters for roll-bearing fault diagnosis. The vibrational HHO outperformed 23 benchmark functions with a faster convergence rate.

Furthermore, the HHO method has been used for various other tasks such as structural design optimization of vehicle components [73], manufacturing optimization problems [74], polluted distribution [75], heat sink design for micro-channel [76], photovoltaic cell optimization [14], and many more.

4.6. Engineering Applications, Electrical Engineering and Renewable Energy. Due to the numerous installations of photovoltaic (PV) power plants, the accurate modeling of PV modules is the need of the hour. Qais [77] proposed a three-diode photovoltaic (TDPV) model for the accurate modeling of photovoltaic losses. The authors have employed the HHO algorithm to extract the unknown parameters of the TTDV model. The authors have utilized datasheet values of PV modules that the industrialists provide to identify 4 of 9 unknown parameters of the TDPV model, and the HHO algorithm is employed to extract the remaining five parameters. The experimental results proved that the proposed HHO-based TDPV model performed better than the existing models. In a similar work, Hussein [31] presented a boosted HHO algorithm to estimate the parameters efficiently for a single-diode PV model. The proposed boosted HHO algorithm enhances the HHO algorithm by integrating it with the exploratory phase of the flower pollination algorithm and mutation step of differential evolution. Chen [14] proposed the HHO-based model for the estimation of solar cell models for single diode, double diode, and PV modules. The proposed HHO-based model is based on opposition-based exploratory strategy and chaotic drifts to identify the optimal agents and unknown PV model parameters.

Mansoor et al. [78] proposed an MMPT controller based on HHO to track the power effectively in solar-powered PV systems in all the weather conditions. The HHO algorithm is proven to achieve faster convergence and track maximum power point. Proton exchange membrane fuel cell (PEMFC) is one of the most importantly environmentally friendly energy sources. Mossa et al. [79] have used a hybrid of atom search optimization and HHO to extract the PEMFC's unknown parameters. The proposed hybrid algorithm is tested on 3 different PEMFC stacks, 250W stack, 500W SR-12 PEM stack, and BCS 500-W PEM stack, respectively, in several operating conditions. In a similar work, Menesy et al. [35] applied a chaotic HHO algorithm to accurately estimate the proton exchange membrane fuel cell's operating parameters that can mimic and simulate its electrical performance. The authors have used an enhanced HHO with 10 chaotic functions to avoid local optima trapping of conventional HHO.

Liu et al. [32] proposed an improved HHO for simulating an efficient PV system and to extract the unknown parameters. To prevent the HHO from falling into local optima, the authors have used a crisscross optimizer and Nelder–Mead simplex algorithm to improve individuals' searching capabilities for achieving a faster convergence rate. In a similar work, Yousri et al. [34] have proposed a modified HHO to relieve the PV systems from the issue of mismatch power loss problems resulting due to the phenomenon of partial shading. The modified HHO provides the optimal reconfiguration pattern for the switching matrix for maximizing the power generated from the array.

The safe operation and rational dispatching of a power system depend on the accurate prediction of wind speed. Fu et al. [33] have employed a hybrid of HHO and GWO for optimizing the parameters of phase space reconstruction and kernel-based extreme learning machine algorithms to predict the wind speeds accurately. The end users can communicate with the operators of the grid through a demand-side management program. This program can help the customers to take the assistance of grid operators to reduce the power consumption of the utilities during peak hours by smartly managing the load. Mouassa et al. [80] employed HHO to schedule energy in smart homes. Abdel Aleem et al. [75] have employed HHO to reduce the harmonic overloading levels of components based on the frequency with optimal planning of C-type harmonic filter, which is resonance free in a non-sinusoidal distribution system. Selim et al. [36] have employed HHO and multi-objective HHO to find the location of distribution generation optimally in a radial distribution system to minimize the voltage deviation and total active power loss and also to increase the voltage stability index under several operational constraints.

4.7. Civil Engineering. Many researchers have used the HHO algorithm to solve some of the critical research problems in the civil engineering (CE) domain, such as predicting the stability of the soil slopes accurately, optimizing structural design problems, air pollutant forecasting, and predicting the blast-induced ground vibration. The rest of this subsection presents recent research works that solved several CE problems.

One of the essential parameters to estimate the settlement of soil layers in CE applications is soil compression coefficient (SCC). Moayedi et al. [37] have proposed a hybrid of HHO and grasshopper optimization algorithm (GOA) to optimize the artificial neural network (ANN) for predicting SCC. The authors have used the ensemble of GOA and HHO to tune and find the optimal parameters of the ANN. The dataset is then trained and tested by the proposed model to predict SSC. To further improve the ANN-GOA prediction model, the authors have proposed a hybrid GOA-HHO algorithm for the prediction of SSC. Another challenging issue in CE is predicting the stability of the soil slopes (SSs). Moayedi et al. [81] proposed the HHO-based convolutional multilayer perceptron model to predict safety factor in constructions with rigid foundations based on SS conditioning factors. The HHO algorithm is used in this work to adjust the computational weights of the SS abovementioned factors.

Guardrail systems are designed to absorb the energy generated by vehicles driven on the roads to increase the safety of motorways. Enes Kurtulus et al. [38] proposed a hybrid HHO-simulated annealing (HHOSA) for optimizing real-world structural design problems. The HHOSA algorithm is used in this work to optimize the design parameters of highway guardrail systems.

Due to the increase in industrialization and motor vehicles, air pollution is increasing daily in several parts of the globe rapidly. Air pollution is affecting the environment badly and endangering all kinds of species. Global warming is also the result of an increase in air pollution. Even though several researchers have conducted several studies, they had significant deficiencies such as insufficient initial parameters and neglecting the significance of predictive stability, which affected the performance of air pollution forecast models. Du et al. [19] proposed the HHO-based extreme learning ML model to overcome the deficiencies of existing models.

Blasting and drilling are the conventional methods for fragmentation of the rock mass in mines as they are efficient and cheap. However, the vibration generated by the blasts can damage the surrounding structures and rock. To control the damage induced by blasting, an efficient prediction model must be designed for controlled blasting in mines. Yu et al. [82] have proposed the HHO-based random forest algorithm to predict the vibration induced by the blast. HHO is used for tuning the parameters of the random forest algorithm. To increase the samples by randomly changing the values of the attributes, the authors have used the Monte Carlo simulation method.

A spillway, which is a structure that regulates discharge flowing from massive hydraulic structures such as dams, plays a pivotal role in the safety of the dams. A spillway also dissipates the extra energy of water with the help of still basins. However, the high flow velocity on the spillway may lead to a serious problem known as bed scouring, resulting in spillway failure and soil erosion. Sammen et al. [63] have proposed a hybrid ANN-HHO model to predict the ski jump spillway's scour depth downstream. The HHO algorithm is used in this work to tune the parameters of the ANN. In a similar work, Khalifeh et al. [83] have proposed a model based on HHO to optimize the water distribution network in Homashahr, Iran, from September 30, 2018, to October 30, 2019. In this article, the researchers have integrated HHO with EPANETE 2, a water distribution network analyzing software. The EPANET 2 software analyzes the velocity of flow for every pipe and the pressure of every node. The diameter of the pipes is the main optimization parameter used in this work. The main objective is to design optimal water distribution systems. The HHO algorithm is used in this work to choose the optimal parameters.

The effects of several uncertainties, such as load, dimension, and material properties, are to be considered in the design and analyzing the risk of structures in CE. Prediction of failure probability in reliability problems with high dimensions is a significant research problem. The first-order reliability method (FORM) is a popular method used in CE for reliability analysis. However, FORM suffers from divergence or convergence issues when dealing with high dimensions with nonlinear limit state function. To address the issue of high dimensionality, Zhong et al. [39] presented an improved FORM based on the HHO algorithm. The HHO algorithm is used to choose optimal algorithmic parameters for FORM.

4.8. Image Processing. Several nature-inspired computing algorithms including HHO have been efficiently used by researchers in solving many problems in image processing, such as digital mammogram segmentation, image thresholding, and removing the noise from the images [84–86]. The recent state-of-the-art works by researchers on applications of HHO in image processing are discussed in the remainder of this subsection. One of the crucial phases in image processing is segmentation, as it simplifies image representation through which it facilitates the analysis. Rodrıguez-Esparza et al. [40] have proposed a HHO-based methodology for multilevel segmentation of images. Minimum cross-entropy thresholding (MCET) is used as a fitness function for HHO in this work. To find the optimal configuration of thresholds for image processing, the HHO algorithm is used in this work.

One of the drawbacks of the traditional HHO algorithm is its limited exploration ability, as it gets completely exhausted when the escape energy is zero. Wunnava et al. [42] have proposed a novel differential evolutionary adaptive HHO (DEAHHO) to address the issue mentioned above. The authors have modified the exploration phase of HHO within the range of [2,0] to limit the escape energy. The authors have also updated the HHO by making the Harris Hawk adaptive to decide when it has to move to a random tall tree or when it has to do perching. Also, to improve the exploration ability of HHO, the authors have used the differential evolutionary concept. Multilevel image thresholding methods based on 1-D histograms have been using Masi entropic function recently. However, the problem with the current approaches is the missing of contextual information in the 1D formulation. To address this issue, a novel 2D practical Masi entropy function is proposed by the authors. DEAHHO has been applied in the proposed 2D practical Masi entropy-based multilevel image thresholding while segmenting the images. To validate the DEAHHO method, the authors have considered 23 popularly used benchmark test functions. For experimentation purposes, 500 images are obtained from the renowned Berkeley segmentation dataset. PSNR, FSI, and SSI metrics are used to evaluate the performance of the proposed method. The proposed method yielded better results when compared to other state-of-the-art algorithms. In a similar work, Wunnava et al. [24] have used the abovementioned updated HHO along with an improved 2D grey gradient method to preserve the edge information of images with high magnitude. The same dataset used by [42] is used in this work to evaluate the proposed method. Designing an efficient automatic brain tumor classification model is the need of the hour as the precision of existing classification models is not satisfactory. Rammurthy and Mahesh [41] proposed a hybrid WOA-HHO (WHHO)-based deep CNN model [87] for classifying brain tumor using MRI images. In this work, rough set theory and cellular automata are used to perform the segmentation of MRI images. The features extracted from the MRI images are variance, mean, local optical-oriented pattern, kurtosis, and tumor size. To tune the parameters of the deep CNN model, the WHHO optimization algorithm is used. The datasets used to train the proposed model are simulated BRATS dataset (dataset 3) and BRATS dataset (dataset 4). The metrics used to evaluate the proposed model are accuracy, specificity, and sensitivity.

HHO algorithm may be struck due to local optima problem and may suffer immature convergence during the exploitation and exploration phases. To address these issues in HHO, Abd Elaziz et al. have proposed [43] a hybrid HHO-SSA optimization algorithm for multilevel image segmentation problems. The proposed model addresses the global optimization problem and also helps in finding optimal threshold values. In the proposed work, the first initial set of solutions is generated. Later, these solutions are divided into two halves. The exploitative and exploratory phases of HHO will be applied to the first half of the solutions, whereas in the second half, the searching stages of SSA will be applied. Later, the optimum solutions from these halves are chosen to continue the rest of the iterative process. 11 natural gray-scale images and IEEE CEC 2005 benchmark functions are used to perform experiments. The measures used to evaluate the proposed method are average fitness value, worst fitness value, best fitness value, and the standard deviation. The experimentation results proved that the hybrid HHO-SSA performs better than the individual algorithms and several other popular algorithms. Removal of noise from images is essential during image processing as the further procedure is not possible with noisy images. Researchers have recently attempted to improve the quantitative and qualitative results by removing the noise from the images. However, the attempts from researchers could not preserve the image quality after the application of de-noising methods. Golilarz et al. [88] used the HHO algorithm for optimal image denoising to tune the parameters of the thresholding functions. In this work, HHO is used to obtain the best-thresholded values for wavelet coefficients before applying the inverse wavelet transforms in the first stage. In the next stage, the authors have presented an improved adaptive generalized Gaussian distribution threshold algorithm that is a data-driven function that has an adaptive threshold value. The proposed function can fit all kinds of images without the usage of a shape tuning parameter. The authors have used six satellite images for experimentation. The results proved that the proposed model achieved better accuracy and less time to process the data when compared to traditional models. In a similar work, Golilarz et al. [44] proposed a hybrid multi-population differential evolution-HHO (CMDHHO) algorithm for optimizing de-noising in satellite images in the wavelet domain. The experimental results have proved that the proposed CMDHHO algorithm yielded better quantitative and qualitative results when compared to several other optimization algorithms and thresholding neural network approaches. CMDHHO also improved the processing time and also proved to be computationally efficient. PSNR and mean SSI are the attributes used to evaluate the performance of the de-noising algorithms considered in this work.

Implementation of multilevel thresholding for color images is time-consuming and complex as the information that has to be processed is high, and also, the number of thresholds is more for color images. Bao et al. [23] have proposed a hybrid algorithm based on HHO for image segmentation of color images. The proposed hybrid model is a combination of HHO and differential evolution (DE). This hybrid algorithm, HHO-DE, extracts optimal features from images for the segmentation of color images. Kapur's entropy and Otsu's method are used as fitness functions to find the threshold values of segmentation. The proposed model divides the entire population into two equal parts assigned to DE and HHO algorithms. During the iterative process, both HHO and DE will update each subpopulation's position simultaneously. The proposed model is implemented on 10 benchmark images. The proposed model is then compared with 7 state-of-the-art methods. For evaluating the performance of the algorithms, 5 measures, FSI, SSI, PSNR, standard deviation, and average fitness values, are used in this work. The comparative analysis proves the superiority of the proposed HHO-DE algorithm. In a similar work, Jia et al. [89] proposed the application of the HHO algorithm for tuning of parameters by pulse coupled neural network (PCNN) method for the segmentation of medical images. The proposed HHO-PCCN method reduced the number of parameters of PCNN without affecting the effect of segmentation. In a similar work, Jia et al. [22] proposed a novel dynamic HHO algorithm with a mutation mechanism for segmentation of satellite images. Compared with the traditional HHO algorithm, the proposed dynamic HHO with a mutation mechanism can overcome the problem of falling into local optimum, increasing the searching capacity. Landslides are one of the most devastating environmental threats that can cause substantial financial and physical damage worldwide. Predicting the landslides reliably can save lives and also reduce damages to property. Bui et al. [90] have used the HHO-based ANN to analyze landslide susceptibility in Western Iran. In addition, HHO is used to tune the parameters of ANN.

4.9. Mechanical Engineering. Essa et al. [64] have proposed an HHO-ANN model to predict the productivity of active solar still, which is used to extract fresh drinkable water from water with a high concentration of salt (briny water). The HHO algorithm is used in this work to select optimal parameters of ANN. The experiments were conducted at the Faculty of Engineering, Kafrelsheikh University, Kafrelsheikh. Song et al. [45] have proposed a hybrid cuckoo search-HHO algorithm to solve three classical mechanical engineering problems: five-stage cantilever beam design problem, welded beam design problem, and tension/compression spring design problem. The cuckoo search-based HHO algorithm is used to optimize the parameters in the engineering abovementioned problems. The proposed algorithms achieved better results when compared with other metaheuristic algorithms.

Friction stir welding has proved to be efficient in welding materials when compared to traditional fusion welding methods. Shehabeldeen et al. [91] have proposed an adaptive neuro-fuzzy inference system integrated model integrated with HHO to predict the mechanical properties of friction stir welding. The HHO algorithm is used to find the optimal parameters of the adaptive neuro-fuzzy inference system and find optimal conditions of the friction stir welding process. Micro-channel heat sinks are one of the popular methods to remove heat and cool the integrated circuits in electronic devices. Entropy generation is a negative factor for micro-channel heat sink systems. Abbasi et al. [76] applied HHO optimization to minimize the entropy generation in micro-channel heat sinks.

The grinding process is a basic shaping method that is used to sharpen weapons and piercing cutting. It is used in the manufacturing of various tools. Providing optimal surface quality is the aim of the grinding process. The optimization of production rate and production cost is crucial in the grinding process to ensure surface quality. Yıldız et al. [74] have used HHO, GOA, and multi-verse optimization algorithm (MVO) to optimize the processing parameters in grinding operations. Jouhari et al. [46] proposed a hybrid HHO-SSA algorithm to address scheduling problems in unrelated parallel machines. In this study, SSA is employed to enhance the performance of HHO by acting as a local search for HHO. As a result, the proposed hybrid algorithm resulted in improved performance and reduced computational time.

To solve shape optimization problems in manufacturing industry, Yıldız et al. [74] employed HHO, SSA, GAO, and dragonfly algorithm. In this work, HHO, SSA, GAO, and dragonfly algorithms are applied for optimizing the shape of the vehicle brake pedal. Singh [47] proposed a hybrid HHO-WOA-based ANN to predict emission properties of a single-cylinder direct injection diesel engine. The proposed hybrid HHO-WOA algorithm optimized ANN to predict the values of several parameters such as hydrocarbon, brake thermal efficiency, carbon monoxide, and carbon dioxide.

4.10. Wireless Communications and Internet of Things. Fiber wireless (FiWi) integrates wireless and optical networks. It can reduce complexity and cost by combining wireless and fiber networks, such as mobility of wireless networks and large bandwidth availability through optical networks [92, 93]. To place multiple optical units at optimal locations in FiWi, Singh and Prakash [94] used the HHO algorithm.

For many applications of wireless sensor networks (WSNs) such as intrusion detection, road traffic tracking, and oil and gas explorations, location information of the sensors is vital so that communication is not disrupted. Bhat and Venkata [95] proposed an HHO area minimization algorithm to improve the location accuracy of sensors in irregular WSN topologies. Area minimization is used in this work to minimize the search area. In a similar work, Houssein et al. [96] have used HHO to identify the ideal location for placing sink nodes in a large-scale WSN. The authors have used HHO to identify the optimal location for a sink node in WSN and used Prim's shortest path algorithm for reconstructing the WSN by choosing minimum transmission paths. In a similar work, the authors in [97] used HHO for selecting an optimal cluster head in an IoT-based network to choose the optimal routing schemes and reduce energy consumption in the IoT network. The simulation results proved the superiority of the proposed approach when compared with other recent models.

The intelligent reflecting surface is one of the techniques that can be used to provide cost-effective and green solutions to enhance the performance of WSN through the smart configuration of propagation of the signals. For example, Xu et al. [98] have employed HHO to maximize the power of the received signal by optimizing the transmit beam forming at the access point and intelligent reflecting surface's reflection coefficient.

5. Conclusions

In this work, we have provided a comprehensive survey on the fundamentals, variants, and applications of the HHO optimizer. Based on the existing reviews, we discovered that HHO is used in various engineering and ML applications such as clustering, classification, and feature selection. Some researchers demonstrated that HHO could efficiently solve critical optimization problems such as pattern recognition, image classification, and unconstrained optimization. At the same time, some of them provided sufficient modifications towards more harmonized exploration and exploitation trends based on the problem's nature. Furthermore, some recent works show that the hybrid variant of HHO has a faster convergence rate, optimal computational accuracy, and greater efficiency than existing metaheuristic algorithms. In our research, we discovered that HHO is efficient in all of the test problems. In the future, the work can be utilized as a guide over the recent developments on the HHO to have a better view of the current state of research on this well-known method. Also, it can be used to categorize different variants of HHO and categorize which variant is suitable for which operation, along with their advantages and disadvantages. This matter can help researchers in addressing appropriate potential problems.

Data Availability

No data are used in this work.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to express their gratitude to Ali Asghar Heidari from the School of Surveying and Geospatial Engineering, University of Tehran and School of Computing, National University of Singapore, for his advice and guidance.

References

[1] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: algorithms and systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[2] Q. V. Pham, D. C. Nguyen, S. Mirjalili et al., "Swarm intelligence for next-generation networks: recent advances and applications," *Journal of Network and Computer Applications*, vol. 191, Article ID 103141, 2021.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," *International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[5] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming*, Springer, Berlin, Germany, 1998.

[6] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[7] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[8] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[9] Q.-V. Pham, T. Huynh-The, M. Alazab, J. Zhao, and W.-J. Hwang, "Sum-rate maximization for UAV-assisted visible light communications using NOMA: swarm intelligence meets machine learning," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 375–410 387, 2020.

[10] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection," *Cognitive Computation*, vol. 12, no. 1, pp. 150–175, 2020.

[11] N. A. Golilarz, A. Addeh, H. Gao et al., "A new automatic method for control chart patterns recognition based on ConvNet and Harris Hawks meta heuristic optimization algorithm," *IEEE Access*, vol. 7, 2019.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[13] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," *Structure*, vol. 25, p. 26, 2020.

[14] H. Chen, S. Jiao, M. Wang, A. A. Heidari, and X. Zhao, "Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts," *Journal of Cleaner Production*, vol. 244, Article ID 118778, 2020.

[15] U. B. Tayab, A. Zia, F. Yang, J. Lu, and M. Kashif, "Short-term load forecasting for microgrid energy management system using hybrid HHO-FNN model with best-basis stationary wavelet packet transform," *Energy*, vol. 203, Article ID 117857, 2020.

[16] H. M. Alabool, D. Alarabiat, L. Abualigah, and A. A. Heidari, "Harris Hawks Optimization: A Comprehensive Review of Recent Variants and Applications," *Neural Computing and Applications*, vol. 33, pp. 1–42, 2021.

[17] L. Lefebvre, P. Whittle, E. Lascaris, and A. Finkelstein, "Feeding innovations and forebrain size in birds," *Animal Behaviour*, vol. 53, no. 3, pp. 549–560, 1997.

[18] K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, S. Bhattacharya, P. Hegde, and S. Singh, "Energy optimization for green communication in iot using Harris hawks optimization," *IEEE Transactions on Green Communications and Networking*, vol. 6, 2022.

[19] P. Du, J. Wang, Y. Hao, T. Niu, and W. Yang, "A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily pm2.5 and pm10 forecasting," *Applied Soft Computing*, vol. 96, Article ID 106620, 2020.

[20] Y. Wei, H. Lv, M. Chen et al., "Predicting entrepreneurial intention of students: an extreme learning machine with Gaussian barebone harris hawks optimizer," *IEEE Access*, vol. 8, pp. 76 841–876 855, 2020.

[21] T. Singh, "A chaotic sequence-guided Harris hawks optimizer for data clustering," *Neural Computing & Applications*, vol. 32, no. 23, pp. 17 789–817 803, 2020.

[22] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation," *Remote Sensing*, vol. 11, no. 12, p. 1421, 2019.

[23] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76 529–576 546, 2019.

[24] A. Wunnava, M. K. Naik, R. Panda, B. Jena, and A. Abraham, "An adaptive Harris hawks optimization technique for two dimensional grey gradient based multilevel image thresholding," *Applied Soft Computing*, vol. 95, Article ID 106526, 2020.

[25] O. M. Ismael, O. S. Qasim, and Z. Y. Algamal, "Improving Harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning," *Journal of Chemometrics*, vol. 34, no. 11, 2020.

[26] R. Sihwail, K. Omar, K. A. Z. Ariffi, and M. Tubishat, "Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection," *IEEE Access*, vol. 8, pp. 121 127–121 145, 2020.

[27] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A Hybrid Harris Hawks Optimization Algorithm with Simulated Annealing for Feature Selection," *Artificial Intelligence Review*, vol. 54, 2020.

[28] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field," *IEEE Access*, vol. 8, pp. 186 638–186 652, 2020.

[29] W. Fu, K. Zhang, K. Wang, B. Wen, P. Fang, and F. Zou, "A hybrid approach for multi-step wind speed forecasting based on two-layer decomposition, improved hybrid DE-HHO optimization and KELM," *Renewable Energy*, vol. 164, pp. 211–229, feb 2021.

[30] K. Shao, W. Fu, J. Tan, and K. Wang, "Coordinated Approach Fusing Time-Shift Multi Scale Dispersion Entropy and Vibrational Harris Hawks Optimization Based SVM for Fault Diagnosis of Rolling Bearing," *Measurement*, vol. 173, Article ID 108580, 2020.

[31] H. M. Ridha, A. A. Heidari, M. Wang, and H. Chen, "Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models," *Energy Conversion and Management*, vol. 209, Article ID 112660, 2020.

[32] Y. Liu, G. Chong, A. A. Heidari et al., "Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models," *Energy Conversion and Management*, vol. 223, Article ID 113211, 2020.

[33] W. Fu, K. Wang, J. Tan, and K. Zhang, "A composite framework coupling multiple feature selection, compound prediction models and novel hybrid swarm optimizer-based synchronization optimization strategy for multi-step ahead

short-term wind speed forecasting," *Energy Conversion and Management*, vol. 205, Article ID 112461, 2020.

[34] D. Yousri, D. Allam, and M. B. Eteiba, "Optimal photovoltaic array reconfiguration for alleviating the partial shading influence based on a modified Harris hawks optimizer," *Energy Conversion and Management*, vol. 206, Article ID 112470, 2020.

[35] A. S. Menesy, H. M. Sultan, A. Selim, M. G. Ashmawy, and S. Kamel, "Developing and applying chaotic Harris hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks," *IEEE Access*, vol. 8, pp. 1146–1159, 2019.

[36] A. Selim, S. Kamel, A. S. Alghamdi, and F. Jurado, "Optimal placement of dgs in distribution system using an improved Harris hawks optimizer based on single and multi objective approaches," *IEEE Access*, vol. 8, pp. 52 815–852 829, 2020.

[37] H. Moayedi, M. Gör, Z. Lyu, and D. T. Bui, "Herding behaviors of grasshopper and Harris hawk for hybridizing the neural network in predicting the soil compression coefficient," *Measurement*, vol. 152, Article ID 107389, 2020.

[38] E. Kurtulus, A. R. Yıldız, S. M. Sait, and S. Bureerat, "A novel hybrid Harris hawks-simulated annealing algorithm and rbf-based metamodel for design optimization of highway guardrails," *Materials Testing*, vol. 62, no. 3, pp. 251–260, 2020.

[39] C. Zhong, M. Wang, C. Dang, W. Ke, and S. Guo, "First-order Reliability Method Based on harris Hawks Optimization for High-Dimensional Reliability Analysis," *Structural And Multidisciplinary Optimization*, vol. 62, 2020.

[40] E. Rodrıguez-Esparza, L. A. Zanella-Calzada, D. Oliva et al., "An Efficient harris Hawks-Inspired Image Segmentation Method," *Expert Systems with Applications*, vol. 155, Article ID 113428, 2020.

[41] D. Rammurthy and P. Mahesh, "Whale harris Hawks Optimization Based Deep Learning Classifier for Brain Tumor Detection Using Mri Images," *Journal of King Saud University-Computer and Information Sciences*, In press, 2020.

[42] A. Wunnava, M. K. Naik, R. Panda, B. Jena, and A. Abraham, "A Differential Evolutionary Adaptive harris Hawks Optimization for Two Dimensional Practical Masi Entropy-Based Multilevel Image Thresholding," *Journal of King Saud University Computer and Information Sciences*, 2020.

[43] M. Abd Elaziz, A. A. Heidari, H. Fujita, and H. Moayedi, "A Competitive Chain-Based harris Hawks Optimizer for Global Optimization and Multilevel Image Thresholding Problems," *Applied Soft Computing*, vol. 95, Article ID 106347, 2020.

[44] N. A. Golilarz, M. Mirmozaffari, T. A. Gashteroodkhani et al., "Optimized wavelet-based satellite image de-noising with multi-population differential evolution-assisted Harris hawks optimization algorithm," *IEEE Access*, vol. 8, p. 133, 2020.

[45] S. Song, P. Wang, A. A. Heidari et al., "Dimension Decided harris Hawks Optimization with Gaussian Mutation: Balance Analysis and Diversity Patterns," *Knowledge-Based Systems*, vol. 215, Article ID 106425, 2020.

[46] H. Jouhari, D. Lei, M. A. A. Al-qaness et al., "Modified Harris hawks optimizer for solving machine scheduling problems," *Symmetry*, vol. 12, no. 9, p. 1460, 2020.

[47] S. Singh, A. Jain, and S. K. Mahla, "Sampled-data model validation: an algorithm and experimental setup of dual fuel ic engine," *Fuel*, vol. 279, Article ID 118517, 2020.

[48] L. A. Zadeh, "Fuzzy sets," *Advances in Fuzzy Systems - Applications and Theory*, pp. 394–432, 1996.

[49] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science Inc, Amsterdam, Netherlands, 1985.

[50] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[51] W. Ding, M. Abdel-Basset, K. A. Eldrandaly, L. Abdel-Fatah, and V. H. C. de Albuquerque, "Smart Supervision of Cardiomyopathy Based on Fuzzy harris Hawks Optimizer and Wearable Sensing Data Optimization: A New Model," *IEEE Transactions on Cybernetics*, vol. 51, 2020.

[52] S. Lathuiliere, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2065–2081, 2019.

[53] V. Olej and P. Hájek, "If-inference systems design for prediction of ozone time series: the case of Pardubice microregion," *Artificial Neural Networks - ICANN 2010*, vol. 6352, pp. 1–11, 2010.

[54] O. Montiel, O. Castillo, P. Melin, and R. Sepulveda, "Mediative fuzzy logic: a new approach for contradictory knowledge management," *Forging New Frontiers: Fuzzy Pioneers II*, vol. 218, pp. 135–149, 2008.

[55] P. Hajek and V. Olej, "Intuitionistic fuzzy inference system with genetic tuning for predicting financial performance," in *Proceedings of the 2018 3rd International Conference on Computational Intelligence and Applications (ICCIA)*, pp. 81–86, IEEE, Hong Kong, China, July 2018.

[56] O. Castillo and P. Melin, "A new method for fuzzy inference in intuitionistic fuzzy systems," in *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2003*, pp. 20–25, IEEE, IL, USA, July 2003.

[57] V. Olej and P. Hajek, "Intuitionistic hierarchical fuzzy inference systems design for air quality modeling," in *Proceedings of the 5th International Conference on Energy, Environment, Ecosystems and Sustainable Development,(EEESD'09)*, pp. 89–94, Vouliagmeni Beach, Greece, September 2009.

[58] D. Sol, R. P. Duncan, T. M. Blackburn, P. Cassey, and L. Lefebvre, "Big brains, enhanced cognition, and response of birds to novel environments," *Proceedings of the National Academy of Sciences*, vol. 102, no. 15, pp. 5460–5465, 2005.

[59] F. Dubois, L. A. Giraldeau, I. M. Hamilton, J. W. A. Grant, and L. Lefebvre, "Distraction sneakers decrease the expected level of aggression within groups: a game-theoretic model," *The American Naturalist*, vol. 164, no. 2, pp. E32–E45, 2004.

[60] J. C. Bednarz, "Cooperative hunting harris' hawks (parabuteo unicinctus)," *Science*, vol. 239, no. 4847, pp. 1525–1527, 1988.

[61] M. Alazab and R. Broadhurst, "Spam and criminal activity," *Trends and issues in crime and criminal justice*, vol. 526, pp. 1–20, 2016.

[62] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, "Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace," *Future Generation Computer Systems*, vol. 117, pp. 205–218, 2021.

[63] S. S. Sammen, M. A. Ghorbani, A. Malik et al., "Enhanced artificial neural network with harris hawks optimization for predicting scour depth downstream of ski-jump spillway," *Applied Sciences*, vol. 10, no. 15, p. 5160, 2020.

[64] F. A. Essa, M. Abd Elaziz, and A. H. Elsheikh, "An enhanced productivity prediction model of active solar still using artificial neural network and Harris Hawks optimizer," *Applied Thermal Engineering*, vol. 170, Article ID 115020, 2020.

[65] Y. Tikhamarine, D. Souag-Gamane, A. N. Ahmed et al., "Rainfall-runoff modelling using improved machine learning methods: harris hawks optimizer vs. particle swarm

optimization," *Journal of Hydrology*, vol. 589, Article ID 125133, 2020.

[66] A. Malik, Y. Tikhamarine, D. Souag-Gamane, O. Kisi, and Q. B. Pham, "Support vector regression optimized by meta-heuristic algorithms for daily streamflow prediction," *Stochastic Environmental Research and Risk Assessment*, vol. 34, no. 11, pp. 1755–1773, 2020.

[67] Q.-V. Pham, N.-N. Dao, T. Huynh-The, J. Zhao, and W.-J. Hwang, "Clustering and Power Allocation for UAV-Assisted NOMA-VLC Systems: A Swarm Intelligence Approach," 2020, https://arxiv.org/abs/2007.15430.

[68] E. Rodrıguez-Esparza, L. A. Zanella-Calzada, D. Oliva et al., "An efficient Harris hawks-inspired image segmentation method," *Expert Systems with Applications*, vol. 155, Article ID 113428, 2020.

[69] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, 2020.

[70] C. Borrego, M. Amadeo, A. Molinaro, and R. H. Jhaveri, "Privacy-preserving forwarding using homomorphic encryption for information-centric wireless ad hoc networks," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1708–1711, 2019.

[71] T. Thaher and N. Arman, "Effiient multi-swarm binary harris hawks optimization as a feature selection approach for software fault prediction," in *Proceedings of the 11th International Conference on Information and Communication Systems (ICICS)*, pp. 249–254, IEEE, Irbid, Jordan, April 2020.

[72] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery," *Computers & Chemical Engineering*, vol. 133, Article ID 106656, 2020.

[73] B. S. Yildiz and A. R. Yildiz, "The Harris hawks optimization algorithm, salp swarm algorithm, grasshopper optimization algorithm and dragon fly algorithm for structural design optimization of vehicle components," *Materials Testing*, vol. 61, no. 8, pp. 744–748, 2019.

[74] A. R. Yıldız, B. S. Yıldız, S. M. Sait, and X. Li, "The Harris hawks, grasshopper and multi-verse optimization algorithms for the selection of optimal machining parameters in manufacturing operations," *Materials Testing*, vol. 61, no. 8, pp. 725–733, 2019.

[75] S. H. E. Abdel Aleem, A. F. Zobaa, M. E. Balci, and S. M. Ismael, "Harmonic overloading minimization of frequency-dependent components in harmonics polluted distribution systems using harris hawks optimization algorithm," *IEEE Access*, vol. 7, pp. 100 824–100 837, 2019.

[76] A. Abbasi, B. Firouzi, and P. Sendur, "On the Application of harris Hawks Optimization (Hho) Algorithm to the Design of Microchannel Heat Sinks," *Engineering with Computers*, vol. 37, pp. 1–20, 2019.

[77] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Parameters extraction of three-diode photovoltaic model using computation and Harris hawks optimization," *Energy*, vol. 195, Article ID 117040, 2020.

[78] M. Mansoor, A. F. Mirza, and Q. Ling, "Harris hawk optimization-based MPPT control for PV systems under partial shading conditions," *Journal of Cleaner Production*, vol. 274, Article ID 122857, 2020.

[79] M. A. Mossa, O. M. Kamel, H. M. Sultan, and A. A. Z. Diab, "Parameter estimation of PEMFC model based on Harris hawks' optimization and atom search optimization algorithms," *Neural Computing & Applications*, vol. 33, pp. 1–16, 2020.

[80] S. Mouassa, T. Bouktir, and F. Jurado, "Scheduling of Smart home Appliances for Optimal Energy Management in Smart Grid Using harris-hawks Optimization Algorithm," *Optimization and Engineering*, vol. 22, pp. 1–28, 2020.

[81] H. Moayedi, A. Osouli, H. Nguyen, and A. S. A. Rashid, "A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability," *Engineering with Computers*, vol. 37, pp. 1–11, 2019.

[82] Z. Yu, X. Shi, J. Zhou, X. Chen, and X. Qiu, "Effective assessment of blast-induced ground vibration using an optimized random forest model based on a Harris hawks optimization algorithm," *Applied Sciences*, vol. 10, no. 4, p. 1403, 2020.

[83] S. Khalifeh, S. Akbarifard, V. Khalifeh, and E. Zallaghi, "Optimization of Water Distribution of Network Systems Using the harris Hawks Optimization Algorithm (Case Study: Homashahr City)," *MethodsX*, vol. 7, Article ID 100948, 2020.

[84] T. R. Gadekallu, G. Srivastava, M. Liyanage, C. L. Chowdhary, S. Koppu, and P. K. R. Maddikunta, "Hand gesture recognition based on a Harris hawks optimized convolution neural network," *Computers & Electrical Engineering*, vol. 100, Article ID 107836, 2022.

[85] T. R. Gadekallu, D. S. Rajput, M. P. K. Reddy et al., "A novel PCA-whale optimization-based deep neural network model for classification of tomato plant diseases using GPU," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1383–1396, 2021.

[86] T. R. Gadekallu, M. Alazab, R. Kaluri, P. K. R. Maddikunta, S. Bhattacharya, and K. Lakshmanna, "Hand gesture classifiation using a novel cnn-crow search algorithm," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1855–1868, 2021.

[87] S. Sriram, R. Vinayakumar, M. Alazab, and K. Soman, "Network flw based iot botnet attack detection using deep learning," in *Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 189–194, IEEE, ON, Canada, July 2020.

[88] N. A. Golilarz, H. Gao, and H. Demirel, "Satellite image denoising with Harris hawks meta heuristic optimization algorithm and improved adaptive generalized Gaussian distribution threshold function," *IEEE Access*, vol. 7, pp. 57 459–57 468, 2019.

[89] H. Jia, X. Peng, L. Kang, Y. Li, Z. Jiang, and K. Sun, "Pulse coupled neural network based on Harris hawks optimization algorithm for image segmentation," *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 28 369–428 392, 2020.

[90] D. T. Bui, H. Moayedi, B. Kalantar et al., "Harris hawks optimization: a novel swarm intelligence technique for spatial assessment of landslide susceptibility," *Sensors*, vol. 19, no. 16, p. 3590, 2019.

[91] T. A. Shehabeldeen, M. A. Elaziz, A. H. Elsheikh, and J. Zhou, "Modeling of friction stir welding process using adaptive neuro-fuzzy inference system integrated with Harris hawks optimizer," *Journal of Materials Research and Technology*, vol. 8, no. 6, pp. 5882–5892, 2019.

[92] A. Balakrishnan, K. Ramana, G. Dhiman et al., "Multimedia concepts on object detection and recognition with f1 car simulation using convolutional layers," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5543720, 2021.

[93] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions,

and challenges," *Vehicular Communications*, vol. 20, Article ID 100184, 2019.

[94] P. Singh and S. Prakash, "Optimizing multiple ONUs placement in fiber-wireless (FiWi) access network using grasshopper and Harris hawks optimization algorithms," *Optical Fiber Technology*, vol. 60, Article ID 102357, 2020.

[95] S. J. Bhat and S. K. Venkata, "An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields," *Computer Networks*, vol. 179, Article ID 107371, 2020.

[96] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal sink node placement in large scale wireless sensor networks based on Harris hawk optimization algorithm," *IEEE Access*, vol. 8, pp. 19 381–419 397, 2020.

[97] A. Seyfollahi and A. Ghaffari, "Reliable data dissemination for the internet of things using Harris hawks optimization," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 1886–1902, 2020.

[98] H. Xu, G. Zhang, J. Zhao, and Q.-V. Pham, "Intelligent reflecting surface aided wireless networks: harris hawks optimization for beamforming design," in *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 200–205, Chengdu, China, December 2020.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9015778, 9 pages
https://doi.org/10.1155/2022/9015778

Research Article

CNN-Based Brain Tumor Detection Model Using Local Binary Pattern and Multilayered SVM Classifier

Morarjee Kolla,[1] Rupesh Kumar Mishra,[1] S Zahoor ul Huq,[2] Y. Vijayalata,[3] M Venu Gopalachari,[4] and KazyNoor-e-Alam Siddiquee[5]

[1] *Department of Computer Science and Engineering, Chaitanya Bharthi Institute of Technology, Hyderabad, Telangana, India*
[2] *Department of Computer Science and Engineering, G. Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India*
[3] *Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, Telangana, India*
[4] *Department of Information Technology, Chaitanya Bharthi Institute of Technology, Hyderabad, Telangana, India*
[5] *Department of Computer Science and Engineering, University of Science & Technology, Chattogram, Bangladesh*

Correspondence should be addressed to KazyNoor-e-Alam Siddiquee; knas11@gmail.com

Received 27 April 2022; Revised 30 May 2022; Accepted 3 June 2022; Published 27 June 2022

Academic Editor: Muhammad Ahmad

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

In this paper, an autonomous brain tumor segmentation and detection model is developed utilizing a convolutional neural network technique that included a local binary pattern and a multilayered support vector machine. The detection and classification of brain tumors are a key feature in order to aid physicians; an intelligent system must be designed with less manual work and more automated operations in mind. The collected images are then processed using image filtering techniques, followed by image intensity normalization, before proceeding to the patch extraction stage, which results in patch extracted images. During feature extraction, the RGB image is converted to a binary image by grayscale conversion via the colormap process, and this process is then completed by the local binary pattern (LBP). To extract feature information, a convolutional network can be utilized, while to detect objects, a multilayered support vector machine (ML-SVM) can be employed. CNN is a popular deep learning algorithm that is utilized in a wide variety of engineering applications. Finally, the classification approach used in this work aids in determining the presence or absence of a brain tumor. To conduct the comparison, the entire work is tested against existing procedures and the proposed approach using critical metrics such as dice similarity coefficient (DSC), Jaccard similarity index (JSI), sensitivity (SE), accuracy (ACC), specificity (SP), and precision (PR).

1. Introduction

Brain tumors develop as a result of unregulated and fast cell proliferation. It can be fatal if not addressed in the early stages. Machine learning techniques are used to assist clinicians in detecting brain tumors and making judgments. The progression in the deep learning procedures involving the best classifiers impacted a significant advance in medical image processing in recent years. A brain tumor develops when brain tissues develop abnormally. The malignant tissues outgrow the healthy cells, resulting in a mass of cells that eventually transform into tumors [1]. Magnetic resonance imaging (MRI) has been the gold standard for noninvasive brain tumor identification in the last few decades due to its improved soft tissue contrast [2, 3]. MRIs have a considerable impact on medical image processing and analysis due to their ability to provide high-resolution information about brain structure and abnormalities [4–6]. A malignant brain tumor grows significantly more quickly than a benign tumor and is more prone to spread to other parts of the brain. Primary malignant brain tumors have poor prognoses and greatly affect cognitive abilities as well as quality of life [7]. The analysis of medical images is critical in assisting people in diagnosing various disorders. The advanced medical imaging modalities are commonly used methods for analyzing anomalies in brain tissues, which can aid in the detection of tumors in their early stages [8]. The data are initially extracted from dataset, which contains MR pictures of the brain. The first stage of the work receives image data from various sources and forwards it to the next layer for preprocessing using linear filters. Further normalization and patch extraction are important procedures performed in the preprocessing layer to prepare the image for use by CNN. Convolution is considered as a mathematical and engineering tool which is involved in the next phase of CNN where the process feature extraction is conducted on the input image combined with multilayered support vector machine (ML-SVM) to provide optimal outcomes in the work.

2. Related Works

Numerous medical imaging techniques are utilized to obtain information regarding tumors (tumor type, shape, size, location, and so on) that is required for diagnosis [9]. The Fuzzy C-Mean (FCM) method was developed by the authors of [10, 11]. As a means of avoiding the difficulty of determining the number of clusters in an FCM, this technique was devised to pick the pixel intensities and cluster them into two groups. System sensitivity, specificity, and accuracy are all measured to determine how well it works. SVM and FCM techniques were used to generate a hybrid approach for brain MRI categorization by the researchers in [12]. In the early stages of the technique, MRI scan quality is improved using image enhancement technologies such as contrast augmentation and midrange stretch. There are 71 features constructed using the intensity profile, the obtained co-occurrence matrix with new values, and also the Gabor functions after the tumor contour is obtained using a semiautomatic approach in [13, 14]. Because of limitations such as parameter selection or the need for prior knowledge of the images as well as significant calculation times, skull stripping is a common preparation stage in classical discriminative approaches [15]. Deep learning was utilized to construct a classification method for multigrade brain tumors, according to the authors of [16, 17]. The CNN model is used to segment the tumor in this strategy, although the results are limited in accuracy and sensitivity because of the limitations of the CNN model [18]. According to the findings of another study [19, 20], brain tumors can be detected using both hand-created and deep learner features.

The authors of [21–23] did a comparison of naive Bayes, J48 decision tree, and neural network; the downside of this strategy is that it is a conventional approach. The use of machine learning to categorize brain tumors, on the other hand, was proposed by [24, 25]. KNN and SVM were used in this comparison, and the model was found to be 0.95 percent accurate. Because of the automated intelligent system outcomes, the accuracy of this model is higher, but the sensitivity is lower. By integrating support vector machine (SVM) and artificial neural network (ANN), the authors [26, 27] proposed a technique for better precise identification and division of mental tumors; however, despite the high accuracy, the precision was not completely satisfactory [28, 29].

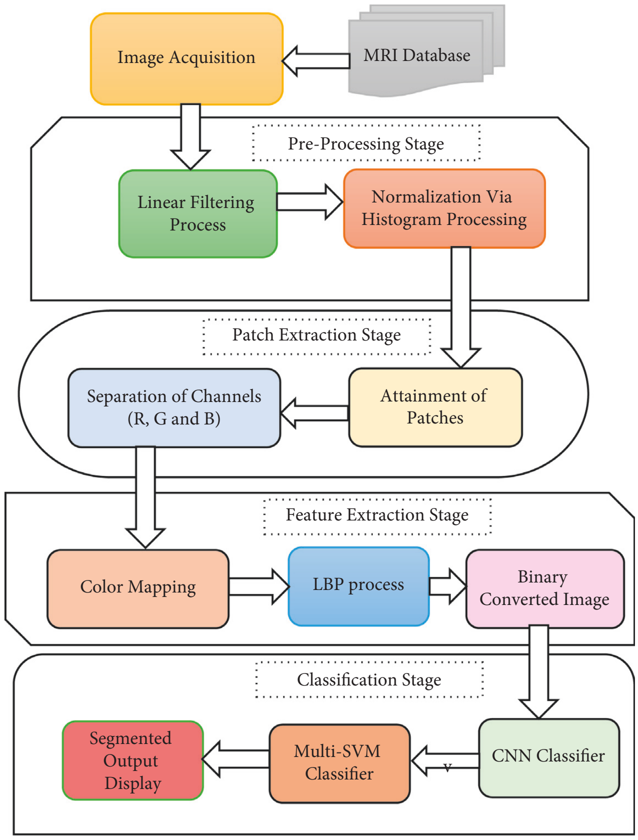


Figure 1: Block diagram of the proposed method.

3. Methodology and Algorithm

This study describes a convolutional neural network (CNN) integrated with a multilayered support vector machine (ML-SVM) algorithm. In this system, there are mainly five distinct blocks: Image Acquisition, Preprocessing, Patch Extraction, Feature Extraction, and CNN Classification and ML-SVM Classifier, as shown in Figure 1. Here, the procedure is explained along with each block and also the successive results.

In this process, images are imported from a dataset. This imported image is MRI scan, which is to be from the glioma region of the brain, and this acquired image is processed to detect tumors. The brain scan images are obtained from the Kaggle medical image database, which is the most preferred and standard database in the field of research. The acquired images are then processed by image filtering techniques followed by image intensity normalization as provided in the proposed block diagram. The input images acquired by the medical imaging modalities consist of artifacts due to inherent properties of modalities. These images must be processed at first hand to remove disturbances that have been added and also normalized. Here, the image is subjected to linear filtering action which is accomplished through a neighborhood operation via weight adjustment procedures.

This filtered image is normalized for uniform intensity. Image normalization is carried out by means of contrast adjustment of the pixel values via histogram processing.

The filtered image is then subjected to the patch extraction stage as shown in the proposed block diagram, to provide

```
Step 1: import MRI from medical database
Step 2: linear filter via Gaussian filter
Step 3: normalization using histogram processing
Step 4: initiate patch extraction process
  (i) Patches obtained
  (ii) RGB channel separation
Step 5: initiate feature extraction process
  (i) Color mapping and obtaining threshold value
  (ii) LBP process to attain binary image
  (a) Image is converted to a grayscale representation.
  (b) For each pixel (gp) in the image, select the P neighbouring pixels. gp's coordinates are specified by
  (c) Set the pixel in the centre (gc) as the threshold for its P neighbours.
  (d) Set to 1 if the adjacent pixel's value is larger than or equal to the centre pixel's value, and 0 otherwise.
  (e) Compute the LBP value now. First, write a binary number comprising digits next to the centre pixel in a counterclockwise
      direction. This binary integer (or its decimal counterpart) is referred to as the LBP-central pixel code and is employed as a
      distinctive local texture.
Step 6: initiate classification process
  (i) Multi-SVM process
    (i) Use training set, group train, and test set as variables for function
    (ii) Classify test cases and map the training data into kernel space
  (ii) CNN classification
    (i) Load train and test data
    (ii) Iterate the process with 100 epochs which yields less error value of 1.2%
    (iii) Create layers and subsampling layers for CNN for varied kernel sizes
    (iv) Classify the data and predict the final output
```

Algorithm 1

patch extracted images, where the image is subdivided to obtain separation between channels; in this process, the image is followed through patches attainment which again subdivides each separated channel based on RGB values for red, green, and blue channels, respectively. In feature extraction, the RGB image is converted to a binary image through grayscale conversion via the colormap process, and later, this process is done by local binary pattern (LBP).

A convolutional network can be used to extract feature information, while a multilayered support vector machine (SVM) can be used to recognize objects. CNN is one of the most extensively used deep learning processes in a wide range of engineering applications. As compared to typical feed-forward neural networks, CNN contains fewer parameters and connections, which makes training more straightforward. In order to give superior outcomes, this includes the training and testing of the required dataset as the primary process. Because it is an iterative method, it is only stopped when the optimal results are achieved. This model was discovered to be capable of extracting characteristics from raw pictures as well as performing classification tasks on its own. The initial phase, as shown in the proposed block diagram, is to train CNN and multilayered SVM models. The second stage is to run the models through their paces and yield the final segmentation results (Algorithm 1).

4. Experimental Investigations

The graphical user interface (GUI) to perform the proposed method in order to segment and detect brain tumor with necessary stages involved is shown in Figure 2.

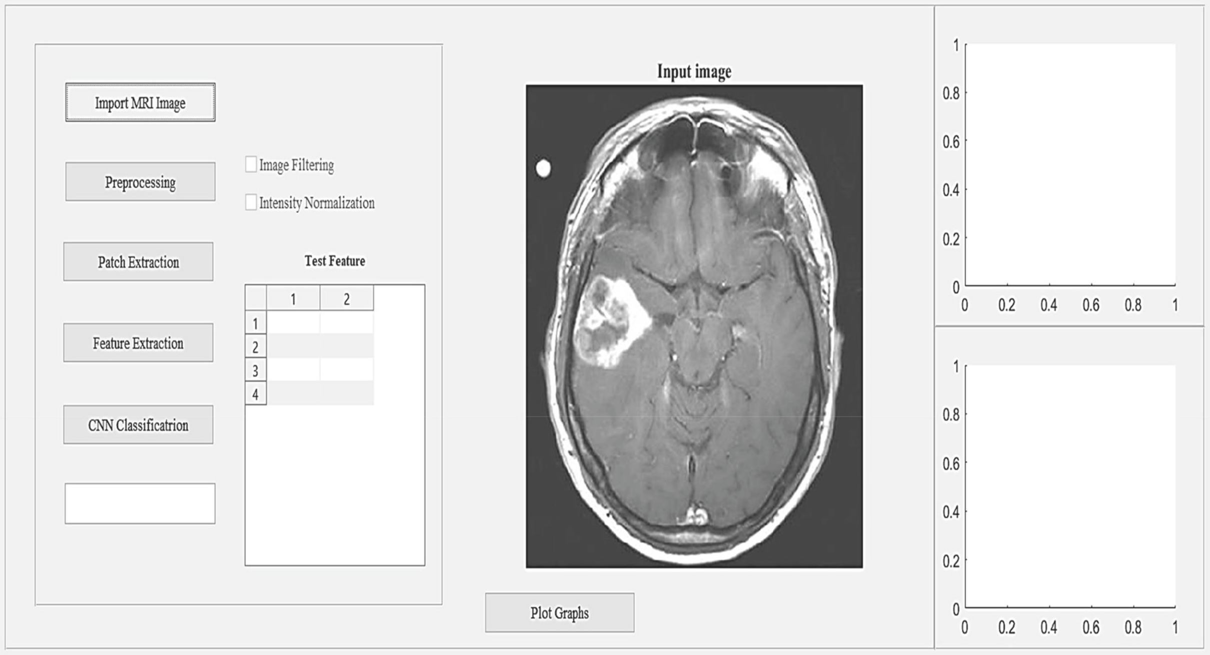
It displays the stages such as image acquisition to import input images from dataset, preprocessing stage to filter the images to remove any unwanted artifacts, patch extraction stage to achieve the patches with respect to channels of RGB, feature extraction stage to carryout color mapping and LBP process to attain the binary image, and CNN classification stage which involves multilayered SVM and later CNN procedures to predict the final output.

The imported input image is MRI scan, as shown in Figure 3, which is to be from the glioma region of the brain, and this acquired image is processed to detect tumors.

The imported image is further fed to the filtering process to attain the filtered output image, as shown in Figure 4. A two-stage filtering procedure is used in this case: first, linear filtering is used, in which the Gaussian filter kernel is employed since it has all of the characteristics of other filters due to the structural organization of its density function; and then, nonlinear filtering is used. It returns a rotationally symmetric Gaussian lowpass filter with a 2D Gaussian smoothing kernel and a positive standard deviation value, resulting in a reduction of artefacts as a result of this filtering.

The resulting image is then subjected to a normalizing process via histogram processing. An operation called mapping is carried out in this scenario to map the intensity values in the resulting grayscale image to new values. The outcome is that normalizing saturates the lowest and highest one percent of all possible pixel values by increasing the contrast values in the resulting image, resulting in the normalized image displayed in Figure 5.

The normalized image is fed into the patch extraction phase, which yields the patch extracted image depicted in



	1	2
1		
2		
3		
4		

Figure 2: Interface of brain tumor segmentation displaying the stages involved.

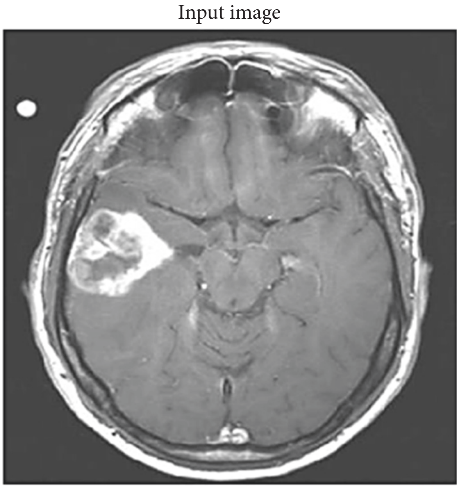


Figure 3: Imported input MRI image from the dataset.

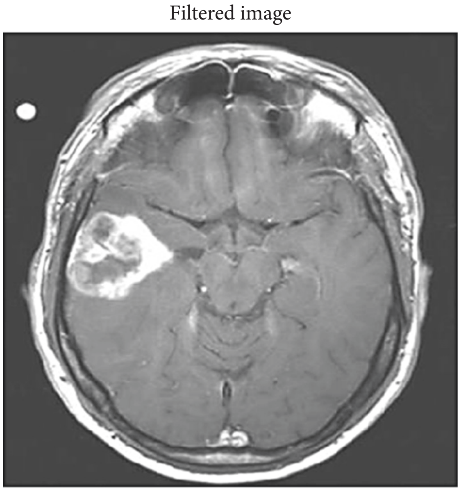


Figure 4: Filtered input MRI image.

Figure 6. Obtaining the patches while keeping the size of each patch in mind is necessary in order to preserve equilibrium. An image patch is a collection of pixels in a photograph that, as the name implies, is a collection of pixels. Patches are divided into groups based on their energy level, with those with a high degree of energy being retained through the use of thresholds.

These patches are divided into red, green, and blue channelized images, as shown in Figures 7–9. In contrast to grayscale images, RGB images include three channels. Each pixel is composed of three channels, each representing a different color. It is necessary to analyze the components of each image's primary colors using the RGB channel separation (which is composed of the three colors: red, green, and blue). A mathematical analysis of the image will be performed, and the results will be presented in gray levels for each color, ranging from black to white and from no color to pure color.

This channel split image is then subjected to the gray thresholding technique, which performs a basic conversion operation. The LBP process is then launched on the resulting image, with LBP being a form of visual descriptor that is utilized for classification purposes in this context. This simple yet efficient texturing operator labels pixels in an image by thresholding the pixels in their immediate vicinity and treating the result as a binary integer. Local binary pattern (LBP) is one of the most often used texturing operators because of its simplicity and effectiveness. At this point, the output image is a binary transformed image, as illustrated in Figure 10.

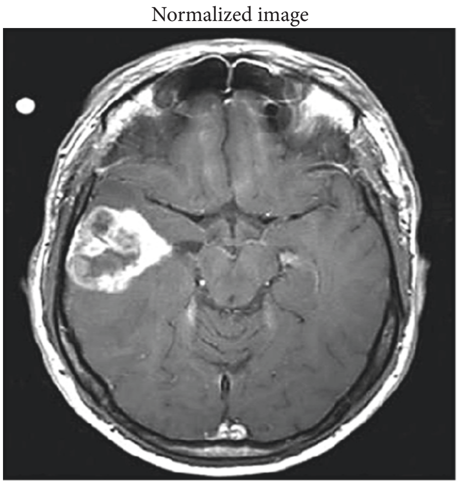


Figure 5: Histogram normalized input MRI image.

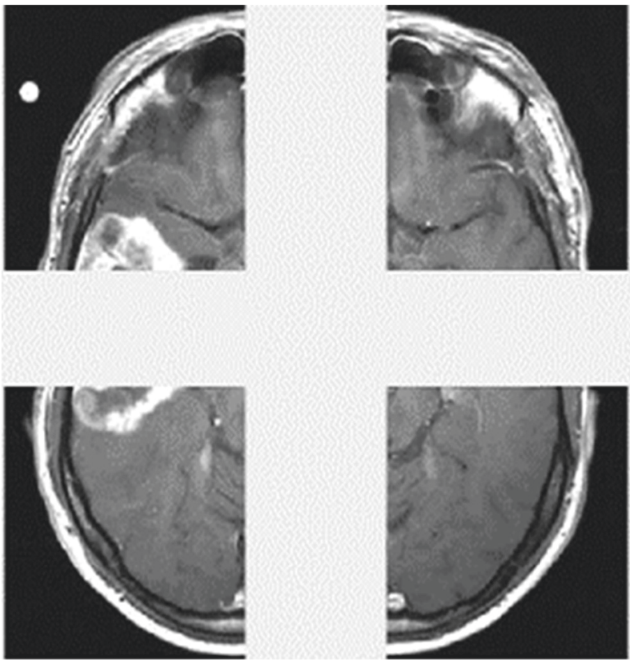


Figure 6: Patch extracted images.

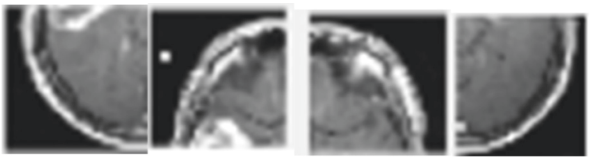


Figure 7: Red channelized images.

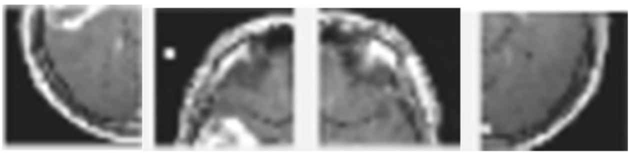


Figure 8: Green channelized images.

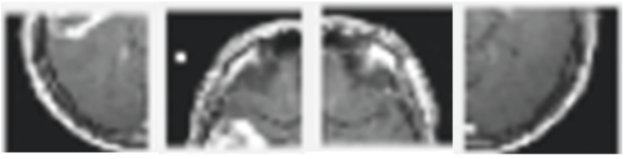


Figure 9: Blue channelized images.

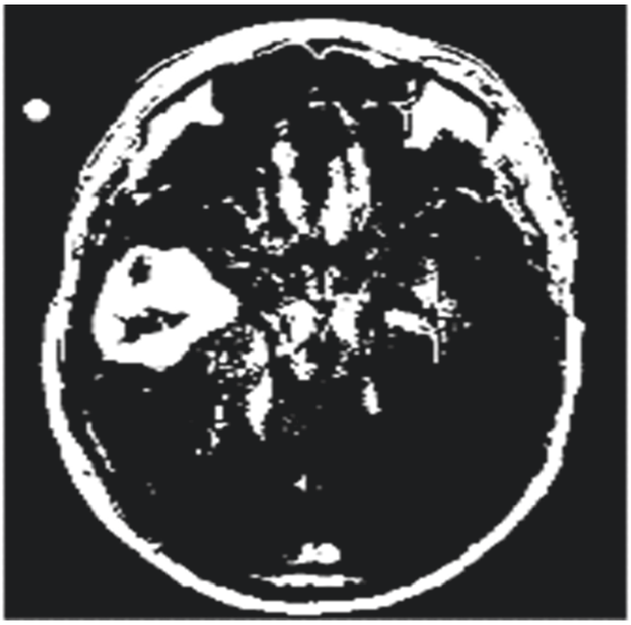
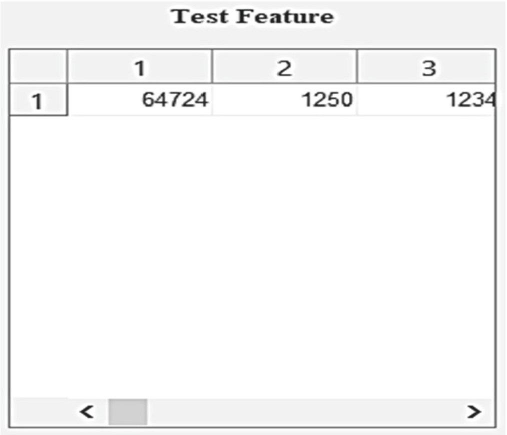


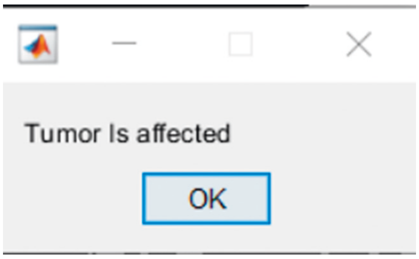
Figure 10: Binary image form of input MRI image.



Test Feature

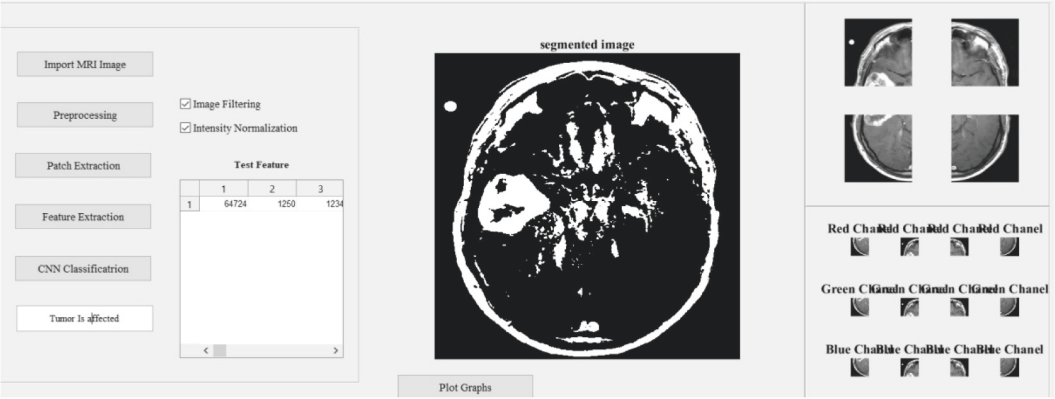
	1	2	3
1	64724	1250	1234

Figure 11: Obtained test features.



Tumor Is affected

Figure 12: Final output from console.



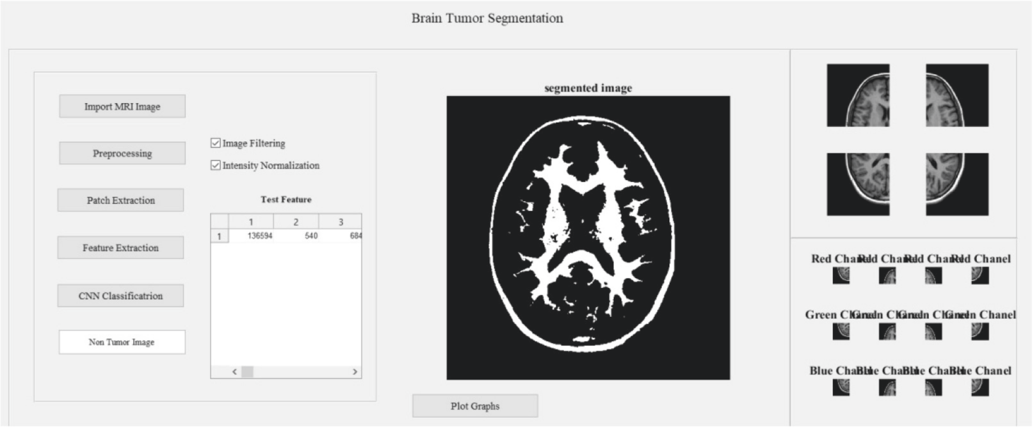
☑ Image Filtering
☑ Intensity Normalization

Test Feature

	1	2	3
1	64724	1250	1234

Tumor Is affected

Figure 13: Interface of brain tumor segmentation displaying the stages involved with final outputs for a tumor affected stage.



☑ Image Filtering
☑ Intensity Normalization

Test Feature

	1	2	3
1	136594	540	684

Non Tumor Image

Figure 14: Interface of brain tumor segmentation displaying the stages involved with final outputs for a nontumor stage.

Table 1: Parametric comparison of DSC and JSI.

Classification methods	Dice similarity coefficient (DSC)	Jaccard similarity index (JSI)
CNN	83.85	89.2
SVM + CNN	88	90.41
Proposed multi-SVM + CNN	96.21	94.32

Figure 11 depicts the appearance of the obtained test features on the constructed GUI. They are crucial and will be utilized in multilayered SVM and CNN classifiers, among other applications. Supervised machine learning (SVM) is a machine learning technique that can be used to aid in the classification or regression of issues. In order to find the optimal potential boundary between the various outputs, this algorithm is used. Using SVM's most basic version, linear separation, the goal is to find a line that optimizes the separation between two classes of 2-dimensional space points in a two-class dataset.

In its most simple type, SVM does not support multilayered classification natively. Multilayered SVMs are usually implemented by combining several two-class SVMs. Therefore, it is a natural step to go from the standard single-layer SVM to the multilayer SVM.

Later, the classified findings will be appropriately labelled using convolutional neural network (CNN) classifiers. CNN is a deep neural network that is commonly used in image classification and machine vision scenarios. Convolutional neural networks (CNN) are complicated feed forward neural networks used in machine learning. Because of its great accuracy, CNNs are employed for image categorization and recognition. Finally, the console shows a simple message indicating whether or not a tumor is present. Figure 12 depicts the console output when a tumor is found.

A final result display of the entire work interface of brain tumor segmentation displaying the stages involved with final outputs for a tumor affected stage and also for a nontumor stage is provided in Figures 13 and 14, respectively.

The entire work is evaluated for existing procedures as well as the proposed process, and the parameters of importance such as dice similarity coefficient (DSC) and Jaccard similarity index (JSI) are shown in Table 1. The graphical representation of the DSC and JSI comparison is also provided in Figure 15.

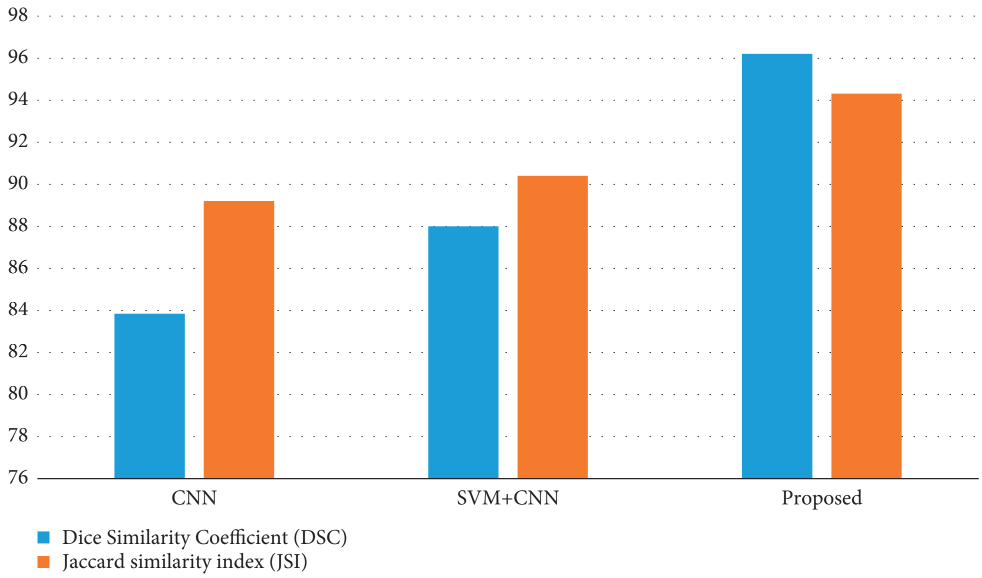


FIGURE 15: Graphical representation of comparison of DSC and JSI.

TABLE 2: Parametric evaluation and comparison.

Classification methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)
CNN	96.45	92	95	94.82
SVM + CNN	95.63	93	95	92.32
Proposed multi-SVM + CNN	99.23	95.73	97.12	97.34

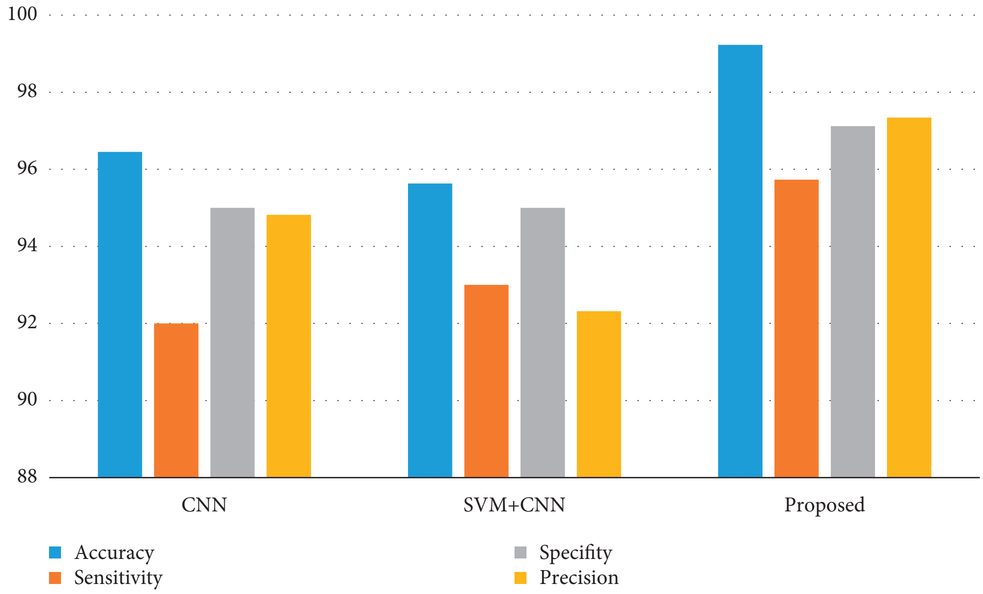


FIGURE 16: Graphical plot of sensitivity, accuracy, specificity, and precision.

In this case, the parameter dice similarity coefficient (DSC) is used to determine the exact amount of ratio of the available real tumor and available nontumor pixels to the anticipated tumor and nontumor pixels and is computed using equation (1), and the Jaccard similarity index (JSI) is used to calculate the percentage of the similarity amid actual tumor pixels in the region of interest and the number of anticipated tumor pixels and is computed as per the standard equation (2).

When the model accurately predicts the positive class, the outcome is known as a “true positive” (TP) in these equations. To denote a result that the model predicted to be positive, the acronym FP stands for false positive, while TN stands for true negative, signifying a result that the model predicted to be negative. When the model inaccurately predicts the negative class, the term “false negative” is used.

$$\text{Dice similarity coefficient } (DSC) = \frac{(2\text{TP})}{(\text{FP} + 2\text{TP} + \text{FN})} X100, \quad (1)$$

$$\text{Jaccard similarity index } (JSI) = \frac{(\text{TP})}{(\text{TP} + \text{FN} + \text{FP})} X100. \quad (2)$$

It is obvious from the table and graphical representation that the proposed method has a leading edge with respect to DSC value being 96.21%, whereas JSI value is 94.32% when compared to previous methods in brain tumor detection and classification. Dice similarity coefficient (DSC) values achieved by the proposed multilayered SVM with CNN are clearly superior to those achieved by earlier approaches.

Similarly, the parameters such as sensitivity, accuracy, specificity, and precision are provided in Table 2.

The graphical representation for the comparison of the parameters such as sensitivity, accuracy, specificity, and precision is also provided in Figure 16.

In this case, the accuracy (ACC) parameter which is of significance is used to calculate the percentage value of the correct tumor region of interest classification rate, which is represented in equation (3), whereas sensitivity (SE) is used to estimate the exact percentage value of how sensitive the method is to compute the corresponding value of the tumor identification rate, and its equation is provided in equation (4).

$$\text{Accuracy } (ACC) = \frac{(\text{TP} + \text{TN})}{(TP + TN) + (FP + FN)} X100, \quad (3)$$

$$\text{Sensitivity (SE)} = \frac{(\text{TP})}{(\text{TP} + \text{FN})} X100. \quad (4)$$

Also, the parameters specificity (SP) and precision (PR) are evaluated, where specificity (SP) discusses the rate value obtained between true negative (TN) values and true positive (TP) values represented as per the formulated equation (5) and further precision (PR) designates about the number of digits in terms of percentage that are used to present a value, and its equation to compute is shown in equation (6).

$$\text{Specificity (SP)} = \frac{(\text{TN})}{(\text{TN} + \text{FP})} X100, \quad (5)$$

$$\text{Precision (PR)} = \frac{(\text{TP})}{(\text{TP} + \text{FP})} X100. \quad (6)$$

It is obvious from the table and graphical representation that the proposed method has a leading edge with respect to accuracy by 99.23%, sensitivity by 95.73%, specificity by 97.12%, and precision by 97.34% when compared to previous methods in brain tumor detection and classification.

5. Conclusion and Summary

Various schemes for detecting and classifying brain tumors have been proposed and investigated in the literature in order to broaden therapy options and patient endurance. Brain tumor segmentation and detection have been processed in steps such as preprocessing, training, testing, and classification in this research work. The dataset’s input images are first filtered and normalized for intensity values. Patch extraction, along with feature extraction, has been viewed as an intermediary stage in this work. After the characteristics were extracted, they were trained and tested in a CNN environment before being given to a multiple layer SVM classifier to display the tumor state in the MR scans. The fundamental advantage of CNN over its predecessors is that it automatically discovers significant features without the need for human intervention. For prior and proposed approaches, critical metrics such as dice similarity coefficient (DSC), Jaccard similarity index (JSI), sensitivity (SE), accuracy (ACC), specificity (SP), and precision (PR) are computed and compared. The proposed strategy proved to be an ideal solution with an accuracy of 99.23 percent. The methods such as gray-level co-occurrence matrix can also be employed in future along with the proposed methodology other than splitting to resolve the issues of losing any small details. The performance and evaluation of the proposed CNN involving multilayered SVM can be improved in the future by undertaking additional research and researching various deep networks. As a future scope, other deep networks can also be investigated for better classification rather than patches losing the details.

Data Availability

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

Conflicts of Interest

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

References

[1] E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem, “Computer-aided diagnosis of human brain tumor through MRI: a survey and a new algorithm,” *Expert Systems with Applications*, vol. 41, no. 11, pp. 5526–5545, 2014.

[2] S. Cha, “Update on brain tumor imaging: from anatomy to physiology,” *AJNR. American journal of neuroradiology*, vol. 27, no. 3, pp. 475–487, 2006.

[3] D. N. Louis, H. Ohgaki, O. D. Wiestler et al., “The 2007 WHO classification of tumours of the central nervous system,” *Acta Neuropathologica*, vol. 114, no. 2, pp. 97–109, 2007.

[4] E. I. Zacharaki, S. Wang, S. Chawla et al., “Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme,” *Magnetic Resonance in Medicine*, vol. 62, no. Issue 4, pp. 1609–1618, 2009.

[5] G. Litjens, T. Kooi, B. E. Bejnordi et al., “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[6] L. Singh, G. Chetty, and D. Sharma, “A novel machine learning approach for detecting the brain abnormalities from MRI structural images,” in *Proceedings of the IAPR international conference on pattern recognition in bioinformatics*, pp. 94–105, Tokyo, Japan, November 2012.

[7] J. A. Schwartzbaum, J. L. Fisher, K. D. Aldape, and M. Wrensch, "Epidemiology and molecular pathology of glioma," *Nature Clinical Practice Neurology*, vol. 2, no. 9, pp. 494–503, 2006.

[8] A. Tiwari, S. Srivastava, and M. Pant, "Brain tumor segmentation and classification from magnetic resonance images: review of selected methods from 2014 to 2019," *Pattern Recognition Letters*, vol. 131, pp. 244–260, 2020.

[9] N. Singh and N. J. Ahuja, "Bug model based intelligent recommender system with exclusive curriculum sequencing for learner-centric tutoring," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 14, no. 4, pp. 1–25, 2019.

[10] A. I şın, C. Direko ğlu, and M. Şah, "Review of MRI-based brain tumor image segmentation using deep learning methods," *Procedia Computer Science*, vol. 102, pp. 317–324, 2016.

[11] N. Singh, A. Kumar, and N. J. Ahuja, "Implementation and evaluation of personalized intelligent tutoring system int," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, pp. 46–55, 2019.

[12] M. Saleck, "Tumor detection in mammography images using fuzzy C- means and GLCM texture features," in *Proceedings of the 14th International Conference on Computer Graphics, Imaging and Visualization*, pp. 122–125, Marrakesh, Morocco, May 2017.

[13] A. J. Zargar, N. Singh, G. Rathee, and A. K. Singh, "Image data-deduplication using the block truncation coding technique," in *Proceedings of the 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 154–158, IEEE, Noida, India, 2015 February.

[14] P. Parveen, "Detection of brain tumor in MRI images, using combination of fuzzy c-means and SVM," in *Proceedings of the 2 nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 98–102, Noida, India, February 2015.

[15] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, "A package-SFERCB-"Segmentation, feature extraction, reduction and classification analysis by both SVM and ANN for brain tumors"," *Applied Soft Computing*, vol. 47, pp. 151–167, 2016.

[16] G. SuryaNarayana, K. Kolli, M. D. Ansari, and V. K. Gunjan, "A traditional analysis for efficient data mining with integrated association mining into regression techniques," in *Proceedings of the 3rd International Conference on Communications and Cyber Physical EngineeringICCCE 2020*, pp. 1393–1404, Hyderabad, India, February 2021.

[17] A. Chaddad and C. Tanougast, "Quantitative evaluation of robust skull stripping and tumor detection applied to axial MR images," *Brain Informatics*, vol. 3, no. 1, pp. 53–61, 2016.

[18] H. Sahu and N. Singh, "Software-defined storage," in *Innovations in Software-Defined Networking and Network Functions Virtualization*, pp. 268–290, IGI Global, Hershey, PA, USA, 2018.

[19] S. Sajjad, "Multi-Grade brain tumor classification using deep CNN with extensive data augmentation," *Journal of Computational Science*, pp. 174–182, 2018.

[20] B. Mishra, N. Singh, and R. Singh, "Master-slave group based model for co-ordinator selection, an improvement of bully algorithm," in *Proceedings of the 2014 International Conference on Parallel, Distributed and Grid Computing*, pp. 457–460, IEEE, Solan, India, 2014 December.

[21] E. Rashid, M. D. Ansari, V. K. Gunjan, and M. Khan, "Enhancement in teaching quality methodology by predicting attendance using machine learning technique," in *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*, pp. 227–235, Springer, New York, NY, USA, 2020.

[22] T. Saba, A. Sameh Mohamed, M. El-Affendi, J. Amin, and M. Sharif, "Brain tumor detection using fusion of hand crafted and deep learning features," *Cognitive Systems Research*, vol. 59, pp. 221–230, 2020.

[23] N. Singh, N. J. Ahuja, and A. Kumar, "A novel architecture for learner-centric curriculum sequencing in adaptive intelligent tutoring system," *Journal of Cases on Information Technology*, vol. 20, no. 3, pp. 1–20, 2018.

[24] M. Al-Ayyoub, G. Husari, O. Darwish, and A. Alabed, "Machine learning approach for brain tumor detection," in *Proceedings of the 3rd International Conference on Information and Communication Systems*, Manhattan, NY, USA, April 2014.

[25] F. Shaik, A. K. Sharma, S. M. Ahmed, V. K. Gunjan, and C. Naik, "An improved model for analysis of Diabetic Retinopathy related imagery," *Indian J Sci Technol*, vol. 9, p. 44, 2016.

[26] N. Subash and J. Rajeesh, "Brain tumor classification using machine learning," *International Science Press, IJCTA*, vol. 8, no. 5, pp. 2335–2341, 2015.

[27] P. Thejaswini, M. B. Bhat, and M. K. Prakash, "Detection and classification of tumour in brain MRI," *International Journal of Engineering and Manufacturing (IJEM)*, vol. 9, no. 1, pp. 11–20, 2019.

[28] V. K. Gunjan, P. S. Prasad, and S. Mukherjee, "Biometric template protection scheme-cancelable biometrics," in *Proceedings of the 2nd International Conference on Communications and Cyber Physical EngineeringICCCE 2019*, pp. 405–411, Singapore, August 2020.

[29] M. D. Ansari, V. K. Gunjan, and E. Rashid, "On security and data integrity framework for cloud computing using tamper-proofing," in *Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering ICCCE 2020*, pp. 1419–1427, Hyderabad, India, December 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3840071, 10 pages
https://doi.org/10.1155/2022/3840071

Research Article

Developing an Intelligent System with Deep Learning Algorithms for Sentiment Analysis of E-Commerce Product Reviews

Mohammad Eid Alzahrani,[1] Theyazn H. H. Aldhyani ,[2] Saleh Nagi Alsubari,[3] Maha M. Althobaiti ,[4] and Adil Fahad[1]

[1] *Department of Engineering and Computer Science, Al Baha University, Al Bahah, Saudi Arabia*
[2] *Applied College in Abqaiq, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*
[3] *Department of Computer Science & Information Technology, Dr. Babasaheb Amedkar Marathwada University, Aurangabad, India*
[4] *Department of Computer Science, College of Computing and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Theyazn H. H. Aldhyani; taldhyani@kfu.edu.sa

Received 26 February 2022; Revised 1 May 2022; Accepted 11 May 2022; Published 28 May 2022

Academic Editor: Muhammad Ahmad

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

Most consumers rely on online reviews when deciding to purchase e-commerce services or products. Unfortunately, the main problem of these reviews, which is not completely tackled, is the existence of deceptive reviews. The novelty of the proposed system is the application of opinion mining on consumers' reviews to help businesses and organizations continually improve their market strategies and obtain an in-depth analysis of the consumers' opinions regarding their products and brands. In this paper, the long short-term memory (LSTM) and deep learning convolutional neural network integrated with LSTM (CNN-LSTM) models were used for sentiment analysis of reviews in the e-commerce domain. The system was tested and evaluated by using real-time data that included reviews of cameras, laptops, mobile phones, tablets, televisions, and video surveillance products from the Amazon website. Data preprocessing steps, such as lowercase processing, stopword removal, punctuation removal, and tokenization, were used for data cleaning. The clean data were processed with the LSTM and CNN-LSTM models for the detection and classification of the consumers' sentiment into positive or negative. The LSTM and CNN-LSTM algorithms achieved an accuracy of 94% and 91%, respectively. We conclude that the deep learning techniques applied here provide optimal results for the classification of the customers' sentiment toward the products.

1. Introduction

Web 3.0 has the main features of the semantic web, artificial intelligence, connectivity, etc., allowing people to use social media to communicate and express their opinions about real-world events. In this context, the analysis of users' reviews is essential for companies to grow worldwide. This makes opinion mining a key player in the analysis of reviews and discussions. Nowadays, companies analyze this type of information to improve the quality and performance of the products and, consequently, survive in a competitive market. Opinion mining can be expressed as the reason behind any action or movement that people use to follow the reason [1].

Within the huge amount of data generated on the Internet, important information is hidden. Data mining techniques are used to extract information and solve various problems. Online product reviews have two important aspects under which data are stored on the Internet. Commercial websites are platforms where users express their sentiment or opinion on several topics. Sentiment analysis refers to a broad area of natural language processing (NLP), computational linguistics, and text mining [2]. The use of these techniques leads to the extraction and analysis of the opinion on a given product. Opinion mining defines an opinion as positive or negative, and sentiment analysis defines the polarity value of a user's opinion on a particular product or service. The current approaches of sentiment analysis are mainly [3] machine learning algorithms [4], lexicon-based methods, [5] and hybrid models [6, 7].

Negation is a prevalent morphological development that impacts polarity and, therefore, must be reflected in the assessment of sentiment. Automatic detection of negation in news articles is required for numerous text processing applications, including sentiment analysis. Here, we explored the role and importance of users' reviews concerning particular products on the decision using sentiment analysis. We present experimental results that demonstrate that sentiment analysis is appropriate to this end. The goal was to determine the polarity of the natural language of texts written in product reviews.

The existing straightforward approaches are statistical, based on frequencies of positive and negative words. Recently, researchers discovered new ways to account for other aspects of content, such as structural or semantic features. The present work focuses on the identification of document-level negation by using multiple computational methods. In recent years, with the exponential growth of smartphone use, many people are connected to social networking platforms, like Facebook, Twitter, and Instagram. Social networks have become a field to express beliefs or opinions, emotions, thoughts, personal issues, places, or personalities.

There are numerous studies applying sentiment analysis, some of which used real-time data from Twitter for extracting patterns by employing the Twitter-streaming application programming interface (API) [8, 9]. The sentiment analyzers are divided into two types: SentiWordNet [10] and WordNet [11]. Sentiment analysis uses positive and negative scores to classify opinions. By developing a model to analyze word sequence disambiguation [12], the Twitter-streaming API was used to gather data concerning the Indonesian presidential elections [13]. Needless tweets were removed, and the remaining data were investigated for sentimental aspects by dividing each tweet into numerous sub-tweets and calculating the sentiment polarity of the sub-tweets for predicting the consequence of the elections. The mean absolute error metric was used to evaluate the results, it noted that the prediction error was 0.6 better than the previous study [14]. To predict the Swedish election outcome by using Twitter data, a system was developed [15]. To predict the outcome of the European elections, a new method was designed that studied the similarity of the structure with the outcome of the vote. Another method was created to test Brazilian municipal elections in six cities [16]. In this methodology, sentiment analysis was applied along with a stratified sample [17] of users to compare the characteristics of the findings with the actual voters.

Many researchers have used machine learning and artificial intelligence to analyze the sentiment of tweets [18, 19]. In [20], the Naive Bayes, support vector machine (SVM) [21], and information entropy-based [22] models were applied to classify product reviews. A hybrid machine learning algorithm based on Twitter opinion mining was proposed in [23]. Heydari et al. [24] proposed time series model for fraudulent sentiment reviewer analysis. Hajek et al. [25] developed a deep feedforward neural network and convolution model to detect fake positive and negative review in an Amazon dataset. Long et al. [26] applied LSTM with multi-head attention network for predicting sentiment-based text using China social media dataset. Dong et al. [27] proposed supervised machine linear regression for predicting sentiment of customers presented in online shopping data using sentiment analysis learning approaches.

Researchers have been focusing on developing powerful models to deal with the ever-increasing complexity of big data [28, 29], as well as expanding sentiment analysis to a wide range of applications [30, 31], from financial forecasting to marketing strategies [32] among other areas [33, 34]. However, only a few of them analyzed different deep learning approaches to give real evidence of their performance [35]. Deep learning techniques are becoming increasingly popular. When assessing the performance of a single approach on a single dataset in a specific area, the results suggest that CNN and RNN have relatively good accuracy. Based on AdaBoost combination, Gao et al. [36] proposed CNN model for sentiment analysis in user-generated text. In this vein, Hassan and Mahmood [37] demonstrated that the CNN and RNN models overcame the problem of short texts in deep learning models.

Some traditional approaches, which are assisted by machine learning techniques, are based on aspects of the used language. Using the domain of movie opinions, Pang et al. [18] studied the performance of various machine learning algorithms, including Naive Bayes, maximum entropy, and SVM. By using SVM with unigrams, they achieved an accuracy of 82.9%. NLP is typically used to extract features used by a sentiment classifier. In this aspect, the majority of NLP strategies are centered on the usage of n-grams but the use of a bag-of-words strategy is also common [38, 39]. Numerous studies have demonstrated significant results when employing the bag-of-words as a text representation for item categorization [40–44].

Researchers have taken advantage of NLP themes to develop deep learning models based on neural networks with more than three layers, according to the journal Nature. Most of these studies found that deep learning models accurately detect sentiment in various situations. The CNN [45, 46], RNN [47], deep neural network [48], recursive neural deep model [49], and the attention-based bidirectional CNN-RNN [50] models are some representative examples. Some researchers combine models, which are then referred to as hybrid neural networks. The hierarchical bidirectional RNN is an example of a hybrid neural network [51]. The main issue with sentiment analysis of product reviews in the e-commerce domain is the existence of fake reviews that lead customers to select undesired products [52].

The main contributions of the proposed research are the following:

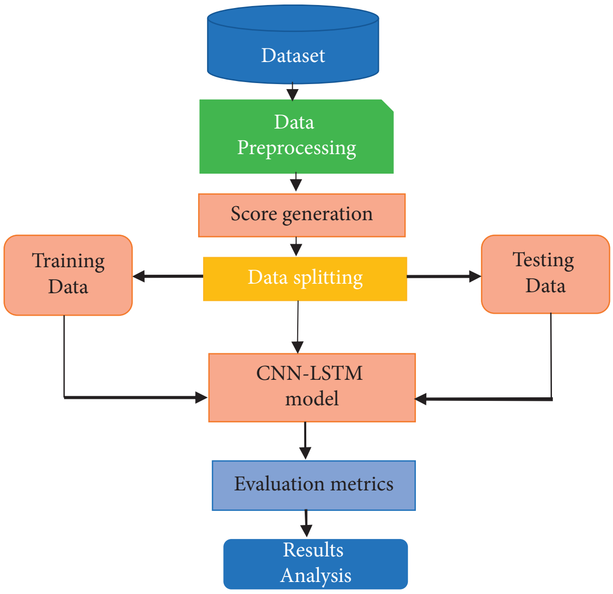


Figure 1: Framework for the proposed methodology.

(1) The generation of a sentiment score using a lexicon-based approach for each product review of the dataset.

(2) Labeling the review texts as negative if the generated sentiment score is <0 or positive if the score is >1.

(3) The combination of all product reviews into a single data frame to obtain more sentiment-related words.

(4) Improving the accuracy by developing a hybrid deep learning model combining the CNN and LSTM models for the product-related sentiment classification.

(5) Comparing the classification performance of the CNN-LSTM and LSTM models.

2. Materials and Methods

The proposed methodology for predicting the review-related sentiments is based on the deep learning algorithms presented here. The phases of the proposed system are the following: dataset collection, data preprocessing, generating the sentiment score, polarity calculation, applying the CNN-LSTM model, evaluation metrics, and analysis of the results. Figure 1 shows the framework of the proposed methodology used in the present study.

2.1. Datasets. To evaluate the proposed system, the dataset [53] was collected from reviews on the Amazon website in JSON file format. Each JSON file comprises a number of reviews (Table 1). The dataset includes reviews of laptops, mobile phones, tablets, televisions, and video surveillance products. The data preprocessing includes various steps, such as lowercase processing with meta-features like the reviewer's ID, the product ID, and the review text.

2.2. Data Preprocessing. We implemented different preprocessing steps aiming at cleaning the review texts so that they are easy to process. The following preprocessing methods were performed on the dataset as a whole.

Table 1: The number of reviews per product category.

Product name	Review count
Laptops	1,946
Mobile phones	1,918
Tablets	1,894
Televisions	1,596
Video surveillance products	2,597

2.2.1. Lowercase. It entails converting whole words of the review text into lowercase words.

2.2.2. Stopword Removal. Stopwords are widely used words in a language, such as "the," "a," "an," "is," and "are". As these words do not carry any information significant for the model, they were removed from the content of the review.

2.2.3. Punctuation Removal. All punctuation marks in the review texts were removed.

2.2.4. One-Word Review Elimination. Reviews that included only one word were eliminated.

2.2.5. Contraction Removal. This process replaces a word originally written in the short form with the respective full form; for instance, "when've" becomes "when have."

2.2.6. Tokenization. Each sentence of the review texts was divided into small pieces of words or tokens.

2.2.7. Part-of-Speech Tagging. This step is used to tag each word present in the sentence with a POS tag, for example, "VB" for a verb, "AJJ" for an adjective, and "NN" for a noun.

2.2.8. Score Generation. The review text was evaluated for sentiment, and a score was generated. For calculating the sentiment score, the dataset was matched with opinion lexicon [53] that consists of 5,000 positive words and 4,500 negative words with their respective scores. The sentiment score was calculated for each review text based on the scores of the lexicon. The review text was labeled as positive if the score was >0; otherwise, it was labeled as negative.

2.2.9. Word Embeddings. We calculated numerical vectors with every preprocessed sentence in the product review dataset using the "Word embeddings" method. To create word indices, we first turned all of the review text terms into sequences. The Keras text tokenizer [54] is being used to obtain those indices. We made sure that no term or word gets a zero index in the tokenizer, and that the vocabulary size is adjusted properly. Then, for each single word in the training and testing sets, a distinctive index is generated,

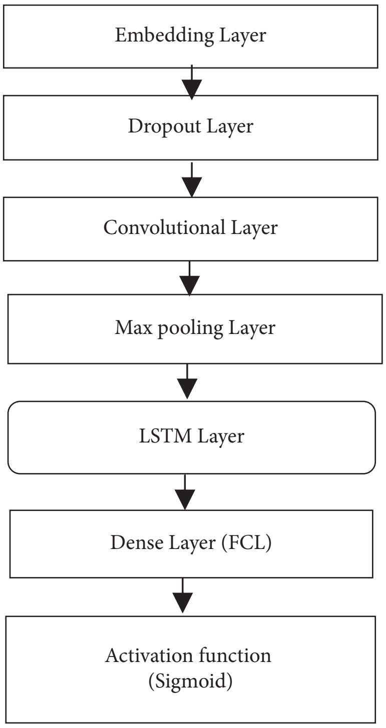


Figure 2: The structure of the CNN-LSTM model.

which is employed to create numeric vectors of all review texts of the dataset.

2.3. *The CNN-LSTM Model.*

Figure 2 presents the structure of the CNN-LSTM model used for sentiment classification of customers' reviews using an Amazon dataset.

2.3.1. Embedding Layer. This is the initial layer of the CNN-LSTM model that is used to transform each word in the training dataset into an actual-valued vector, meaning that a set of sentiment-related words are constructed and transformed into a numerical form. This process is known as word embedding. The embedding layer consisted of three components: the vocabulary size (maximum features; 15,000 words), the embedding dimensions (50), and the input sequence length (400 words).

2.3.2. Dropout Layer. The main task of this layer is to avoid the overfitting of the model [52]. Here, we assigned the value 0.4 to the dropout rate parameter, where this value has a range between 0 and 1. The main function of the dropout layer is to arbitrarily deactivate a set of neurons in the embedding layer, where every neuron denotes the dense exemplification of a sentiment word in a review text.

CNN is a deep learning technique used in different areas such as natural language preprocessing tasks, computer vision, and medical image processing.

2.3.3. Convolution Layer. The third layer of the CNN-LSTM model is used for the extraction of features from the input matrix. It uses n convolution filters that operate over the elements of the input sequence matrix to find the convolutions for each sequence. We set the number of filters to 64 and the size of the filter kernel to 3×3.

2.3.4. Max Pooling Layer. This layer performs downsampling beside the spatial dimensionality of the given input sequences. It considers the maximum value of all input features in the pool of each filter kernel. It has assigned to 5×5 kernel.

2.3.5. LSTM Layer. LSTM is a type of RNN capable of learning long-term dependence [52]. We used an LSTM layer and assigned it to 50 hidden units toward the next layer. One of the most notable advantages of employing a convolutional neural network as feature extraction technique beyond a traditional LSTM is the reduction in the aggregating amount of features. Throughout the feature extraction process, a sentiment classification model uses these features (words) for prediction of the product review text as positive or negative sentiment. LSTM executes precalculations for the input sequences before providing an output to the last layer of the network. In every cell, four discrete computations are conducted based on four gates: input (i_t), forget (f_t), candidate (c_t), and output (o_t). The structure of the LSTM model is presented in Figure 3. The equations for these gates are as follows:

$$\begin{aligned}
f_t &= \mathrm{sig}\left(W f_{xt} + U f_{ht} - 1 + b_f\right),\\
i_t &= \mathrm{sig}\left(W i_{xt} + U i_{ht} - 1 + b_i\right),\\
O_t &= \mathrm{sig}\left(W o_{xt} + U o_{ht} - 1 + b_o\right),\\
c \sim t &= \tanh\left(w c_{xt} + U c_{ht} - 1 + bc\right),\\
C_t &= \left(f_{to} ct - 1 + i_{to} c \sim t\right),\\
h_t &= O_{to} * \tanh\left(C_t\right),\\
\tanh(x) &= \frac{1 - e^{2x}}{1 - e^{2x}},
\end{aligned} \tag{1}$$

where sig and tanh are the sigmoid and tangent activation functions, respectively, X is the input data, W and b represent the weight and bias factor, respectively, C_t is the cell state, $c \sim t$ is the candidate gate, and h_t refers to the output of the LSTM cell.

2.3.6. Dense Layer (Fully Connected Layer). This is a hidden layer in the CNN-LSTM model. It consists of 512 artificial connected neurons that connect all neurons of the network. The function applied to this layer is the rectified linear unit described by the following equation:

$$f(x) = \max(o, x). \tag{2}$$

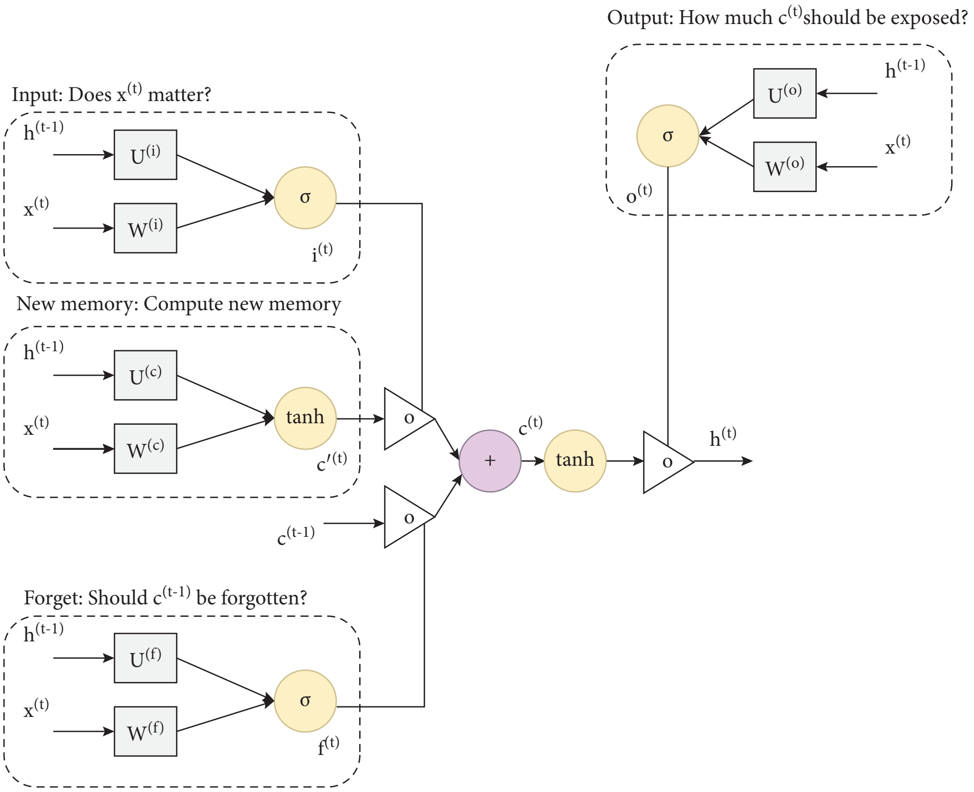


Figure 3: The structure of the LSTM model.

2.3.7. Sigmoid Activation Function. It is the first layer that detects and classifies the output classes (positive or negative sentiment). The sigmoid function formula is given as follows (Algorithm 1):

$$\sigma = \frac{1}{1 - e^{2x}}. \tag{3}$$

2.4. Evaluation Metrics. To evaluate the proposed models (CNN-LSTM and LSTM), the accuracy, precision, recall, F1-score, and specificity metrics were used. The performance measurements are presented below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \times 100\%,$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%,$$

$$\text{F1} - \text{score} = 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \times 100\%, \tag{4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%,$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%,$$

where true positive (TP) represents the total number of samples that are successfully classified as positive sentiment, false positive (FP) is the total number of samples that are incorrectly classified as negative sentiments, true negative (TN) denotes the total number of samples that are successfully classified as negative sentiment, and false negative (FN) represents the total number of samples that are incorrectly classified as positive sentiments.

3. Experimental Results

In this section, we present the experimental results of the application of the CNN-LSTM and LSTM models for the analysis and prediction of sentiment in the e-commerce domain. We used hardware with 4 GB RAM and an i7 2800 CPU and ran the experiments on the Jupyter environment. The evaluation metrics (accuracy, precision, F1-score, recall, and specificity) were employed to examine the proposed system. The word cloud (sentiment words and product names) of the dataset is presented in Figure 4, which shows graphical representations of words (large font words) that give greater importance to that seem more repeatedly in the used product review dataset.

3.1. Data Splitting. In this phase, we divided the dataset that consisted of 13,057 product reviews into 70% training, 10% validation, and 20% testing datasets. Then, the CNN-LSTM and LSTM models were applied to detect and classify the review texts into positive or negative. Table 2 shows the splitting of the dataset.

```
Step 1: input training set as x_train, targets as y_train
Step 2: assign the hyperparameters as embedding_dimension = 50, number_filters = 64, vocabulary_size = 15000 words, input_length = 400, dropout_rate = 0.4, strides = 5, activation_function = Relu, kernel_size = 3 × 3, pool_size = 5 × , lstm_units = 50, batch_size = 32, number_epochs = 5, num_classes = 2, optimizer = (Adam).
Step 3: initialize sequential model ()
Step 4: set embedding layer as input layer
Model = model.add(embedding(embedding_dimension, vocabulary_size, input_length))
Step 5: add convolutional layer
Model = model.add(convolution 1D(number_filters, kernel_size))
Step 6: add max pooling layer
Model = model.add(max_pool layer(pool_size, strides))
Step 7: add LSTM layer
Model = model.add(LSTM_layer(lstm_units, activation_function, recurrent_activation, dropout_rate, return_sequences))
Step 8: add dropout layer
Model-model.add(Dropout(dropout_rate))
Step 9: add dense layer
Model = model.add(Dense_layer(num_classes, activation_function = "sigmoid"))
Step 10: compilation
model.compile(e (loss_function, optimizer)
model.fit (y_train, y_train, number_epochs, batch_size)
```

Algorithm 1: CNN-LSTM.

Figure 4: Word cloud of the dataset.

Table 2: The splitting of the dataset.

Total number of reviews	Training set 80%	Validation set 10%	Testing set 20%
13,057 (11,184 positive; 1,873 negative)	9,400	1,045	2,612

Table 3: Results of the deep learning models.

Models	Specificity	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
LSTM	95	91.03	92.07	97.73	95.50
CNN-LSTM	96	94	94	99	96.03

3.2. Results and Discussion. Table 3 shows the results of the deep learning approaches. The CNN-LSTM model achieved high accuracy (96%).

The confusion matrix of the CNN-LSTM and LSTM models is shown in Figure 5. The confusion matrix is used to present the rates of TP, FP, TN, and FN of the sample. Based on these rates, the evaluation metrics (specificity, accuracy, recall, precision, and F1-score) were calculated to evaluate the CNN-LSTM model using unseen data to predict the sentiment of customers. LSTM resulted in 82.24% TP, while

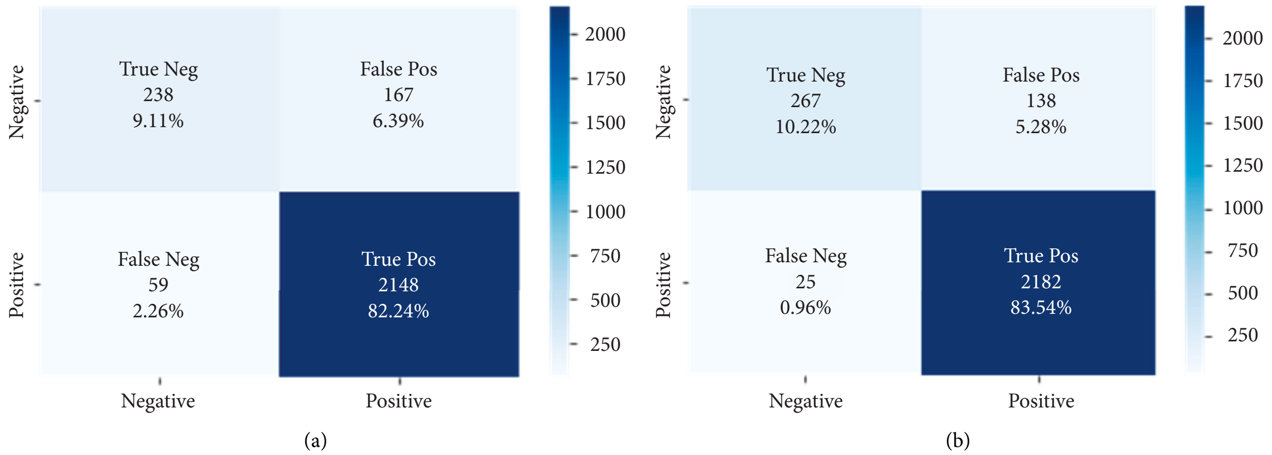


Figure 5: Confusion matrix of the (a) LSTM and (b) CNN-LSTM models.

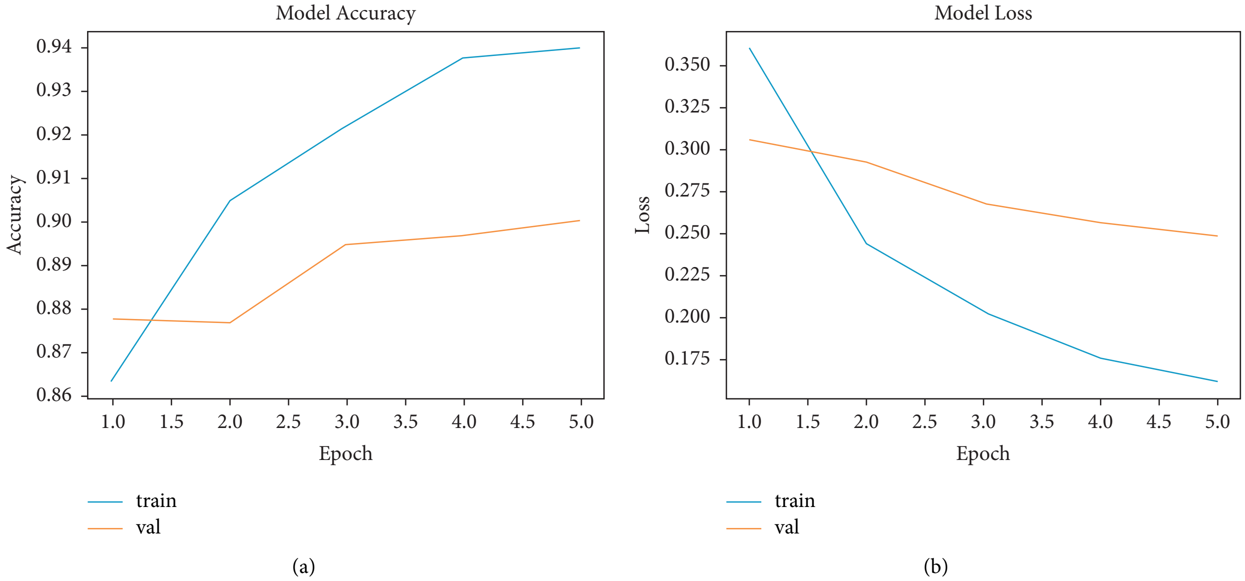


Figure 6: The performance of the LSTM model: (a) accuracy and (b) loss.

CNN-LSTM resulted in 83.54% TP. As for misclassification, LSTM resulted in 6.39% FP and CNN-LSTM in 5.28% FP, indicating that the CNN-LSTM model was slightly better than the LSTM model.

The accuracy performance of LSTM for the training and validation datasets is presented in Figure 6. The LSTM model presented increasing accuracy during the training phase (from 86% to 94%), whereas in the testing phase, it achieved 91% accuracy with 10 epochs. The loss of the LSTM model in the training phase decreased from 5 to 0.35, while in the validation phase, the model loss decreased from 0.3 to 0.27.

The accuracy performance of the CNN-LSTM during the training phase increased from 87.50% to 97%. In the validation phase, the accuracy performance reached 94% (Figure 7(a)). The loss of the CNN-LSTM model in the validation phase was 0.20 (Figure 7(b)).

The dataset developed by Rajkumar et al. [53] proposed SVM and Naive Bayes methods to predict sentiment analysis. They collected data from Amazon concerning mobile phones, tablets, cameras, and televisions. They applied the SVM method to each dataset individually. Here, we applied deep learning models to all the datasets combined. The empirical results of our system were compared with the results of [28] and are shown in Table 4. The CNN-LSTM model achieved an accuracy of 94%.

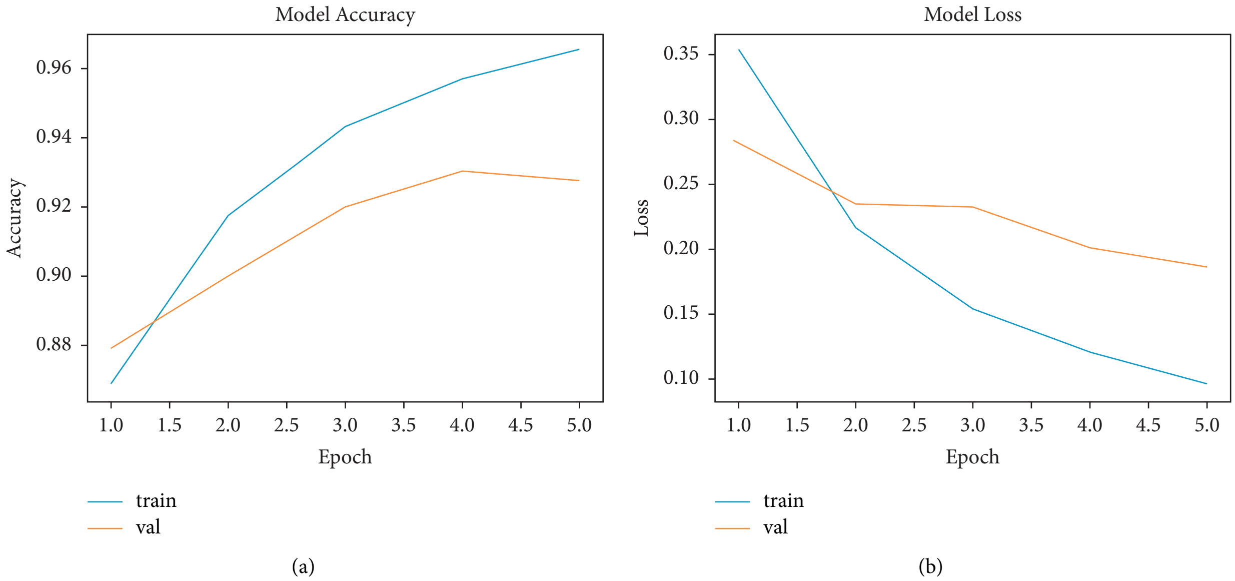


Figure 7: The performance of the CNN-LSTM model: (a) accuracy and (b) loss.

Table 4: Significant results of the CNN-LSTM model compared to the SVM method.

Models	Datasets	Accuracy (%)
Support vector machine [28]	Televisions, tablets, mobile phones, laptops, and video surveillance	88, 84, 92, 88, and 93
Proposed system (CNN-LSTM)	All dataset	94

4. Conclusion

Recently, sentiment analysis has become a valuable tool for the generation and evaluation of different types of data, helping the decision-making processes that lead to the improvement of businesses and companies. Social networking creates a large amount of data that require processing and analysis to obtain relevant insights. In the present study, the experimental dataset was collected from the Amazon website and included reviews of laptops, mobile phones, tablets, televisions, and video surveillance products. The lexicon-based approach was used for the calculation of the sentiment score for each review text. The output of the preprocessed data was classified with the LSTM and CNN-LSTM models. The experimental results showed that our model was satisfactory in all the measurement metrics.

Data Availability

We have collected dataset from authors of research article: https://www.researchgate.net/publication/326985579_Sentiment_Analysis_ on_Product_Reviews_Using_ Machine_Learning_ Techniques_Proceeding_of_CISC_ 2017

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors deeply acknowledge Taif University for supporting this research through Taif University Researchers Supporting Project Number (TURSP-2020/328), Taif University, Taif, Saudi Arabia. This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Project No: GRANT674).

References

[1] E. Cambria and D. S. A. Das, “Affective computing and sentiment analysis,” *A Practical Guide to Sentiment Analysis*, vol. 31, pp. 1–10, 2017.

[2] O. Jagdale, V. Harmalkar, S. Chavan, and N. Sharma, “Twitter mining using R,” *Int. J. Eng. Res. Adv. Tech.*vol. 3, pp. 252–256, 2017.

[3] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: a survey,” *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[4] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[5] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based methods for sentiment analysis,” *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[6] R. Prabowo and M. Thelwall, “Sentiment analysis: a combined approach,” *Journal of Informetrics*, vol. 3, no. 2, pp. 143–157, 2009.

[7] Y. Dang, Y. Zhang, and H. Chen, “A lexicon-enhanced method for sentiment classification: an experiment on online product reviews,” *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 46–53, 2010.

[8] R. Jose and V. S. Chooralil, “Prediction of election result by enhanced sentiment analysis on twitter data using word sense disambiguation,” in *Proceedings of the 2015 International*

Conference on Control Communication & Computing India (ICCC), pp. 638–641, Trivandrum, India, November 2015.

[9] A. Esuli and F. S. Sebastiani, *A High-Coverage Lexical Resource for Opinion Mining*, Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR), Pisa, Italy, 2006.

[10] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[11] R. Navigli, "Word sense disambiguation," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, 2009.

[12] M. Ibrahim, O. Abdillah, A. F. Wicaksono, and M. Adriani, "Buzzer detection and sentiment analysis for predicting presidential election results in a twitter nation," in *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1348–1353, Atlantic City, NJ, USA, November 2015.

[13] N. Dokoohaki, F. Zikou, D. Gillblad, and M. Matskin, "Predicting Swedish elections with twitter: a case for stochastic link structure analysis," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1269–1276, Paris, France, August 2015.

[14] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[15] R. Miranda Filho, J. M. Almeida, and G. L. Pappa, "Twitter population sample bias and its impact on predictive outcomes: a case study on elections," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1254–1261, Paris, France, August 2015.

[16] E. Foreman, *Survey Sampling Principles*, CRC Press, Boca Raton, FL, USA, 1991.

[17] S. N. Alsubari, S. N. Deshmukh, A. Abdullah Alqarni et al., "Data analytics for the identification of fake reviews using supervised learning," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 3189–3204, 2022.

[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?" in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing—EMNLP '02'*, vol. 10, pp. 79–86, Philadelphia, PA, USA, July 2002.

[19] G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," in *Proceedings of the 2014 Seventh International Conference on Contemporary Computing (IC3)*, pp. 437–442, Noida, India, August 2014.

[20] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, pp. 137–142, Chemnitz, Germany, April 1998.

[21] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing Comput," *Linguist*, vol. 22, pp. 39–71, 1996.

[22] F. H. Khan, S. Bashir, and U. Qamar, "Tom: twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245–257, 2014.

[23] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance, "What yelp fake review filter might be doing?" in *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pp. 409–418, Cambridge, MA, USA, July, 2013.

[24] A. Heydari, M. Tavakoli, and N. Salim, "Detection of fake opinions using time series," *Expert Systems with Applications*, vol. 58, pp. 83–92, 2016.

[25] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Computing & Applications*, vol. 32, no. 23, Article ID 17259, 2020.

[26] F. Long, K. Zhou, and W. Ou, "Sentiment Analysis of Text Based on Bidirectional LSTM with Multi-Head Attention," *Advanced Optical Imaging for Extreme Environments*, vol. 7, Article ID 141960, 2019.

[27] J. Dong, Y. Chen, A. Gu et al., "Potential Trend for Online Shopping Data Based on the Linear Regression and Sentiment Analysis," *Mathematical Problems in Engineering*, vol. 2020, Article ID 4591260, 11 pages, 2020.

[28] S. S. Roy, M. Biba, R. Kumar, R. Kumar, and P. Samui, "A new SVM method for recognizing polarity of sentiments in twitter," in *Handbook of Research on Soft Computing and Nature-Inspired Algorithms*, pp. 281–291, IGI Global, Hershey, Pennsylvania, 2017.

[29] M. J. S. Keenan, *Advanced Positioning, Flow, and Sentiment Analysis in Commodity Markets*, Wiley, Hoboken, NJ, USA, 2018.

[30] R. Satapathy, E. Cambria, and A. Hussain, *Sentiment Analysis in the Bio-Medical Domain*, Springer, Berlin, Germany, 2017.

[31] A. Rajput, "natural language processing, sentiment analysis, and clinical analytics," in *Innovation in Health Informatics*, pp. 79–97, Elsevier, Amsterdam, The Netherlands, 2020.

[32] J. Qian, Z. Niu, and C. Shi, "Sentiment analysis model on weather related tweets with deep neural network," in *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pp. 31–35, Macau, China, February 2018.

[33] D.-H. Pham and A.-C. Le, "Learning multiple layers of knowledge representation for aspect based sentiment analysis," *Data & Knowledge Engineering*, vol. 114, pp. 26–39, 2018.

[34] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha, and S. Yenduri, "Application of deep learning to sentiment analysis for recommender system on cloud," in *Proceedings of the 2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 93–97, Dalian, China, July 2017.

[35] Q. Tul, M. Ali, A. Riaz et al., "Sentiment analysis using deep learning techniques: a review," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, p. 424, 2017.

[36] Y. Gao, W. Rong, Y. Shen, and Z. Xiong, "Convolutional neural network based sentiment analysis using Adaboost combination," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1333–1338, Vancouver, BC, Canada, July 2016.

[37] A. Hassan and A. Mahmood, "Deep learning approach for sentiment analysis of short texts," in *Proceedings of the Third International Conference on Control, Automation and Robotics (ICCAR)*, pp. 705–710, Nagoya, Japan, April 2017.

[38] M. Kraus and S. Feuerriegel, "Sentiment analysis based on rhetorical structure theory:Learning deep neural networks from discourse trees," *Expert Systems with Applications*, vol. 118, pp. 65–79, 2019.

[39] L. Li, T.-T. Goh, and D. Jin, "How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis," *Neural Computing & Applications*, vol. 32, no. 9, pp. 4387–4415, 2018.

[40] P. Singhal and P. Bhattacharyya, *Sentiment Analysis and Deep Learning: A Survey*, Center for Indian Language Technology Indian Institute of Technology, Bombay, Indian, 2016.

[41] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019.

[42] A. S. M. Alharbi and E. de Doncker, "Twitter sentiment analysis with a deep neural network: an enhanced approach using user behavioral information," *Cognitive Systems Research*, vol. 54, pp. 50–61, 2019.

[43] G. Beigi, X. Hu, R. Maciejewski, and H. Liu, "An overview of sentiment analysis in social media and its applications in disaster relief," in *Sentiment Analysis and Ontology Engineering*, pp. 313–340, Springer, Cham, Switzerland, 2016.

[44] K. Sikka, T. Wu, J. Susskind, and M. Bartlett, "Exploring bag of words architectures in the facial expression domain," in *Proceedings of the European Conference on Computer Vision*, pp. 250–259, Florence, Italy, October 2012.

[45] R. Bekkerman and J. Allan, *Using Bigrams in Text Categorization*, Center of Intelligent Information Retrieval, UMass: Amherst, MA, USA, 2004.

[46] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with Fisher vectors for image categorization," in *Proceedings of the International Conference on Computer Vision*, pp. 1487–1494, Barcelona, Spain, November 2011.

[47] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: phrase and topic discovery, with an application to information retrieval," in *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, pp. 697–702, Omaha, Nebraska, October 2007.

[48] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 959–962, Santiago, Chile, August 2015.

[49] L. Yanmei and C. Yuda, "Research on Chinese micro-blog sentiment analysis based on deep learning," in *Proceedings of the 8th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 358–361, Hangzhou, China, December 2015.

[50] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: an attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[51] L. Arras, G. Montavon, K. R. Müller, and W. Samek, "Explaining Recurrent Neural Network Predictions in Sentiment Analysis,".

[52] S. N. Alsubari, S. N. Deshmukh, M. H. Al-Adhaileh, F. W. Alsaade, and T. H. Aldhyani, "Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5522574, 11 pages, 2021.

[53] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques. Cognitive informatics and soft computing," *Advances in Intelligent Systems and Computing*, vol. 768, 2018.

[54] Tensorflow and text preprocessing, Accessed 20 January 2022, https://www.tensorflow.org/api_docs/python/tf/keras/preprocess /text/Tokenize.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8512469, 13 pages
https://doi.org/10.1155/2022/8512469

Research Article

Enhancement of Detection of Diabetic Retinopathy Using Harris Hawks Optimization with Deep Learning Model

Nagaraja Gundluru,[1] Dharmendra Singh Rajput,[2] Kuruva Lakshmanna ,[2] Rajesh Kaluri,[2] Mohammad Shorfuzzaman ,[3] Mueen Uddin,[4] and Mohammad Arifin Rahman Khan [5]

[1]*School of Computer Science and Engineering, VIT, Vellore, India*
[2]*School of Information Technology and Engineering, VIT, Vellore, India*
[3]*Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*
[4]*School of Digital Science, University Brunei Darussalam, Gadong BE1410, Brunei Darussalam*
[5]*Department of Computer Science and Engineering, Bangladesh University, Dhaka 1207, Bangladesh*

Correspondence should be addressed to Mohammad Arifin Rahman Khan; arifin.khan@bu.edu.bd

Received 4 March 2022; Accepted 29 April 2022; Published 26 May 2022

Academic Editor: Muhammad Ahmad

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

In today's world, diabetic retinopathy is a very severe health issue, which is affecting many humans of different age groups. Due to the high levels of blood sugar, the minuscule blood vessels in the retina may get damaged in no time and further may lead to retinal detachment and even sometimes lead to glaucoma blindness. If diabetic retinopathy can be diagnosed at the early stages, then many of the affected people will not be losing their vision and also human lives can be saved. Several machine learning and deep learning methods have been applied on the available data sets of diabetic retinopathy, but they were unable to provide the better results in terms of accuracy in preprocessing and optimizing the classification and feature extraction process. To overcome the issues like feature extraction and optimization in the existing systems, we have considered the Diabetic Retinopathy Debrecen Data Set from the UCI machine learning repository and designed a deep learning model with principal component analysis (PCA) for dimensionality reduction, and to extract the most important features, Harris hawks optimization algorithm is used further to optimize the classification and feature extraction process. The results shown by the deep learning model with respect to specificity, precision, accuracy, and recall are very much satisfactory compared to the existing systems.

1. Introduction

Diabetes mellitus (DM) is a most important worldwide health concern, which causes a range of long-term complete impairments that have a significant influence on the patient and society, as the illness usually upsets people in their best fruitful ages. As per the latest alarming statistical facts released by the International Diabetes Federation (IDF) Diabetes Atlas 2021 document [1], there are 537 million (1 in 10) adults (aged 20–79 years) living with diabetes mellitus worldwide in 2021. This number is anticipated to rise to 643 million (1 in 9 adults) by 2030 and 784 million (1 in 8 adults) by 2045. Also, 81% (4 in 5 adults) with diabetes are living in middle- and low-income countries. Around 6.7 million deaths (1 every 5 seconds) occurred in 2021 due to DM. An assessed 44% of adults (240 million) living with diabetes in middle and low-income countries are left undiagnosed. In 2021, the global health expenditure caused by diabetes alone is evaluated at USD 996 billion with an increase of 316% over the last 15 years. Around 541 million (1 in 10) adults worldwide have been positioned at high risk of emerging type 2 diabetes due to diminished glucose tolerance levels. The top 10 countries with the maximum number of people with diabetes account for 68% of adults with diabetes [2].

The typical symptoms of type 1 diabetes are frequent urination or bedwetting, constant hunger, excessive thirst, lack of energy or fatigue, blurred vision, sudden weight loss, and diabetic ketoacidosis. Type 2 diabetes is the most common type of diabetes, accounting for over 90% of all diabetes worldwide. Type 2 diabetes also have similar symptoms to type 1 diabetes people but, in general, the status of people may be symptomless and less dramatic. Consequently, one-half of the people suffer from type 2 diabetes due to undiagnosed and continue with prediabetes condition. If this persists for a long duration, the people will develop the health complications such as kidney disease, neuropathy, retinopathy, foot ulceration, peripheral artery disease, heart disease or stroke, very poor healing of lower-limb ulcers, and visual impairment. An effective lifestyle management and consistent health check-ups or screening are a couple of the best precautionary treatments for preventing the development and progress of the mentioned complications [3].

People who have diabetes for a longer duration and poorly controlled blood sugar level, the more prospective they can develop diabetic retinopathy (DR), which causes loss of vision by the gradual destruction of the blood vessels of the retina (a light-sensitive tissue at the back of the eye) over a period of time [4]. The symptoms of DR include blurred vision, difficulty seeing well at night, seeing floaters or spots, and having a dark or empty spot in the centre of the vision. DR is categorized into two types as follows: nonproliferative diabetic retinopathy (NPDR) is the premature period of the disease in which symptoms will be insignificant or nonexistent. In NPDR, tiny blood vessels leak blood and other fluids due to weakness. Fluid might leak into the macula (a retinal tissue liable for clear central vision), which causes the macula to swell, resulting in cloudy or blurred vision. Proliferative diabetic retinopathy (PDR) is the further progressive form of the disease. At this phase, circulation complications deprive the retina of oxygen. As a result, new, fragile blood vessels can begin to grow in the retina and into the vitreous, and the gel-like fluid fills the back of the eye. The new blood vessels may leak blood into the vitreous (centre of eye), causing cloudy vision.

As far as health issues are concerned, the prevention is better than cure. Moreover, the treatment of a specific disease can be easy and effective if it is detected at the premature stage itself. Repetitive medical check-ups play a key role in DR because it exhibits mild symptoms until it is too late for actual treatment [5–7]. If you have either type 1 or type 2 diabetes, your physician may endorse that you have a comprehensive eye assessment straightaway once diabetes is diagnosed. A comprehensive eye scrutiny by an ophthalmologist or optometrist can detect edema (swelling) in the macula at the back of eye. The macula is critical to our central vision, which allows us to see in fine detail. Optical coherence tomography (to check the current status of the retina) and fluorescein angiography (to assess unusual blood vessel growth) are a couple of diagnosis approaches used by physicians through direct fundus examination or fundus photographs. Nevertheless, there is a shortage of knowledgeable ophthalmologists who can assess the fundus photographs to detect DR, which sometimes results in misdiagnosis. Moreover, there is a scarcity of knowledgeable physicians in local areas where there are more diabetic patients and the investigation procedure is onerous. Thus, the need for an automated diagnosis infrastructure for effective time and cost saving instead of manual diagnosis.

Deep learning (DL) is a slice of machine learning techniques, which has been extensively applied for the detection and classification of DR. The DL-based approaches were considered for computer-aided medical diagnosis of DR include support vector machine (SVM), convolutional neural networks (CNNs), restricted Boltzmann machines, sparse coding, and auto encoder [8–10]. All these approaches follow the common procedure, collect the retina image data set, do preprocessing, extract features, and classify using the DL technique. The retinal image data set consists two types of images, i.e., optical coherence tomography (OCT) and fundus colour images [11]. OCT images are either 2 or 3-dimensional, which gives significant evidence about retina thickness and its structure. The fundus images are large 2-dimensional image views of the top layer of retina [12]. The algorithm, which is used for extracting features from the image data set, plays a significant role in the outcome of the experiment. It is better to apply the best optimizer to find the significant features from the data set in the public domain, which might contain some insignificant features also. The principal component analysis (PCA) is the well-established unsupervised machine learning technique for the feature engineering process, which includes extraction and dimensionality reduction [13–15].

In reference [16], the authors proposed a hybrid model for early detection of DR. This model consists of feature selection using PCA, dimensionality reduction using firefly algorithm (FA) [17], and classification using deep learning technique. The FA was the meta-heuristic algorithm and was motivated by the flashing behaviour of fireflies. The algorithm mimics how fireflies interact using their flashing lights. The algorithm assumes that all fireflies are unisex, which means any firefly can be attracted by any other firefly; the attractiveness of a firefly is directly proportional to its brightness, which depends on the objective function. The FA algorithm suffers from trapping itself in the local optimum and has a sluggish convergence speed. So, FA is not a perfect solution to achieve our objective to optimize the dimensionality reduction process during the feature extraction from the image data set. The latest Harris hawks optimization (HHO) algorithm resolves hitches of a feature space comprising multi-modality, local optimal solutions, and misleading optima [18]. HHO is a population-based metaheuristic algorithm, inspired by the hunting strategy and cooperative behaviour of Harris hawks. As far as our knowledge, HHO does not apply so far for DR detection and classification during the feature engineering process. The main objectives of this research work are optimizing the feature set extracted from the image data set and proposing the best classification algorithm through experiments. The contributions of this work are as follows:

(i) A principal component analysis algorithm has been used for feature extraction and selection from the image data set.

(ii) A dimensionality reduction using the HHO algorithm has been proposed in order to optimize the feature set further.

(iii) A deep convolution neural network has been used for DR detection and classification.

(iv) The proposed work, a combination of deep neural network, PCA, and HHO, has been implemented and compared with various machine learning models such as KNN (k-nearest neighbour), SVM (support vector machine), and XGBOOT classification algorithm.

(v) The numerical outcomes of this experiment along with comparisons are encouraging and are better than entrenched metaheuristic methods in terms of accuracy, precision, recall, specificity, and sensitivity.

The subsequent sections of this study are organized as follows: a noticeable literature review has been presented in Section 2. Section 3 deliberates the proposed methodology and experimental setting details. Section 4 discusses the results achieved with the proposed method and compared with existing approaches and Section 5 articulates the conclusion and scope of future work.

2. Literature Review

Today's world of advances in deep learning (DL) has changed the way, in which healthcare is handled recently, which allows the medical practitioners effectively diagnose and treat diseases. Several researchers across the globe attempted to address the task at hand effectively. From the evolution of various DL-based classification, detection models have been drawn in this century. Many researchers are working in this area across the globe. Table 1 lists the review on deep learning applications in diabetic retinopathy and other datasets.

Currently, DL has begun to have an immense impact in various fields of health care. The rapid development of variations in DL techniques and the increased availability of data in health care have allowed the recording of impressive health care results [26, 27]. DL approaches can uncover details contained in a vast volume of health care data that are clinically important, which can be used for treatment, monitoring, prevention, and decision-making of health conditions. EHR processing, health behaviour reaction, and sound treatment retrieval from eye-related research, text, and classification are some of the implementation areas of DL. This will lead to simpler treatment for patients, with quicker and more effective monitoring. Usage of DL in medicine has converted the use of basic instruments, such as stethoscopes and thermometers, into computed tomography (CT), lithotripsy, ventilators, radio nuclear imaging, radiation therapy, ultrasound diagnostic devices, and dialysis, which has been used for highly adaptive treatment for traditional medical care capable of dealing with many dreaded illnesses. There is no question that health care treatment and facilities will see greater changes in even other sectors in the coming years to make them more competitive with qualitative programs.

Many international collaborative works have focused on applying DL-based algorithms for the diagnosis of diabetic retinopathy disease. The authors in reference [28] explored DL applications for a range of biomedical issues, including simple biological processes, patient classification, and patient care, and addressed if the DL can transform the mentioned activities or if the biomedical sphere faces special challenges. They notice that DL is yet to definitively overcome or revolutionize biomedicine in any of the important problems in the field after a comprehensive literature review, but promising progress is made on the previous significant works. While changes over previous works have been typically small, recent developments suggest that deep learning approaches can offer useful means to speed up or sustain human science. While progress has been made in connecting the prediction of a particular NN to input features, it remains an open challenge to understand how users can view these models to generate testable predictions of the system under research. However, DL is expected to show promising results in biological applications.

In the last few decades, diabetic retinopathy become a global medical problem among elderly people. The authors in reference [29] have explored DNN to predict diabetic retinopathy. They have proposed a principal component analysis (PCA)-based DNN model for the classification. The grey wolf optimization (GWO) algorithm is used to extract features of diabetic retinopathy dataset. The proposed model has compared their prominent results with pre-existing techniques like XGBoost, k-NN, Naive Bayes, and support vector machine (SVM). Reference [30] presented an approach for multimodal fusion in the contourlet domain based on weighted PCA. The main purpose of using contourlet transform is because of capability to capture visual geometrical anisotropy and structures. Further, weighted PCA minimizes the dimensionality of the source images and improves better selection of principal components.

Maximum and minimum fusion methods are used to fuse the decomposed coefficients. Image quality is assessed quantitatively using conventional fusion metrics to evaluate the fused image in terms of both information content and reconstruction quality. Reference [31] extracted the optimal features from the heart disease dataset through the proposed dimensionality reduction technique. The dataset used in this work was obtained from the heart disease dataset from the publicly available UCI machine learning repository. This dataset has 74 features. They have used 6 ML classifiers to validate the proposed model. Random forest (RF) integrated with chi-square-based PCA (CHIPCA) yielded the highest accuracy of 99.4% for Cleveland Hungarian (CH), 99.0% for Hungarian, and 98.7% for Cleveland datasets.

Reference [32] developed a model based on each patient's risk factors. For each stage of DR advancement, the author proposes a model to estimate the time and rate of progression. The proposed model could aid physicians in creating a customised follow-up program for patients depending on their disease stage and risk factors. Reference [33] attempted to optimize the energy utilization in the IoT networks through an optimal CH selection using a nature-inspired algorithm, HHO. The performance of the HHO-based CH model was analysed through several parameters such as load, Temperature, number of alive nodes, delay, and residual energy.

The authors in reference [34] used a model to process the images in order to distinguish the ocular structure and detect

Table 1: Review of deep learning applications in diabetic retinopathy and other datasets.

Reference	Dataset	Method used	Evaluation metrics	Research challenges
[19]	Diabetic retinopathy (DR) dataset consisted of 75137 images	5-Fold cross-validation and data-driven deep learning algorithm	Sensitivity, specificity, and AUC score	The results were not properly evaluated using typical state-of-the-art models
[20]	73 patients (122 eyes) were evaluated, 50.7% men and 49.3% women	RBM-1000, RBM-500, and OPF-1000	Sensitivity measured, specificity, and accuracy	More in-depth analysis on larger datasets was missing and accuracy may also be improved
[21]	14,186 retinal images and Messidor dataset with 1200 images	Deep learning algorithm	Accuracy, sensitivity, specificity, positive and negative predictive values, and AUC	Dataset is fixed and is not compared with other technique
[22]	128175 retinal images, EyePACS-1 dataset consisted of 9963 images, and Messidor-2 dataset with 1748 images	Deep convolutional neural network	The algorithm had 97.5% and 96.1% sensitivity and 93.4% and 93.9% specificity in the 2 validation sets	Limited dataset, system maybe failed to learn more complex features
[23]	Heart disease dataset	Effective heart disease prediction system using enhanced deep genetic algorithm and adaptive Harris hawks optimization-based clustering	Accuracy, precision, recall, specificity, and F-score	Requires more improvement in the learning process
[24]	COVID-CT-dataset: 349 and 397 images and CT scans for COVID-19 classification: 4,001 and 9,979 images	Hybrid learning and optimization approach CovH2SD-CovH2SD uses DL. HHO algorithm to optimize the hyperparameters	Accuracy, precision, recall, F1-score, and AUC performance metrics	Not good for multiclass classification
[25]	Hand gesture dataset from Kaggle repository	HHO is used for hyperparameter tuning of CNN for enhancing hand gesture recognition	Reduction of the burden on the CNN by reducing the training time and 100% accuracy for hand gesture classification is attained	Requires more improvement in the learning process

the existence of diabetic retinopathy in this study. For mapping an image with the relevant label, the model's parameters were optimized using the transfer-learning process. The author has used the medical fundus oculi images dataset for training and testing a model. The proposed study has a 97.78% accuracy rate for the accurate prediction of diabetic retinopathy in fundus oculi images. Many authors in reference [35] applied this dataset to DNN and CNN.

For these kinds of applications, image data alone is not sufficient. Many works have been done in this area but they did not find out a prominent method or model to improve the detection of diabetic retinopathy.

3. Proposed Method

3.1. Harris Hawks Optimization (HHO). In this section, we describe the mathematical model of the proposed method along with its usage for optimal results. In general, population-based metaheuristic optimization algorithms mimic the natural concept by considering a set of solutions (populations) during the optimization phase of each iteration [36]. The latest such algorithm, the Harris hawks optimization (HHO) technique, is a gradient-free, metaheuristic, swarm-based, nature-inspired algorithm [18]. The key concept of this algorithm is the utilization of dynamic and natural cooperative hunting behaviour of Harris' hawks for victims (medium-sized preys such as rabbits, hares, reptiles, ground squirrels, quail, birds, and other rodents) [37]. The hunting patterns exhibit the Harris' hawk's intelligent behaviour despite the complex dynamic environment and escaping nature (zig-zag gestures) by victims. So, an optimization technique, which simulates the behaviour of this hunting pattern, can give better results compared to the existing techniques. The main advantages of the HHO algorithm are the possibility of getting a global optimal solution, high convergence speed, high accuracy, and better quality. Consequently, the HHO algorithm can be applied to solve various optimization problems in the engineering domain such as feature extraction, design and development of a model, pattern recognition, and electrical and electronics optimal design applications [33].

The working mechanism of the HHO algorithm in different stages is depicted in Figure 1 and the next subsections follow its description in a stage-by-stage manner. Broadly, the algorithm consists of two stages: exploration and exploitation. The exploration stage models the behaviour of the Harris hawks search process to detect and spot the victims. The exploitation stage models the intelligent hunt.

3.1.1. Exploration Stage. Usually, the hawks who have powerful eyes will spend hours of time patiently to track and detect the victims by waiting, observing, and

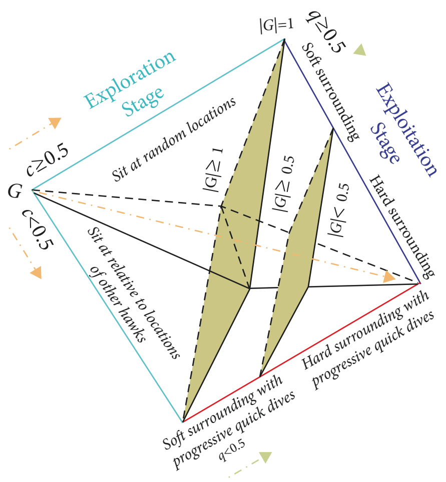


Figure 1: Various stages during the HHO algorithm.

monitoring the desert spot. In the context of HHO for optimization purposes, the hawks are observed as candidate solutions and the best of the candidate solutions at each iteration process are observed as intended optimal solution or victim. Initially, the hawks can perch at location of the spot in two possible ways, either close enough relative to the location of the other family hawks and the victim or at random locations on high trees. Both the ways are considered with an equal chance of probability and the same is modelled in equation (1) with $c \geq 0.5$ for random locations and $c < 0.5$ for relative locations. Several hawks cooperatively move towards the victim from different directions to surprise it. The change in the location of the hawks at each iteration during the exploration stage is mathematically modelled as follows:

$$C(i+1) = \begin{cases} C_{\text{random}}(i) - n_1\left|C_{\text{random}}(i) - 2n_2C(i)\right| \\ c \geq 0.5 \\ \left(C_{\text{victim}}(i) - C_{avg}(i)\right) - n_3\left(L + n_4(U - L)\right) \\ c < 0.5 \end{cases}, \tag{1}$$

where $C(i)$ is the current location vector of hawks. $C(i+1)$ is the location vector of hawks in the next iteration i. $C_{\text{victim}}(i)$ is the location of victim. C_{avg} is the average location of the current population of hawks. The c, n_1, n_2, n_3, and n_4 are random variables whose values are to be updated at each iteration between 0 and 1. The upper and lower bounds of these variables are considered as U and L, respectively. The randomly picked hawk from the current population is represented by $C_{\text{random}}(i)$. The average location of the hawks is calculated using equation (2):

$$C_{\text{avg}}(i) = \frac{1}{M}\sum_{k=1}^{M} C_k(i), \tag{2}$$

where $C_k(i)$ denotes the position of hawks after iteration i and M represents the number of hawks.

3.1.2. Exploitation Stage.

In this stage, the hawks execute the sudden pounce on the envisioned victim, which was spotted in the exploration stage. In this context, several styles of chasing will take place due to the victims may execute different escaping strategies according to the dynamic environment. In the HHO, four potential tactics were provided based on chasing styles of the hawks and escaping attitudes of the victim. Let q represents escaping chance of a victim from sudden pounce, successful escape with $q < 0.5$, and unsuccessful escaping with $q \geq 0.5$. However, the hawks have their own strategies such as soft or hard surround to catch the victim. In other words, the retained energy of the victim gives directions to the hawks to do either soft or hard encircle the victim. To win this hunting process, the hawks will try to reach closer and closer to the envisioned victim and then cooperatively executes the sudden pounce to kill the same. Simultaneously, the victim looses its energy while using escaping strategies. After some time, the victim energy will be exhausted and this context leads to catch the same easily by hawks.

The modelling of the victim energy plays a vital role here and it is defined as follows:

$$G = 2G_0\left(1 - \frac{i}{I}\right), \tag{3}$$

where G represents the victim escaping energy. I is the maximum possible iterations. G_0 is the initial energy state and it is a random variable whose value changes between -1 and 1 in each iteration. The victim is strengthening if the value of G_0 increases from 0 to 1 and weakening if the value of G_0 decreases from 0 to -1. However, the dynamic value of escaping energy G is always in the downtrend during iteration by iteration. The HHO moves to the exploration stage to search for another victim if $|G| \geq 1$ and continues in the exploitation stage if $|G| < 1$. During the exploitation stage, the algorithm switches between soft (if $|G| \geq 0.5$) and hard (if $|G| < 0.5$) surrounding of the victim. So, the hawks will execute any one of the following four promising tactics based on the G, the escaping energy of the victim, and q, the escaping chance of a victim.

3.1.3. Soft Surrounding.

This scenario is applicable, if $|G| \geq 0.5$ and $q \geq 0.5$. Here, the victim tries to escape from hawks through confusing zig-zag movements/jumps with having sufficient energy. At this juncture, the hawks follow a soft surrounding approach to do sudden pounce in several rounds of attempts by making the victim energy exhausted. This soft surrounding approach has been modelled as follows:

$$C(i+1) = \Delta C(i) - G\left|SC_{\text{victim}}(i) - C(i)\right|. \tag{4}$$

$$\Delta C(i) = C_{\text{victim}}(i) - C(i), \tag{5}$$

where $\Delta C(i)$ is the difference between the location vector of the victim and the present location in iteration i. The strength of the victim to do zig-zag movement or jump is represented by $S = 2(1 - \text{n}5)$. Here, S value updates dynamically in each iteration to mimic the movements or jumps by victims. N5 is a random variable whose value is between 0 and 1.

3.1.4. Hard Surrounding. This scenario is applicable, if $|G| < 0.5$ and $q \geq 0.5$. Here, the victim energy is so exhausted and its escaping energy is low. At this juncture, the hawks follow a hard surrounding approach to do sudden pounce to catch the intended victim. Here, the current locations of the hawks are updated using equation (6) and this concept with one hawk is illustrated in Figure 2:

$$C(i+1) = C_{\text{victim}}(i) - G|\Delta C(i)|. \tag{6}$$

3.1.5. Soft Surrounding with Progressive Quick Dives. This scenario is applicable, if $|G| \geq 0.5$ and $q < 0.5$. This means, the hawks construct a more intelligent soft surrounding strategy due to the victim still has sufficient energy to escape successfully [37]. In real time, the victim chooses random escaping patterns and leapfrog movements [38]. To model this concept, the levy flight (LF) random walk notion is adopted in to the HHO algorithm [39,40]. The LF concept helps to mimic the real-time zig-zag movements of victim and the hawks' sudden, irregular, and quick dives around the escaping victim. The hawks dynamically execute many sudden quick dives around the victim, and update their location and direction of attack progressively according to the escaping behaviour of victim. The literature proved that the LF-based actions are the optimal searching strategies for hunters in nondestructive searching circumstances. Moreover, the LF-based movement of patterns is common in the victims such as rabbit, monkeys, and sharks. This motives the utilization of LF-based movements within the HHO algorithm.

To catch the victim in the adverse condition, the hawks' real behaviour is to choose the best possible dive at every step and reach closer by closer to the victim. To mimic this behaviour, the hawks can compute their next movement based on the following rule:

$$A = C_{\text{victim}}(i) - G\left|SC_{\text{victim}}(i) - C(i)\right|. \tag{7}$$

After computing the next move, the hawks compare the latest move with the previous one. If the latest is better than the previous one, then they choose and execute the same. Otherwise, they choose sudden, irregular, and quick dives around the escaping victim. Sometimes, the hawks choose to dive according to LF-based patterns as defined in the following equation:

$$B = A + V * LF(D), \tag{8}$$

where D is the problem dimension, V is a random 1Xd dimensional vector, and the levy flight (LF) function is computed using the following equation:

$$LF(x) = 0.01 \times \frac{p \times \sigma}{|q|^{1/\beta}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma(1+\beta/2) \times \beta \times 2^{(\beta-1/2)}}\right)^{1/\beta}, \tag{9}$$

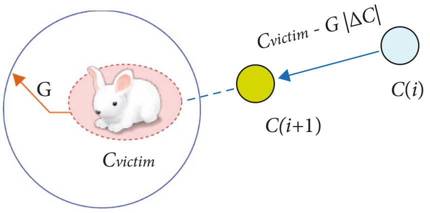


Figure 2: Illustration of overall vectors in the strategy of hard surrounding with one hawk.

where β is a constant with a value of 1.5 and p and q are random variables with a value between 0 and 1.

Therefore, in the soft surrounding phase, the latest positions of the hawks can be defined in the final strategy using the following equation:

$$C(i+1) = \begin{cases} A & if\ F(A) < F(C(i)) \\ B & if\ F(B) < F(C(i)) \end{cases}, \tag{10}$$

where A and B are attained using equations (7) and (8), respectively. An instance of the above modelling concept for one hawk is illustrated in Figure 3. This demonstration also contains the leapfrog movements based on LF through possible iterations. The LF-based patterns are depicted with coloured dots on trial and then the HHO algorithm touches position B. In every stage, the next better location is chosen as A or B. The same concept is applicable to all hawks during searching.

3.1.6. Hard Surrounding with Progressive Quick Dives. This scenario is applicable, if $|G| < 0.5$ and $q < 0.5$. This means, the hawks execute hard surrounding before the sudden pounce to catch and kill the victim due to the victim energy has been exhausted. At victim side, this condition is similar to the soft surrounding, but the hawks move closer by closer by reducing the average position with the escaping victim. Hence, the hawks update their locations in the context of hard surrounding using the following equation:

$$C(i+1) = \begin{cases} A & if\ F(A) < F(C(i)) \\ B & if\ F(B) < F(C(i)) \end{cases}, \tag{11}$$

where A and B are attained using equations (12) and (13), respectively:

$$A = C_{\text{victim}}(i) - G\left|SC_{\text{victim}}(i) - C_{\text{avg}}(i)\right|. \tag{12}$$

$$B = A + V \times LF(D), \tag{13}$$

where $C_{avg}(i)$ is computed using equation (2). An instance of the above modelling concept with overall vectors for one hawk is illustrated in Figure 4. This demonstration also contains the leapfrog movements based on LF through some iterations. The LF-based patterns are depicted with coloured dots and the next better location is provided by A or B for the next iteration.

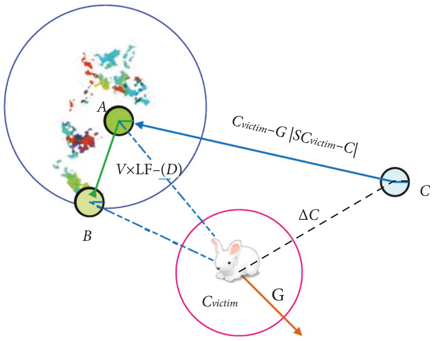


Figure 3: Illustration of overall vectors in the strategy of soft surrounding with progressive quick dives.

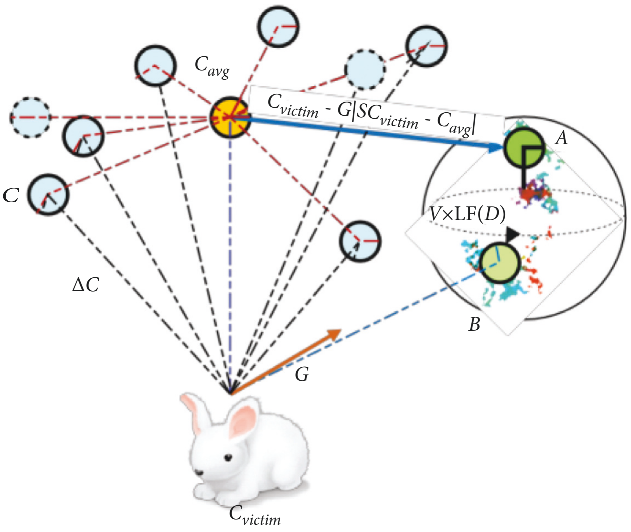


Figure 4: Illustration of overall vectors in the strategy of hard surrounding with progressive quick dives.

The key feature of HHO algorithm, which has a productive impact during the exploitation stage, is that it exhibits a series of search strategies and picks the best move at each iteration. The HHO algorithm tries to improve the superiority of the solution throughout the optimization process by progressive choice arrangement by search agents. The usage of adaptive and time-varying constituents allows the HHO algorithm to resolve the hitches of a feature space comprising multimodality, local optimal solutions, and misleading optima. The candidate solutions take help from the strength of randomized moves in congruent during the stages of exploration and exploitation learning.

Finally, the computational complexity of the HHO algorithm has been analysed and computed based on three tasks: initialization, fitness evaluation, and updating of hawks. If there are N number of hawks, then the computational complexity of the initialization process is O(N). If D is the dimension of definite problem and M is the maximum number of iterations then the computational complexity of the updating procedure is $\mathrm{O}(M \times N) + \mathrm{O}(M \times N \times \mathrm{D})$, which is a sum of searching for the best location and updating the location vector of all hawks. Therefore, the computational complexity of HHO is $\mathrm{O}(N \times (M + \mathrm{MD} + 1))$.

4. Results and Explanation

The dataset, the experimental framework, the metrics, and the experimental outcomes are all discussed in this section. There were 1151 instances and 20 attributes in the diabetic retinopathy dataset used in this study. Table 1 [16] lists the characteristics of the dataset used in this study. Except for the output layer, all of the layers used the Softsign activation function. The experiment was conducted using the Diabetic Retinopathy Debrecen dataset [41] from the UCI ML library. The features extracted from the image dataset were used to create the attributes in this dataset. The Python experimentation was carried out on a personal machine with 8 GB of RAM. The results of the current proposed work are outperformed than the existing methodologies discussed in the literature review.

4.1. Metrics Used in the Evaluation of the Model.

The suggested model is evaluated using the metrics listed below.

Accuracy. In the testing phase, it is the percentage of proper predictions made by a classifier and the actual value of the label. It is also known as the ratio of the number of right assessments to the total number of assessments. The following equation can be used to calculate accuracy:

$$\text{Accuracy} = \frac{(TNeg + TPos)}{(TNeg + TPos + FNeg + FPos)}. \tag{14}$$

Here, TPos is true positives, TNeg is true negatives, FPos is false positives, and FNeg is false negatives.

When the class label of a record is available in the given dataset and the classifier indicates positive for that record, then it is called a true positive. When the class label of a record is not available and the classifier forecasts the class label, then it is called a true negative. When the class label of a record is accessible and the classifier expects a negative for that record, then it is called a false negative. When the class label of a record is not available in a dataset and the classifier estimates a positive class, then it is called a false positive.

Specificity. It is the percentage of true negatives successfully detected by the classifier while taking the test. The following equation is used to calculate it:

$$\text{Specificity} = \frac{(Tneg)}{(Tneg + Fpos)}. \tag{15}$$

Sensitivity. During testing, it is the percentage of true positives successfully detected by the classifier. The following equation is used to calculate it:

$$\text{Sensitivity} = \frac{(TPos)}{(TPos + FNeg)}. \tag{16}$$

Precision. Precision is an important metric for measuring exactness. It expresses how much of the total forecasted positive occurrences the classifier identified as positive, as shown in the following equation:

```
Inputs: All features extracted from fundus image data set.
Outputs: Optimal feature set
          Initialize the population randomly, $C_i(i = 1,2,\ldots,M)$
    Set C          while *until end condition met* do Compute the hawk's fitness values. Victim as the victim best location
    end
          for *each hawk* $(C_i)$ do
          Update the strength of jump $S$ and initial energy $G_0$
    end
          $G_0 = 2\mathrm{rand}() - 1,\ S = 2(1 - \mathrm{rand}())$
          Update the energy $G$ using (3)
    end
          if $|G| \geq 1$ then
          The vector of location is updated using Eq.
    end
          if $|G| < 1$ then
          if $|G| \geq 0.5$ *and* $q \geq 0.5$ then
          The vector of location is updated using (4)
    end
          else if $|G| < 0.5$ *and* $q \geq 0.5$ then
          The vector of location is updated using (6)
    end
          else if $|G| \geq 0.5$ *and* $q < 0.5$ then
          The vector of location is updated using (10)
          else if $|G| < 0.5$ *and* $q < 0.5$ then
          The vector of location is updated using (11)
    end
return $C_{victim}$
```

Algorithm 1: HHO Algorithm Pseudocode [18].

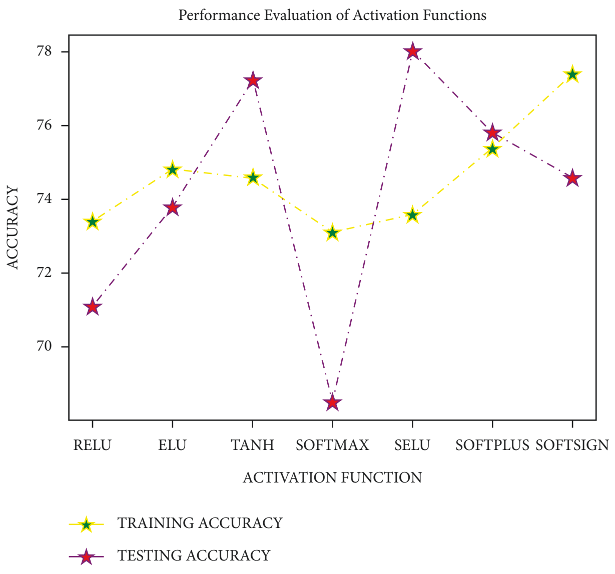


Figure 5: Analysis of activation functions.

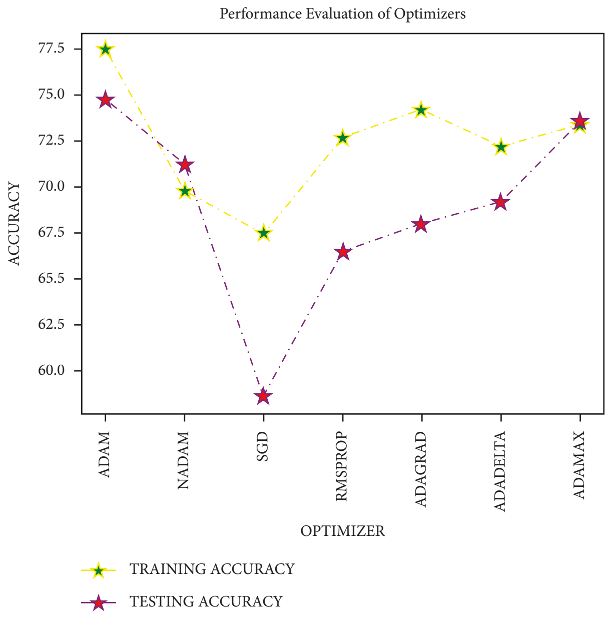


Figure 6: Analysis of optimizers.

$$\text{Precision} = \frac{(Tpos)}{(Tpos + Fpos)}. \quad (17)$$

Recall. The percentage of positive cases classified as positive by the classifier is determined by recall. When there is a large cost connected with the false negative, as stated in equation (18), the recall is a performance parameter used to predict the optimal model:

$$\text{Recall} = \frac{(Tpos)}{(Tpos + Fneg)}. \quad (18)$$

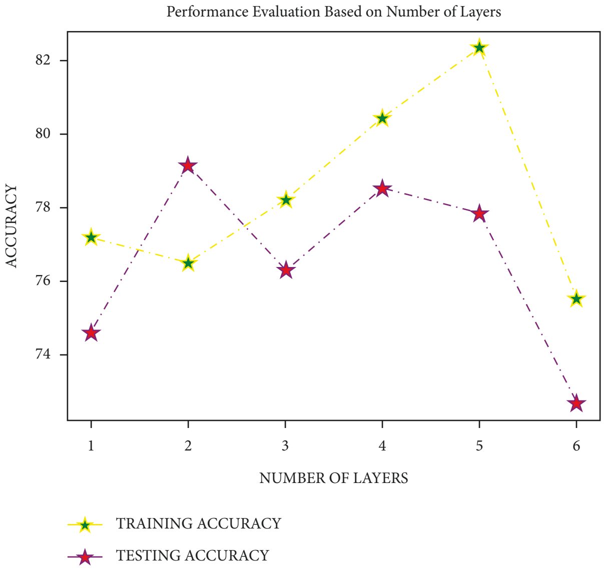


Figure 7: Analysis based on the number of layers.

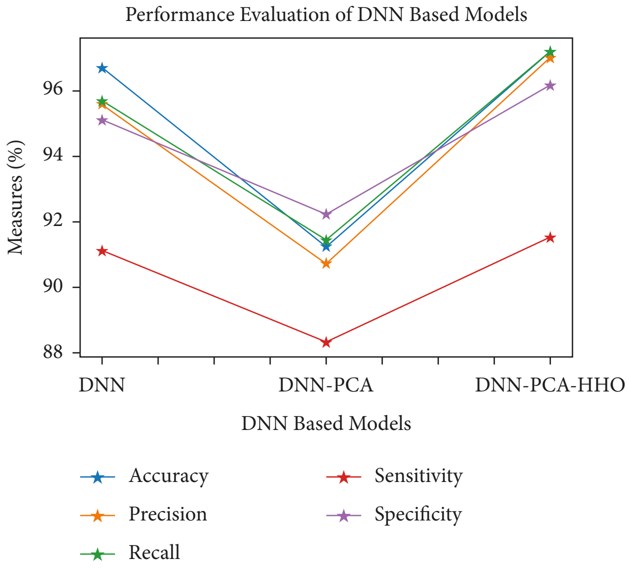


Figure 9: Analysis of DNN-based models.

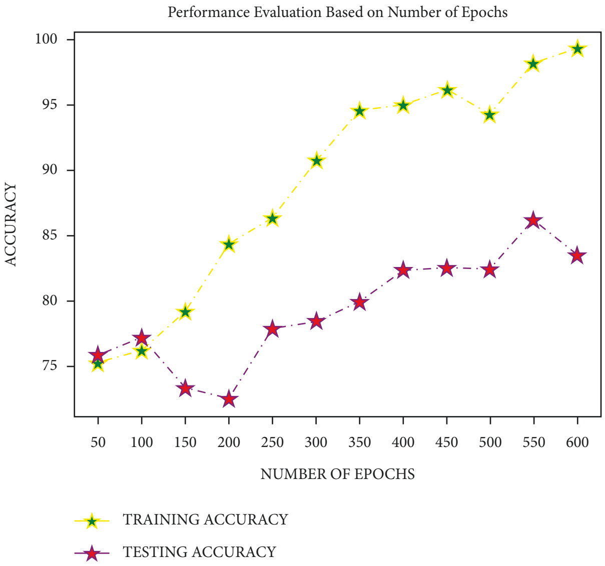


Figure 8: Analysis based on the number of epochs.

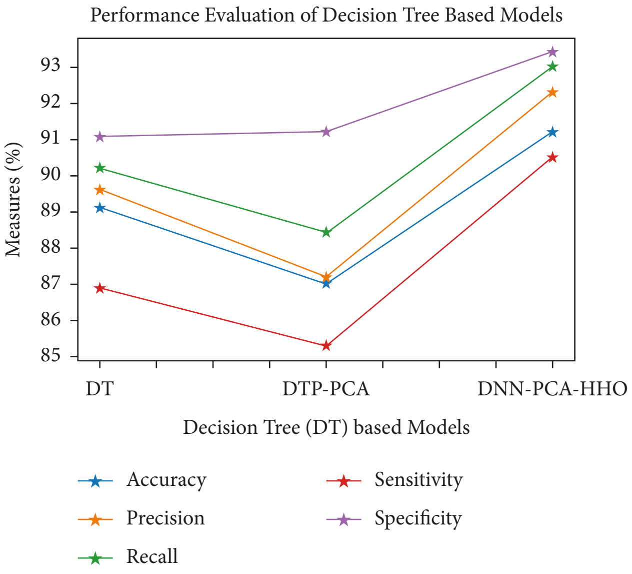


Figure 10: Analysis of DT-based models.

4.2. Performance Analysis. The DNN-PCA model was built using a sequential strategy for testing the proposed model. The dataset was divided into two sections for cross validation, with 80% utilised for training and 20% used for verifying every 64 records (batch size). To determine the optimal activation function for a dataset of 50 epochs and a batch size of 64, testing was performed on many activation functions such as ReLU, ELU, tanh, Softmax, SELU, Softplus, and Softsign. The Softsign activation function, as shown in Figure 5, provided the best average training and testing accuracy. As a result, the Softmax activation function is used to evaluate the model on dense layers.

The investigation was completed on the dataset with 50 epochs and a batch size of 64 to obtain the optimal optimizer in the layers of deep neural networks using many optimizers such as Adam, NAdam, SGD, rmsprop, adagrad, adadelta, and adamax. As shown in Figure 6, the Adam optimizer offered the highest level of accuracy. As a result, the Adam optimizer is selected for input and other dense layers experimentation. For the output layer, a sigmoid optimizer is employed.

The DR dataset was explored with many layers with 50 epochs, Adam optimizer at input and dense layers, Softsign activation function, sigmoid optimizer at the output layer, and 64 as batch size to pick the number of layers in DNN for experimenting. The model showed an excellent training and testing accuracy with 5 layers, as shown in Figure 7, with the accuracy level starting to decrease with 6 levels. As a result, a five-layer deep neural network was deployed in the experiment.

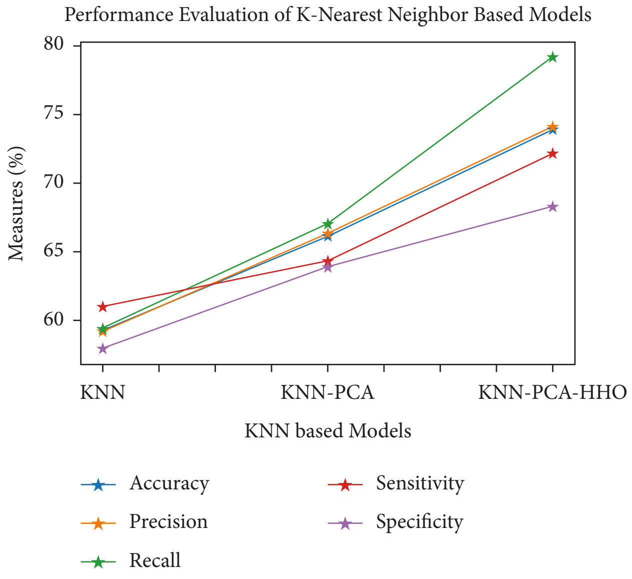


Figure 11: Analysis of KNN-based models.

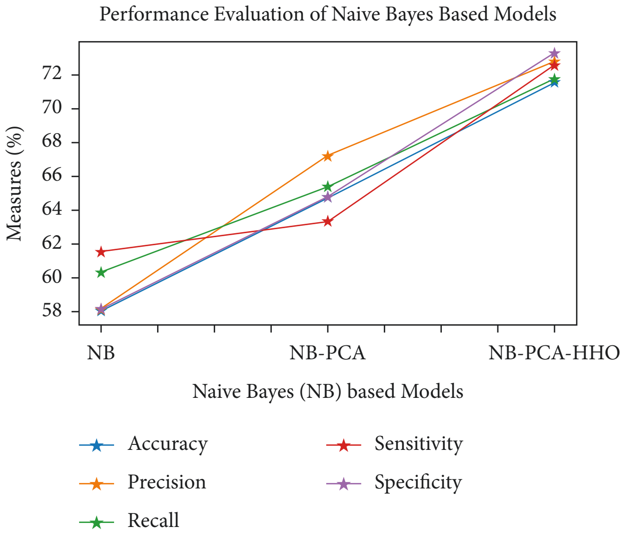


Figure 12: Analysis of NB-based models.

The DR data set was tested utilising Softsign activation with five intermediate layers, sigmoid optimizer at the output layer, Adam optimizer at input and dense layers, and 64 as batch size to determine the number of epochs. The model provided better results in providing a good average training and testing accuracy with 600 epochs, as shown in Figure 8, with the testing accuracy beginning to dip with 650 epochs. As a result, 600 epochs were used to train a deep neural network.

The number of components chosen for the PCA in the experimental study was 0.9%, that is, to retain 99% of the information

Figures 9–14 show how the accuracy, precision, recall, sensitivity, and specificity of machine learning models are evaluated. These graphs show that ML models based on PCA-HHO provided the best results rather than the other

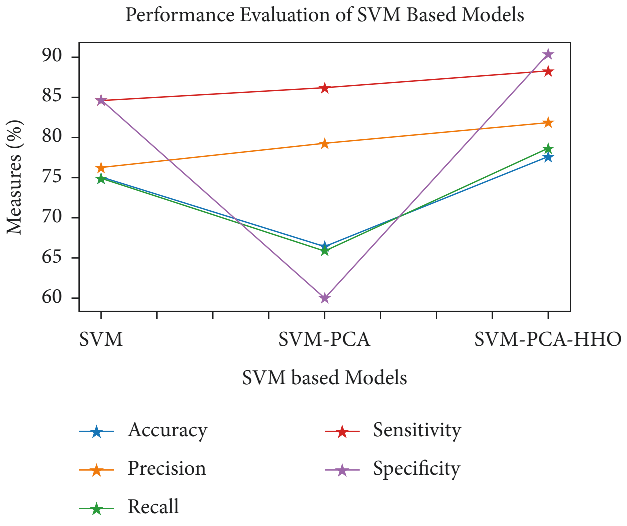


Figure 13: Analysis of SVM-based models.

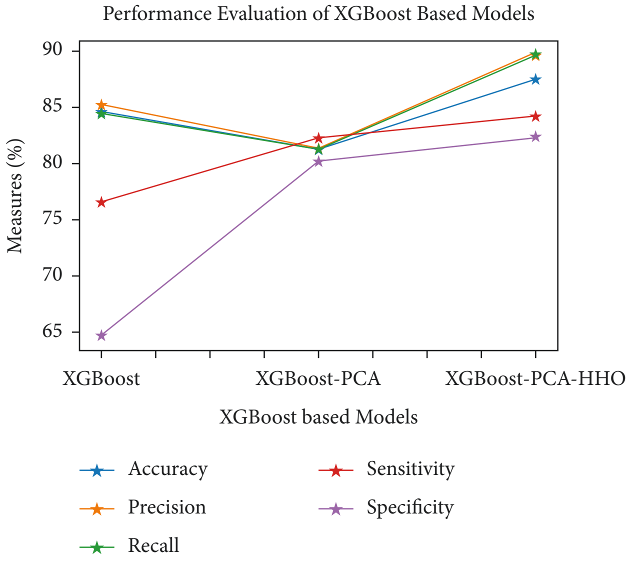


Figure 14: Analysis of XGBoost-based models.

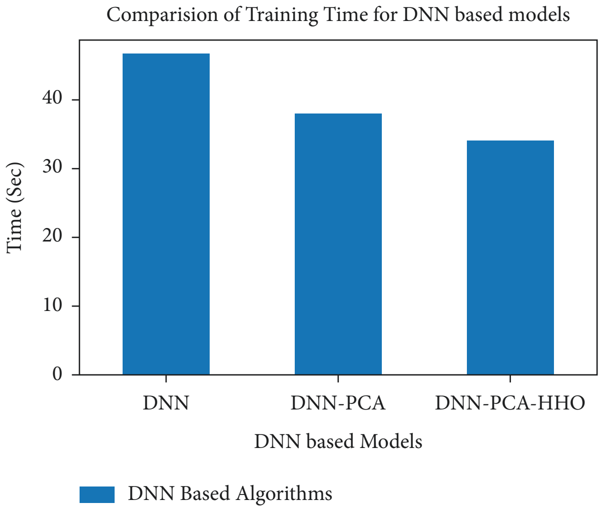


Figure 15: Training time analysis for DNN-based models.

Table 2: Summary of the experimental results.

Metric ⟶method ↓	DNN	DNN-PCA	DNN-PCA-HHO	DT	DT-PCA	DT-PCA-HHO	KNN	KNN-PCA	KNN-PCA-HHO	NB	NB-PCA	NB-PCA-HHO	SVM	SVM-PCA	SVM-PCA-HHO	XGBoost	XGBoost-PCA	XGBoost-PCA-HHO
Accuracy	96.7	91.2	97	89.1	87	91.2	59.4	66	73.9	58	65	71.5	75	66	77.5	84.5	81	88
Precision	95.6	90.7	97	89.6	87	92.3	59.2	66	74.1	58	67	72.7	76	79	81.7	85.2	81	90
Recall	95.7	91.4	97	90.2	88	93.0	59.3	67	79.1	60	65	71.6	75	66	78.6	84.3	81	90
Sensitivity	91.1	88.3	91	86.9	85	90.5	60.9	64	72.1	61	63	72.5	84	86	88.2	76.5	82	84
Specificity	95.1	92.2	96	91.1	91	93.4	57.9	64	68.2	58	65	73.2	85	60	90.3	64.6	80	82

two scenarios: ML with PCA and ML without dimensionality reduction. When dimensionality reduction and feature engineering concepts are included or excluded from ML methods, it is found that the suggested model, DNN-PCA-HHO, outperforms the other hybrid ML techniques.

The following are the highlights of the results relevant to the suggested model:

(1) When compared to other prominent ML hybrid models, the DNN-PCA-HHO model surpasses them.

(2) When PCA is used alone on DNN and other ML algorithms, the performance measurements deteriorate slightly. However, the amount of time spent training is minimised.

(3) PCA + HHO, on the other hand, improves the performance of ML algorithms while cutting training time in half, as shown in Figure 15.

(4) When the original dataset was employed, it fell victim to overfitting, which had a negative impact on the results. The performance has been improved even though the size of the data set was increased by two times and the same is shown in the Figure 13.

Table 2 describes the findings of the analysis.

5. Conclusion

The proposed system used principal component analysis for extracting the best features. The dataset for the proposed model is gathered from the publicly accessible UCI machine learning repository, which contained redundant and unnecessary features in its raw form. The Harris hawks optimization algorithm outperformed the process of selection and extraction of the required features from the dataset. The proposed model results were compared to the outcomes of the most popular machine learning algorithms, with the findings demonstrating the model's superiority in terms of specificity, precision, accuracy, recall, and sensitivity. Nevertheless, in the event of a low-dimensional dataset, the model's ability to perform well may be limited by the possibility of overfitting. Therefore, our proposed model encourages the researchers to pursue similar research in a variety of other health disciplines using high-dimensional data.

Data Availability

The data can be provided based on the request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the researchers of Taif University,Taif, Saudi Arabia with project number TURSP-2020/79.

References

[1] D. Atlas, "International diabetes federation," in *IDF Diabetes Atlas*, International Diabetes Federation, Brussels, Belgium, 10th edn edition, 2021.

[2] S. D. Solomon, E. Chew, E. J. Duh et al., "Diabetic retinopathy: a position statement by the american diabetes association," *Diabetes Care*, vol. 40, no. 3, pp. 412–418, 2017.

[3] E. J. Duh, J. K. Sun, and A. W. Stitt, "Diabetic retinopathy: current understanding, mechanisms, and treatment strategies," *JCI insight*, vol. 2, no. 14, 2017.

[4] A. W. Stitt, T. M. Curtis, M. Chen et al., "The progress in understanding and treatment of diabetic retinopathy," *Progress in Retinal and Eye Research*, vol. 51, pp. 156–186, 2016.

[5] D. S. W. Ting, G. C. M. Cheung, and T. Y. Wong, "Diabetic retinopathy: global prevalence, major risk factors, screening practices and public health challenges: a review," *Clinical and Experimental Ophthalmology*, vol. 44, no. 4, pp. 260–277, 2016.

[6] K. Lakshmanna, N. Khare, and N. Khare, "Fdsmo: frequent dna sequence mining using fbsb and optimization," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 4, pp. 157–166, 2016.

[7] K. Lakshmanna and N. Khare, "Mining dna sequence patterns with constraints using hybridization of firefly and group search optimization," *Journal of Intelligent Systems*, vol. 27, no. 3, pp. 349–362, 2018.

[8] G. Reddy, N. Khare, and N. Neelu, "Hybrid firefly-bat optimized fuzzy artificial neural network based classifier for diabetes diagnosis," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 4, pp. 18–27, 2017.

[9] K. Lakshmanna, N. Khare, and N. Khare, "Constraint-based measures for dna sequence mining using group search optimization algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 3, pp. 91–100, 2016.

[10] S. M. Basha and D. S. Rajput, "A roadmap towards implementing parallel aspect level sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29463–29492, 2019.

[11] M. Soares, C. Neves, I. P. Marques et al., "Comparison of diabetic retinopathy classification using fluorescein angiography and optical coherence tomography angiography," *British Journal of Ophthalmology*, vol. 101, no. 1, pp. 62–68, 2017.

[12] J. Sahlsten, J. Jaskari, J. Kivinen et al., "Deep learning fundus image analysis for diabetic retinopathy and macular edema grading," *Scientific Reports*, vol. 9, no. 1, pp. 1–11, 2019.

[13] B. Xiao, "Principal component analysis for feature extraction of image sequence,"vol. 1, pp. 250–253, in *Proceedings of the 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering*, vol. 1, IEEE, Chengdu, China, June 2010.

[14] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 374, no. 2065, p. 20150202, Article ID 20150202, 2016.

[15] S. Bhattacharya, S. R. K. S, P. K. R. Maddikunta, R. Kaluri, and S. Singh, "A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu," *Electronics*, vol. 9, no. 2, p. 219, 2020.

[16] T. R. Gadekallu, N. Khare, S. Bhattacharya et al., "Early detection of diabetic retinopathy using pca-firefly based deep learning model," *Electronics*, vol. 9, no. 2, p. 274, 2020.

[17] H. Wang, W. Wang, X. Zhou et al., "Firefly algorithm with neighborhood attraction," *Information Sciences*, vol. 382-383, pp. 374–387, 2017.

[18] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[19] R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," *Ophthalmology*, vol. 124, no. 7, pp. 962–969, 2017.

[20] M. Bader Alazzam, F. Alassery, and A. Almulihi, "Identification of diabetic retinopathy through machine learning," *Mobile Information Systems*, vol. 2021, 2021.

[21] M. Baget-Bernaldiz, R.-A. Pedro, E. Santos-Blanco et al., "Testing a deep learning algorithm for detection of diabetic retinopathy in a Spanish diabetic population and with messidor database," *Diagnostics*, vol. 11, no. 8, p. 1385, 2021.

[22] V. Gulshan, L. Peng, M. Coram et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.

[23] R. B. Ramasamy, R. Selvaraj, and Y. Venila, "Classification of heart disease using adaptive Harris hawk optimization-based clustering algorithm and enhanced deep genetic algorithm," *Soft Computing*, vol. 11, 2021.

[24] S. M. Balaha Hm and E. M. El-Gendy, "Covh2sd: A covid-19 detection approach based on Harris hawks optimization and stacked deep learning," *Expert Systems with Applications*, vol. 186, pp. 1–20, 2021.

[25] "Hand gesture recognition based on a Harris hawks optimized convolution neural network," *Computers & Electrical Engineering*, vol. 100, Article ID 107836, 2022.

[26] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[27] 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.

[28] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones et al., "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 141, Article ID 20170387, 2018.

[29] T. R. Gadekallu, N. Khare, S. Bhattacharya, Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *Journal of Ambient Intelligence and Humanized Computing*, vol. 83, pp. 1–14, 2020.

[30] A. Aisha Moin, V. Vikrant Bhateja, and A. Anuja Srivastava, "Weighted-PCA based multimodal medical image fusion in contourlet domain," in *Proceedings of the international congress on information and communication technology*, pp. 597–605, Springer, Baghdad Iraq, January 2016.

[31] A. K. Garate-Escamila, A. H. El Hassani, and E. Andres, "Classification models for heart disease` prediction using feature selection and pca," *Informatics in Medicine Unlocked*, vol. 19, Article ID 100330, 2020.

[32] D. J. Chen, J. C. Kuo, A. J. Wright et al., "Determining risk factors that affect progression in patients with nonproliferative diabetic retinopathy," *Journal of ophthalmology*, vol. 2021, 2021.

[33] K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, S. Bhattacharya, P. Hegde, and S. Singh, "Energy optimization for green communication in iot using Harris hawks optimization," *IEEE Transactions on Green Communications and Networking*, p. 1, 2022.

[34] A. Ayala, T. Ortiz Figueroa, B. Fernandes, and F. Cruz, "Diabetic retinopathy improved detection using deep learning," *Applied Sciences*, vol. 11, no. 24, Article ID 11970, 2021.

[35] S. Dutta, B. Manideep, S. M. Basha, R. D. Caytiles, and N. Iyengar, "Classification of diabetic retinopathy images by using deep learning models," *International Journal of Grid and Distributed Computing*, vol. 11, no. 1, pp. 89–106, 2018.

[36] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, N J, United States, vol. 74, 2009.

[37] J. C. Bednarz, "Cooperative hunting harris' hawks (parabuteo unicinctus)," *Science*, vol. 239, no. 4847, pp. 1525–1527, 1988.

[38] N. E. Humphries, N. Queiroz, J. R. M. Dyer et al., "Environmental context explains Lévy and Brownian movement patterns of marine predators," *Nature*, vol. 465, no. 7301, pp. 1066–1069, 2010.

[39] M. F. Shlesinger, "Levy flights: variations on a theme," *Physica D: Nonlinear Phenomena*, vol. 38, no. 1-3, pp. 304–309, 1989.

[40] G. Viswanathan, V. Afanasyev, S. V. Buldyrev et al., "Levy flights in random searches," *Physica A: Statistical Mechanics and Its Applications*, vol. 282, no. 1-2, pp. 1–12, 2000.

[41] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowledge-Based Systems*, vol. 60, pp. 20–27, 2014.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9867270, 1 page
https://doi.org/10.1155/2023/9867270

Retraction

Retracted: IoT-Based Hybrid Ensemble Machine Learning Model for Efficient Diabetes Mellitus Prediction

Computational Intelligence and Neuroscience

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Padhy, S. Dash, S. Routray, S. Ahmad, J. Nazeer, and A. Alam, "IoT-Based Hybrid Ensemble Machine Learning Model for Efficient Diabetes Mellitus Prediction," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2389636, 11 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2389636, 11 pages
https://doi.org/10.1155/2022/2389636

RETRACTED

Research Article

IoT-Based Hybrid Ensemble Machine Learning Model for Efficient Diabetes Mellitus Prediction

Sasmita Padhy,[1] Sachikanta Dash,[2] Sidheswar Routray ,[3] Sultan Ahmad ,[4] Jabeen Nazeer ,[4] and Afroj Alam [5]

[1]*School of Computing Science and Engineering, VIT Bhopal University, Bhopal, Madhya Pradesh, India*
[2]*Department of Computer Science and Engineering, GIET University, Gunupur, Odisha, India*
[3]*Department of Computer Science and Engineering, School of Engineering, Indrashil University, Rajpur, Mehsana, Gujarat, India*
[4]*Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia*
[5]*Department of Computer Science, Bakhtar University, Kabul, Afghanistan*

Correspondence should be addressed to Afroj Alam; aalam@bakhtar.edu.af

Received 1 April 2022; Revised 25 April 2022; Accepted 30 April 2022; Published 18 May 2022

Academic Editor: Muhammad Ahmad

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

Nowadays, there is a growing need for Internet of Things (IoT)-based mobile healthcare applications that help to predict diseases. In recent years, several people have been diagnosed with diabetes, and according to World Health Organization (WHO), diabetes affects 346 million individuals worldwide. Therefore, we propose a noninvasive self-care system based on the IoT and machine learning (ML) that analyses blood sugar and other key indicators to predict diabetes early. The main purpose of this work is to develop enhanced diabetes management applications which help in patient monitoring and technology-assisted decision-making. The proposed hybrid ensemble ML model predicts diabetes mellitus by combining both bagging and boosting methods. An online IoT-based application and offline questionnaire with 15 questions about health, family history, and lifestyle were used to recruit a total of 10221 people for the study. For both datasets, the experimental findings suggest that our proposed model outperforms state-of-the-art techniques.

1. Introduction

Diabetes, often known to be diabetes mellitus (DM), is a group of metabolic illnesses characterized by persistently elevated blood sugar levels. Excessive urination, continuous thirst, and an increase in hunger are all symptoms of high blood sugar [1]. Diabetes, if not treated promptly, can lead to significant health problems in a person, such as hyperglycaemic, hyperosmolar condition, diabetic ketoacidosis, or even one of the results for death. Long-term effects include stroke, cardiovascular disease, foot ulcers, renal failure, and vision problems [2]. When the body's pancreas is unable to produce enough insulin, diabetes develops, or even if the insulin generated is not appropriately used by the body's cells and tissues. The diabetes mellitus can be categorized into the following three types [3].

(i) "Insulin-subordinate diabetes mellitus" (ISDM) is a disorder in which the pancreas produces less insulin than the body demands, resulting in type 1 diabetes. To compensate for the pancreas' lower insulin production, type 1 diabetics require supplementary insulin.

(ii) Type-2 diabetes is defined as an insulin resistive body, which occurs when the body's cells react to the insulin differently than they would ordinarily. "Adult starting diabetes" or "noninsulin subordinate diabetes mellitus" (NISDM) is other term for this condition. This kind of diabetes is more common in those with a high BMI or who have a sedentary lifestyle.

(iii) During the time of pregnancy, the third type of diabetes called gestational diabetes may develop.

A typical human's sugar levels may vary from range 70 to 99 mg/dL. A person is classified as having diabetes when her or his fasting glucose level reached 126 mg/dL. From the healthcare point of view, someone with a higher glucose level between 100 and 125 mg/dL may be considered prediabetic [4]. In such an individual, type 2 diabetes is more prone to develop. GDM (gestational diabetes mellitus) is a kind of diabetes that develops during pregnancy that is no clear evidence of diabetes during the 2nd and 3rd trimesters of pregnancy. Diabetic may be caused by other factors, such as monogenic diabetes syndromes, and exocrine pancreas diseases.

Diabetes disorders have the capacity to harm several sections of the human body. The followings are some of the human body components that are impacted by diabetes: the heart, the eye, the kidney, and the nerves of humans [5, 6]. As the name implies, it is simple to estimate how much chronic and serious illnesses shorten human life. Machine learning algorithms have varying degrees of categorization and prediction capacity [7]. According to [8], no one strategy is superior in terms of performance and accuracy for all diseases; although one classifier performs best in a certain dataset, another method or approach outperforms the others for other diseases. The new or proposed study focuses on a novel combination or hybridization of multiple classifiers for diabetic mellitus (DD) classification and prediction, solving the difficulty of single or individual classifiers. The new study proposes using several machine learning methods (MLTs) to detect diabetic mellitus (DM) at an early stage in order to save human lives. The major goal of this research is to create an information system that can forecast diabetes with greater accuracy.

1.1. Symptoms. The symptoms of diabetes may vary depending on the blood glucose level. Some people, particularly those with type-2 diabetes or prediabetes, may not show any signs at all. Symptoms of type-1 diabetes appear more quickly and are more severe. Some of the signs and symptoms of type 1 and type 2 diabetes are as follows:

(i) Availability of ketones in urine
(ii) Thirst rises
(iii) Frequent urination
(iv) Hunger to the point of death
(v) Frequent weight loss
(vi) Fatigue
(vii) Cloudy vision
(viii) Long-lasting sores
(ix) Infections that recur often, such as gum or skin infections, as well as vaginal infections
(x) Obesity is defined as a BMI greater than 25

Diabetes is a familial disease that affects several members of the family. People have HDL cholesterol levels of less than 40 milligrams per deciliter in their blood. People with polycystic ovary syndrome over 45 years old from ethnic groupings such as African Americans, Native Americans, Latin Americans, and Asian Pacific live a sedentary lifestyle.

The IoT in genetic terms is used for a collection of connected bodily objects that may be accessed over the Internet. The "thing" in the Internet of Things can be an object with sensors that have been assigned an IP address [9]. It can build and share data over a network without requiring any human assistance. Individuals are becoming increasingly conscious of and committed to their own health. A large portion of hospital expenditures is spent on medical examinations. There is an unrivaled opportunity to improve the quality of care and the efficacy of therapies by adopting technology-based healthcare procedures [10–13].

There are a variety of advantages to implementing IoT, including real-time applications and data collection and analysis. Figure 1 depicts how this significant shift in medical practice will be examined in an IoT hospital. An ID card will be issued to a diabetic patient that, once scanned, will help to connect them to a secure cloud where their electronic health-related data and medical records would be stored. On a tablet or computer, doctors and attendants will have no trouble using the record.

The remainder of the paper is laid out as follows: Section 2 focuses on the related work reviewed during the proposed work. Section 3 briefly describes the traditional models which were implemented for prediction and comparison. In section 4, the proposed methodologies along with the implementation are presented, and experimental results along with a discussion are carried out in section 4. Lastly, the conclusion of the proposed work is presented in section 5.

2. Related Work

Diabetes may be a major disease, with an affected adult population of more than 70%. To anticipate diabetes symptoms, several researchers have utilized approaches such as data mining and machine learning [14]. Only a handful has utilized both neural networks and genetic algorithms. Because diabetes prediction is a supervised problem, supervised techniques such as machine learning, data mining, and artificial neural networks have been employed by numerous researchers.

Numerous scientific researchers have utilized the Pima Indians dataset for diabetes (PIDD) to predict diabetes. Weka and machine learning approaches were used in [15–17]. Data mining, machine learning, neural network, and hybrid techniques are among the methodologies used by researchers. In diabetes prediction, artificial neural networks (ANN) are commonly employed. Komi et al. [18] described several data mining approaches that were used for showing information for type 2 diabetes. Swapna et al. [19] used electrocardiogram (ECG) data to detect diabetes using deep learning algorithms. They retrieved features using a convolution neural network (CNN), and then, a support vector machine algorithm is used to extract the features. Finally, they determined that the accuracy rate was 95.7%. To represent knowledge-based systems, fuzzy cognitive maps (FCM) have been used. Tuppad et al. [20] proposed a strategy for predicting gestational diabetes using the case-based fuzzy cognitive maps decision-making system. Saeedi

RETRACTED

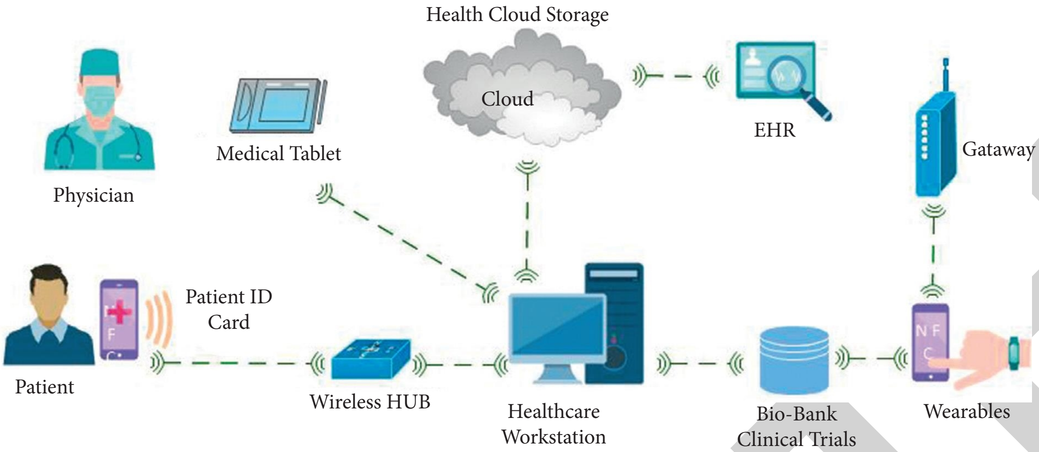


Figure 1: Health monitoring in IoT hospitals.

RETRACTED

et al. [8] proposed a framework to detect the presence or absence of diabetes mellitus. This framework is based on a delicate registering technique, specifically fuzzy cognitive maps (FCM). The product instrument was tested on 50 cases, with 96% accuracy in predicting outcomes.

A significant advancement in medical imaging technology has occurred in the last decade as a result of the application of iris image detection. Furthermore, the machine learning approaches are useful to improve the determining capacity of iridologists. Systemic disease with ocular consequences was linked to the proposed model [21]. The random forest classifier achieved 89.66% accuracy by analyzing 200 subject data from 100 diabetic and nondiabetic people. To predict diabetes using PIDD, Sisodia et al. [22] utilized three machine learning algorithms: decision tree (DT), support vector machine (SVM), and naive Bayes (NB). The accuracy of 76.3 percent was determined for the naive Bayes classifier. Wu et al. [23] employed a data mining technique to determine an individual's development of risk factors for type-2 diabetes with an accuracy of 95.42%. Experimentally, the initial seed point value resulted in the modification. Choubey et al. [24] utilized J48, random forest, and ANN for classification and utilising unsupervised techniques like principal component analysis (PCA) after feature reduction.

Siddiqui et al. wanted to see if there was a link between diabetes and metabolic syndrome [25]. For forecasting, the authors employed the Naive Bayes and J48 decision tree models. The training set was balanced by using k-medoids sampling. In their study, NB surpassed the competition. The effects of different machine learning techniques on the determination of diabetes are summarized by Wittenbecher et al. [26] and Zhou et al. [27]. The proposed work first records the patient information through sensors and then transmitted it to the cloud server. The proposed concept got a 0.045 correlation coefficient value which increases the strength of the algorithm [28].

All age groups finally saw a linear connection between BMI and diabetes. BMI and age were shown to be good predictors of diabetes risk. Zou et al. [29] developed a nomogram based on the seven diabetes risk factors to help people predict their type 2 diabetes risk. A robot is intelligent in the sense that it has built-in watching and detecting capabilities, as well as the ability to gather sensor data from various sources and fuse it for the device's "acting" purpose. Mall et al. [30] introduced the e-health mind stage by employing robots that were connected via IoT to provide personalized varied care methods, particularly to diabetes patients. The robot is equipped with sensors that monitor the diabetic's medical and dietary status, providing them with comprehensive multidimensional care.

According to statistical analysis and the multivariate Cox regression method [31], the TG/HDL-C ratio was positively associated with the prevalence of diabetes in the Chinese population. The author proposed an MSSO-ANFIS model for the diagnosis of heart disease which uses a levy flight algorithm. The proposed model obtains 99.45 accuracies and 96.54 precision [32]. It is concluded that those in their 30s and 40s with elevated ALT (alanine aminotransferase) are at a higher risk than those with low ALT. Choi et al. [33] employed machine learning (ML) algorithms on people with nondiabetics and a high risk of cardiovascular disease. In this paper, the author proposed an MDCNN classifier that collects data from IoT sensors. The proposed model obtains 98.2 accuracies as compared with existing classifiers [34]. Over the last five years, Korea University Guro Hospital has accumulated data in the form of an EMR (electronic medical record) [35]. Various ML methods were then employed with the help of cross-validation. The most accurate model is the logistic regression model [36–38].

3. Different Machine Learning Approaches

Once the data are available, we use machine learning techniques to analyze it. We use a number of classification algorithms to predict diabetes. The strategies were tested using a diabetic dataset from Pima Indians. The major purpose is to assess the results of these methods and determine their validity, as well as who was accountable, using machine learning techniques. This is a crucial characteristic that plays a big part in prediction. The methods are as follows.

RETRACTED

3.1. Logistic Regression (LR). The sigmoid function is used to evaluate probabilities in LR, which is a sort of supervised learning method. The sigmoid function calculates the relationship between at least one independent variable and a binary-dependent variable. The LR model is a form of machine learning classification model that has binary values like 0 or 1, −1 or 1, true or false as the dependent variable and the independent variable such as interval, ordinal, binominal, or ratio level. The logistic/sigmoid equation function is as follows:

$$y = \frac{1}{1 + e^{-x}}, \tag{1}$$

where y is denoted as the outcome of the weighted sum with variables x as input. Here, the output is estimated as 1 if it is more than 0.5; else, it is 0.

3.2. Support Vector Machine. Out of many supervised classification techniques, the SVM is one of them that may be used for regression and classification in machine learning techniques. It is mostly used to solve classification difficulties. The main goal of SVM is to categorize the data point using a suitable hyperplane in a multidimensional space. A hyperplane is considered as a boundary of classification for data values. In this technique, each data item in n-dimensional space is represented as a point, with the value of each feature matching the value of a certain coordinate. We would plot these two components in two-dimensional space, with two layouts for each point if we only knew two qualities about an individual, such as height and hair length (these directions are known as support vectors). Because the two closest focuses are the furthest distance from the line in Figure 1, the dark line divides the data into two different organized groupings. Our classifier is represented by this line. Based on the falling of testing data on both sides of the line, the new data are able to be categorized into one of two categories.

3.3. K-Nearest Neighbor. Both regression and classification issues may be solved using the K-nearest neighbor (KNN) technique [39]. However, in the industry, it is more commonly utilized in classification issues. KNN is a straightforward computation that stores all existing examples and ranks new ones based on the votes of its k neighbors. To place the case in the class with the most people among its K nearest neighbors, distance work is used. The Manhattan, Hamming, Euclidean, and Makowski distances are among the distance capabilities. The first 3 numbers of features are used for indefinite functions, whereas the 4th one is used for absolute variables. If $K = 1$, the case is essentially assigned to the class of the next closest neighbor. Selecting K for KNN modeling might be challenging at times.

3.4. Random Forest. The random forest (RF) classifier technique generates several decision trees from a portion of the randomly chosen dataset used for training purposes. The votes from several decision trees are combined to establish the final class of test items [29]. Each tree offers a classification to a new object based on characteristics, and for that class, we say the tree as "votes." The classification employing the utmost votes is selected by the forest. The random forest has several options that produce accurate predictions for a variety of applications. The following is how each tree is planted and grown:

Table 1: Possible answers for different features.

Sl. No	Features	Possible answers
1	Age	More than 18 years
2	Gender	Male 643, Female 429
3	Family history	Yes/No
4	Physical_Activity	1. More_than_one_hour 2. Less_than_one_hour 3. Never
5	Urination_Frequency	1. Frequently 2. Not much
6	Junc_Food_Consumption	Yes/No
7	Blood_Pressure	1. Normal 2. High 3. Low
8	BMI	Numeric
9	Diabetes	Yes/No
10	Reglar_Intake_Of_Medicne	Yes/No
11	No_Of_Pregnancies	Numeric
12	Smoking	Yes/No
13	Alcohol_Consumption	Yes/No
14	Hour_Of_Sleep	Numeric
15	Stress	Yes/No

(1) If N instances are there in the training set then, an N cases random sample is chosen with replacement and that can be utilized for training the tree.

(2) If there are M inputs and out of which m inputs are randomly chosen at each node out of the M variables, where $m < M$, with the finest split on this input m being utilized to divide the node. Here, m is kept constant throughout the growth of the forest.

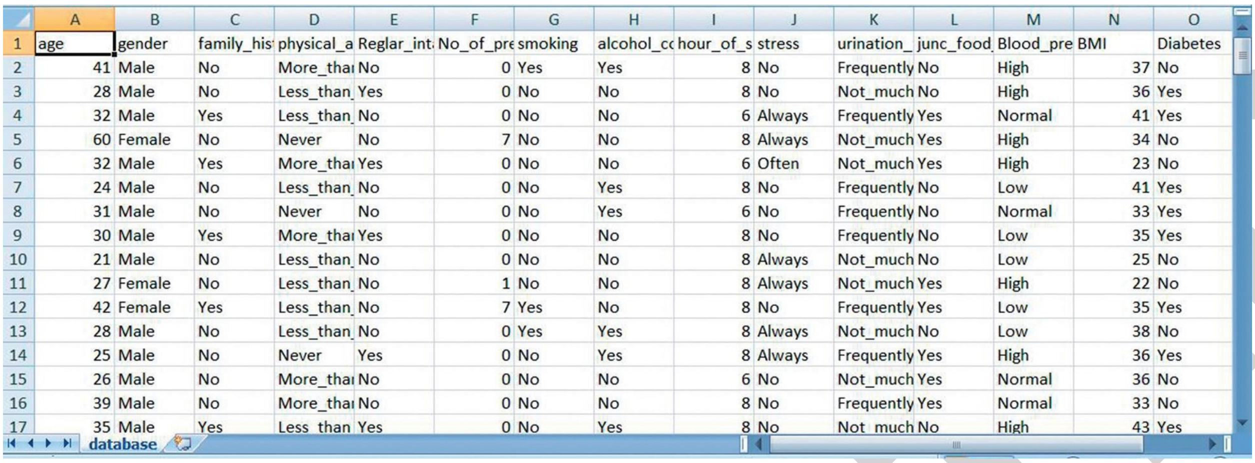
(3) Every tree is brought to its full potential. Pruning is out of the question.

4. Proposed Methodology

A total of 10221 individuals aged 18 and above were chosen for this study, including 6031 men and 4190 females. The participants were invited to complete an online IoT sensing operation and a questionnaire (Table 1) that they had developed themselves based on the factors that might contribute to diabetes. The same tests were carried out on another database, the PIMA Indian Diabetes database [31–33], to validate the model's validity. Figure 2 depicts a sample dataset gathered by a questionnaire.

4.1. M-Health Systems Using Web-Based IoT Service and Sensors for Diabetes Monitoring. When the reading rises, an update automatically is sent to the doctor via voice calls or text messages. This may be accomplished through the use of a web application that establishes worldwide communication between the patient's online portal and the IoT sensor of the patient, which updates the patient's personal information

RETRACTED



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	gender	family_his	physical_a	Reglar_int	No_of_pre	smoking	alcohol_cc	hour_of_s	stress	urination_	junc_food	Blood_pre	BMI	Diabetes
2	41	Male	No	More_thai	No	0	Yes	Yes	8	No	Frequently	No	High	37	No
3	28	Male	No	Less_than_	Yes	0	No	No	8	No	Not_much	No	High	36	Yes
4	32	Male	Yes	Less_than_	No	0	No	No	6	Always	Frequently	Yes	Normal	41	Yes
5	60	Female	No	Never	No	7	No	No	8	Always	Not_much	Yes	High	34	No
6	32	Male	Yes	More_thai	Yes	0	No	No	6	Often	Not_much	Yes	High	23	No
7	24	Male	No	Less_than_	No	0	No	Yes	8	No	Frequently	No	Low	41	Yes
8	31	Male	No	Never	No	0	No	Yes	6	No	Frequently	No	Normal	33	Yes
9	30	Male	Yes	More_thai	Yes	0	No	No	8	No	Frequently	No	Low	35	Yes
10	21	Male	No	Less_than_	No	0	No	No	8	Always	Not_much	No	Low	25	No
11	27	Female	No	Less_than_	No	1	No	No	8	Always	Not_much	Yes	High	22	No
12	42	Female	Yes	Less_than_	No	7	Yes	No	8	No	Frequently	Yes	Low	35	Yes
13	28	Male	No	Less_than_	No	0	Yes	Yes	8	Always	Not_much	No	Low	38	No
14	25	Male	No	Never	Yes	0	No	Yes	8	Always	Frequently	Yes	High	36	Yes
15	26	Male	No	More_thai	No	0	No	No	6	No	Not_much	Yes	Normal	36	No
16	39	Male	No	More_thai	No	0	No	No	8	No	Frequently	Yes	Normal	33	No
17	35	Male	Yes	Less than	Yes	0	No	Yes	8	No	Not much	No	High	43	Yes

Figure 2: Screenshot of the collected dataset.

such as blood sugar level and remaining medicines. This is one method for managing diabetes remotely that has been proposed.

One of the most extensively utilized technologies is using IoT devices to monitor diabetes patients. By just registering in the programme that talks with the IoT sensors, one may keep track of their diabetes state. This application simplifies the monitoring process for new members, diabetes patients, their family members, and anybody else who is interested. The user must have their user name and password. After the member's information has been verified and the registration has been completed, the user may log in and access the extra services that are offered. It is vital to keep track of the user profile that was generated when you signed up. It is vital that their sensor readings be automatically enrolled. Here, the RFID tag must be linked with the sensors that are attached to the patient. The IoT can keep track on the patient remotely irrespective of the availability of the patient either in the home or at the hospital. A number of sensors are used in this technique. Arduino is an open-source microcontroller that makes things more flexible and accessible, allowing you to develop transdisciplinary projects. Body temperature sensors, OPS2 (oxygen and pulse sensor), and blood pressure sensors are all examples of e-health sensors that use Arduino. A glucometer sensor is a medical gadget that measures glucose levels in the blood. By pricking the skin with a lancet, a small drop of blood is sufficient to compute the level of blood sugar in the patient.

All the above-mentioned sensors must be linked to the body of the patient so that the necessary detailed reading of the patient can be monitored by the e-health sensor. The login credentials of the patient are verified whenever the patient logged in using an RFID tag. The patients' detailed data are then immediately updated. Sensors affixed to the body take the readings, which are then connected utilising IoT tools. It will immediately send a message or a phone call to the patient's doctor regarding the details condition of the patient. The data are subsequently entered into a diabetic patient management website. In Figure 3, the different sensors used to monitor the patient and record their information for further prediction are depicted.

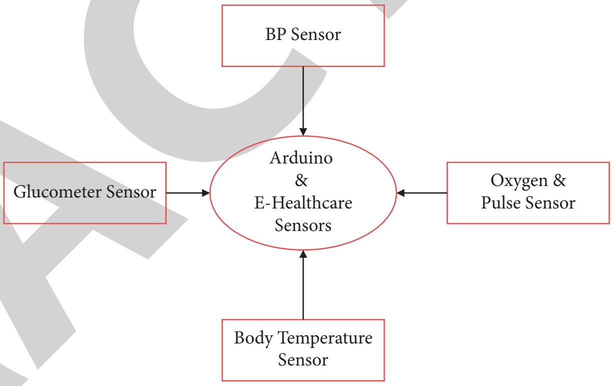


Figure 3: Sensors and services for diabetes monitoring.

Once the data are collected through the IoT sensor and questionnaires, we applied a hybrid bagging and boosting, ensemble methodology to the data. The proposed work is divided into two stages. During the first stage, the training data are fit into three different traditional machine learning models which are logistic regression, K-nearest neighbor, and support vector machine individually. Then, a voting process is applied to the resultant prediction which elects the output among them. This whole process is known as bagging. In the second stage, the identified output is then fit through the random forest model to boost the prediction. This process is known as boosting. The detailed flow of the proposed model is presented in Figure 4.

4.2. Implementation. The study's implementation was done with Google Colab, and the coding was done with the python programming language. Both the Pima dataset and the gathered dataset were used to forecast the availability of diabetes. After then, each classifier's predictions are compared with the proposed model.

4.3. Available Pima Dataset. Parameters used in Pima datasets are as follows:

(1) Age

(2) Glucose

RETRACTED

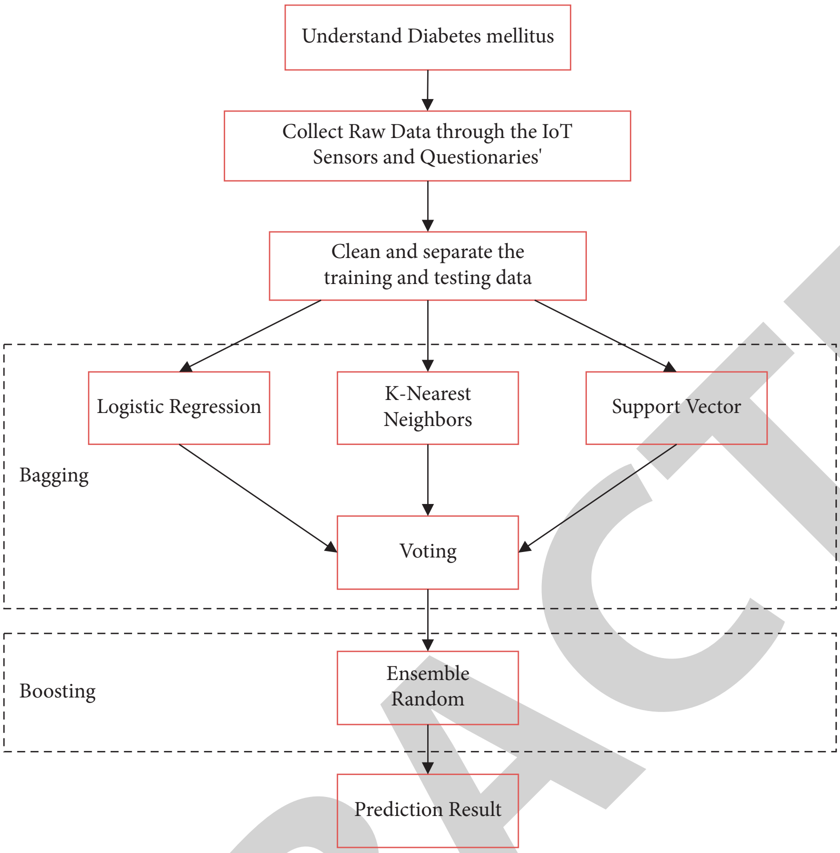

Understand Diabetes mellitus
Collect Raw Data through the IoT Sensors and Questionaries'
Clean and separate the training and testing data
Logistic Regression
K-Nearest Neighbors
Support Vector
Bagging
Voting
Boosting
Ensemble Random
Prediction Result

Figure 4: Flow of the proposed model.

(3) Blood pressure
(4) BMI
(5) Insulin
(6) Skin thickness
(7) Diabetes pedigree function
(8) Pregnancies
(9) Outcome

5. Experimental Results and Discussions

The data set used to predict diabetes is shown in Tables 2 and 3. The diabetes parameters serve as the variable, which is dependent, whereas the other factors served as independent ones. For dependent diabetes features, only two values are accepted, with a “zero” indicating no diabetes and a “one” signifying the availability of diabetes. The whole sample is divided into two groups, with a ratio of 70 : 30 for the training and testing dataset. All four methods of classification were used for prediction. The training data were then used to predict the test set outcomes using SVM, k-nearest neighbor, RF, and LR classifications, resulting in the confusion matrix given in Table 2.

The measure provided in equations (2)–(8) may be computed using the obtained confusion matrices. True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) were the results of these matrices (TP). Because there are more nondiabetic cases than diabetic cases in both datasets, the TN is greater than the TP. As a consequence, all of the techniques provide positive results. The following measurements have been calculated using the following formulae [34] to determine the precise accuracy of each method:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{3}$$

$$\text{Specificity} = \frac{TN}{TP + FP}, \tag{4}$$

$$MCC = \frac{(TP * TN) - (FP + FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{5}$$

$$\text{Error Rate} = \frac{FN + FP}{TP + TN + FN + FP}, \tag{6}$$

$$F - \text{Measure} = \frac{2 * (\text{Precision} * \text{Sensitivity})}{\text{Precision} + \text{Sensitivity}}, \tag{7}$$

$$\text{Accuracy} = \frac{TN + TP}{TP + TN + FN + FP}. \tag{8}$$

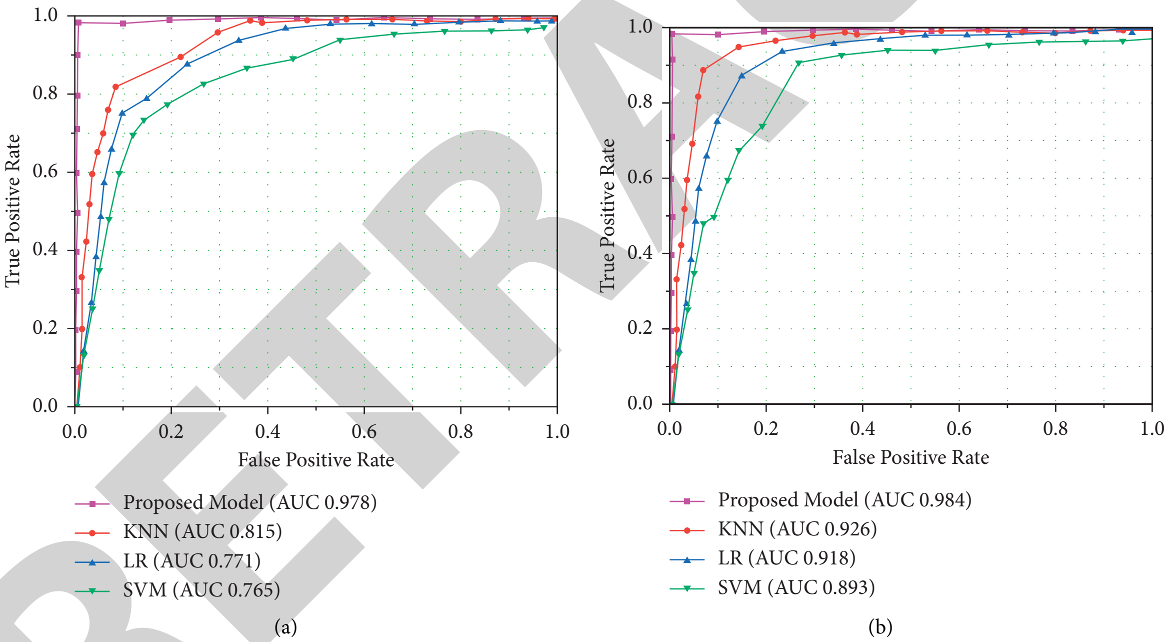
RETRACTED

Table 2: Matrix of confusion for different classification methods.

Dataset/Models	Logistic regression	K-nearest neighbor	Support vector machine	Proposed model
PIMA dataset	[[138 12] [42 41]]	[[132 20] [39 44]]	[[143 9] [47 36]]	[[128 20] [36 45]]
Collected dataset	[[192 24] [15 87]]	[[164 38] [46 74]]	[[195 15] [24 93]]	[[224 3] [4 95]]

Table 3: Comparison of statistical measurement for various classification techniques.

	Logistic regression		K-nearest neighbor		Support vector machine		Proposed model	
	Collected dataset	Pima dataset	Collected dataset	Pima dataset	Collected dataset	Pima dataset	Collected dataset	Pima dataset
Accuracy	0.872	0.744	0.739	0.708	0.888	0.744	0.984	0.750
Error	0.127	0.255	0.261	0.291	0.112	0.255	0.016	0.250
Sensitivity	0.923	0.775	0.778	0.748	0.898	0.775	0.987	0.789
Specificity	0.764	0.666	0.702	0.603	0.816	0.666	0.916	0.661
Precision	0.885	0.856	0.816	0.832	0.933	0.856	0.991	0.840
F-measure	0.903	0.813	0.797	0.787	0.915	0.813	0.989	0.813
MCC	0.732	0.416	0.503	0.331	0.764	0.416	0.963	0.436
Kappa	0.727	0.470	0.516	0.419	0.713	0.466	0.922	0.488
AUC	0.908	0.765	0.916	0.815	0.893	0.771	0.984	0.978



(a)

(b)

Figure 5: (a) ROC curve with AUC for PIMA dataset. (b) ROC curve with AUC for the collected dataset.

Another finding is that the accuracy level as per Table 3 among all individual techniques is higher on our collected dataset than on the used PIMA dataset, owing to the former's greater number of variables relevant to assessing diabetes risk. The random forest classifier outperforms all others in terms of accuracy (98.4%), sensitivity, specificity, precision, and F-measure, proving that it is the best technique for our dataset. Furthermore, in the case of random forest, the AUC value is 1, indicating that this model performs exceptionally well in classification. Figure 5 depicts the clear graph for the ROC curve and AUC for both the collected dataset and PIMA datasets. Here, it indicates that in both cases, the ensemble RF boosting classifier gives the highest result with a value of 1.

The significance of each parameter in the dataset is depicted in Table 4. On the classifier model construction, the python function "summary" is used to perform this analysis. The star beside each parameter indicates the significance of that variable. The ratings are in the following order: where "***" denotes the highest priority, "*" denotes the least important, and a feature without any symbol denotes the least concerned with diabetes. Figure 6 depicts the correlation matrices of the different parameters, and Figure 7 depicts the comparison of different classification algorithms. There is no statistical significance for the variable with no rating. Variable importance is studied to find which parameter has the greatest impact on the forecast.

Table 4: Importance of parameters.

	Count	Mean	Std	min	0.25	0.5	0.75	Max
Age***	10221	32.9026	11.09011	21.00000	24.00000	29.50000	40.00000	66.00000
Gender**	10221	0.50649	0.50158	0.00000	0.00000	1.00000	1.00000	1.00000
family_history***	10221	0.29220	0.45626	0.00000	0.00000	0.00000	1.00000	1.00000
physical_activity***	10221	1.38311	0.78547	0.00000	1.00000	2.00000	2.00000	2.00000
Regular_intake_of_medicine***	10221	0.26623	0.44343	0.00000	0.00000	0.00000	1.00000	1.00000
No_of_pregnancies**	10221	1.88311	3.08686	0.00000	0.00000	0.00000	3.00000	14.00000
Smoking	10221	0.07792	0.26892	0.00000	0.00000	0.00000	0.00000	1.00000
alcohol_consumption	10221	0.33766	0.47445	0.00000	0.00000	0.00000	1.00000	1.00000
hour_of_sleep*	10221	6.59740	0.91836	6.00000	6.00000	6.00000	8.00000	8.00000
Stress	10221	0.44805	0.57214	0.00000	0.00000	0.00000	1.00000	2.00000
Urination_frequency	10221	0.40909	0.49327	0.00000	0.00000	0.00000	1.00000	1.00000
Junc_food_consumption	10221	0.52597	0.50095	0.00000	0.00000	1.00000	1.00000	1.00000
Blood_pressure**	10221	0.86363	0.80900	0.00000	0.00000	1.00000	2.00000	2.00000
BMI	10221	32.3701	7.40424	0.00000	28.00000	33.00000	36.00000	67.00000
Diabetes	10221	0.44155	0.49819	0.00000	0.00000	0.00000	1.00000	1.00000

```
<matplotlib . axes . _subplots . AxesSubplot at 0×7f76f62a6790>
```

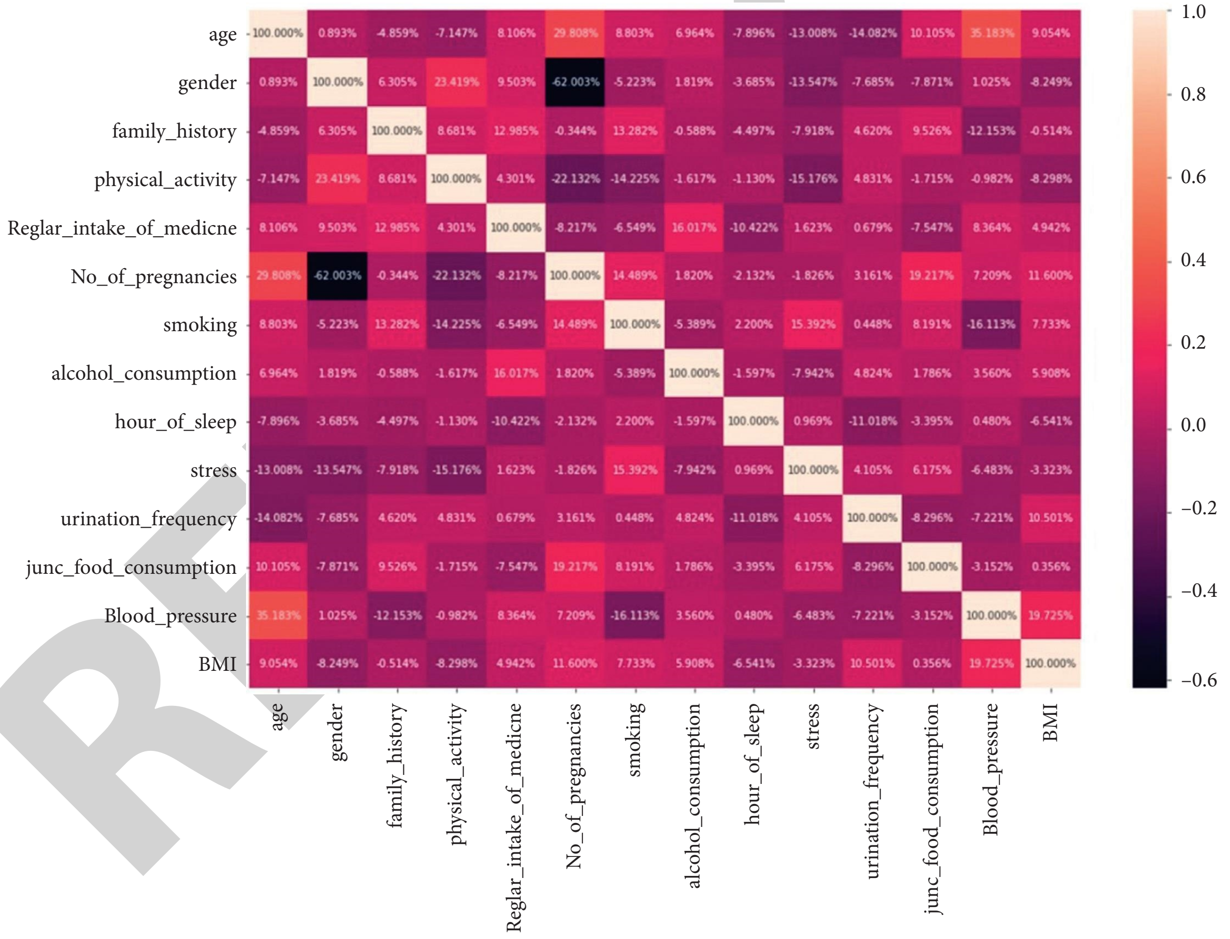


Figure 6: Correlation matrix.

5.1. Comparative Analysis. Table 5 shows a comparison between the current state of the art and our suggested technique. The author of [19] employed deep learning algorithms to predict diabetes, providing a maximum accuracy of 95.7 percent. Bhatia et al. [6] employed a more accurate genetic algorithm fuzzy cognitive maps and achieved an accuracy of 96 percent. Samant et al. [21] used an improvised random forest technique to achieve 89.66 percent accuracy, whereas Sisodia et al. [22] used modified machine learning algorithms with efficient coding to get 76.3 percent accuracy. Wu et al. [23] have employed improved data mining techniques to get an accuracy of

RETRACTED

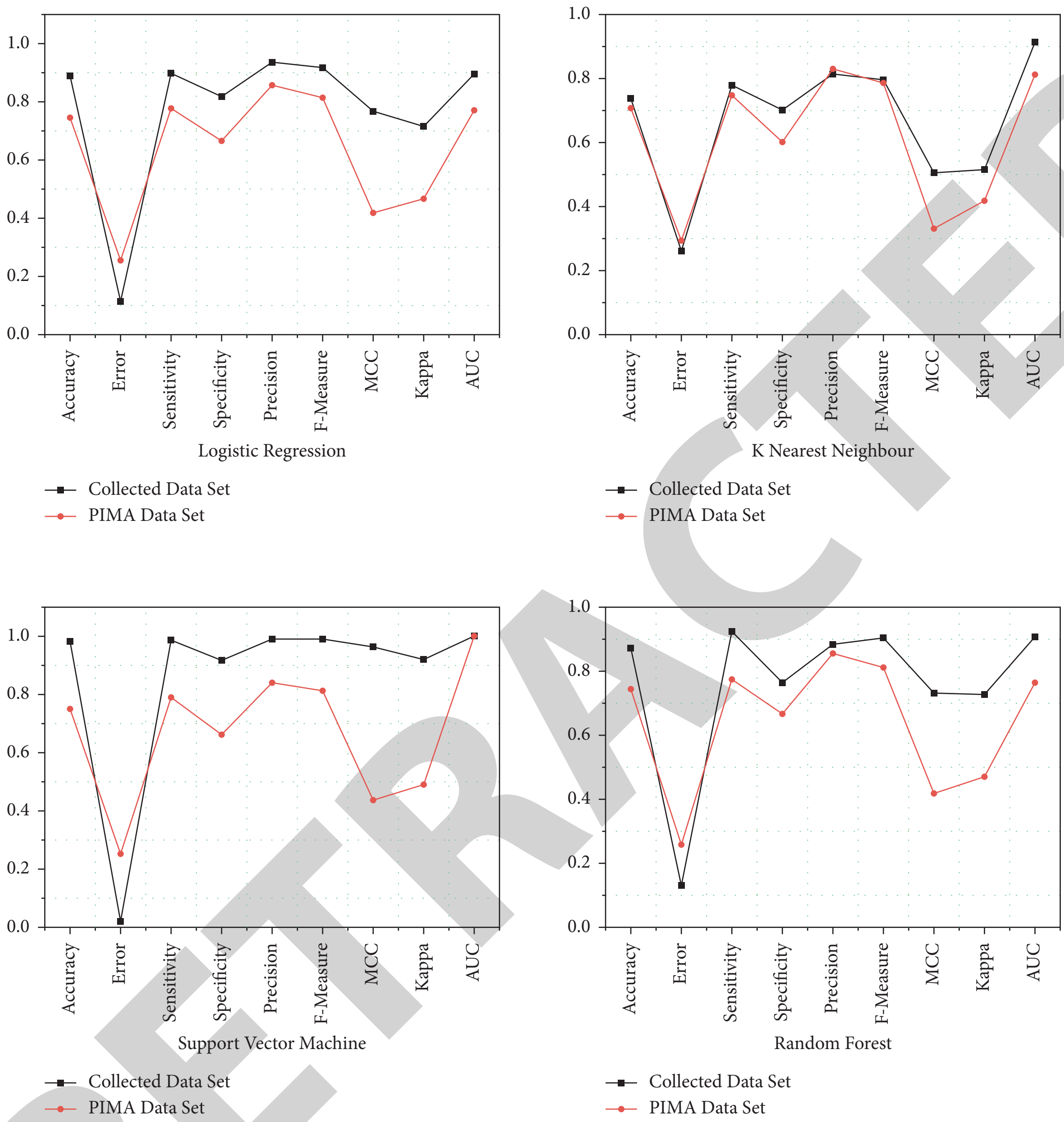


Figure 7: Comparison of traditional classification models and proposed model when implemented on the collected dataset and PIMA dataset.

Table 5: Comparison study of our suggested work with the current state of the art in terms of accuracy.

Author(Year)	Method	Accuracy (%)
Swapna et al. [19] (2018)	Deep learning algorithms	95.7
Bhatia et al. [6] (2019)	Fuzzy cognitive maps	96
Samant et al. [21] (2017)	Improvised random forest technique	89.66
Sisodia et al. [22] (2018)	Modified machine learning algorithms	76.3
Wu et al. [23] (2018)	Improved data mining techniques	95.42
Our approach	IoT-based hybrid ensemble machine learning model	**98.4**

RETRACTED

95.42 percent. Our method achieved a 98.4 percent accuracy by utilising an IoT-based hybrid ensemble machine learning model that is superior to the current state of the art.

6. Conclusion

One of the most pressing worldwide health concerns is detecting diabetes risk at an early stage. Our research aims to build up a system for predicting the risk of diabetes mellitus. Three traditional machine learning techniques and the proposed hybrid ensemble model for classification were used in this work, and the results were compared to several statistical metrics. The prediction has been done using ML algorithms on collected 15 diabetes-related data from IoT sensors as well as questionnaires. Also, the four algorithms were used on the PIMA database for prediction. The accuracy level of the proposed classification in our dataset is 98.4 percent, which is the greatest among the others, according to the testing results. For the PIMA dataset, the proposed model also provides the greatest accuracy. All described models generated appreciable results for different parameters such as accuracy and recall sensitivity using four different machine learning methods. It is observed from the results that among all factors, "age," "family_history," "physical_activity," and "regular_intake_of_medicine" have the highest significance. These variables have a larger influence on diabetes prediction than the others. This result can be used to forecast any other illness in the future. This study is currently researching and improving various ML approaches for forecasting diabetes along with other health conditions.

Data Availability

The data used to support the findings of this study are available from the author upon request (pinky.sasmita@gmail.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] B. Farajollahi, M. Mehmannavaz, H. Mehrjoo, F. Moghbeli, and M. J. Sayadi, "Diabetes diagnosis using machine learning," *Frontiers in Health Informatics*, vol. 10, no. 1, p. 65, 2021.

[2] K. Ogurtsova, J. D. da Rocha Fernandes, Y. Huang et al., "IDF Diabetes Atlas: global estimates for the prevalence of diabetes for 2015 and 2040," *Diabetes Research and Clinical Practice*, vol. 128, pp. 40–50, 2017.

[3] H. Qin, Z. Chen, Y. Zhang et al., "Triglyceride to high-density lipoprotein cholesterol ratio is associated with incident diabetes in men: a retrospective study of Chinese individuals," *Journal of Diabetes Investigation*, vol. 11, no. 1, pp. 192–198, 2020.

[4] J. Xie and Q. Wang, "Benchmarking machine learning algorithms on blood glucose prediction for type I diabetes in comparison with classical time-series models," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 11, pp. 3101–3124, 2020.

[5] J. Chaki, S. T. Ganesh, S. K Cidham, and S. Ananda Theertan, "Machine learning, and artificial intelligence-based Diabetes Mellitus detection and self-management: a systematic review," *Journal of King Saud University - Computer and Information Sciences*, 2020, ISSN 1319-1578.

[6] N. Bhatia and S. Kumar, "Prediction of severity of diabetes mellitus using fuzzy cognitive maps," *Advances in Life Science and Technology*, vol. 29, pp. 71–78, 2015.

[7] A. U. Haq, J. P. Li, A. Saboor et al., "Detection of breast cancer through clinical data using supervised and unsupervised feature selection techniques," *IEEE Access*, vol. 9, pp. 22090–22105, 2021.

[8] P. Saeedi, I. Petersohn, P. Salpea et al., "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: results from the international diabetes federation diabetes atlas, 9th edition," *Diabetes Research and Clinical Practice*, vol. 157, Article ID 107843, 2019.

[9] S. K. K. Sahtyagi, P. Goswami, S. R. Pokhrel, and A. Mukherjee, "Internet of things for healthcare: an intelligent and energy efficient position detection algorithm," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[10] P. P. Malla, S. Sahu, and S. Routray, "Investigation of breast tumor detection using microwave imaging technique," *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–4, 2020.

[11] S. Rajasoundaran, A. V. Prabu, G. S. Kumar, P. P. Malla, and S. Routray, "Secure opportunistic watchdog production in wireless sensor networks: a review," *Wireless Personal Communications*, vol. 120, no. 2, pp. 1895–1919, 2021.

[12] A. U. Haq, J. P. Li, S. Ahmad, S. Khan, M. A. Alshara, and R. M. Alotaibi, "Diagnostic approach for accurate diagnosis of COVID-19 employing deep learning and transfer learning techniques through chest X-ray images clinical data in E-healthcare," *Sensors*, vol. 21, no. 24, p. 8219, 2021 Jan.

[13] S. Routray, A. K. Ray, and C. Mishra, "An efficient image denoising method based on principal component analysis with learned patch groups," *Signal, Image and Video Processing*, vol. 13, no. 7, pp. 1405–1412, 2019.

[14] N. Fazakis, O. Kocsis, E. Dritsas, S. Alexiou, N. Fakotakis, and K. Moustakas, "Machine learning tools for long-term type 2 diabetes risk prediction," *IEEE Access*, vol. 9, pp. 103737–103757, 2021.

[15] A. H. Syed and T. Khan, "Machine learning-based application for predicting risk of type 2 diabetes mellitus (T2DM) in Saudi arabia: a retrospective cross-sectional study," *IEEE Access*, vol. 8, pp. 199539–199561, 2020.

[16] R. D. Howsalya Devi, A. Bai, and N. Nagarajan, "A novel hybrid approach for diagnosing diabetes mellitus using farthest first and support vector machine algorithms," *Obesity Medicine*, vol. 17, p. 100152, 2020.

[17] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.

[18] M. Komi, J. Li, Y. Zhai, and X. Zhang, "June). Application of data mining methods in diabetes prediction," *2017 2nd international conference on image, vision and computing (ICIVC)*, pp. 1006–1010, 2017.

[19] G. Swapna, R. Vinayakumar, and K. P. Soman, "Diabetes detection using deep learning algorithms," *ICT Express*, vol. 4, no. 4, pp. 243–246, 2018.

RETRACTED

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 5475313, 11 pages
https://doi.org/10.1155/2022/5475313

Research Article

Exploration of Black Boxes of Supervised Machine Learning Models: A Demonstration on Development of Predictive Heart Risk Score

Mirza Rizwan Sajid,[1] **Arshad Ali Khan**,[2] **Haitham M. Albar**,[3]
Noryanti Muhammad,[4] **Waqas Sami**,[5,6] **Syed Ahmad Chan Bukhari**,[7]
and Iram Wajahat[8]

[1]*Department of Statistics, University of Gujrat, Hafiz Hayat Campus Jalalpur Road, Gujrat 50700, Pakistan*
[2]*Faculty of Computing, Universiti Malaysia Pahang, Pekan 26600, Malaysia*
[3]*Department of Surgery, College of Medicine, Majmaah University, Almajmaah 11952, Saudi Arabia*
[4]*Centre of Excellence for Artificial Intelligence & Data Science,*
Department of Knowledge Management & Information Technology, Universiti Malaysia Pahang, Gambang 26300, Malaysia
[5]*Department of Community Medicine and Public Health, College of Medicine, Majmaah University, Almajmaah 11952, Saudi Arabia*
[6]*Azra Naheed Medical College, Superior University, Lahore 54000, Pakistan*
[7]*Division of Computer Science, Mathematics and Science, Collins College of Professional Studies, St. Johns University, New York, NY 11439, USA*
[8]*Allied Institute of Medical Sciences, Gujranwala 52250, Pakistan*

Correspondence should be addressed to Mirza Rizwan Sajid; mirza.rizwan@uog.edu.pk

Received 4 March 2022; Accepted 26 April 2022; Published 12 May 2022

Academic Editor: Muhammad Ahmad

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

Machine learning (ML) often provides applicable high-performance models to facilitate decision-makers in various fields. However, this high performance is achieved at the expense of the interpretability of these models, which has been criticized by practitioners and has become a significant hindrance in their application. Therefore, in highly sensitive decisions, black boxes of ML models are not recommended. We proposed a novel methodology that uses complex supervised ML models and transforms them into simple, interpretable, transparent statistical models. This methodology is like stacking ensemble ML in which the best ML models are used as a base learner to compute relative feature weights. The index of these weights is further used as a single covariate in the simple logistic regression model to estimate the likelihood of an event. We tested this methodology on the primary dataset related to cardiovascular diseases (CVDs), the leading cause of mortalities in recent times. Therefore, early risk assessment is an important dimension that can potentially reduce the burden of CVDs and their related mortality through accurate but interpretable risk prediction models. We developed an artificial neural network and support vector machines based on ML models and transformed them into a simple statistical model and heart risk scores. These simplified models were found transparent, reliable, valid, interpretable, and approximate in predictions. The findings of this study suggest that complex supervised ML models can be efficiently transformed into simple statistical models that can also be validated.

1. Introduction

Machine learning (ML) models have gained significant importance in recent times due to their flexible nature to gauge complex phenomena accurately. However, this high performance often leads to the complexity of ML methods. Interestingly, the higher complexity of models brings higher accuracy, but it comes at the cost of the interpretability of the models [1, 2]. The trade-off between accuracy and interpretability of models has directed that generally, artificial neural network (ANN) models yield more accuracy but with the least interpretability [3]. On contrary, linear models are easy to interpret but with less accuracy. This issue of interpretability associated with ML models is termed a "black box." It simply means that in ML processes, data go in, decisions come out, but the processes between input and output are opaque [4, 5]. However, the interpretability of models is as important as accuracy [6], especially in clinical decision-making [7].

In the domain of medical science, the understanding of how the models make predictions is the key to practitioners' trust in the model and its output [8]. Generally, the interpretable models are trustworthy because they are consistent with the prior knowledge and experience of practitioners. Additionally, the decision-makers can identify the unusual patterns within the data sets and explain them in a particular scenario. The black box nature of ML models has attained significant attention from experts in various fields. And technology giants such as Google, IBM, and Microsoft have been investigating the techniques and methods that can enhance the interpretability of these models [9]. The ML models focus on predictions and usually provide high accuracy, while the statistical models focus on drawing statistical inferences from these models [10]. Theoretically, this huge difference needs proper justifications when implementing ML models. Therefore, understanding the transparent process of developing ML models is always a key challenge for practitioners, especially when we want to prioritize them over traditional statistical models. Moreover, in the case of fatal diseases such as cardiovascular diseases (CVDs), it is and should be the fundamental requirement of clinical and public health experts while developing risk assessment models. Moreover, the need to have an accurate and validated heart risk score will have an impact on the estimation and prediction of the need for cardiac services and cardiac surgery in different settings with a low financial burden [11, 12].

A risk assessment or prediction model acts as a preventive strategy [13] and helps establish decision support systems [14]. Because of its simplicity and interpretability, logistic regression (LR) is a traditional statistical technique that is commonly used in developing such cardiovascular disease (CVD) related models [13]. However, the existing well-known risk prediction models fail, like the Framingham heart risk scores, which could not detect a large number of individuals accurately [15]. This failure increased the cost of treatment by identifying many individuals as members of the risky group who did not have the predicted level of risk [16, 17]. Therefore, researchers suggest that ML models can solve these accuracy-related issues [13, 18, 19]. In recent years, an extensive literature has been published regarding the utilization of ML models for CVD-related problems [15, 20–22]. These studies found that the performance of the complex ML models improved the prediction accuracy of the process compared to the existing statistical models. However, their "black box" nature is a potential hindrance in their utilization [1, 4, 7], and practitioners could not get an explanation for their prediction results and the role of features [23, 24]. The complexity and behavior of features within these opaque models pose a question about the interpretive abilities of the models. It motivates designing such methodologies that can help improve the interpretability of complex ML models with the minimum loss in performance. This study aims to propose and implement a methodology for transforming complex supervised ML-based risk prediction models into simple interpretable yet accurate risk prediction models. This proposed methodology was tested on the primary dataset, especially collected for the development of a predictive heart risk score using nonlaboratory features. The details of this data set have been provided in the subsequent Section.

2. Materials and Methods

The proposed methodology is like a stacking ensemble that combines different simple and weak ML models for better performance [25]. However, in this study, we adopted it to improve ML models' interpretation and transparency and concluded with a simple LR model (see Figure 1). Mainly, it trained the ML models and derived the importance of the features through best-performed models. Furthermore, these extracted feature weights are used in the simple statistical model. The used ML models were from the class of artificial neural networks (ANN), support vector machines (SVM), and decision trees (DT). This methodology consists of a long process, and the same process has been followed for the derivation of novel nonlaboratory-based heart risk score (NLHRS) and their related risk prediction models. This NLHRS or heart risk score is specifically developed in this study. For easy implementation, in the development of this heart risk score, we only focused on such factors whose data are observational and can be collected without laboratory settings.

(1) In the first step, supervised ML models that are DT, SVM with different kernels, and ANN were trained and implemented on the data set.

(2) In the second step, the performance of these ML models was compared with that of an LR-based model. It is because LR is a commonly used statistical technique in the literature for prediction models. The ML models, which outperformed the LR model, will be continued in the next steps. This comparison is needed because weak ML models should be scrutinized to make this approach more efficient. Therefore, weak ML models were dropped for further processing. In this study, accuracy, sensitivity, specificity, area under the curve (AUC), root mean square error (RMSE), and kappa statistic were used to assess the performance of risk prediction models using tenfold cross-validation. We need to extract relative feature weights from the best-performed models in the next steps; therefore, this cross-validation was adopted because, in the training and testing approach, we can have different relative feature weights at various splits, that is, 80:20, 70:30, and 60:40, which can be problematic in further usage [26].

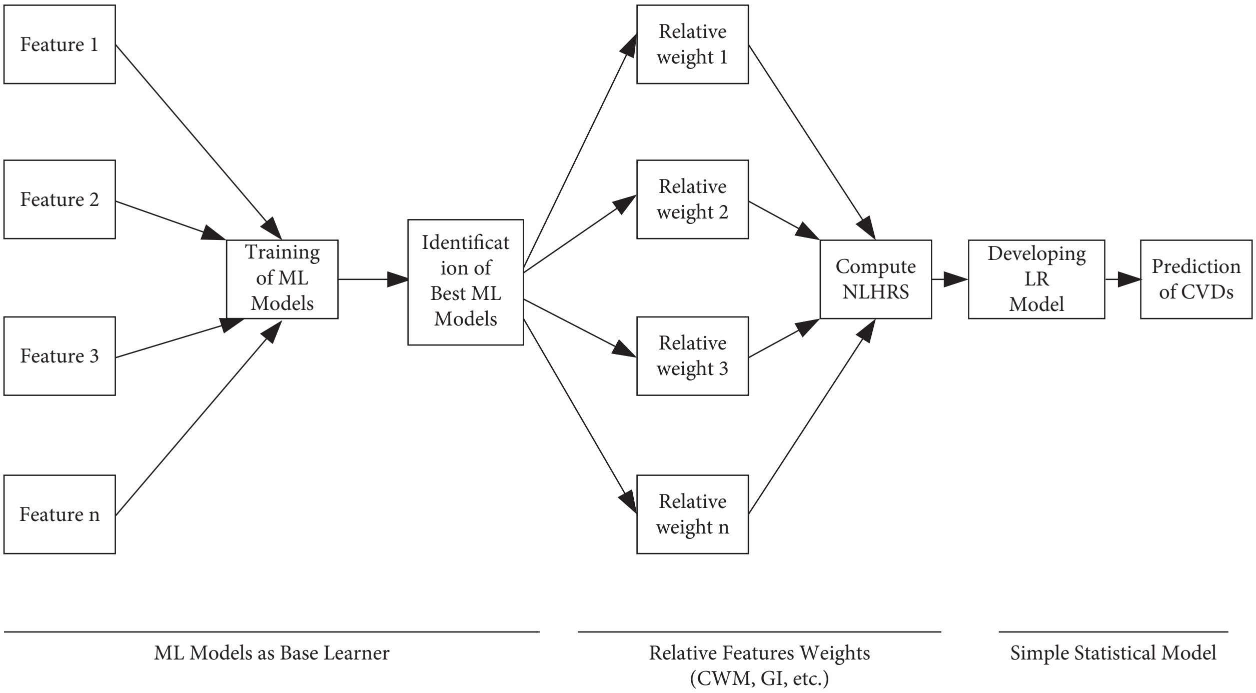


Figure 1: Proposed methodology for exploration of supervised ML models.

(3) In the next step, relative feature weights were computed using a model-driven approach. It is also called a learner-based approach to extracting feature weights (see Figure 1). The best-performed ML models were employed to compute relative feature weights or importance. Input-hidden-output connection weight methodology (CWM) was used to compute relative feature weights in the finalized ANN model as suggested in the literature [27, 28]. DT uses information gain (IG) and the Gini index (GI) for assigning relative weights to features as discussed in the recent literature [26]. In addition, orthogonal vector coordinates orthogonal to the hyperplane are used by SVM to figure out the relative weights of features in the trained model. However, if the SVM model utilizes the radial basis function (RBF) kernel, then relative weights of features can be extracted through the recursive feature elimination (RFE) method as discussed in the literature [29].

(4) Next, we transformed the original binary data set (explained in the next section) with newly computed relative feature weights. Initially, feature values in the data set were in the form of “0” and “1” and were labelled with “no” and “yes,” respectively. Yes = 1 indicated the exposure or presence of a particular feature. In the data transformation process, “1” was replaced with newly computed relative weights of features derived through the best ML models. Additionally, an additive approach was used to compute an index or score. However, the new score ranges from 0 to 100 while developing heart risk scores. In this study, these scores were denoted as ML-based NLHRS and were further used as input in the simple LR model for the development of NLHRS-based simple risk prediction models.

(5) Finally, complete processes of performance assessment and validation were followed for these newly developed simple risk prediction models. The process of validation mainly evaluates the discrimination and calibration strengths of these simple risk prediction models are explained in the subsequent subsection. The valid NLHRS and associated risk prediction models can be used for screening of CVDs in the asymptomatic population.

2.1. Validation of NLHRS-Based Risk Prediction Models. In contrast to ML models, the proposed methodology will provide statistical models that can be validated through existing well-known methods. Therefore, in addition to the common performance matrices mentioned above, newly developed NLHRS-based risk prediction models were also validated. The basic purpose of this validation is to assess the discrimination and calibration capabilities of the models. Various statistical tests were used to evaluate the validity of these models. AUC, Brier score (BS), Spiegelhalter’s Z-statistic, and Hosmer and Lemeshow (H) statistic were used for this purpose. AUC is used to assess the discriminative capability of the risk prediction model [30]. The relative BS provides an overview of the discriminative strength and calibration of the models [31]. In multiple models, Spiegelhalter’s Z-statistic is used to make a comparison and help identify a better model [32]. Calibration is another important measure that assesses the agreement between observed and predicted outcomes [30]. H-statistic was used to measure the calibration of the model as suggested in the guidelines [33]. H-statistic is a goodness-of-fit (GoF) criteria in risk prediction models; therefore, its large value indicates the large difference between observed and predicted outcomes, leading to less calibrated models. For a good risk prediction model, the value of the H-statistic should be less than 20 [34]. In addition to H-statistic, calibration charts are also recommended to visualize the model's performance in various groups of probability [30].

2.2. Collection and Description of Data Set. In the development of NLHRS and its related risk prediction models, the same fifteen features were used as discussed in the literature [35, 36]. The features were gender, age in years, parental history of CVDs, dietary habits, physical activity, smoking, passive smoking, self-reported general stress, obesity, diabetes mellitus, and hypertension. Initially, except for age, all features were binary. However, we categorized age into two highly discriminative data-driven categories for scoring purposes as discussed in the literature [35, 36]. The QUEST algorithm, a type of DT, was used to derive the highly discriminative age categories, and the new binary feature is called "age groups." This binary feature is utilized in the formulation of NLHRS and their corresponding risk prediction models. The data on these selected features were collected from the Punjab Institute of Cardiology (PIC), Pakistan, through a gender-matched case-control study. Male and female cases were matched with their corresponding male and female controls. Approval from the ethical review committee was also sought before the execution of the study. A sample of 460 subjects (with 1:1 matching) having ages >30 years were selected for this study in the duration of September 2018 to February 2019. Only those cases were selected who had their first CVD event (except congenital and rheumatic heart diseases) in their lifetime. The details of the data collection process and designs can be viewed in previous works [26, 37]. The empirical study was performed on Weka version 3.8 and Python 3.9.0.

3. Results

This study included 460 subjects with an equal distribution of cases and controls (230 cases and 230 controls). Among these subjects, 32.2% were females in both groups: cases and controls, because, it is already mentioned in the previous section that it is a gender-matched case-control study. The average age of subjects was 48 years with 11.31 years of standard deviation.

3.1. Implementation of Proposed Methodology for Computation of NLHRS. Selected supervised ML models were implemented to train the models on the CVDs data set. These models were trained without gender as it was found insignificant due to the matching strategy and its insignificance can affect the performance of models. However, its confounding effect would be adjusted at a later stage. Passive smoking was an insignificant feature in the development of the LR model. It was found that the two ML models that are linear SVM and ANN outperformed LR and other ML models in the majority of the performance matrices (see Table 1). Therefore, it was excluded from further processing. We tried more than 50 configurations that include various numbers of hidden layers, number of nodes, and so on. However, the ANN model of a single hidden layer and 7 hidden nodes performed better than other ANN- and LR-based models (baseline model). Figure 2 is the illustration of the ANN model and showed the complexity of the association between features and outcome variables. Various cost function values (c_t) were implemented to optimize the linear SVM model, and $c_t = 0.5$ provided a good risk prediction model. It was found that ANN with a single hidden layer yielded the highest amount of accuracy (82.61%), specificity (0.848), AUC (0.881), and kappa statistic (0.653). Similarly, linear SVM also provided better performance matrices than the LR model. Therefore, linear SVM and ANN models were selected to evaluate model-driven relative feature weights. These two ML models fulfilled most performance criteria with good discrimination and calibration. These extracted feature weights were subsequently used to form the NLHRS and its corresponding risk prediction models. The NLHRS produced through the ANN model was named the "artificial neural network-based risk score" and is denoted by ANN-RS. Similarly, NLHRS produced through linear SVM is named support-vector-machine-based risk score (SVM-RS).

3.2. Extraction of Relative Feature Weights Using Best-Performed ML Models. The finalized relative feature weights were computed through CWM methodology in the ANN model and given in Table 2. The relative weights of features range from 5.138% to 11.944% in the overall prediction process (see Table 2). It is found that the feature "age groups" is crucial for the ANN-based risk prediction model since it receives the highest weight (11.944%). Hypertension is the second most important feature, with 10.527% relative weight in the ANN network. The least important feature identified by the ANN-based model with a relative weight of 5.138% is self-reported general stress.

Linear SVM is another ML best-performed model that was used for the computation of relative feature weights. Table 2 shows the relative feature weights computed through this model, ranging from 2.683% to 11.283%. Linear SVM-based feature weights identified that hypertension was the most discriminating feature for CVD status. Physical inactivity was the second most important feature in the linear SVM-based model. Contrary to the ANN-based model, the linear SVM ranked parental history of CVDs was the least discriminating feature for the model. Overall, it was found that both ML models have different patterns of features and different ranks. This is possibly due to the different nature of these two ML models.

3.3. Development of NLHRS-Based Risk Prediction Models and Their Performance Assessment. Now, the two types of NLHRS were represented as ANN-RS and SVM-RS and were used in the simple LR equation as proposed in methodology to formulate NLHRS-based risk prediction models (see Figure 1). ANN-RS and SVM-RS acted as a

Table 1: Performance assessment of ML models for the development of NLHRS.

Models	ANN		Linear SVM		RBF SVM		Random forest		LR	
Confusion matrix	Case	Control	Case	Control	Case	Control	Case	Control	Case	Control
Case	185	45	187	43	187	43	191	39	188	42
Control	35	195	46	184	50	180	56	174	50	180
Accuracy	82.61		80.65		79.80		79.30		80.00	
Sensitivity	0.791		0.813		0.813		0.830		0.817	
Specificity	0.848		0.800		0.783		0.757		0.783	
Kappa statistic	0.653		0.613		0.595		0.587		0.600	
AUC	0.883		0.881		0.870		0.857		0.873	
RMSE	0.365		0.372		0.379		0.411		0.380	
Number of criteria fulfilled	5/6		5/6		1/6		1/6		Baseline risk prediction model	

single covariate or prognostic index (PI) in two different LR equations. These PIs were denoted as $\mathrm{PI_A}$ and $\mathrm{PI_S}$ for ANN-RS and SVM-RS, respectively. $\mathrm{PI_A}$ and $\mathrm{PI_S}$ are the indexes that will be computed for an individual using the ANN- and SVM-based relative feature weights. For elaboration, an example is discussed here.

Example 1. For example, if a person is physically inactive, has a smoking history, consumes high salty foods, and has a history of hypertension and diabetes mellitus, then PI_A and PI_S will be computed as follows:

$\mathrm{PI}_A = 8.024$ (physical inactivity = 1) + 9.067 (smoking history = 1) + 6.128 (consumption of high salty foods = 1) + 10.527 (hypertension = 1) + 6.371 (diabetes mellitus = 1) = 40.12

$\mathrm{PI}_S = 9.499$ (physical inactivity = 1) + 7.887 (smoking history = 1) + 8.338 (consumption of high salty foods = 1) + 11.283 (hypertension = 1) + 7.961 (diabetes mellitus = 1) = 44.91

It was found that the regression coefficient of each type of PI was significant in their respective model. Furthermore, these equations were also adjusted for gender effects. These equations were referred to as NLHRS-based risk prediction model equations (see equations (1)–(4)). These simple LR-based equations with the PIs need to be validated before being used as a risk prediction model. Therefore, the predictive capability and validity of each NLHRS-based model were computed in the subsequent section. Equations (1) and (3) are the z_i (logits) of the ANN-RS and SVM-RS, respectively. However, equations (2) and (4) were used to estimate the probability of having CVDs based on ANN-RS and SVM-RS, respectively.

$$z_i = -5.659 + 0.107(\text{Gender}) + 0.160(\mathrm{PI}_A), \tag{1}$$

$$P(\mathrm{CVD}s = 1|\mathrm{PI}_A) = \frac{1}{1 + e^{-z_i}} = \frac{1}{1 + e^{-(-5.659+0.107(\text{Gender})+0.160\ (\mathrm{PI}_A))}}, \tag{2}$$

$$z_i = -6.131 + 0.415(\text{Gender}) + 0.174\ (\mathrm{PI}_S), \tag{3}$$

$$P(\mathrm{CVD}s = 1|\mathrm{PI}_S) = \frac{1}{1 + e^{-z_i}} = \frac{1}{1 + e^{-(-6.131+0.415(\text{Gender})+0.174\ (\mathrm{PI}_S))}}. \tag{4}$$

The performance of newly developed NLHRS-based risk prediction models was assessed and reported in Table 3. The results showed that the SVM-RS-based model had a relatively better accuracy of 83.50% with 0.888 AUC and the lowest RMSE value. The ANN-RS-based model provided 81% accuracy with 0.876 AUC. The accuracy of the ANN-RS-based risk prediction model was lower than the SVM-RS-based model. SVM-RS-based risk prediction model yielded improved findings than the original ML-based RPM provided in Table 1. However, the ANN-RS-based model provided a slightly compromised performance than the original ML-based risk prediction models. This slight loss of performance can be justified and possibly due to the small to medium size sample of the study. However, it shows that ML-based complex models are “wrapped up” within the newly developed simple NLHRS-based models.

3.4. Validation Process of NLHRS-Based Risk Prediction Models.

In the medical sciences, the validity of the risk prediction model needs to undergo more rigorous evaluation. Therefore, there is a need for at least valid internal predictions of individuals from the underlying population [38]. This section specifically answers the question, “Are NLHRS-based risk prediction models valid?” Various

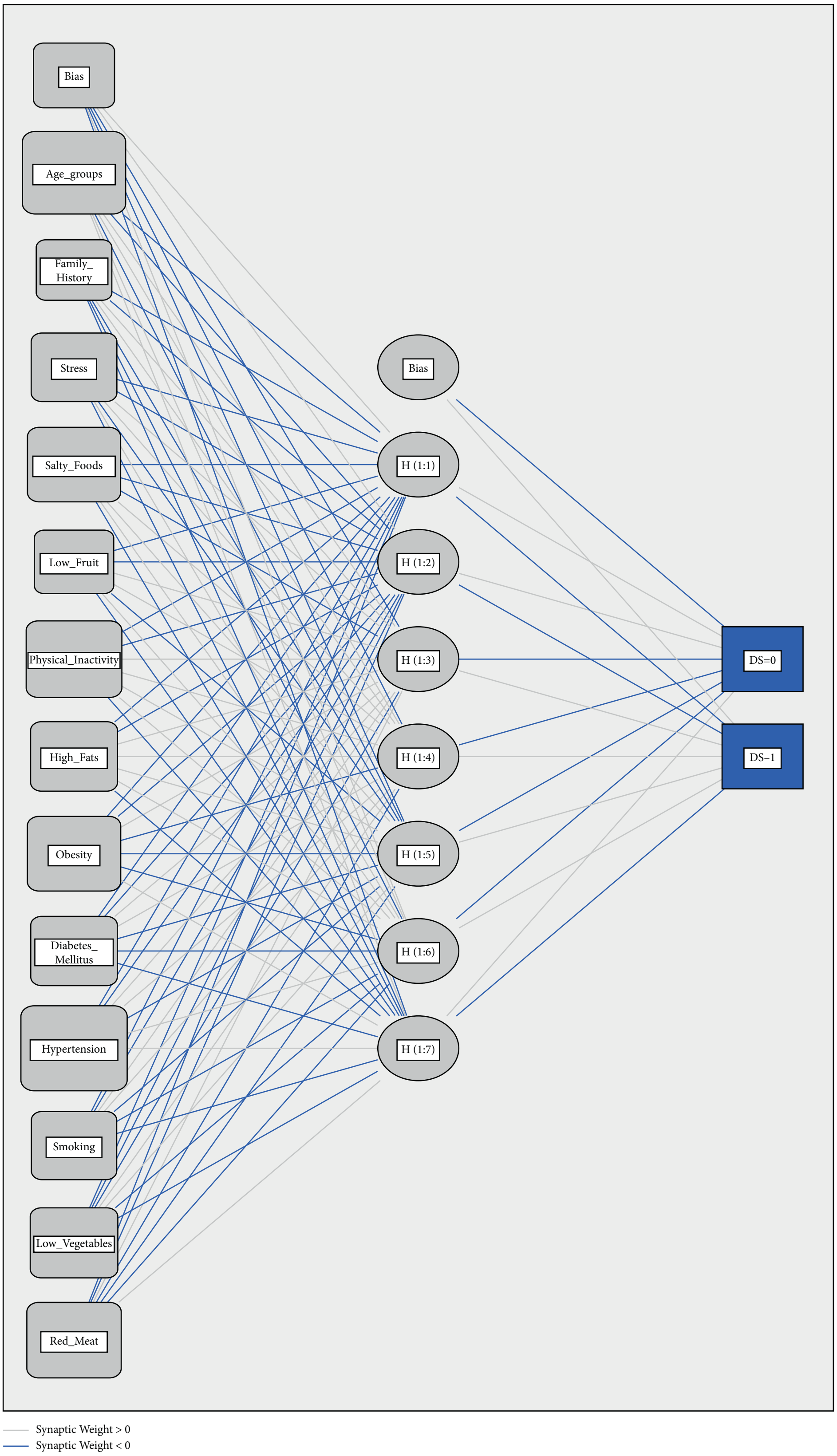


Figure 2: Illustration of finalized ANN model for prediction of disease status (DS).

Table 2: Extraction of relative feature weights using ANN and linear SVM.

Features	Artificial neural network		Linear support vector machine	
	Sum of input feature contribution	Relative feature weights (%)	Original feature weights	Relative feature weights (%)
Age groups	0.836	11.944	1.085	8.828
Parental history of CVDs	0.401	5.735	0.330	2.683
Self-reported general stress	0.360	5.138	0.732	5.957
Consumption of high salty foods	0.429	6.128	1.024	8.338
Low fruit consumption	0.666	9.517	0.832	6.773
Physical inactivity	0.562	8.024	1.167	9.499
High fried foods/trans fats	0.404	5.770	0.857	6.976
Abdominal obesity	0.406	5.794	1.046	8.512
Diabetes mellitus	0.446	6.371	0.978	7.961
Hypertension	0.737	10.527	1.386	11.283
Smoking history	0.635	9.067	0.969	7.887
Low vegetables consumption	0.588	8.394	0.894	7.273
Red meat/poultry consumption	0.531	7.591	0.987	8.030
Total	**7.000**	**100.000**	**12.286**	**100.000**

Table 3: Performance assessment of NLHRS-based risk prediction models.

Models	ANN-RS		SVM-RS	
Confusion matrix	Case	Control	Case	Control
Case	190	40	190	40
Control	48	182	36	194
Accuracy	81.000		83.500	
Sensitivity	0.826		0.826	
Specificity	0.791		0.843	
Kappa statistic	0.620		0.670	
AUC	0.876		0.888	
RMSE	0.378		0.362	

statistical tests were used and reported here to evaluate the internal validity of NLHRS and its corresponding risk prediction models. The ideal values of these tests are provided in Table 4. The value of the BS score, which is close to 0, is ideal. It is found that the SVM-RS-based model has the lowest BS value of 0.115. The ANN-RS-based model provided a BS value of 0.142 that is far from the SVM-RS-based model. The lower value of BS indicates it has relatively well discriminated and calibrated NLHRS, but it also needs support from a test of significance. Therefore, Spiegelhalter's Z-statistic was used to assess the H_0, which is similar to the H-statistic. The smaller value of Spiegelhalter's Z-statistic reflected a good overall calibration by the model. ANN-RS- and SVM-RS-based models provided −1.791 and −1.443 values of Spiegelhalter's Z-statistic, respectively, with p-value > 0.05. It means SVM-RS and ANN-RS provide well-calibrated NLHRS-based models.

The SVM-RS-based risk prediction model achieved the highest value of AUC, but the ANN-RS-based model also depicts comparable findings with a point estimate of 0.876. Figure 3 is the presentation of the discriminating capability of these two finalized models. These values indicate that the SVM-RS-based model had a better discriminating ability than the other model.

From Table 4, ANN-RS and SVM-RS showed good calibration as both models have H-statistic <20. Both NLHRS-based risk prediction models have acceptable levels of calibration. Furthermore, the required level of significance (>0.05) in both models was also achieved. It can also be observed from Figure 4 that ANN-RS-based risk prediction models showed better closeness with the ideal line as compared to SVM-RS models. Overall, the calibration plots graphically augmented the H-statistic findings and found the slight superiority of ANN-RS over the SVM-RS-based risk prediction model. However, irrespective of good calibration, the ANN-RS-based model had slightly compromised accuracy as discussed previously.

Overall, the performance and validity of NLHRS-based risk prediction models showed that both ANN-RS- and SVM-RS-based models yielded stable results, especially in the tests of significance such as H-statistic, Spiegelhalter's Z-statistic, and AUC. Except for calibration measurement, the SVM-RS-based risk prediction model was found to be more consistent in the performance and validation process. Therefore, these simple models with almost the same accuracy can reinforce the interpretability of initially developed complex ML models.

Table 4: Validation of NLHRS-based risk prediction models.

Assessment	Test statistic/criteria	Ideal value	ANN-RS	SVM-RS
Overall discrimination and calibration	Brier mean probability score	0	0.142	0.115
Overall comparison of models	Spiegelhalter's Z-statistic (p-value)	0 ($p > 0.05$)	−1.791 (0.073)	−1.443 (0.150)
Calibration	H-statistic (p-value)	<20 ($p > 0.05$)	13.719 (0.089)	14.427 (0.071)
Discrimination	AUC	1	0.876	0.888

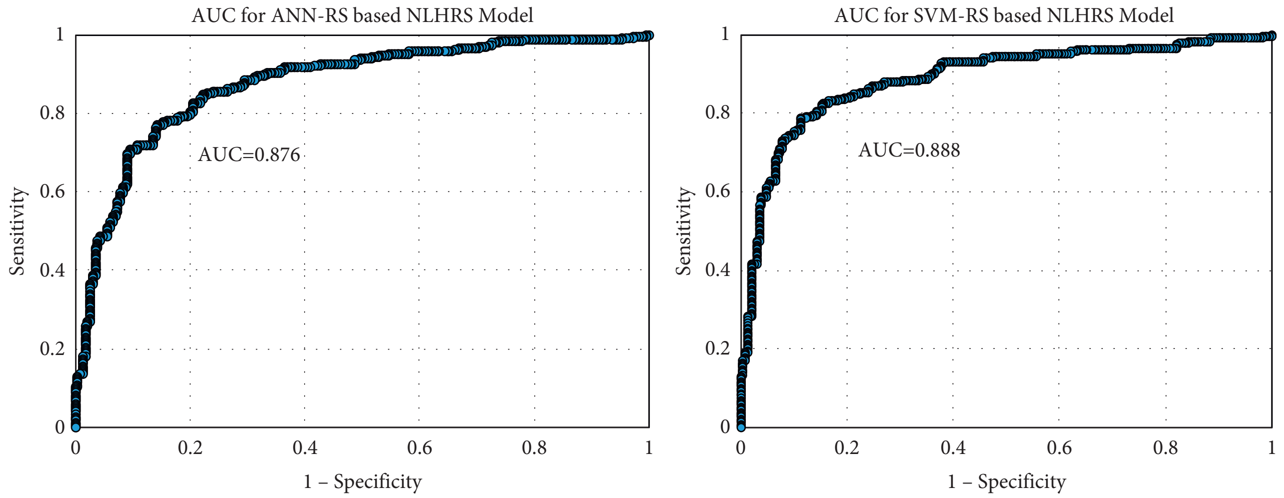


Figure 3: Discrimination strength of ANN-RS- and SVM-RS-based risk prediction models.

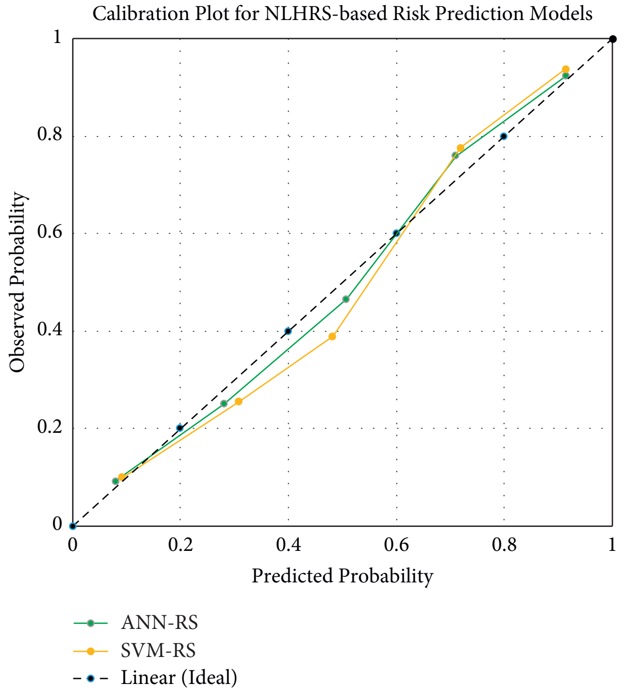


Figure 4: Calibration chart for NLHRS-based risk prediction models.

4. Discussion

The scope and application of complex ML models in decision support systems are continuously increasing due to their high performance. However, the usage of these effective but complex opaque models is a real challenge for practitioners. To help them, in this study, we proposed and implemented a methodology that efficiently transformed the complex supervised ML models into simple interpretable but approximately accurate statistical models. This methodology has been tested on a CVD-related data set. Two newly developed heart risk scores (ANN-RS and SVM-RS) and their corresponding models were found consistent in performance matrices, specifically with good discriminative and predictive strength. In addition to these matrices, the validation process of these models also augmented our proposed methodology. The small difference between the performance of complex ML-based risk prediction models and the transformed simple NLHRS-based risk prediction models is the key output of this proposed methodology.

Literature has shown an upward surge in the development and implementation of new ML algorithms. The performance of these algorithms is almost equal to humans or even surpasses them [1]. However, this constant improvement in the performance of these algorithms also increases their complexity, which attracted the attention of researchers [5, 39, 40]. The interpretability of models is an attractive characteristic, but the requirement for high performance in models cannot be undermined. Therefore, the balance of this trade-off is the ultimate requirement of researchers in recent works [1]. The literature has discussed that the simplicity of risk scores or models can come at the expense of performance, especially in terms of discrimination and calibration [41, 42, 43], and could increase the cost to the health system by providing wrong predictions

[16]. This study provides a methodology that extracts the relative weights of features from best-performed complex ML models and utilizes them in a simple LR model to maintain the balance between interpretability and high performance of models. This unique combination of complex ML and simple LR models provides a hybrid approach of high performance and interpretability.

Our proposed methodology has provided models that are interpretable, can maintain performance, are transparent in the process, can be validated, and can be presented in statistical equations. Interestingly, these multiple characteristics are acquired at multiple stages of the proposed methodology. In the first stage, the actual input features were processed through complex ML models by considering possible problems of nonlinearity and interaction effects. This step provided higher accuracy than traditional statistical models such as LR. Using the relative feature weights approach, it tried to estimate the real weights utilized in the best ML models. This extraction of the relative weights of features highlighted the individual role of features in complex ML models and provided a transparent mechanism. The index of these relative feature weights acts as a single covariate in the LR model equation to form a simple statistical model. This step leads to simplicity, interpretability, and statistical form in the newly derived models. This simple LR model equation can easily be used due to its excellent interpretative capabilities, especially for medical scientists. Finally, parallel to existing LR-based models, newly developed models can also be validated along the same lines. These models also offer a continuous form of probability, but in simplicity, they are comparable to the existing semiquantitative models [44], which simply use an index of features for risk estimation of CVDs.

The newly developed heart risk scores, NLHRS, and their associated risk prediction models fulfill the majority of the criteria of a valid model, especially in terms of discrimination and calibration. Therefore, these two NLHRS-based risk prediction models can be used as screening tools for future purposes. In addition, this validated score can help predict the need for cardiac services in the different areas, especially in the absence of a good database. It also helps guide decision-makers to establish a cardiac unit or cardiac centre. The predictive performance and discrimination strength of the linear SVM-based NLHRS and its risk prediction model are relatively better than the ANN-RS-based model. Furthermore, the performance of linear SVM-based NLHRS and its related model has been improved from the original risk prediction models. However, the slightly compromised performance of ANN-based NLHRS and its model can be improved by using large data sets [45] and new methodologies for the extraction of feature weights. The findings of this study should be interpreted in the following limitations. Firstly, the proposed methodology has been tested on the CVDs data set that has been specifically collected for this purpose. Future researchers can test this methodology on other domains as well. Secondly, this study mainly focused on binary independent features. It is because the majority of risk factors of CVDs have binary nature. However, the proposed methodology can also be implemented on other types of data sets as well.

5. Conclusions

The conversion of complex ML models into simple but valid statistical models is the pertinent objective of the study. A novel methodology was adopted that uses the philosophy of stacking ensemble ML methods that combine different models for better performance. However, this complex process has been unfolded and yielded simple statistical models, which are interpretable, transparent, and valid. The proposed methodology had been tested on the primary data set and efficiently provided two valid heart risk scores, NLHRS, and their corresponding risk prediction models. Performance matrices generated by NLHRS-based models are approximately equal to the original complex ML-based risk prediction models. Therefore, it is suggested that this simple but transparent methodology can be used to develop interpretative models that use the efficient output of complex ML models as input. Furthermore, this validated risk score will have the potential for future use to predict cardiac diseases that will have an impact on clinical settings and future decision-making.

Data Availability

The source of primary data used in this study is given in the manuscript. However, it is available on request to the corresponding author.

Ethical Approval

This study followed the guidelines of the declaration of Helsinki and was approved by the ethical review committee (approval number: RTPGME-Research-91) of the Punjab Institute of Cardiology (PIC) hospital, Lahore, Pakistan.

Consent

Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors would like to thank the Ministry of Higher Education, Malaysia, providing motivational support under the Fundamental Research Grant Scheme (FRGS) No. FRGS/1/2019/STG06/UMP/02/11 (University reference RDU1901190). The authors also thank Dr. Ahmad Shahbaz (Professor of Cardiac Surgery at PIC Hospital, Pakistan) for his support in the data collection process.

References

[1] A. Gosiewska, A. Kozak, and P. Biecek, “Simpler is better: Lifting interpretability-performance trade-off via automated feature engineering,” *Decision Support Systems*, vol. 150, Article ID 113556, 2021.

[2] Y. Lou, R. Caruana, and J. Gehrke, “Intelligible models for classification and regression,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150–158, Beijing, China, August 2012.

[3] D. Gunning, *Explainable Artificial Intelligence (XAI)*, Defense Advanced Research Projects Agency (DARPA), 2017, https://www.darpa.mil/attachments/XAIProgramUpdate.pdf.

[4] The Lancet Respiratory Medicine, “Opening the black box of machine learning,” *Lancet Respiratory Medicine*, vol. 6, no. 11, p. 801, 2018.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ““Why should i trust you?” Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, California, San Francisco, USA, August 2016.

[6] J.-P. Kucklick, *Towards a Model-And Data-Focused Taxonomy of XAI Systems*, https://aisel.aisnet.org/wi2022/business_analytics/business_analytics/2, 2022.

[7] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[8] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, NSW, Sydney, Australia, August 2015.

[9] S. Mohseni, N. Zarei, and E. D. Ragan, “A multidisciplinary survey and framework for design and evaluation of explainable AI systems,” arXiv preprint arXiv:1811.11839, 2018.

[10] S. J. Cunningham, *Machine Learning and Statistics: A Matter of Perspective*, University of Waikato, Department of Computer Science, Hamilton, New Zealand, 1995.

[11] Y. Jiang, X. Zhang, R. Ma et al., “Cardiovascular disease prediction by machine learning algorithms based on cytokines in Kazakhs of China,” *Clinical Epidemiology*, vol. 13, pp. 417–428, 2021.

[12] Y. Yan, J.-W. Zhang, G.-Y. Zang, and J. Pu, “The primary use of artificial intelligence in cardiovascular diseases: What kind of potential role does artificial intelligence play in future medicine?” *Journal of geriatric cardiology: JGC*, vol. 16, no. 8, pp. 585–591, 2019.

[13] J. A. A. G. Damen, L. Hooft, E. Schuit et al., “Prediction models for cardiovascular disease risk in the general population: Systematic review,” *BMJ*, vol. 353, p. i2416, 2016.

[14] F. Kavaler and R. S. Alexander, *Risk Management in Healthcare Institutions: Limiting Liability and Enhancing Care*, Jones & Bartlett Publishers, Burlington, Massachusetts, United States, 2014.

[15] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, “Can machine-learning improve cardiovascular risk prediction using routine clinical data?” *PLoS One*, vol. 12, no. 4, Article ID e0174944, 2017.

[16] D. Han, K. K. Kolli, H. Gransar et al., “Machine learning based risk prediction model for asymptomatic individuals who underwent coronary artery calcium score: Comparison with traditional risk prediction approaches,” *Journal of cardiovascular computed tomography*, vol. 14, no. 2, pp. 168–176, 2020.

[17] P. M. Ridker, E. Danielson, F. A. H. Fonseca et al., “Rosuvastatin to prevent vascular events in men and women with elevated C-reactive protein,” *New England Journal of Medicine*, vol. 359, no. 21, pp. 2195–2207, 2008.

[18] J. H. Chen and S. M. Asch, “Machine learning and prediction in medicine—beyond the peak of inflated expectations,” *New England Journal of Medicine*, vol. 376, no. 26, pp. 2507–2509, 2017.

[19] S. Sajeev and A. Maeder, “Cardiovascular risk prediction models: A scoping review,” in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–5, NSW, Sydney, Australia, January 2019.

[20] A. M. Alaa, T. Bolton, E. Di Angelantonio, J. H. F. Rudd, and M. van der Schaar, “Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants,” *PLoS One*, vol. 14, no. 5, Article ID e0213653, 2019.

[21] M. R. Sajid, B. A. Almehmadi, W. Sami et al., “Development of nonlaboratory-based risk prediction models for cardiovascular diseases using conventional and machine learning approaches,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 23, p. 12586, 2021a.

[22] S. Suzuki, T. Yamashita, T. Sakama et al., “Comparison of risk models for mortality and cardiovascular events between machine learning and conventional logistic regression analysis,” *PLoS One*, vol. 14, no. 9, Article ID e0221911, 2019.

[23] G. Luo, “Automatically explaining machine learning prediction results: A demonstration on type 2 diabetes risk prediction,” *Health Information Science and Systems*, vol. 4, no. 1, pp. 2–9, 2016.

[24] G. Luo, “PredicT-ML: A tool for automating machine learning model building with big clinical data,” *Health Information Science and Systems*, vol. 4, no. 1, pp. 5–16, 2016.

[25] A. K. Verma and S. Pal, “Prediction of skin disease with three different feature selection techniques using stacking ensemble method,” *Applied Biochemistry and Biotechnology*, vol. 191, no. 2, pp. 637–656, 2020.

[26] M. R. Sajid, N. Muhammad, R. Zakaria et al., “Nonclinical features in predictive modeling of cardiovascular diseases: A machine learning approach,” *Interdisciplinary Sciences: Computational Life Sciences*, vol. 13, no. 2, pp. 201–211, 2021b.

[27] J. De Oña and C. Garrido, “Extracting the contribution of independent variables in neural network models: A new approach to handle instability,” *Neural Computing & Applications*, vol. 25, no. 3-4, pp. 859–869, 2014.

[28] Z. Zhang, M. W. Beck, D. A. Winkler, B. Huang, W. Sibanda, and H. Goyal, “Opening the black box of neural networks: Methods for interpreting neural network models in clinical applications,” *Annals of Translational Medicine*, vol. 6, no. 11, p. 216, 2018.

[29] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, “SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels,” *BMC Bioinformatics*, vol. 19, no. 1, p. 432, 2018.

[30] E. W. Steyerberg and F. E. Harrell Jr, “Prediction models need appropriate internal, internal-external, and external validation,” *Journal of Clinical Epidemiology*, vol. 69, pp. 245–247, 2016.

[31] E. W. Steyerberg, *Clinical Prediction Models*, Springer, New York, US, 2019.

[32] K. Rufibach, “Use of Brier score to assess binary predictions,” *Journal of Clinical Epidemiology*, vol. 63, no. 8, pp. 938-939, 2010.

[33] E. W. Steyerberg and Y. Vergouwe, “Towards better clinical prediction models: Seven steps for development and an ABCD for validation,” *European Heart Journal*, vol. 35, no. 29, pp. 1925–1931, 2014.

[34] R. B. D'Agostino Sr, M. J. Pencina, J. M. Massaro, and S. Coady, "Cardiovascular disease risk assessment: Insights from Framingham," *Global heart*, vol. 8, no. 1, pp. 11–23, 2013.

[35] P. Joseph, S. Yusuf, S. F. Lee et al., "Prognostic validation of a non-laboratory and a laboratory based cardiovascular disease risk score in multiple regions of the world," *Heart*, vol. 104, no. 7, pp. 581–587, 2018.

[36] C. McGorrian, S. Yusuf, S. Islam et al., "Estimating modifiable coronary heart disease risk in multiple regions of the world: The INTERHEART Modifiable Risk Score," *European Heart Journal*, vol. 32, no. 5, pp. 581–589, 2011.

[37] M. R. Sajid, N. Muhammad, R. Zakaria, A. Shahbaz, and A. Nauman, "Associated factors of cardiovascular diseases in Pakistan: Assessment of path analyses using warp partial least squares estimation," *Pakistan Journal of Statistics and Operation Research*, vol. 16, pp. 265–277, 2020.

[38] D. G. Altman and P. Royston, "What do we mean by validating a prognostic model?" *Statistics in Medicine*, vol. 19, no. 4, pp. 453–473, 2000.

[39] P. Hall, "On the art and science of machine learning explanations," arXiv preprint arXiv:1810.02909, 2018.

[40] N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy, "Magix: Model agnostic globally interpretable explanations," arXiv preprint arXiv:1706.07160, 2017.

[41] H. F. AlFaleh, A. A. Alsheikh-Ali, A. Ullah et al., "Validation of the Canada acute coronary syndrome risk score for hospital mortality in the gulf registry of acute coronary events-2," *Clinical Cardiology*, vol. 38, no. 9, pp. 542–547, 2015.

[42] T. Huynh, S. Kouz, A. Yan et al., "Canada Acute Coronary Syndrome Risk Score: A new risk score for early prognostication in acute coronary syndromes," *American Heart Journal*, vol. 166, no. 1, pp. 58–63, 2013.

[43] A. T. Yan, R. T. Yan, M. Tan et al., "Risk scores for risk stratification in acute coronary syndromes: Useful but simpler is not necessarily better," *European Heart Journal*, vol. 28, no. 9, pp. 1072–1078, 2007.

[44] H. Bueno and F. Fernández-Avilés, "Use of risk scores in acute coronary syndromes," *Heart*, vol. 98, no. 2, pp. 162–168, 2012.

[45] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science & Business Media, New York, US, 2009.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9828465, 1 page
https://doi.org/10.1155/2023/9828465

Retraction

Retracted: Emotion Analysis Model of Microblog Comment Text Based on CNN-BiLSTM

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. Li and S. Yi, "Emotion Analysis Model of Microblog Comment Text Based on CNN-BiLSTM," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1669569, 10 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1669569, 10 pages
https://doi.org/10.1155/2022/1669569

RETRACTED

Research Article

Emotion Analysis Model of Microblog Comment Text Based on CNN-BiLSTM

Aichuan Li[1] **and Shujuan Yi**[2]

[1] *College of Information and Electrical Engineering, Heilongjiang Bayi Agricultural University, Daqing, Heilongjiang 163319, China*
[2] *Engineering Research Center of Processing and Utilization of Grain By-Products, Ministry of Education, Heilongjiang Engineering Technology Research Center for Rice Ecological Seedlings Device and Whole Process Mechanization, Daqing, Heilongjiang 163319, China*

Correspondence should be addressed to Shujuan Yi; yishujuan@byau.edu.cn

Received 18 February 2022; Revised 24 March 2022; Accepted 19 April 2022; Published 30 April 2022

Academic Editor: Muhammad Ahmad

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

Aiming at the problems of over reliance on labor and low generalization of traditional emotion analysis methods based on dictionary and machine learning, an emotion analysis model of microblog comment text based on deep learning is proposed. Firstly, text is obtained through microblog crawler program. After data preprocessing, including data cleaning, Chinese word segmentation, removal of stop words, and so on, the Skip-gram model is used for word vector training on a large-scale unmarked corpus, and then the trained word vector is used as the text input of CNN-BiLSTM model, which combines Bidirectional Long-Short Term Memory (BiLSTM) neural network and Convolution Neural Network (CNN). Considering the historical context information and the subsequent context information, BiLSTM can better use the temporal relationship of text to learn sentence semantics. CNN can extract hidden features from the text and combine them. Finally, after Adamax optimization training, the emotion type of microblog comment text is output. The proposed model combines the learning advantages of BiLSTM and CNN. The overall accuracy of text emotion analysis has been greatly improved, with an accuracy of 0.94 and an improvement of 8.51% compared with the single CNN model.

1. Introduction

With the rapid development of information technology and the growing number of Internet users, the Internet has been inseparable from people's daily life and information has begun to show explosive growth [1]. Microblog is one of the main public opinion platforms in China, carrying a large number of information released by the public. Most of these information have emotional tendencies [2, 3]. Analyzing the text of microblog hot topics and mining the emotional tendencies are of great significance for government units or enterprises to understand the trend of public opinion [4].

Text emotion analysis refers to the analysis of subjective text with emotional color, mining the contained emotional tendency, and dividing the emotional attitude [5]. Microblog text emotion analysis belongs to a specific research field of text emotion classification. In the research field of natural language processing, the analysis and research of text emotion is an important branch. The process of text emotion analysis includes the acquisition of original data, data preprocessing, feature extraction, classifier, and emotion category output [6–8]. The acquisition of original data is generally through web crawlers to obtain relevant data; data preprocessing refers to data cleaning to remove noise. Common methods include removing invalid characters and data, unifying data categories, using word segmentation tools for word segmentation, stop word filtering, etc. Feature extraction has different implementation methods according to the different methods used, and the feature extraction of deep learning method is generally automatic [9]. The final

RETRACTED

emotional category of the text is obtained from the output of the classifier. The common classifiers are Support Vector Machine and Softmax.

In foreign countries, many achievements have been made in the emotional analysis of Twitter texts. However, in China, the research on Chinese text begins relatively late, and the research on Chinese microblog text is relatively few. Due to the differences of language expression and the diversity and complexity of Chinese expression, the research on microblog text is full of challenges [10, 11]. Reference [12] analyzed and expounded the level of emotion analysis, various emotion models, emotion analysis, and text emotion detection process, and discussed the challenges faced in the process of emotion analysis. Reference [13] proposed a dictionary-based text emotion analysis system. This method is the main standard method for modeling text for machine learning in emotion analysis methods, which can effectively solve the problem of data overload. But the method is more traditional and difficult to apply to the complex text environment of microblog. Reference [14] developed a classification model based on machine learning algorithm to realize the social website automatically filtering out the messages from the visitors for avoiding excessive speech, but the performance of the single model was slightly deficient. The traditional model based on motion dictionary is simple, fast, and stable, but it also has limitations. The traditional model needs to prepare a relatively complete motion dictionary, but this step needs a lot of human and material resources for careful operation, which has low practicability and is not conducive to the development of natural language processing.

With the continuous development of machine learning, some inherent advantages of deep learning have solved some problems. Deep learning can establish a classification model for the emotion classification of microblog text without background language knowledge to realize the emotion classification of microblog text with high accuracy [15]. For example, reference [16] proposed an image text emotion analysis model based on deep multimodal attention fusion, which used the hybrid fusion framework for emotion analysis, and realized emotion classification and recognition by combining the discriminant features and internal correlation between visual and semantic content. Reference [17] proposed a Chinese text emotion analysis method based on Convolutional Neural Network (CNN), which realized feature preprocessing by normalizing features, and optimized text feature size to improve analysis performance. However, the CNN model ignores the correlation between the whole and the part, and the analysis accuracy of comment text needs to be further improved. Reference [18] proposed a network text emotion analysis method based on the improved attention mechanism and two-way gated loop unit model. Reference [19] proposed a text emotion analysis method combining dictionary language model and deep learning to solve the problem of accurate and rapid emotion analysis of comment text in the network big data environment.

Aiming at the problems of single word quantization and poor analysis accuracy in most of the existing deep learning algorithms in text processing, an emotion analysis model of microblog comment text based on deep learning is proposed. Compared with the traditional dictionary-based and machine learning models, the innovations of the proposed model are as follows.

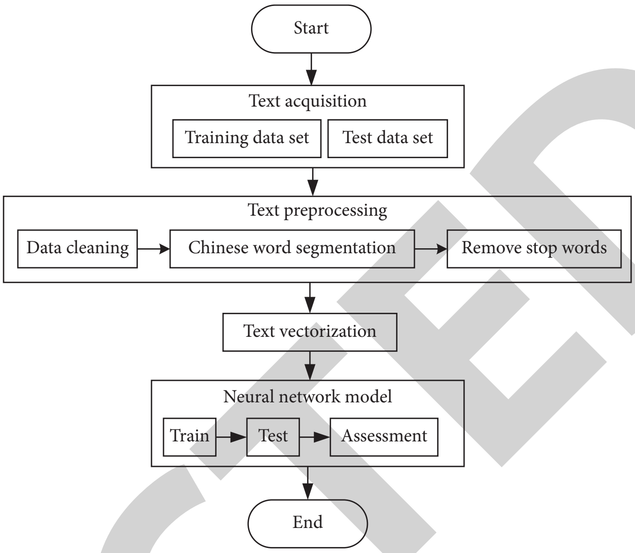


Figure 1: Text emotion analysis model process based on deep learning.

(1) In order to better analyze emotion, the Skip-gram model is used to quantify the word of the preprocessed text data set. Part of the speech features are integrated, which can significantly improve the effect of emotion analysis.

(2) Because the output of LSTM hidden layer only depends on the output of the previous time, resulting in inaccurate analysis results, the proposed model combines Bidirectional Long Short-Term Memory (BiLSTM) and CNN to construct CNN-BiLSTM model. BiLSTM can obtain previous and subsequent information, and CNN can deeply mine text features. Thus, the accuracy of text emotion analysis is improved.

2. Proposed Model

2.1. Model Process. The process of microblog comment text emotion analysis model based on deep learning is shown in Figure 1, which mainly includes four parts: text data acquisition, text preprocessing, text vectorization, and neural network model.

After the text data is obtained, it is divided into training data set and test data set through manual annotation, and the data set is preprocessed through data cleaning, Chinese word segmentation, and removal of stop words, and then the text vectorization is completed. Finally, the quantified text is sent to CNN-BiLSTM model for emotional analysis of microblog comment text.

2.1.1. Text Acquisition. In order to construct the word vector of words and emoticons, a microblog crawler program is

RETRACTED

implemented by using the open-source crawler framework WebCollecter, and the crawled microblog text is segmented by using Natural Language Processing and Information Retrieval (NLPIR) tool. In the process of word segmentation, the emotional symbols in the constructed microblog emotional symbol database are used as a word segmentation dictionary to make them exist as a basic language unit after word segmentation [20, 21]. In addition, a series of microblog text preprocessing has been carried out, such as complex and simple conversion, replacement of Uniform Resource Locator (URL), elimination of short and meaningless microblogs, etc. Finally, a Word2Vec word vector training corpus is constructed, including 40302879 microblogs and 1432646813 words. Skip-gram model is used in the training process, and other relevant parameters are set by default. After training, a word vector space containing 850599 words is obtained, in which the word vector dimension of each word is 250.

2.1.2. Text Preprocessing. Generally, the obtained text data has many invalid contents or dirty data that affect the later classification, so text preprocessing is an essential work, and good text preprocessing can improve the accuracy of later classification [22]. Text preprocessing generally includes three steps: data cleaning, Chinese word segmentation, and removing stop words.

(1) Data Cleaning. Data cleaning is to clean and remove characters and data irrelevant to text content. Taking the microblog text as an example, there are some irrelevant characters and data in the microblog text, such as forwarding symbol //, designated user symbol @, topic symbol #, microblog expression, HTML tag, URL link, and so on, which are irrelevant to the content expressed by the microblog and can affect the result of emotion analysis task. For these data, the strategy of deletion or replacement can be adopted. The regular expression is used to delete the character data other than the microblog expression, and the microblog expression is replaced with the corresponding Chinese text.

(2) Chinese Word Segmentation. Because Chinese text is not like English text, which is naturally separated by spaces, word segmentation must be done for Chinese text, and a Chinese text must be divided into individual words. At present, Chinese word segmentation can be divided into two methods: machine learning-based and rule-based (rule refers to word segmentation specification and thesaurus). In 2002, researchers treated Chinese word segmentation as part of speech tagging. After tagging each word, they used a supervised machine learning model for word segmentation. Then, the fusion model of BiLSTM and Conditional Random Field (CRF) was used for Chinese word segmentation. At the same time, the research showed that the machine learning method based on statistics was better than the traditional rule-based method, especially in the recognition of unlisted words (words not included in the dictionary). With the development of deep learning, deep learning has begun to be applied in the field of emotion analysis. Although deep learning has not shown excellent technical advantages over the traditional supervised machine learning method, it still has potential. In the proposed method, word segmentation is only a pre-step in the text emotion analysis task, so it is decided to use the existing Jieba Chinese word segmentation tool for word segmentation.

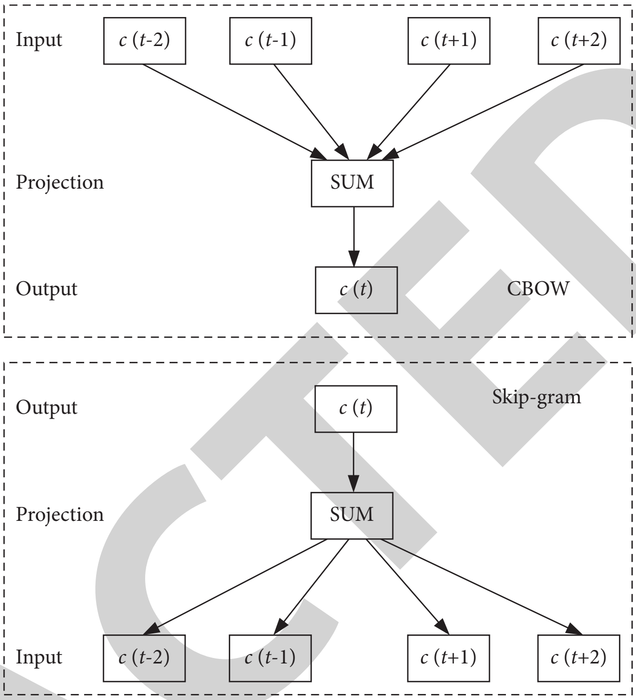


Figure 2: Structures of CBOW model and Skip-gram model.

(3) Remove Stop Words. Similar to information retrieval, in which some words shielded in order to improve query efficiency and save storage space, in the text emotion analysis task, some words unrelated to emotional tendency will also be removed, which are called stopwords. In Chinese text, in addition to nouns, adjectives, verbs, adverbs, and other emotional words, there are also some meaningless words, such as "do, bar, do, also, to, from, then, in addition, then". These meaningless words generally belong to prepositions, mood auxiliaries, and conjunctions. Removing these meaningless stopwords through the stopwords list can reduce the dimension of feature vector and reduce data noise and data volume.

2.1.3. Text Vectorization. The word sequence obtained through text preprocessing needs to be transformed into a vector used as the final input of the model [23]. Word2Vec launched by Google is a commonly used word vector tool, which is mainly based on Continuous Bags of Words (CBOW) model and Skip-gram model. The structures of the two models are shown in Figure 2. CBOW model uses the context of the current word c_t to predict the probability of the current word c_t, and Skip-gram model uses c_t to predict the probability of its context. The proposed model uses Skip-

RETRACTED

gram model to train word vectors on large-scale unmarked corpus, and takes the trained word vectors as the input word vectors of the model.

Skip-gram neural network model can be constructed and implemented through two frameworks: Hierarchical Softmax and Negative Sampling (NEG). The proposed model adopts the method of Negative Sampling, including three-layer structure of input layer, projection layer, and output layer. The text samples $c_{t-n}, \ldots, c_{t-1}, c_t, c_{t+1}, \ldots, c_{t+n}$ in Corpus are recorded as samples $(c, \text{context}(c))$. The Negative Sampling method divides words into positive samples and negative samples. The central word c is negative samples, and the rest are positive samples. In the output layer, the Negative Sampling technology is adopted. For any $\varpi \in \{u\} \cup \text{NEG}(u)$, $\text{NEG}(u)$ represents negative sample set generated when processing word u, the posterior probability of which is as follows:

$$p(\varpi|c) = \begin{cases} \sigma\left(v(c)^T \theta^{\varpi}\right), L^u(\varpi) = 1, \\ 1 - \sigma\left(v(c)^T \theta^{\varpi}\right), L^u(\varpi) = 0, \end{cases} \tag{1}$$

where σ represents Sigmoid activation function; $v(c)$ represents the target word; and θ represents the ϖ-th word in the window except the target word.

The Skip-gram language model requires to maximize the probability of words appearing in the context. The optimized objective function is in the following form:

$$\begin{aligned} G &= \prod_{c\in C} \prod_{u\in \text{Context}(c)} \prod_{\varpi\in\{u\}\cup\text{NEG}(u)} p(\varpi|c), \\ L &= \sum_{c\in C} \sum_{u\in C(c)} \sum_{\varpi\in\{u\}\cup\text{NEG}(u)} \begin{array}{c} \left\{L^u(\varpi)\cdot\log\left[\sigma\left(v(c)^T\theta^{\varpi}\right)\right]\right\}+ \\ \left[1 - L^u(\varpi)\right]\cdot\log\left[\sigma\left(v(c)^T\theta^{\varpi}\right)\right] \end{array}. \end{aligned} \tag{2}$$

2.2. Text Emotion Analysis Based on CNN-BiLSTM

2.2.1. CNN-BiLSTM Model Structure. The hidden layer output of the standard LSTM at a certain time only depends on the output of the previous step, so it is only related to the pre-context, that is, it can only see the historical information. However, in the actual situation of text emotion analysis, it should be related to both the pre-context and the post-context, that is, it can not only see the historical information, but also pay attention to the future information [24]. BiLSTM can exactly do this because it can obtain both positive and reverse semantic information. Its model structure is shown in Figure 3.

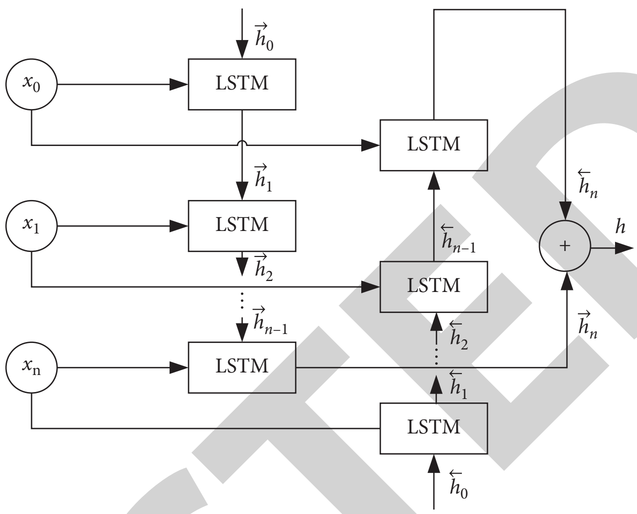


Figure 3: Structure of BiLSTM model.

Therefore, the proposed model uses the fusion of BiLSTM and CNN to construct the CNN-BiLSTM model, and its structure is shown in Figure 4. The input of the model is the word vector matrix integrating part of speech features trained by Word2Vec tool, and the output is the text emotion classification result.

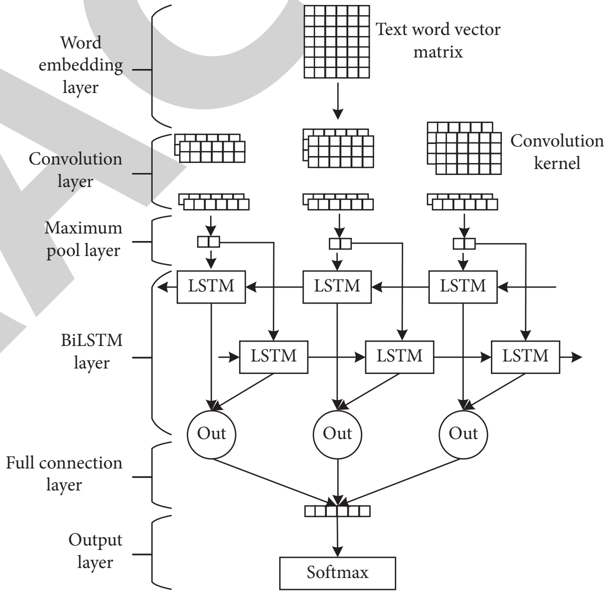


Figure 4: Structure of CNN-BiLSTM model.

2.2.2. Convolution. The convolution layer uses convolution kernel to perform convolution abstract operation on text word vector, so that the original word vector sequence becomes a convoluted abstract meaning sequence. For a given text sample $\mathbf{x}$, it can be expressed as $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, where $\mathbf{x}_i \in \mathbf{R}^d$ represents the d dimension word vector of the i-th word in the text sample $\mathbf{x}$. n represents the number of words in the text, so the text sample can be expressed as $\mathbf{x} \in \mathbf{R}^{n\times d}$. The number of words contained in the convolution kernel is k, and its vector is $\mathbf{m} \in \mathbf{R}^{k\times d}$. Usually in image processing, the convolution kernel is square, such as 6×6. Then, the convolution operation is performed by gradually moving the convolution kernel along the width and height on the picture. However, since the input in the natural language processing task is a $n \times d$ dimensional word matrix, the convolution kernel slides only in height, but in width, it is consistent with the dimension of the word vector [25]. This ensures that the position of each window sliding is a complete word vector

RETRACTED

without convolution of some word vectors of several words, and ensures the rationality of words as the minimum granularity in the language. Therefore, for each position i in the text sample $\mathbf{x}$, there is a window vector $\mathbf{s}_i$ containing a continuous k word vectors, which is expressed as follows:

$$\mathbf{s}_i = \left[\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+k+1}\right]. \tag{3}$$

The feature map $\varphi \in \mathbf{R}^{n-k+1}$ is calculated by sliding the windows one by one through the convolution kernel, $\varphi = [\varphi_1, \varphi_2, \ldots, \varphi_{n-k+1}]$. The calculation of φ_i in each sliding window is as follows:

$$\boldsymbol{\varphi}_i = \delta\left(\boldsymbol{\omega}_i * \mathbf{m} + \mathbf{b}\right), \tag{4}$$

where $*$ represents the multiplication of corresponding matrix elements, $\mathbf{b}$ is the offset vector, function δ is the activation function, and ω is the weight matrix. The proposed model selects ReLU function as the activation function.

The designed convolution layer generates multiple feature maps through j convolution kernels, and all convolution kernels have the same size. j feature maps are obtained from the window vector $\mathbf{s}_i$ at each position i in the text sample $\mathbf{x}$, and a new feature representation Ψ_i is obtained by combining the feature vectors φ_i at the same position:

$$\boldsymbol{\Psi}_i = \left[\boldsymbol{\varphi}_{i1}, \boldsymbol{\varphi}_{i2}, \ldots, \boldsymbol{\varphi}_{ij}\right]. \tag{5}$$

The final feature representation is $\Psi \in \mathbf{R}^{(n-k+1)\times j}$. $\Psi = [\Psi_1, \Psi_2, \ldots, \Psi_j]$ is used as the input of the next Max-pooling layer.

2.2.3. Pooling Layer. In the convoluted feature maps, the average pooling or Max pooling is usually used to reduce the amount of data, so as to reduce the parameters and reduce the amount of calculation. The proposed model adopts the Max-pooling method, and the mathematical expression is as follows:

$$\tilde{\boldsymbol{\varphi}} = \max\left(\boldsymbol{\varphi}_i\right). \tag{6}$$

It can be seen from equation (6) that for the feature maps obtained by convolution kernel, only the feature map with the highest score is retained.

There are two reasons for using the Max-pooling method. First, the calculation of the upper layer is reduced by eliminating the non-maximum value, and the most significant information can be retained by extracting the local dependencies of different convolution kernels.

2.2.4. BiLSTM Layer. The deep features after Max pooling will be used as the input of BiLSTM in this layer. Compared with LSTM, BiLSTM can focus on the future context on the basis of considering the past context. In addition, when accepting text vectors, the order is from left to right, which will lead to the more backward words that are more likely to affect the emotion of the whole sentence, which is unreasonable. BiLSTM can better alleviate this. The hidden layer of BiLSTM includes forward connection and backward connection, which are used to learn the forward information and backward information of text, respectively, and are both connected to the output layer unit. The backward connection output vector is obtained through the following operations:

$$\begin{aligned}
\overleftarrow{\mathbf{f}}_t &= \sigma\left(\overleftarrow{\boldsymbol{\omega}}_{1f}\mathbf{x}_t + \overleftarrow{\boldsymbol{\omega}}_{2f}\mathbf{h}_{t+!} + \overleftarrow{\mathbf{b}}_f\right),\\
\overleftarrow{\boldsymbol{\phi}}_t &= \sigma\left(\overleftarrow{\boldsymbol{\omega}}_{1i}\mathbf{x}_t + \overleftarrow{\boldsymbol{\omega}}_{2i}\mathbf{h}_{t+!} + \overleftarrow{\mathbf{b}}_i\right),\\
\overleftarrow{\mathbf{C}}_t &= \overleftarrow{\mathbf{f}}_t * \overleftarrow{\mathbf{C}}_{t+1} + \overleftarrow{\boldsymbol{\phi}}_t * \tanh\left(\overleftarrow{\boldsymbol{\omega}}_{1c}\mathbf{x}_t + \overleftarrow{\boldsymbol{\omega}}_{2c}\mathbf{h}_{t+!} + \overleftarrow{\mathbf{b}}_c\right),\\
\overleftarrow{\mathbf{o}}_t &= \sigma\left(\overleftarrow{\boldsymbol{\omega}}_{1o}\mathbf{x}_t + \overleftarrow{\boldsymbol{\omega}}_{2o}\mathbf{h}_{t+!} + \overleftarrow{\mathbf{b}}_o\right),\\
\overleftarrow{\mathbf{h}}_t &= \overleftarrow{\mathbf{o}}_t * \tanh\left(\overleftarrow{\mathbf{C}}_t\right),
\end{aligned} \tag{7}$$

where $\overleftarrow{\mathbf{f}}_t 1/2$, $\overleftarrow{\phi}_t$, $\overleftarrow{\mathbf{o}}_t$ are the outputs of backward LSTM forgetting gate, input gate and output gate at time t, respectively; $\overleftarrow{\mathbf{C}}_t$ represents the output of the backward memory control unit after time t. $\overleftarrow{\omega}_1$ and $\overleftarrow{\omega}_2$ are the weight matrices of backward categories, and its subscript represents the specific category; $\mathbf{x}_t$ and $\mathbf{h}_{t+1}$ represent the input and hidden layer vectors at time t and, $t+1$ respectively; $\mathbf{b}$ represents the backward offset vector, and its subscript represents the specific category.

Finally, the output vector representation $\mathbf{h}$ of BiLSTM hidden layer is the combination of forward connection output $\overrightarrow{h}$ and backward connection output $\overleftarrow{\mathbf{h}}$, and also serves as the input of the next layer:

$$\mathbf{h} = [\overrightarrow{\mathbf{h}} \oplus \overleftarrow{\mathbf{h}}]. \tag{8}$$

2.2.5. Full Connection Layer and Classifier. The full connection layer receives the output features of the BiLSTM layer as the input, and then outputs them to the final output layer after comprehensive processing. Each neuron in the full connection layer is connected to all neurons in the previous layer in order to integrate features.

The output layer selects Softmax as the classifier, which outputs the results of text emotion analysis. The Softmax classifier is calculated as follows:

$$p_i = \frac{e^{\left(\vartheta_i^T x\right)}}{\sum_{q=1}^{Q} e^{\left(\vartheta_q^T x\right)}}, \tag{9}$$

where p_i represents the probability that the classification result is category i; Q indicates the number of all categories; and $\vartheta_i^T x$ represents multiple inputs. Each output is in the interval [0, 1], the probability sum of all categories is 1, and the category with the highest probability is taken as the final result of the text emotion analysis task.

2.3. Optimization Algorithm. As everyone knows, the optimization algorithm plays an important role in the success or failure of the deep neural network model. Because the main role of the optimization algorithm is to find the global

```
Input
Step length $\alpha$;
Exponential decay rate $\beta_1, \beta_2 \in [0, 1)$;
Random objective function $\mu(\tau)$;
Initial parameter quantity $\tau_0$.
Begin
(1)    $t = 0$
(2)    While $\tau_0$ does not converge,
           $t = t + 1$;
         Calculate the random target gradient of time step $t$: $g_t = \nabla_\tau \mu_t(\tau_{t+1})$;
         Update estimated deviation: $\lambda_t = \beta_1 \cdot \lambda_{t-1} + (1 - \beta_1)$;
         Update exponentially weighted infinite norm;
         Update parameters: $\tau_t = \tau_{t-1} - (\alpha/(1 - \beta_1^t)) \cdot \lambda_t/u_t$
(3)    Return $\tau_t$.
(4)    Output parameters $\tau_t$.
    End
```

Algorithm 1: Pseudo code of Adamax optimization algorithm.

optimal solution by gradient descent in the back propagation of neurons and update the connection weight of neurons, an excellent optimization algorithm can accurately find the global optimal solution in the shortest time and accurately update the weight matrix of connected neurons. Different optimization algorithms are selected for the same model to produce different results. Common optimization algorithms include Stochastic Gradient Descent (SGD), RMSprop, Adadelta, Adam, etc. The proposed model adopts Adamax optimization algorithm, which is a variant of Adam algorithm. Adam optimization algorithm is a random objective function optimization algorithm based on adaptive estimation of low-order matrix proposed by Diederik P. Kingma et al. in 2015. This method has the advantages of simple implementation, high calculation efficiency, and less memory requirements. It is very suitable to solve the problem of large data and parameter scale.

In Adam, the gradient of each individual weight in the update rule is inversely proportional to the L2 norm of the gradient at the current time and the previous time, so the update rule based on L2 norm can be extended to the update rule based on LP norm. With the increase of p, it begins to become numerically unstable. In the special case of $p \longrightarrow \infty$, the original algorithm becomes a very simple and stable algorithm, which is Adamax. Its description is shown in algorithm 1.

3. Experiment and Analysis

The specific parameters of the microblog data set used in the experiment, such as sentence length, batch size, and Dropout value, are shown in Table 1. The server hardware environment used in this experiment is: graphics card GTX1080Ti dual card, 22 G video memory, CPU2.6 GHz, 16 threads, 16 G memory capacity, and 240 G hard disk capacity; and the operating system is Ubuntu 16.04.

Table 1: Model parameter setting.

Parameter	Value
Sentence length	80
Batch size	128
Word vector dimension	300/330
Convolution window size	3,4,5
Number of convolution kernels	256
Number of BiLSTM hidden units	256
Activation function	ReLU
Loss function	Cross entropy
Learning rate	0.0001

3.1. Evaluation Criteria.

3.1. Evaluation Criteria. The proposed model uses Precision and Recall as the indicators to evaluate the results of emotion classification in positive emotion samples and negative emotion samples, respectively, and Accuracy is used as the evaluation index for the overall performance. TP (True Positive) represents the number of samples with positive emotion and correctly predicted as positive samples; FP (False Positive) indicates the number of samples with negative emotion and incorrectly predicted as positive samples; TN (True Negative) indicates the number of samples with negative emotion and correctly predicted as negative samples; FN (False Negative) indicates the number of samples with positive emotion and incorrectly predicted as negative samples.

(1) Precision, representing the proportion of text correctly classified into this category in text actually classified into this category, is calculated as follows:

$$\text{Precision}_p = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{Precision}_n = \frac{\text{TN}}{\text{TN} + \text{FN}}. \tag{10}$$

(2) Recall, representing the proportion of text correctly classified into this category in all text of this category, is calculated as follows:

RETRACTED

$$\text{Recall}_p = \frac{\text{TP}}{\text{TP} + \text{FN}},$$
$$\text{Recall}_n = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{11}$$

(3) Accuracy, the proportion of correct prediction, is calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{12}$$

3.2. Effect of Epoch Parameter on Model Results. In deep learning, Epoch refers to a complete iteration, including a forward transfer and a reverse transfer of all the training samples. The experiment is based on CNN-BiLSTM model. Except for Epoch parameters, other parameters remain unchanged. The accuracy under different Epochs is shown in Figure 5.

As can be seen from Figure 5, with the increase of Epoch, the accuracy of the training set in the text emotion analysis task continues to improve, while the test set first increases and then decreases. When the 7th Epoch is trained, the accuracy on the test set reaches the maximum, and then the accuracy begins to decline, which may be due to over-fitting at the beginning of training. It can be concluded that too much or too little Epoch values have an impact on the results of text emotion analysis. When there is too little Epoch, the optimal results will not be obtained, but when there are too many training times, the model will over fit the training data and perform poorly in the test set. Therefore, the value of Epoch is very important for the evaluation of model performance. According to the experimental results, Epoch is set to 7. At this time, the analysis performance of the model is ideal.

3.3. Dropout Value Comparison Experimental Results. The introduction of Dropout can remove some neurons and update the weight and bias term through the remaining neurons in the process of forward calculation and back propagation. Then, the removed neurons are restored. Next, some neurons are removed again according to a certain probability, update the weight and bias term, and repeat this process until the end of neural network training. This can improve the generalization ability of the model to a certain extent and prevent the model from over-fitting. On the premise that other parameters remain unchanged, change the Dropout value. The accuracy results of the proposed model for the emotion analysis of microblog comment text are shown in Figure 6.

As can be seen from Figure 6, when the Dropout value is 0.3, the accuracy is the highest, up to 0.93. When the Dropout value is too low or too high, the accuracy performance is not high. This is because when the Dropout value is too low, too many neurons participate in training and are easy to fall into over-fitting. When the Dropout value is too high, too few neurons are left and easy to fall into under-fitting. Therefore, the Dropout value of the model proposed in the subsequent comparative experiment is set to 0.3.

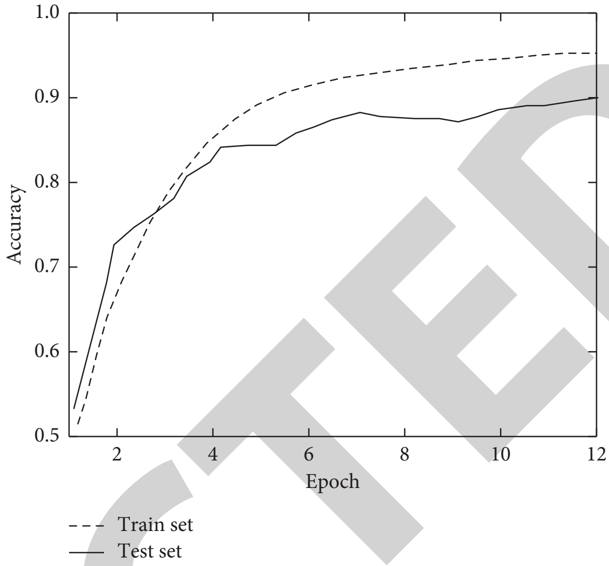


Figure 5: Comparison of training results under different Epochs.

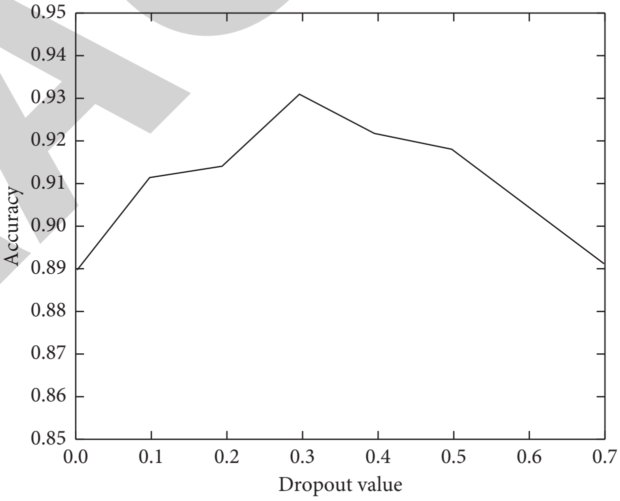


Figure 6: Comparison results of Dropout values.

3.4. Comparison of Training Results of Unidirection and Bidirection Long-Term and Short-Term Memory Models. The CNN-BiLSTM model is based on the combination of CNN and BiLSTM. On the one hand, because the convolution layer of CNN can effectively extract deeper and more abstract emotional features. On the other hand, the model combining the advantages of the two neural networks is better than the single neural network model. BiLSTM has a higher accuracy of emotion classification than unidirection LSTM because it can better take into account the previous and subsequent information. In order to intuitively show the advantages of CNN-BiLSTM model, compare it with the analysis results of CNN, LSTM, and BiLSTM models, and their accuracies are shown in Figure 7.

RETRACTED

RETRACTED

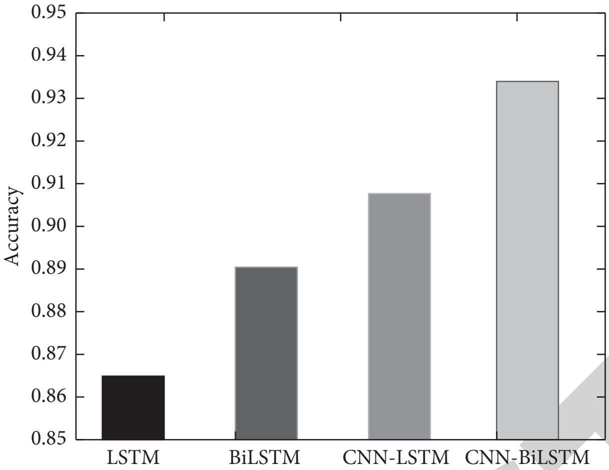


Figure 7: Comparison results of unidirectional and bidirectional LSTM models.

Table 2: Experimental results of different models.

		Ref. [13]	Ref. [14]	Ref. [17]	Proposed model
Positive emotion	Precision	0.71	0.78	0.86	0.95
	Recall	0.83	0.84	0.82	0.89
Negative emotion	Precision	0.76	0.82	0.91	0.96
	Recall	0.69	0.75	0.80	0.88
Accuracy		0.74	0.78	0.86	0.94

The experimental results of crawling Microblog data set show that BiLSTM can better analyze the information before and after the text, so as to improve the accuracy of emotion analysis. Compared with LSTM, its accuracy is improved by 2.89%. Similarly, the analysis accuracy of the BiLSTM model combined with CNN is higher than that of CNN-LSTM, about 0.935, which also demonstrates the feasibility of the proposed model.

3.5. Comparison with Other Algorithms. In order to demonstrate the performance of the proposed model, compare and analyze it with the models in references [13, 14] and [17], optimize the training parameters of each comparative experiment for many times, and select the experimental data with the best effect. The statistics of the experimental results are shown in Table 2.

As can be seen from Table 2, compared with other models, the analysis performance of the proposed model is the best, and its overall analysis accuracy is 0.94. Taking the analysis of positive emotion as an example, its analysis precision and recall are 0.95 and 0.89, respectively. Because the proposed model combines CNN and BiLSTM, it not only uses CNN to extract the deep features of the text, but also obtains the context information of the text based on BiLSTM, which is more conducive to the analysis of text emotion. Reference [13] implemented text emotion analysis based on dictionary model. Compared with deep learning algorithm, its analysis accuracy is low, lower than 0.80. Similarly, reference [14] used machine learning algorithm for text classification, which used automatic filtering to preprocess the text, but lacked comprehensive analysis and extraction of text features. Therefore, its accuracy is only 0.78. Reference [17] proposed a text emotion analysis method based on CNN, which can extract higher level sequences of text features through convolution layer, and the analysis accuracy is improved to 0.86. However, due to the lack of consideration of text context information, the overall accuracy of emotion analysis is reduced by 8.51% compared with the proposed model. Therefore, the effectiveness of the proposed CNN-BiLSTM model can be demonstrated.

4. Conclusion

The massive growth of data increases the complexity of the network environment. As a research hotspot of natural language processing, text emotion analysis has great research significance in public opinion analysis, user portrait, and recommendation system. Therefore, at this stage, with the continuous progress of artificial intelligence, it has important research value to realize effective emotion analysis through emotion calculation. Aiming at the problems of single word quantization and poor analysis accuracy in most of the existing deep learning algorithms in text processing, an emotion analysis model of microblog comment text based on deep learning is proposed. Because the output of LSTM hidden layer only depends on the output of the previous time, resulting in inaccurate analysis results, the proposed model combines BiLSTM and CNN to construct CNN-BiLSTM model. BiLSTM can obtain previous and subsequent information, and CNN can deeply mine text features. Thus, the accuracy of text emotion analysis is improved. Based on the crawled microblog data set, the experimental analysis of the proposed model shows that its overall accuracy is 0.94, which is better than other comparison models, and has a certain practical application value.

The proposed CNN-BiLSTM model includes a BiLSTM layer that can take into account the previous and subsequent information. Because the Gated Recurrent Unit (GRU) simplifies the gate structure in the LSTM, in the next research, it will consider using a bidirectional GRU network to replace the BiLSTM to improve the analysis efficiency of the model.

Data Availability

The data included in this paper are available without any restriction.

Disclosure

A preprint of this article has previously been published [26].

Conflicts of Interest

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

Acknowledgments

This work was supported by the Central Government Directs Special Projects for the development of local science and technology (no. ZY20B05), Heilongjiang Agricultural Reclamation Administration's Projects (no. HKKY190201-02), Heilongjiang Innovative Talent Project (no. CXRC2017014), and the University's Talent Research Program (no. XDB201813).

References

[1] A. Braoudaki, E. Kanellou, and C. Kozanitis, "Hybrid data driven and rule based sentiment analysis on Greek text," *Procedia Computer Science*, vol. 178, no. 3, pp. 234–243, 2020.

[2] S. K. Hong and S. K. Chong, "Analysis of IT service quality elements using text sentiment analysis," *Journal of Society of Korea Industrial and Systems Engineering*, vol. 43, no. 4, pp. 33–40, 2020.

[3] W. Li, B. Jin, and Y. Quan, "Review of research on text sentiment analysis based on deep learning," *Open Access Library Journal*, vol. 07, no. 3, pp. 1–8, 2020.

[4] N. Chintalapudi, G. Battineni, M. D. Canio, G. G. Sagaro, and F. Amenta, "Text mining with sentiment analysis on seafarers' medical documents," *International Journal of Information Management*, vol. 1, no. 1, pp. 1–9, 2021.

[5] M. Anandarajan, C. Hill, and T. Nolan, "[Advances in analytics and data science] practical text analytics volume 2 (maximizing the value of text data) || sentiment analysis of movie reviews," *Using R*, vol. 10, no. 13, pp. 193–220, 2019.

[6] W. Liao, B. Zeng, X. Yin, and P. Wei, "An improved aspect-category sentiment analysis model for text sentiment analysis based on RoBERTa," *Applied Intelligence*, vol. 6, no. 4, pp. 1–12, 2020.

[7] S. K. Singh and M. Sachan, "SentiVerb system: classification of social media text using sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 22, Article ID 32109, 2019.

[8] H. He and B. Chen, "A smart sentiment analysis system in word, sentence and text level," *Advanced Engineering Forum*, vol. 1, no. 2, pp. 338–342, 2011.

[9] H. J. Pang, "Text sentiment analysis-oriented commodity review detection," *Journal of Computer Applications*, vol. 32, no. 7, pp. 2038–2037, 2012.

[10] Y. Kirelli and S. Arslankaya, "Sentiment analysis of shared tweets on global warming on twitter with data mining methods: a case study on Turkish language," *Computational Intelligence and Neuroscience*, vol. 2020, no. 7, Article ID 1904172, 2020.

[11] M. Li, L. Xie, A. Zhang, and F. Ren, "Reinforcement emotion-cognition system: a teaching words task," *Computational Intelligence and Neuroscience*, vol. 2019, no. 4, 13 pages, Article ID 8904389, 2019.

[12] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 1–19, 2021.

[13] P. Ajitha, A. Sivasangari, R. I. Rajkumar, and S. Poonguzhali, "Design of text sentiment analysis tool using feature extraction based on fusing machine learning algorithms," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 1, pp. 1–9, 2020.

[14] M. Alqmase, H. Al-Muhtaseb, and H. Rabaan, "Sports-fanaticism formalism for sentiment analysis in Arabic text," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 1–24, 2021.

[15] A. Razzaq, M. Asim, Z. Ali, S. Qadri, and I. Mumtaz, "Text sentiment analysis using frequency-based vigorous features," *China Communications*, vol. 16, no. 12, pp. 145–153, 2019.

[16] F. Huang, X. Zhang, Z. Zhao, J. Xu, and Z. Li, "Image-text sentiment analysis via deep multimodal attentive fusion," *Knowledge-Based Systems*, vol. 167, no. 3, pp. 26–37, 2019.

[17] F. Xu, X. Zhang, Z. Xin, and A. Yang, "Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 697–709, 2019.

[18] X. Lu and H. Zhang, "Sentiment analysis method of network text based on improved AT-BiGRU model," *Scientific Programming*, vol. 2, no. 12, pp. 1–11, 2021.

[19] B. Liu, "Text sentiment analysis based on CBOW model and deep learning in big data environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 451–458, 2020.

[20] G. Xu, Z. Yu, H. Yao, F. Li, Y. Meng, and X. Wu, "Chinese text sentiment analysis based on extended sentiment dictionary," *IEEE Access*, vol. 7, no. 7, Article ID 43749, 2019.

[21] S. Al-Azani and E. El-Alfy, "Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text," *Procedia Computer Science*, vol. 109, no. 5, pp. 359–366, 2017.

[22] Y. Cheng, L. Yao, G. Xiang, G. Zhang, T. Tang, and L. Zhong, "Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism," *IEEE Access*, vol. 8, no. 2, Article ID 134964, 2020.

[23] Y. Kim, M. Kang, and S. R. Jeong, "Text mining and sentiment analysis for predicting box office success," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 8, pp. 4090–4102, 2018.

RETRACTED

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9005278, 11 pages
https://doi.org/10.1155/2022/9005278

Research Article

AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease

T. R. Mahesh,[1] **V. Dhilip Kumar**,[2] **V. Vinoth Kumar**,[1] **Junaid Asghar**,[3] **Oana Geman**,[4] **G. Arulkumaran**,[5] **and N. Arun**[1]

[1]*Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bangalore, India*
[2]*Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India*
[3]*Faculty of Pharmacy, Gomal University, Dera Ismail Khan 29050, Khyber Pakhtunkhwa, Pakistan*
[4]*Stefan Cel Mare University of Suceava, Suceava, Romania*
[5]*Department of Electrical and Computer Engineering, Bule Hora University, Bule Hora, Ethiopia*

Correspondence should be addressed to G. Arulkumaran; erarulkumaran@gmail.com

Received 24 February 2022; Revised 24 March 2022; Accepted 30 March 2022; Published 18 April 2022

Academic Editor: Muhammad Ahmad

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

As a result of technology improvements, various features have been collected for heart disease diagnosis. Large data sets have several drawbacks, including limited storage capacity and long access and processing times. For medical therapy, early diagnosis of heart problems is crucial. Disease of heart is a devastating human disease that is quickly increasing in developed and also developing countries, resulting in death. In this type of disease, the heart normally fails to provide enough blood to different body parts in order to allow them to perform their regular functions. Early, as well as, proper diagnosis of this condition is very critical for averting further damage and also to save patients' lives. In this work, machine learning (ML) is utilized to find out whether a person has cardiac disease or not. Both the types of ensemble classifiers, namely, homogeneous as well as heterogeneous classifiers (formed by combining two separate classifiers), have been implemented in this work. The data mining preprocessing using Synthetic Minority Oversampling Technique (SMOTE) has been employed to cope with the imbalance problem of the class as well as noise. The proposed work has two steps. SMOTE is used in the initial phase to reduce the impact of data imbalance and the second phase is classifying data using Naive Bayes (NB), decision tree (DT) algorithms, and their ensembles. The experimental results demonstrate that the AdaBoost-Random Forest classifier provides 95.47% accuracy in the early detection of heart disease.

1. Introduction

Heart disease is mainly observed as the world's most dangerous and life-threatening chronic disease. During heart illness, the heart generally fails to deliver enough blood to different body regions so as to allow them to operate normally. The narrowing and occlusion of coronary arteries can cause heart failure. Heart disease is one among the leading reasons for death nowadays across the globe [1]. This leads to crucial requirement of monitoring the functioning organs in the human body and a critical aspect in monitoring health records of cardiovascular system. The coronary arteries control the entire circulation of blood to the heart. According to the latest survey, United States is one of the severely affected countries with relatively high ratio of heart disease observed in patients. The symptoms like breathing problem, physical body weakness, exhaustion, and swollen feet among various other symptoms are the most typical markers of heart disease [2]. Most of the cardiovascular diseases affecting people across the world are usually fatal. So, to overcome this problem, development of new technique may aid in detection of heart diseases in early stages as there is huge growth in the technology. Also, before causing substantial damage to avoid advantageous problems in terms of time, cost, and saving human lives machine learning techniques are used to focus on monitoring the heart diseases. Machine learning involves emerging techniques in manipulating and extracting features or relevant data information in possible way [3]. Machine learning is one of the complex fields and also has huge scope in various applications which is expanding all the time. Machine learning techniques consist of supervised learning, unsupervised learning, and also ensemble learning classifiers, which are mainly used to forecast the heart diseases in early stages with increase in accuracy results [4].

In the past years, academicians and researchers attempted to create and implement many intelligent programs by applying predefined procedures, which are similar to regular existing program works [2]. But, still there is a lag in monitoring many observations and instances in timely manner to overcome many societal challenges. Nowadays, very challenging tasks include photo tagging, identification of web-based ranking, identification of spam, or no spam Emails. To overcome these tasks or objectives, one of the options includes development of a program generating relevant rules to evaluate the data samples. It is also called training set, and one of the common emerging fields used for this is machine learning methods. Since 2010–2015, many intelligence software-based machine learning methods are applied including recognition systems on patient images to improve the accuracy results from 72% to 95% [5].

Most of the machine learning applications are evolving in present days and affecting every aspect in our daily lives. Machine learning is applicable in many emerging areas like healthcare monitoring systems, pattern recognition and feature extraction, text and speech recognition, education systems, military and defense applications, fraud detection, etc. Artificial intelligence takes the main lead in the development of ML technology systems. ML technology also simulates human learning systems from the input dataset or information. Many machine learning algorithms from firms such as Facebook, Amazon, or Flipkart are boosting the business trends in developing various brands [6]. With the help of past data or information, machine learning tries to discover new patterns in applying algorithms to achieve feasible outcome results. Also adding value to the business trends or organizations mainly focuses on monitoring future situations and outcome [7].

2. Related Work

A lot of research work is carried out using machine learning methods in achieving more accurate results and predicting outputs based on input dataset [8]. Machine learning plays a very important role in view of new trends and new techniques based on customer behavior or various input patterns, in the development of new products and new brands [9]. Enterprises can understand the customer needs at deeper level to overcome their needs using machine learning algorithms depending, for various applications, on their outcomes [10]. Machine learning also increases the importance in business operations and artificial intelligence is becoming practically high using today's ML models.

One of the new strategies for detecting cardiac diseases, mainly based on Co-Active Neuro-Fuzzy Interference Systems (CANFIS), is applied in one of the research work [11]. Most of the research study is based on regularity in detection of heart diseases based on their strategies as well as on their difficulties. Classifier strategies for the detection of heart diseases are demonstrated using machine learning algorithm, Naive Bayes classifier model. Most of the survey is carried out on various applications, in many research papers, by using data mining algorithms for prediction of heart diseases [12]. But traditional invasive-based approach is carried out using machine learning algorithms. The classifier models for diagnosing heart diseases are based on medical history of patients, patient test results, or scan results so that researchers or doctors can research on connected symptoms [13].

Alternatively, one more disadvantage is that the dye used is harmful as it affects kidneys, as it increases creatinine, including a high cost, a different kind of adverse effects, and a very good level of technological knowledge [14]. The traditional method is comparatively costly and also computationally intensive method for disease diagnosis which takes time to assess [15]. Researchers have tried to create various noninvasive smart healthcare systems which are based on predictive ML techniques, namely, SVM, K-NN, Naive Bayes (NB), and, also, decision tree (DT), among others, to overcome the challenges in conventional invasive-based methods for the identification of heart disease [16].

In the medical field, one of the most used classifiers is the decision tree. In this work [17] SEER medical datasets were used to predict the disease survivorship using classification and regression trees (CART).

In this work [18], use of neural networks was introduced to diagnose and forecast heart disease as well as blood pressure. A Deep Neural Network was built using the given disease attributes to generate an output that was accomplished by the output perceptron and almost included 120 hidden layers, which is the most basic and relevant method for ensuring an accurate result of having heart disease if the model is using the test dataset [19]. The use of a supervised algorithm for cardiac disease diagnostics is being recommended [17]. When the attributes of data are associated, the random forest approach has a tendency to favor the smaller group [20]. This is why, in order to alleviate the challenge of imbalanced data and limit the probability of bias against minorities in the dataset, the SMOTE method is being used. In this study [21], a combination of SMOTE and Artificial Neural Network (ANN) has been used to diagnose ovarian cancer using a publicly available dataset of ovarian cancer. The research demonstrates that, by using the preprocessing methodology of SMOTE to decrease the impact of data imbalance, we can improve the performance and efficiency of neural networks in cancer classification. On large datasets, most single classifier algorithms have the drawbacks of being computationally expensive and difficult. For large datasets, in particular, classification approaches do not give consistent and reliable results, making some individual classifier systems wasteful and unreliable [22]. For example, the DT approach is particularly good at managing intervariable interactions, but it struggles with linear relationships between variables [23].

In recent years, ensemble classifiers have become a popular strategy in machine learning and pattern recognition. In a nutshell, it is a method for combining the findings of many classifiers. The ensemble method's main goal is to improve classification efficiency by weighing multiple independent classifiers and thereby combining them into a single or an individual classifier that outperforms each one individually [22, 24, 25].

3. Exploratory Knowledge

One of the most well-known areas of medical research is the research for heart disease. Early identification and accurate projections of heart diseases have a significant impact on therapy and reduce patient mortality rates. The sections that follow provide brief descriptions of the algorithms used to detect heart disease in this study.

3.1. Decision Tree (DT) Classifier. A decision tree is a supervised ML algorithm that makes decisions based on a set of rules, very similar to how normally people do. A ML classification method is designed to make judgments, in one sense. Classification and regression problems can both be solved with this classifier [26].

There are different notions that define the model. They are given below.

(i) Entropy: Entropy is a measurement of a system's unpredictability or disorder. In the year 1850, a German physicist named Rudolf Clausius proposed this hypothesis. It is computed as shown in

$$\text{Entropy} = -\sum p(X)\log\ p(X), \tag{1}$$

where p(X) is a fraction of examples in a given class.

(ii) Gini Index: It is also called the Gini coefficient, which is a measure of income distribution in a population. Corrado Gini, an Italian statistician, created it in 1912. The Gini impurity is computed using

$$\text{Gini Inpurity} = 1 - \sum_{i=1}^{C} (p_i)^2. \tag{2}$$

(iii) Information Gain: The reduction in entropy achieved by changing a dataset is known as information gain, and it is frequently utilized in the training of decision trees. The entropy of a dataset before and after a transformation is used to calculate information gain. It is computed using

$$IG\left(D_p, f\right) = I\left(D_p\right) - \frac{N_{\text{left}}}{N} I\left(D_{\text{left}}\right) \text{-} \frac{N_{\text{right}}}{N} I\left(D_{\text{right}}\right), \tag{3}$$

where f is feature split on $D_p$which is parent dataset; D_{left} is left child node dataset;D_{right} is right child node dataset; I is impurity criterion; N is total number of samples; N_{left} is samples number of left child node; N_{right} is samples number of right child node.

3.2. The CART Algorithm. The CART algorithm was first introduced by Breiman et al. [27]. Hunt's algorithm is used to create the CART. To build a DT, it can process categorical as well as continuous attributes. It also accounts for missing data and constructs the DT by making use of Gini Index as an attribute selection criterion. CART divides the given datasets (training set) into binary segments and builds binary trees as a result. The Gini Index is not employed in the ID3 and C4.5 probabilistic assumptions. In order to increase accuracy of classification, CART algorithm increases the accuracy by making use of cost-complexity pruning for removing unpredictable branches from the DT.

3.3. Alternating Decision Tree (AltDTree). AltDTree is a classification ML method. It is related to boosting and generalizes decision trees. An AltDTree is made up of a series of decision nodes that indicates a predicate condition and prediction nodes that hold a single number [28]. Classic DTs, Voted DTs, and Voted Decision Stumps are all generalized into AltDTree. It allows any boosting implementation to extract the AltDTree model from the data as a learning method. In the context of the decision tree, AltDTree is an appealing extension of boosting. It enables the use of various boosting strategies to create an AltDTree model with unique properties that can handle a wide range of applications.

3.4. Random Forest (RF) Classifier. RF works by using the training data to create several decision trees. In the case of classification, every tree suggests output as a class; also the class with greatest number of outputs is selected as the final outcome [29]. In order to build, number of trees must be specified. RF is such a technique for aggregating or even bagging bootstrap data. This method is used to reduce an important parameter called variance in the outcomes.

3.5. Reduced Error Pruning Tree (RedEPTree). Top-down induction of decision trees has been observed to be hampered by the pruning phase's poor performance. It is known, for example, that the size of the resulting tree rises linearly with the sample size, despite the fact that the tree's accuracy does not improve. Errors are reduced. The RedEPTree technique is based on the notion of calculating information gain using entropy and backfitting to minimize variance-induced error [30].

3.6. Naive Bayes (NB) Classifier. There are two steps of classified data in the Naive Bayesian approach [31]. The first stage involves evaluating the parameters of a probability distribution using the training input data. In the second stage, the test dataset is categorized based on the greatest posterior probability. The NB classifier's pseudocode is shown below.

3.7. AdaBoost. AdaBoost makes it possible to merge various "weak classifiers" into a single classifier which is called "strong classifier." Decision trees with one level, or decision trees with only one split, are the most popular algorithm used with AdaBoost. Decision Stump is another name for these trees [32]. This approach creates a model by assigning equal weights to all of the data points. It then gives points that are incorrectly categorized with a higher weight. In the next model, all points with greater weights are given more importance. It will continue to train models till a lower error is received [33].

The weight of the training set is used to start the AdaBoost algorithm. Let us consider training set $(x_1, y), \ldots (x_n, y_n)$, in which each x_i is in instance space X and each label y_i is in collection of labels Y, that is very much similar to the collection of $\{-1, +1\}$. Weight on training instance I on the round t is assigned as $D_{\mathrm{It}}(i)$. At the start, the same weight is used $(D_{\mathrm{It}}(i)) = 1/\mathrm{M}$, $i = 1, \ldots, \mathrm{M})$, where It is the iteration number. Then, weight of the misclassified case from the base learning algorithm is then increased in each round. The AdaBoost algorithm's pseudocode is shown below.

And

$$\alpha_{\mathrm{It}} = \frac{1}{2} \mathrm{In} \left[\frac{P_{+1} - P_{-1}}{P_{-1} + P_{-1}} \right]. \tag{4}$$

C_{It} is the normalization constant, α_{It} is used to allow the outcome to be generalized and to solve the problem of overfitting and noise sensitive situations [33]. The real value of $\alpha_{It} h_{It}\ (x)$ is built using a class probability estimate (P).

4. Proposed Methodology

The proposed approach contains two phases in this section. SMOTE is used in the initial phase to lessen the impact of data imbalance. Then, the second phase entails classification using Naive Bayes and DT methods (AltDTree, CART, RedEPTree, and RF) [33]. After that, AdaBoost Ensembles of the aforementioned algorithms are constructed and their performance is evaluated. Then, heterogeneous classifiers that are formed by combining two different individual classifiers are evaluated against different performance metrics to figure out the best model. Figure 1 depicts the flow of the suggested technique.

4.1. Dataset. The UCI repository provided the Heart Disease dataset. This dataset comprises 13 medical variables for 304 patients, which helps to determine whether the patient is in the danger of developing heart disease or not, as well as categorize patients who are at risk and those who are not. The pattern that leads to the discovery of patients at risk for heart disease is retrieved from this dataset. There are two aspects to these records: training and testing. Each row corresponds to a single record in this dataset, which has 303 rows and 14 columns. Table 1 lists all of the qualities and the heatmap is depicted in Figure 2.

4.2. Data Preprocessing. Most classification algorithms aim to gather pure samples to learn and make the borderline of each class as definitive as possible in order to perform better prediction. Synthetic instances that are far from the borderline are easier to categorize than those that are near to the borderline, which present a significant learning difficulty for the majority of classifiers. The authors in [32] describe an advanced strategy (A-SMOTE) for preprocessing imbalanced training sets based on these findings. It aims to clearly characterize the borderline and create pure synthetic samples from SMOTE generalization. This approach is divided into two parts, as follows.

Step 1. The SMOTE technique is used to create a synthetic instance using

$$N = 2 * (r - z) + z, \tag{5}$$

where r denotes majority class samples, z denotes minority class samples number, and N is the initial synthetic instance number (which is newly generated).

The synthetic instances generated by SMOTE can be approved or rejected based on two criteria, which correspond to the first stage: For example, consider $\widehat{x} = \{\widehat{x}_1, \widehat{x}_2, \widehat{x}_3, \ldots . \widehat{x}_N\}$ which is the collection of new synthetic instances, and $\widehat{x}_i{}^{(j)}$ is the jth attribute value of $\widehat{x}_i$, $j \in\in [1, M]$.Let$S_m = \{S_{m1}, S_{m2}, \ldots S_{mz}\}$ and $S_\alpha = \{S_{\alpha 1}, S_{\alpha 1}, S_{\alpha 1}, \ldots . S_{\alpha r}\}$ be the set of the minority samples as well as majority samples [32]. In order to make the rejection or acceptance decision, distance is computed between $\widehat{x}_i$ and S_{mk}, D D_{minority} $(\widehat{x}_i, S_{mk})$ and the distance between $\widehat{x}_i$ and $S_{\alpha l}$, D D_{majority} $(\widehat{x}_i, S_{\alpha l})$. For I from N steps, we calculate the distances as stated below, using equations (6) and (7).

$$DD_{\mathrm{minority}}\left(\widehat{x}_i, S_{mk}\right) = \sum_{j=1}^{M} \sqrt{\left(\widehat{x}_i{}^{(j)} - \widehat{S}_{mk}{}^{(j)}\right)^2}, \ k \in [1, z], \tag{6}$$

$$DD_{\mathrm{minority}}\left(\widehat{x}_i, S_{al}\right) = \sum_{j=1}^{M} \sqrt{\left(\widehat{x}_i{}^{(j)} - \widehat{S}_{al}{}^{(j)}\right)^2}, \ l \in [1, r]. \tag{7}$$

As per (6) and (7), we compute arrays A_{minority} and A_{majority} using (8) and (9).

$$A_{minority} = \left(DD_{\min ority}\left(\widehat{x}_i, S_{m1}\right), \ldots DD_{\min ority}\left(\widehat{x}_i, S_{mz}\right)\right), \tag{8}$$

$$A_{majority} = \left(DD_{majority}\left(\widehat{x}_i, S_{a1}\right), \ldots DD_{majority}\left(\widehat{x}_i, S_{ar}\right)\right). \tag{9}$$

Then we choose the minimum value out of A_{minority}, $\min\left(A_{\mathrm{minority}}\right)$and the minimum value out of A_{majority}, $\min\left(A_{\mathrm{majority}}\right)$. If *min* $\left(A_{\mathrm{minority}}\right)$ is lesser than $\min\left(A_{\mathrm{majority}}\right)$, the new samples are accepted else, rejected.

$$\min\left(A_{\mathrm{majority}}\right) < \min\left(A_{\mathrm{majority}}\right) \ \text{(Accepted)}.$$

```
Input required: TDS: Training Dataset TDS = $u_i\,(1, 2, 3, 4, \ldots\ldots n)$ Output expected: Class Labels YES and NO
Step_1: The Given Dataset TDS, consists of symptoms pertaining to different classes, say YES and NO
Step_2: Calculate prior-probability of "YES" class = No of attributes of class YES/Total no of attributes Compute prior-probability of
    "NO" class = No of attributes of class NO/Total no of attributes
Step_3: Compute $enum_i$, total no. of attributes that are frequent for each class
    $num_{\text{yes}}$ = Sum of frequent attributes of class YES
    $num_{\text{no}}$ = Sum of frequent attributes of class NO
Step_4: Compute the conditional probability
    $P_{(\text{attribute1/class YES})} = \text{attributeCount}/n_i\,(\text{YES})$
    $P_{(\text{attribute1/class NO})} = \text{attributeCount}/n_i\,(NO)$
    $P_{(\text{attribute2/class NO})} = \text{attributeCount}/n_i\,(\text{YES})$
    $P_{(\text{attribute2/class NO})} = \text{attributeCount}/n_i\,(\text{NO})$
      . . .
    $P_{(\text{attribute2/class NO})} = \text{attributeCount}/n_i\,(\text{NO})$
Step_5: Classify a new record of attributes of a patient based on the probability P (NEW/feature).
    Compute $P_{(\text{YES/attribute})} = P_{(\text{YES})} * P_{(\text{attribute1/class YES})} * P_{(\text{attribute1/class YES})} \cdots P_{(\text{attributen/class YES})}$
    Compute $P_{\text{NO/attribute}} = P_{(\text{NO})} * P_{(\text{attribute1/class NO})} * P_{(\text{attribute1/class NO})} \cdots P_{(\text{attributen/class NO})}$
Step_6: Assign the new record of patient to either class YES or NO which has higher probability.
```

Algorithm 1: Pseudocode of NB classifier.

```
    Input required: TDS: Training Dataset TDS = $x_i\,(1, 2, 3, 4, \ldots\ldots n)$ labels $y_i \in$ Y
    It: Iteration Number
    Steps a to h
(a) Assign TDS sample $(x_1, y_1), \ldots (x_n, y_n)$; $x_i \in X$, $y_i \in \{-1, +1\}$
(b) Initialize weights of $D_{\text{It}}(i) = 1/\text{M}$, $i = 1, \ldots,$ M
(c) for It = 1, ... , T
(d) Train the learner that is weak using distribution $D_{\text{It}}$
(e) Get hypothesis of weak $h_{\text{It}}: X \longrightarrow \{-1, +1\}$ along with its error = $\varepsilon_{\text{It}} = \sum_{h_{\text{It}(x)} \neq y_i}^{n} D_{\text{It}}(i)$
(f) Update distribution $D_{\text{It}}$: $D_{\text{It}+1}(i) = D_{\text{It}}(i) exp(-\alpha_{\text{It}} y_{\text{It}} T_{\text{It}}(x_{\text{It}}))/C_{It}$
(g) Next It that, It + 1
(h) Final hypothesis Outputs: $H_{(x)} = \text{sign}[\sum_{\text{It}=1}^{T} \alpha_{\text{It}} h_{\text{It}}]\ (x)]$
```

Algorithm 2: Pseudocode of AdaBoost classifier.

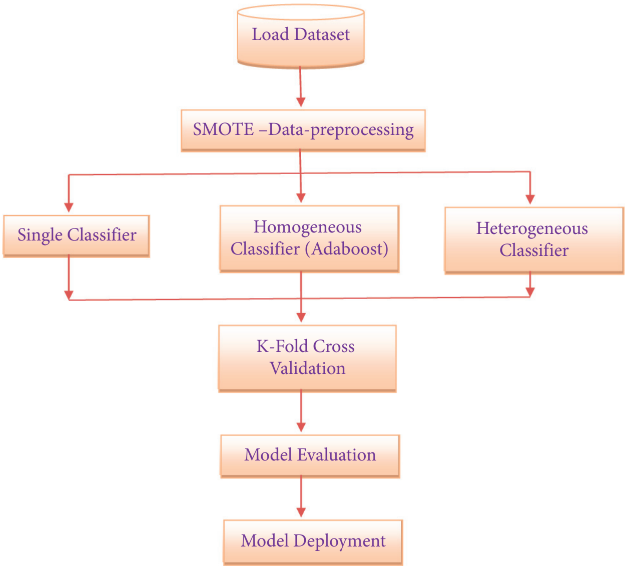


Figure 1: Proposed flow diagram.

Table 1: Attributes of the dataset.

Sl. No.	Features	Description	Values
1	Age	Age in years	Continuous
2	Sex	Gender of patient	Male/female
3	CP	Chest pain	Four types
4	Trestbps	Resting blood pressure	Continuous
5	Chol	Serum cholesterol	Continuous
6	FBS	Fasting blood sugar	<, or >120 mg/dl
7	Restecg	Resting electrocardiograph	Five values
8	Thalach	Maximum heart rate achieved	Continuous
9	Exang	Exercise induced angina	Yes/no
10	Oldpeak	ST depression when working out compared to the amount of rest taken	Continuous
11	Slope	Slope of peak exercise ST segment	Up/flat/down
12	Ca	Gives number of major vessels colored by fluoroscopy	0–3
13	Thal	Defect type	Reversible/fixed/normal
14	Num (disorder)	Heart disease	Not present ("NO")/present in the four major types ("YES")

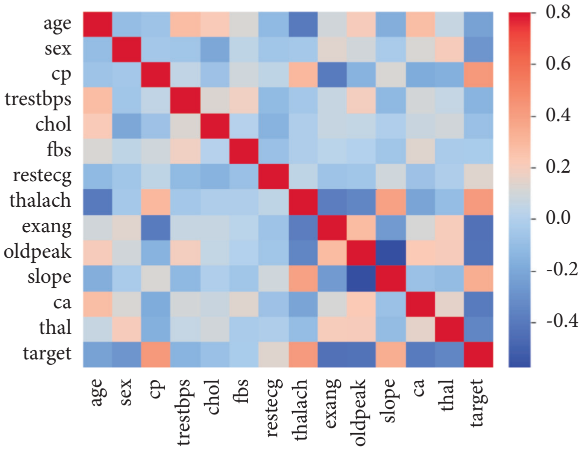


Figure 2: Heatmap depiction of the dataset.

$$\min(A_{\text{minority}}) \geq \min(A_{\text{majority}}) \text{ (Rejected).}$$

Step 2. Then, using the accepted synthetic instances, the following steps are taken to remove the noise.

Suppose $\widehat{S} = \{\widehat{S}_1, \widehat{S}_2, \widehat{S}_3, \ldots . . \widehat{S}_n\}$ is a new synthetic minority received by Step 1. We then compute the distance between $\widehat{S}_i$ with each original minority S_m, $\text{Min}_{\text{Rap}}(\widehat{S}_i, \widehat{S}_m)$, defined using

$$S_m, \text{Min}_{\text{Rap}}\left(\widehat{S}_i \widehat{S}_m\right) = \sum_{k=1}^{z} \sum_{j=1}^{M} \sqrt{\left(\widehat{S}_i^{\;(j)} - S_{mk}^{\;(j)}\right)^2}, \qquad (10)$$

where

S_m, $\text{Min}_{\text{Rap}}(\widehat{S}_i.\widehat{S}_m)$ samples rapprochement including all minority and as per (10), L is obtained as follows:

$$L = \sum_{i=1}^{n}\left(\text{Min}_{\text{Rap}}\left(\widehat{S}_i, S_m\right)\right). \qquad (11)$$

Step 3. Compute the distance between $\widehat{S}_i$, and each original majority S_a, $Maj_{\text{Rap}}(\widehat{S}_i S_a)$, described using

$$Maj_{\text{Rap}}\left(\widehat{S}_i S_a\right) = \sum_{i=1}^{r} \sum_{j=1}^{M} \sqrt{\left(\widehat{S}_i^{\;(j)} - S_{al}^{\;(j)}\right)^2}. \qquad (12)$$

$Maj_{\text{Rap}}(\widehat{S}_i, S_a) \longrightarrow$ samples rapprochement including all majority and as per equation (13) H is obtained as follows:

$$H = \sum_{i=1}^{n}\left(Maj_{\text{Rap}}\left(\widehat{S}_i, S_a\right)\right). \qquad (13)$$

Then, we remove half of synthetic samples which have most likely less distance between $\widehat{S}_i$and S_a to obtain the data, that is, of high purity.

5. Performance Evaluations

The different ML algorithms, namely, Naive Bayes, AltD-Tree, RedEPTree, CART, and RF, are applied on the dataset as individual classifiers. Their performance is compared in terms of several metrics as described in the next section.

5.1. Performance Metrics. If the dataset is not balanced, accuracy may not be a good measure [34]. The number of accurately classified examples divided by total number of data instances is referred to as accuracy. The accuracy is computed using

$$\text{Accuracy} = \frac{TNs + TPs}{TNs + TPs + FPs + FNs}. \tag{14}$$

Precision is one of the performance metrics that is going to measure how many correct positive forecasts have been done. So, precision estimates the accuracy of the minority class; then, the ratio of accurately predicted positive instances divided by the total number of positive examples predicted, is used to compute it using

$$\text{Precision} = \frac{TPs}{TPs + FPs}. \tag{15}$$

A good classifier should have a precision of 100% (high); only when both numerator and denominator are identical, i.e., TP = TP + FP, can precision become 100% [33].

Recall is a metric that measures how many correct positive predictions were produced out of all possible positive predictions. Unlike precision, which only considers the right positive predictions out of all positive predictions, recall considers the positive predictions that were missed. In this approach, recall provides some indication of the positive class' coverage. The recall is computed using

$$\text{recall} = \frac{TPs}{TPs + FNs}. \tag{16}$$

We want both accuracy and recall to be of the value one in a good classifier, which also means FP and FN should be zero. As a result, we require a statistic that considers both precision and recall. The F1-score is a measure that takes precision and recall into account and is defined as follows:

$$F1\ \text{Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{17}$$

To compute error rates in forecasted value, let P^N denote a collection of data having the form (t_1, r_1) , (t_2, r_2),. . . (t_p, r_p)such that t_i denotes n-dimensional tuples of test with respective values of r_i for a given response r and denotes count of tuples in P^N.

In all test instances, the mean-absolute-error (MAE) is the mean of the difference among the projected and guanine value. It is the standard deviation of the prediction error calculated using

$$MAE = \sum_{i=1}^{p} \left| r_i - {r_i}^T \right|. \tag{18}$$

The root mean squared error (RMSE) is a well-known approach for calculating numeric prediction success. The mean of the squared discrepancies among every value is computed and its matching true value is used to calculate this value using

$$RMSE = \sqrt{\frac{\sum_{r=1}^{p} \left(r_i - {r^T}_i\right)^2}{p}}. \tag{19}$$

The total absolute mistake is made relative to what the error would have been if the prediction had just been the average of the actual numbers known as Relative Absolute Error (RAE). It is computed using

$$RAE = \frac{\sum_{r=1}^{p} \left(r_i - {r^T}_i\right)^2}{\sum_{r=1}^{p} \left(r_i - \overline{r}_i\right)^2}. \tag{20}$$

The total squared error made is compared to what the error would have been if the prediction had been the average of the absolute value, known as relative squared error (RRSE). It is computed using

$$RRSE = \sqrt{\frac{\sum_{r=1}^{p} \left(r_i - {r^T}_i\right)^2}{\sum_{r=1}^{p} \left(r_i - \overline{r}_i\right)^2}}. \tag{21}$$

Table 2 depicts that Random Forest is the best model as it takes only 2.27 seconds for model building (TTBM: Time to Build Model), while the AltDTree has taken 60.18 seconds for model building.

Figure 3 shows the accuracy forecast for individual classifiers. Among all the aforementioned classifiers being used in the current research work, AltDTree provides the best accuracy of 93.56%. Random Forest provides 92.45% accuracy and NB classifier prediction is the lowest with 78.67% accuracy.

Figure 4 depicts the rates of errors obtained from the individual classifiers. AltDTree MAE rate is 0.28 and RMSE rate value is 0.41. This demonstrates that there is low error recorded during the prediction procedures. However, NB has a higher error rate, i.e., 0.60 MAE and 0.83 RMSE, respectively.

Table 3 demonstrates that AdaBoost-RF is the best model, as it has taken only 10.34 seconds to build the model. But the AdaBoost-CART is the worst model as it takes 295.45 seconds to build the model. Also, AdaBoost-RF has highest F1-value of 0.98 and AdaBoost-NB has the lowest F1-value of 0.81.

From Figure 5, AdaBoost-RF predictions are better than any other mentioned classification algorithm with an accuracy of 95.47%. However, AdaBoost-AltDTree provides 93.56% prediction accuracy and stands second. The AdaBoost-NB provides the least prediction rate of 80.6%.

Figure 6 depicts the different error rates that were recorded. AdaBoost Ensemble classifiers provide the lowest error rate of 0.14 for MAE and 0.38 for RMSE. However, AdaBoost-NB has a higher error rate, i.e., 0.54 and 0.76 for MAE and RMSE, respectively, whose values are almost the same as that of NB individual classifier.

Table 4 depicts the results of ensemble classifiers which are heterogeneous in nature. RF-CART and RF-RedEPTree take 7.34 seconds and 7.89 seconds for building the model,

Table 2: Single classifier evaluation comparison.

Performance metrics	Naive Bayes	AltDTree	RF	RedEPTree	CART
TTBM (sec)	4.56	60.18	2.11	10.25	52.24
Accuracy (%)	78.6	93.56	92.45	79.23	78.67
MAE	0.60	0.28	0.27	026	0.27
RMSE	0.83	0.41	0.42	0.42	0.56
RAE	120	67.71	77.87	79.12	68.91
RRSE	127.41	95.33	82.92	97.89	98.34
F1-score	0.3	0.85	0.84	0.83	0.81

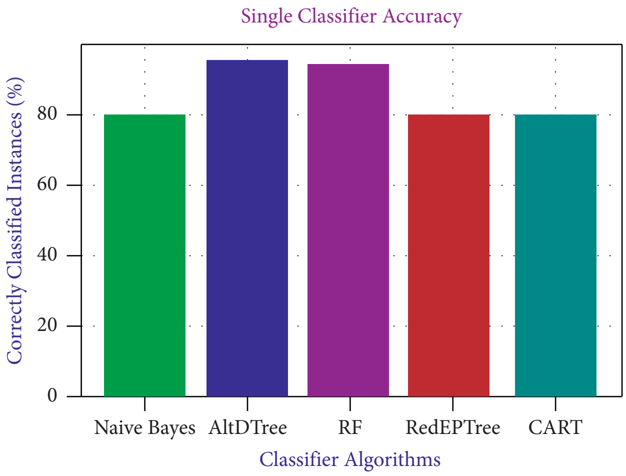


Figure 3: Accuracy prediction for single classifiers.

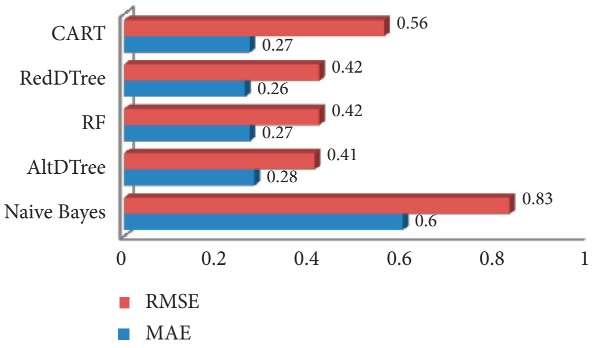


Figure 4: Error rates of individual classifier.

Table 3: AdaBoost classifier.

Performance metrics	AB-NB	AB-AltDTree	AB-RF	AB-RedEPTree	AB-CART
TTBM (sec)	18.32	30.01	10.34	64.35	295.45
Accuracy (%)	80.6	93.56	95.47	82.23	81.67
MAE	0.54	0.21	0.14	0.21	0.20
RMSE	0.76	0.43	0.38	0.41	0.41
RAE	129.79	57.78	35.87	45.19	41.61
RRSE	155.62	96.23	65.47	91.03	91.08
F1-score	0.81	0.94	0.98	0.83	0.87

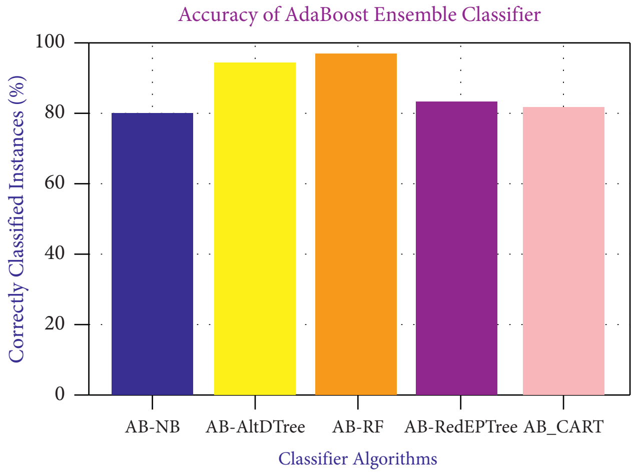


Figure 5: Accuracy of AdaBoost classifier.

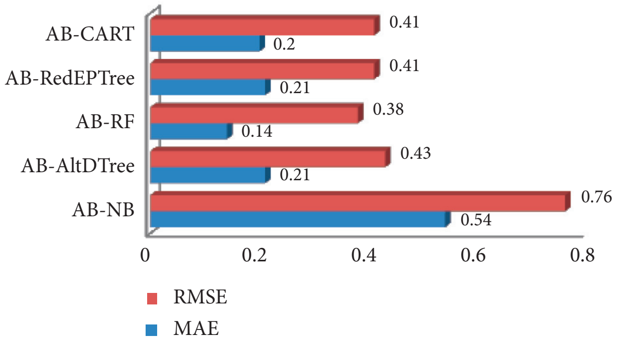


Figure 6: AdaBoost classifier error rate.

Table 4: Ensemble classifiers, heterogeneous.

Performance metrics	NB + AltDTree	NB + RF	AltDTree + RF	RF + RedEPTree	RF + CART	AltDTree + RedEPTree	AltDTree + CART
TTBM (sec)	30.03	32.05	398.12	7.89	7.34	357.77	598.02
Accuracy (%)	76.45	76.05	70.12	85.45	86.29	74.49	71.29
MAE	0.42	0.43	0.37	0.35	0.34	0.37	0.41
RMSE	0.42	0.39	0.49	0.36	0.36	0.37	0.42
RAE	99.23	92.23	80.12	71.01	70.89	73.23	89.23
RRSE	98.23	97.49	101.22	91.29	90.12	93.37	99.34
F1-score	0.74	0.75	0.68	0.84	0.85	0.73	0.69

respectively, which are very low. However, AltDTree-CART has taken 598.02 seconds being the worst time for building the model. So, it can be said that RF-CART has a higher F1-score of 0.85, and RF-RedEPTree is second with F1-score of 0.84. AltDTree-RF and AltDTree-CART have the worst F1-scores of 0.68 and 0.69, respectively.

From Figure 7, RF-CART provides the best accuracy of 86.29% in comparison to others, followed by RF-RedEPTree with 85.45% prediction accuracy. AltDTree-RF has the lowest accuracy value of 70.12%.

Figure 8 depicts error rates obtained by ensemble classifiers are heterogeneous in nature. RF-CART exhibits

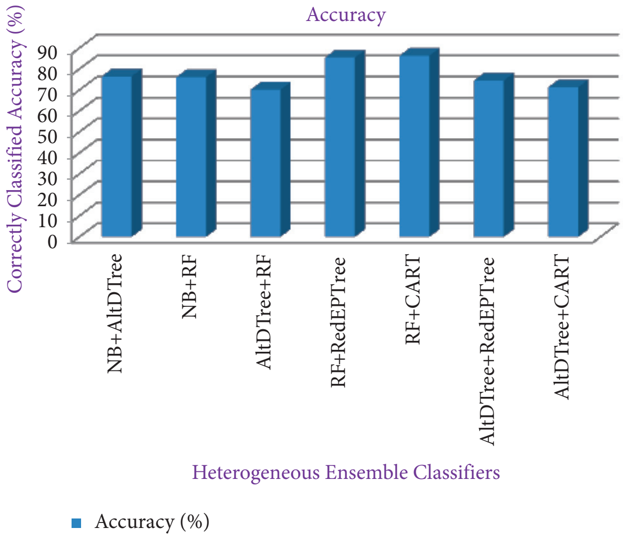


Figure 7: Accuracy of heterogeneous ensemble classifiers.

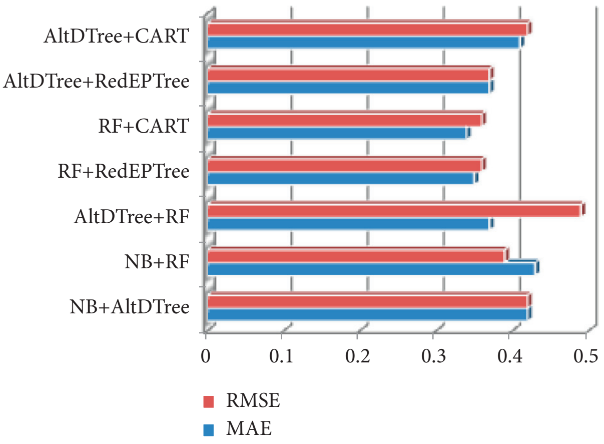


Figure 8: Error rates for heterogeneous ensemble classifiers.

the lowest error rate of 0.34 (MAE) and RMSE of 0.36. However, NB-RF has the highest MAE rate of 0.43 and AltDTree-RF has the highest RMSE rate of 0.49.

6. Conclusion

The AdaBoost Ensemble model for heart disease prediction has been proposed in this work, which is based on recognized feature patterns. In the diagnosis of cardiac disease, it can be compared with classic data mining methods. Ensemble classification approaches replace traditional methods of extracting meaningful information during the feature extraction step. The homogeneous classifiers and ensemble classifiers which are formed by combining multiple methods called heterogeneous classifiers were employed in this study. The data mining preprocessing technique using Synthetic Minority Oversampling Technique (SMOTE) is used to cope with the problem of class imbalance as well as noise present in the heart disease dataset. The best time to build the model for heterogeneous ensemble classifiers is 7.34 seconds for RF-CART and 7.89 seconds for RF-RedEPTree ensemble, according to the experimental results. NB-AltDTree has been observed to have taken the worst time of 598.02 seconds to build the model. With 86.29% prediction accuracy, RF-CART outperforms other classification algorithms, followed by RF-RedEPTree with 85.45% prediction accuracy. As per the results, AdaBoost-RF classifier exhibits 0.14 error rate for MAE which is the lowest and 0.38 for RMSE among the other AdaBoost Ensemble classifiers. In all the overall experiments, the performances of classifiers were compared, and the findings revealed that AdaBoost-RF is the best among other classifiers with 95.47% accuracy.

Data Availability

The [UCI repository] data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] World Health Organization, *Cardiovascular Diseases*, WHO, Geneva, Switzerland, 2020, https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1.

[2] P. A. Heidenreich, J. G. Trogdon, O. A. Khavjou et al., “Forecasting the future of cardiovascular disease in the United States,” *Circulation*, vol. 123, no. 8, pp. 933–944, 2011 Mar 1.

[3] J. Soni, U. Ansari, D. Sharma, and S. Soni, “Predictive data mining for medical diagnosis: an overview of heart disease prediction,” *International Journal of Computer Application*, vol. 17, no. 8, pp. 43–48, 2011.

[4] S. H. Jee, Y. Jang, D. J. Oh et al., “A coronary heart disease prediction model: the Korean Heart Study,” *BMJ Open*, vol. 4, no. 5, e005025 pages, 2014.

[5] The Economist, *From Not Working to Neural Networking*, http://www.economist.com/news/specialreport/21700756-artificialintelligence-boom-based-old-idea-modern-twist-not, 2016.

[6] S. Ben-David and S. Shalev-Shwartz, *Understanding Machine Learning," from Theory To Algorithms*, Cambridge University Press, Cambridge, UK, 2020.

[7] M. P. M., D. S. Bote and S. D. Deshmukh, "Heart disease prediction system using naive Bayes," *Int J. Enhanced Res. Sci. Technol. Eng*, vol. 2, no. 3, 2013.

[8] A. Ganna, P. K. E. Magnusson, N. L. Pedersen et al., "Multilocus genetic risk scores for coronary heart disease prediction," *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 33, no. 9, pp. 2267–2272, 2013.

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

[10] A. Dandapath and M. K. Raja, "Heart disease prediction using machine learning techniques: a survey," *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 684–687, 2018.

[11] J. Soni, S. Soni, D. Sharma, and U. Ansari, "Intelligent and effective heart disease prediction system using weighted associative classifiers," *International Journal on Computer Science and Engineering*, vol. 3, no. 6, pp. 2385–2392, 2011.

[12] R. Subramanian and L. Parthiban, "Intelligent heart disease prediction system using CANFIS and genetic algorithm," *International Journal of Biological, Biomedical and Medical Sciences*, vol. 3, no. 3, 2008.

[13] O. W. Samuel, A. K. Asogbon, P. Fang, and G. Li, "An integrated decision support system based on ANN and Fuzzy_ AHP for heart failure risk prediction," *Expert Systems with Applications*, vol. 68, pp. 163–172, 2017.

[14] Y. Kumaraswamy and S. B. Patil, "Intelligent and effective heart attack prediction system using data mining and artificial neural network," *European Journal of Scientific Research*, vol. 31, pp. 642–656, 2009.

[15] J. Singaraju and K. Vanisree, "Decision support system for congenital heart disease diagnosis based on signs and symptoms using neural networks," *International Journal of Computer Application*, vol. 19, pp. 6–12, 2015.

[16] B. Edmonds, *Proceedings of AISB Symposium on Socially Inspired Computing*, pp. 1–12, Hatfield, 2005.

[17] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: a comparison of three data mining methods," *Artificial Intelligence in Medicine*, vol. 34, no. 2, pp. 113–127, 2005 Jun.

[18] J. Y.. , S. Kiyasu, *Patent No. 4,338*, p. 396, U.S. Patent and Trademark Office, Washington, DC, 1982.

[19] M. Raihan, S. Mondal, A. More et al., "Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design," in *Proceedings of the 19th International Conference on Computer and Information Technology (ICCIT)*, pp. 299–303, IEEE, Dhaka, Bangladesh, December 2016.

[20] L. Tolo and T. Lengauer, "Classification with correlated features: un- reliability of feature ranking and solutions," *Bioinformatics*, vol. 27, no. 14, p. 1986, 1994.

[21] A. Hambali Moshood and D. Gbolagade Morufat, "Ovarian cancer classification using hybrid synthetic minority oversampling technique and neural network," *Journal of Advances in Computer Research (JACR)*, vol. 7, no. 4, pp. 109–124, 2016.

[22] J. Vandar Kuzhali and S. Vengataasalam, "A novel ensemble classifier based classification on large datasets with hybrid feature selection approach," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 17, pp. 3633–3642, 2014.

[23] D. Lavanya, R. K. Usha, Ensemble decision tree classifier for breast cancer data," *International Journal of Information Technology and Computer Science*, vol. 2, no. 1, pp. 17–24, 2012.

[24] C. A. Shipp and L. I. Kuncheva, "Relationships between combination methods and measures of diversity in combining classifiers," *Information Fusion*, vol. 3, no. 2, pp. 135–148, 2002.

[25] R. Lior, "Ensemble-based classifiers. Artificial intelligence," *Review*, vol. 33, pp. 1–39, 2010.

[26] J. Leskovec and A. Grover, "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 2016.

[27] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, C. A, 1984.

[28] K. Sharma, T. R. Mahesh, and J. Bhuvana, "Big data technology for developing learning resources," *Journal of Physics: Conference Series*, Conference Series IOP Publishing Ltd, vol. 1979, no. 1, p. 012019, May 2021.

[29] M. R. Sarveshvar, A. Gogoi, A. K. Chaubey, S. Rohit, and T. R. Mahesh, "Performance of different machine learning techniques for the prediction of heart diseases," in *Proceedings of the International Conference on Forensics, Analytics, Big Data, Security (FABS)*, pp. 1–4, Bengaluru, India, December 2021.

[30] I. H. Witten and E. Frank, "Data Mining Practical Machine Learning Tools and Techniques," *The United States of America, Morgan Kaufmann Series in Data Management Systems*, 2nd Edition, 2005.

[31] H. K. Shashikala, T. R. Mahesh, V. Vivek, M. G. Sindhu, C. Saravanan, and T. Z. Baig, "Early detection of spondylosis using point-based image processing techniques," in *Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 655–659, Bangalore, India, August 2021.

[32] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: a new pre-processing approach for highly imbalanced datasets by improving SMOTE international journal of computational intelligence systems," vol. 12, no. 2, p. 1412, 2019.

[33] P. Chaitanya Reddy, R. M. S. Chandra, P. Vadiraj, M. Ayyappa Reddy, T. R. Mahesh, and G. Sindhu Madhuri, "Detection of plant leaf-based diseases using machine learning approach," in *Proceedings of the 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 1–4, Bangalore, India, December 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 9810910, 1 page
https://doi.org/10.1155/2023/9810910

Retraction

Retracted: Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques

Computational Intelligence and Neuroscience

Received 10 October 2023; Accepted 10 October 2023; Published 11 October 2023

Copyright © 2023 Computational Intelligence and Neuroscience. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. P. Rodrigues, R. Fernandes, A. Aakash et al., "Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5211949, 14 pages, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 5211949, 14 pages
https://doi.org/10.1155/2022/5211949

RETRACTED

Research Article

Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques

Anisha P Rodrigues,[1] **Roshan Fernandes**,[1] **Aakash A**,[1] **Abhishek B**,[1] **Adarsh Shetty**,[1] **Atul K**,[1] **Kuruva Lakshmanna**,[2] **and R. Mahammad Shafi**[3]

[1] *Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte, Karkala, India*
[2] *SITE, Vellore Institute of Technology, Vellore, Tamilnadu, India*
[3] *Department of Electrical and Computer Engineering, College of Engineering and Technology, Tepi Campus, Mizan-Tepi University, Tepi, Ethiopia*

Correspondence should be addressed to R. Mahammad Shafi; mahammadshafi.r@mtu.edu.et

Received 9 March 2022; Revised 29 March 2022; Accepted 1 April 2022; Published 15 April 2022

Academic Editor: Muhammad Ahmad

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

In this modern world, we are accustomed to a constant stream of data. Major social media sites like Twitter, Facebook, or Quora face a huge dilemma as a lot of these sites fall victim to spam accounts. These accounts are made to trap unsuspecting genuine users by making them click on malicious links or keep posting redundant posts by using bots. This can greatly impact the experiences that users have on these sites. A lot of time and research has gone into effective ways to detect these forms of spam. Performing sentiment analysis on these posts can help us in solving this problem effectively. The main purpose of this proposed work is to develop a system that can determine whether a tweet is "spam" or "ham" and evaluate the emotion of the tweet. The extracted features after preprocessing the tweets are classified using various classifiers, namely, decision tree, logistic regression, multinomial naïve Bayes, support vector machine, random forest, and Bernoulli naïve Bayes for spam detection. The stochastic gradient descent, support vector machine, logistic regression, random forest, naïve Bayes, and deep learning methods, namely, simple recurrent neural network (RNN) model, long short-term memory (LSTM) model, bidirectional long short-term memory (BiLSTM) model, and 1D convolutional neural network (CNN) model are used for sentiment analysis. The performance of each classifier is analyzed. The classification results showed that the features extracted from the tweets can be satisfactorily used to identify if a certain tweet is spam or not and create a learning model that will associate tweets with a particular sentiment.

1. Introduction

In recent times, the use of microblogging platforms has seen huge growth, one of them being Twitter. As a result of this growth, businesses and media outlets are increasingly looking for methods to use Twitter to gather information on how people perceive their products and services. Although there has been research on how sentiments are communicated in genres such as news articles and online reviews, there has been far less research on how sentiments are expressed in microblogging and informal language due to message length limits. In recent years, many businesses have used Twitter data and have obtained upside potential for businesses venturing into various fields. On the other hand, scammers and spambots have been actively spamming Twitter with malicious links and false information, causing real users to be misled. Our goal is to gather an arbitrary amount of data from a prominent social media site, namely, Twitter, and perform spam detection and sentiment analysis. This research work aims to create a model that can extract information from tweets, identify them as spam or not, and link the collected tweets to a specific sentiment. The features required are extracted using vectorizers like TF-IDF and the Bag of Words model. The extracted features are passed into classifiers. For spam detection, decision tree, logistic regression, multinomial naïve Bayes, support vector machine,

RETRACTED

random forest, and Bernoulli naïve Bayes are used, whereas, for sentiment analysis, stochastic gradient descent, support vector machine, logistic regression, random forest, naïve Bayes, and deep learning methods such as simple recurrent neural network (RNN) model, long short-term memory (LSTM) model, bidirectional long short-term memory (BiLSTM) model, and convolutional neural network (CNN) 1D model are used. Classification results and performance are evaluated and contrasted in terms of overall accuracy rate, recall, precision, and F1-score. To assess the efficiency of our model, we put it to the test using real-time tweets.

1.1. Contributions of the Proposed Work. The main contributions of the proposed work are given as follows:

(i) Most of the existing work showed the use of manual labeling on the dataset used, although very accurate, there was a limit on the size of the dataset. In the proposed spam detection, we took a large SMS dataset for training and testing our models with live tweets.

(ii) In the existing works, no major distinctions between various topics and keywords of tweets while analyzing the sentiment are seen. In the proposed sentiment analysis, we wish to observe the differences in prediction when taking numerous general and topical subjects.

(iii) The proposed work has experimented on real-time data directly from Twitter.

(iv) The proposed work analyzed the performance measures of many of the classification models by using different stemmers and lemmatizes on real-time data and compared the results based on evaluation parameters.

(v) The multinomial Naïve Bayes classifier achieved a classification accuracy of 97.78% and the deep learning model, namely, LSTM, achieved a validation accuracy of 98.74% for the Twitter spam classification. The support vector machine classifier achieved a classification accuracy of 70.56% and the deep learning model, namely, LSTM, achieved a validation accuracy of 73.81% for the Twitter sentiment analysis for the randomly chosen tweets.

The rest of the content is organized as follows. Section 2 discusses the related work, Section 3 gives the detailed methodology used in the proposed work, Section 4 discusses the results, and the concluding observations on the proposed work and the future work are discussed in Section 5.

2. Related Work

Spam classification is performed using real-time Twitter data. Text mining techniques are used for preprocessing, and machine learning techniques such as backpropagation neural network and naïve Bayes are used as classifiers. Twitter API is used to collect real-time datasets from publicly available Twitter data. It is found that naïve Bayes performs better than backpropagation neural network [1]. A system is proposed that uses tweet-based features and the user to classify tweets. The benefits of these tweet text features include the ability to detect spam tweets even if the spammer attempts to create a new account. For the evaluation, it was run through four different machine learning algorithms and their accuracy was determined [2]. The spam detection system is developed for real-time or near-real-time Twitter environments. The method used is to capture the bare minimum of features available in a tweet. The two datasets used are the Social Honeypot Dataset and 1KS-10KN. The usage of several feature sets has the advantage of increasing the possibilities of capturing diverse types of spam and making it harder for spammers to exploit all of the spam detection system's feature sets [3]. The support vector machine method is used to classify the tweets as spam. The Waikato Environment for Knowledge Analysis and the Sequence Minimal Optimization Algorithm were utilized. To train the model, a dataset of tweets from Twitter was taken. When compared to other spam models, this model has a high level of reliability based on the correctness of the system [4]. The decision tree induction algorithm, the naïve Bayes algorithm, and the KNN algorithm are used to detect spam on Twitter. The research work compiled a dataset by picking 25 regular Twitter users at random and crawling tweets from publishers they follow. The proposed solution has the advantage of being practical and delivering much better classification results than other methodologies now in use. One problem with the proposed strategy is that it takes longer to train models, and the feature extraction procedure may be inefficient and expensive [5]. The naïve Bayes and logistic regression are used for Twitter spam detection. The dataset was obtained by utilizing spam words, and some labeling was performed on it. The advantage of using both the tweet and account-based features is that it boosts the accuracy rate even more [6].

The features of spam profiles on Twitter are investigated to improve social spam detection. Relief and information gain are the two approaches used for feature selection. Four classification methods are used and compared in this study: multilayer perceptrons, decision trees, naïve Bayes, and k-nearest neighbors. A total of 82 Twitter profiles have been gathered in this dataset. The benefit of this strategy is that promising detection rates can be attained independent of the language of the tweets. The disadvantage of this strategy is that they employed a small dataset for training, which results in poor accuracy [7]. The support vector machine, K-nearest neighbor (KNN), naïve Bayes, and bagging algorithms are used for spam detection on Twitter. The UCI machine learning data repository was utilized as the dataset. The benefit is that the performance of different cutting-edge text classification algorithms, including naïve Bayes, was compared against bagging (an ensemble classifier) to filter out spam comments. Ensemble classifiers have been discovered to generate better outcomes in the vast majority of cases [8]. Various strategies are discussed to acquire the best accuracy achievable utilizing the dataset. The classifiers employed were naïve Bayes classifier (NB), support vector machine (SVM), KNN, artificial neural network (ANN), and random

RETRACTED

forest (RF). The datasets utilized were SMS Spam Corpora (UCI repository) and Twitter Corpora (public live tweets). The benefit is that these classical classifiers performed well in terms of accuracy in spam classification in both datasets [9]. The RF, Maximum-Entropy (MaxEnt), C-Support Vector Classification (SVC5), Extremely Randomized Trees (ExtraTrees), gradient boosting, spam post detection (SPD), and multilayer perceptron (MLP) algorithms are used to classify the spam tweets. The automatically annotated spam posts detection dataset (SPD automated) named Honeypot and manually annotated spam posts detection dataset were used (SPD manual). Automated spam accounts, according to the study, follow a well-defined pattern with periodic activity spikes. Any real-time filtering application can benefit from this strategy. The performance of the various models is consistent, and there is a considerable improvement over the baseline. The problem is that distinguishing between genuine human users and legitimate social bots, as well as human spammers and social bot spammers, is difficult [10].

Spam detection methods include supervised, unsupervised, and semisupervised. The product dataset reviews are used as the dataset and it has been discovered that combining unlabeled data with a small amount of labeled data (which will be challenging to produce effectively) can enhance accuracy [11]. A survey of sentiment classification, opinions, opinion mining process, opinion spam detection, and rules to identify the spam is performed. The techniques used are Sentiment Classification and Opinion mining. To classify social media networks and website review dataset opinions, machine learning algorithms such as Naïve Bayes and SVM are utilized. The benefit is that the usefulness of a review may be established using a regression model and providing a utility value to each review, allowing review ranking to be further trained and tested [12]. A model for sentiment analysis is built, which predicts the box office performance of films in India on their opening weekend. The technique used is lexicon-based filtering and trend analysis using agglomerative hierarchical clustering for the movie review dataset. The advantage is that the lexicon method is simpler than the methods available in machine learning. The disadvantages include limitations of Twitter API, sampling bias, noise, promotion and spam, and infringement of privacy [13]. A method for making opinion mining easier is performed by combining linguistic analysis and opinion classifiers to predict positive, negative, and neutral sentiments for political parties using Naïve Bayes and SVM. It was observed that SVM performed better for the given contextual data [14]. Sentiword was utilized to recognize nouns, adjectives, and verbs, while bespoke software was built to determine other parts of speech using POS tags to analyze iPhone 6 reviews. The filtered tweets were scored and inserted into a MySQL database, which was then exported to Rapid Miner and the NamSor add-on was installed. For each matched tweet, NamSor's list of genders was then put into the database. The implementation of these methods was relatively easy as many software tools were used. However, NamSor used for gender identification is not very accurate [15, 16].To deal volatility of spam contents and spam drift, a framework is introduced. The framework uses the strength of the unsupervised machine learning approach that learns from unlabeled tweets. Experimental results show that the proposed unsupervised learning method achieves a recall value of 95% to learn the pattern of new spam activities [17].

The major challenge in the supervised learning approach for sentiment analysis is domain-dependent feature set generation, which is addressed in the study and a novel approach is proposed to identify unique lexicon set in Twitter sentiment analysis. The study shows that the Twitter-specific lexicon set is small in size and domain-dependent. The vectorization used in traditional approaches generates a highly sparse matrix, which produces low accuracy measures. The study feature set is hierarchically reduced and to reduce sparsity, a small set of seven metafeatures is used. Twitter domain refunded feature set produces excellent sentiment classification results [18]. To identify the review's semantic orientation Bayesian classifier (NB), SVM, part-of-speech tagging, and SVM and scoring-based hybrid approach (called HS-SVM) are used in scientific article reviews. The HS-SVM classifier produces the best results, while the scoring system performs marginally better than the supervised approaches in the 5-point scale classification. Handling multilingual reviews is a drawback [19]. A study and comparison analysis of existing sentiment analysis techniques such as lexicon-based approaches and machine learning and evaluation metrics are performed on Twitter data. The techniques used are Max Entropy, naïve Bayes, and SVM. It supports various domains such as medical, social media, and sports. The drawbacks include identification of the subjective part of the text, domain dependency, detection of sarcasm, explicit negation of sentiments, recognition of entity, and handling comparisons [20, 21, 22]. The dragonfly algorithm is used for a swarm-based improvement system to examine high-recommendation websites for the online E-shopping sites and Fuzzy C-means (FCM) datasets. The advantage is that it helps expand consumer loyalty by identifying highlights of specific items and better feature identification. The disadvantage is that it does not support characterization procedures for positive and negative groups [23]. The Waikato Environment for Knowledge Analysis (WEKA) was utilized to construct data mining methods for preprocessing, classification, clustering, and outcome analysis of the Twitter Sentiment System for SemEval 2016 and Sanders Analytics Twitter sentiment corpus. The advantage is that it uses WEKA to classify sentiments from Twitter data and provides improved accuracy. The downside is that the result could be impacted by the training features and sentiment classification method [24]. The people's opinions and sentiments concerning Syrian refugees are analyzed. WordCloud is used to visualize a massive amount of data with the use of a sentiment analysis lexicon [25]. Machine learning techniques can be extended to classify fake reviews, fake news, aspect analysis, and DNA sequence mining [16, 26, 27, 28]. The text classification is improved using the two-stage text feature selection algorithm [29, 30]. The multiobjective genetic algorithm and CNN-based algorithms are used to detect spam messages on Twitter [31]. According to the detailed survey made on Twitter spam detection, there are limited labeled datasets available to train

RETRACTED

the spam detection algorithm. This survey has given an insight into various vectorization techniques used in representing the text [32]. Researchers have used the metadata along with the dataset to increase the accuracy of sentiment analysis [33]. Machine learning algorithms have been applied for spam detection in e-mail and IoT platforms too [34]. The summary of Twitter spam detection and sentiment analysis is given in Table 1.

To conclude, from the literature survey, we observe that many of the researchers have contributed to the Twitter sentiment analysis. The researchers have used different datasets and applied different machine learning and deep learning algorithms. The main research gap observed is the lack of dataset used for Twitter spam detection and comparing various machine learning and deep learning models on spam classification. Also, the proposed work has contributed to analyzing the real-time tweets for spam detection and sentiment analysis. Hence, we believe that the proposed methodology makes a unique contribution to Twitter spam detection and sentiment analysis in terms of the type of dataset used, algorithms applied for classification, and various analyses used on the results.

3. Methodology

The proposed system architecture shown in Figure 1 follows the principles used in natural language processing tasks and these include all the steps of preprocessing, training the model, and testing it on live tweets. Tweets are pulled from the Twitter database via the tweepy API. Using vectorizers, we build a feature vector which is then used for testing the models. We use the classification models that have already been trained by our text datasets and then we select the model with the highest accuracy and predict the live tweets with the given model.

The initial step in the proposed methodology is to collect the dataset. The dataset used for the spam detection has a size of 5572, in which 4825 ham and 747 spam contents are present. The dataset used for the sentiment analysis has 31015 tweets, in which 12548 are labeled neutral, 9685 are labeled positive, and 8782 are labeled negative class. Further, the proposed methodology has analyzed the live tweets for classifying the tweets as positive, negative, and neutral. This dataset must be preprocessed for further analysis. The main stages included in the preprocessing include filtering, tokenization, stop word removal, and stemming/lemmatization. Then, the dataset has to be represented in vector form, namely, TF-IDF or Bag of Words. This step is followed by training the classification models on the given features. Choose models suited for multiclassification for sentiment analysis and binary classification for spam detection. The results will be evaluated and compared using the various evaluation parameters. The analysis will be performed on the live Twitter data too.

3.1. Cleaning and Visualizing Data.

3.1. Cleaning and Visualizing Data. One of the more rudimentary ways to find the sentiment of a given tweet is by analyzing the emojis present in a tweet. Popular websites like Twitter and Quora have so much data that a great deal of effort is spent automating the spam removal process. Also, it is important to filter out fake news or reviews on these sites. Organizations will be particularly interested in the opinion of various users of their products. To perform these tasks, it is first imperative that we perform some form of text preprocessing. Four steps need to be taken for preprocessing:

(1) Filtering: this entails the removal of URL links, e.g., http:Google.com, also removing tags to other usernames, which in Twitter often begin with an @ symbol.

(2) Tokenization: the next step involves building a Bag of Words, by removing any punctuation or question marks. This allows large amounts of data to be represented in a proper format.

(3) Removing stop words:remove articles and prepositions such as a, an, and the.

(4) Constructing n-grams: this is one of the most crucial steps. An n-gram is defined as follows: it is an n-item contiguous sequence from a particular text or speech sample. Depending on the application, the elements can be letters, phonemes, words, syllables, or base pairs.

It is observed that the decision on whether a unigram or a bigram needs to be constructed is taken on the result we wish to accomplish. Unigrams by themselves provide good coverage of data, but bigrams and trigrams lend themselves to sentiment analysis and product reviews; for example, bigrams like “not good” convey sentimentality quite succinctly. For the proposed model, we have only used unigram tokens for tweet preprocessing and instead have focused on comparing various stemmers and lemmatizers mostly reviewing their accuracy. Even though lemmatizers are guaranteed to derive the base word of a composite word found in our text document, such a task does not create a massive push in accuracy and the classification models used were more important. After cleaning up the text documents, we can proceed with further analysis by splitting our texts into tokens. These tokens must be converted into feature vectors. Feature vectors are a method of representation that is to be used while training the various classification models.

In the proposed work, we have mainly compared two techniques, namely, Bag of Words and TF-IDF methods. The Bag of Words is a very simple method of conversion wherein all the different words in the corpus are considered as features. Each column represents the number of times a particular term appears in the text. Although it is inexpensive to compute, it does not provide much information other than the number of occurrences of the given word. Term frequency-inverse document frequency (TF-IDF) method assigns a score for each word in the text-based not only on the number of times its occurrence but also on how likely it can be found in texts of other classifications. This means that words that are common in almost all texts, irrespective of their classifications, are assigned a lower score. These feature vectors can now be used by the different classification models for training.

Table 1: Summary of Twitter spam detection and sentiment analysis.

Techniques used	Key findings
Backpropagation neural network and naïve Bayes are used as classifiers [1] for spam detection.	Spam classification is performed on real-time Twitter data. Naïve Bayes performs better than backpropagation neural network.
Support vector machine method and sequence minimal optimization algorithm [4] are used for spam detection.	When compared to other spam detection models, this model has a high level of reliability based on the correctness of the system.
The decision tree induction algorithm, the naïve Bayes algorithm, and the KNN algorithm are used for spam detection [6].	The proposed solution has the advantage of being practical and delivering much better classification results than other methodologies now in use.
Relief and information gain are the two approaches used for feature selection. Classifiers used for spam detection are multilayer perceptrons, decision trees, naïve Bayes, and k-nearest neighbors [7].	A total of 82 Twitter profiles have been gathered in this dataset. The proposed work uses different language tweets but fails to give better accuracy as the dataset size is small.
The support vector machine, K-nearest neighbor (KNN), naïve Bayes, and bagging algorithms are used for spam detection [8].	Naïve Bayes was compared against bagging (an ensemble classifier) to filter out spam comments. Ensemble classifiers have been discovered to generate better outcomes in the vast majority of cases.
Naïve Bayes classifier (NB), support vector machine (SVM), K-nearest neighbor (KNN), artificial neural network (ANN), and random forest (RF) are used for spam detection [9].	SMS spam corpora (UCI repository) and Twitter corpora (public live tweets) datasets are used for analysis. The benefit is that these classical classifiers performed well in terms of accuracy in spam classification in both datasets.
The random forest, maximum-entropy (MaxEnt), C-Support vector classification (SVC5), extremely randomized trees (ExtraTrees), gradient boosting, spam post detection (SPD), and multilayer perceptron (MLP) algorithms are used for spam detection [10].	The automatically annotated spam posts detection dataset (SPDautomated) named Honeypot and manually annotated spam posts detection dataset was used (SPDmanual) and the different algorithms are evaluated and compared.
Agglomerative hierarchical clustering is used for spam detection [13].	The movie review dataset is used for the analysis. The lexicon method used is simpler than the methods available in machine learning.
Naïve Bayes and SVM are used for spam detection [14].	The political dataset is used for analysis. It was observed that SVM performed better for the given contextual data.
Rapid miner and the NamSor are used for tweet classification [15].	NamSor, which was used for gender identification, is not very accurate.
An unsupervised machine learning approach is used for tweet spam classification and sentiment analysis [17].	The proposed unsupervised learning method achieved a recall value of 95% to learn the pattern of new spam activities.
Lexicon-based sentiment analysis [18].	A small Twitter-specific lexicon set is used, which gives good accuracy. For general tweet analysis, the accuracy is reduced.
Bayesian classifier (NB), support vector machines (SVM), part-of-speech tagging, and SVM and scoring-based hybrid approach (called HS-SVM) are used in scientific article reviews classification [19].	The HS-SVM classifier produces the best results.
Max entropy, naïve Bayes, and support vector machine are used for sentiment classification [20].	The tweets are analyzed on domains such as medical, social media, and sports.

3.2. Machine Learning Algorithms Used for Twitter Spam Detection and Sentiment Analysis. Various machine learning algorithms used for Twitter spam detection and sentiment analysis are discussed in this section.

3.2.1. Decision Tree. Decision tree is a supervised classifier that can be employed to tackle classification and regression issues; however, it is most commonly used for classification. In this tree-structured classifier, internal nodes provide dataset features, branches reflect decision rules, and each leaf node delivers the result. The decision node and the leaf node are the two nodes in the Decision Tree. Decision Nodes are used to make a decision and have numerous branches, whereas Leaf Nodes are the outcome of such decisions and have no more branches. Entropy controls how a Decision Tree decides how to partition data. It influences the way a Decision Tree constructs its boundaries. Its formula is given as follows:

$$H(s) = -\text{probability of } \log_2(p+) - -\text{probability of } \log_2(p-), \quad (1)$$

where $(p+)$ represents the percentage of the positive class and (p-) represents the percentage of the negative class.

3.2.2. Logistic Regression. In logistic regression, the sigmoid function is a binary classification function that is used for binary classifications. Given an initial feature vector x, it gives an output probability of the classification of the given text. Its formula is given as follows:

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}, \quad (2)$$

where P is the probability of a 1 (the proportion of 1s), e is the natural logarithm base, and a and b are model parameters. When X is 0, the value of a yields P and b controls how

RETRACTED

RETRACTED

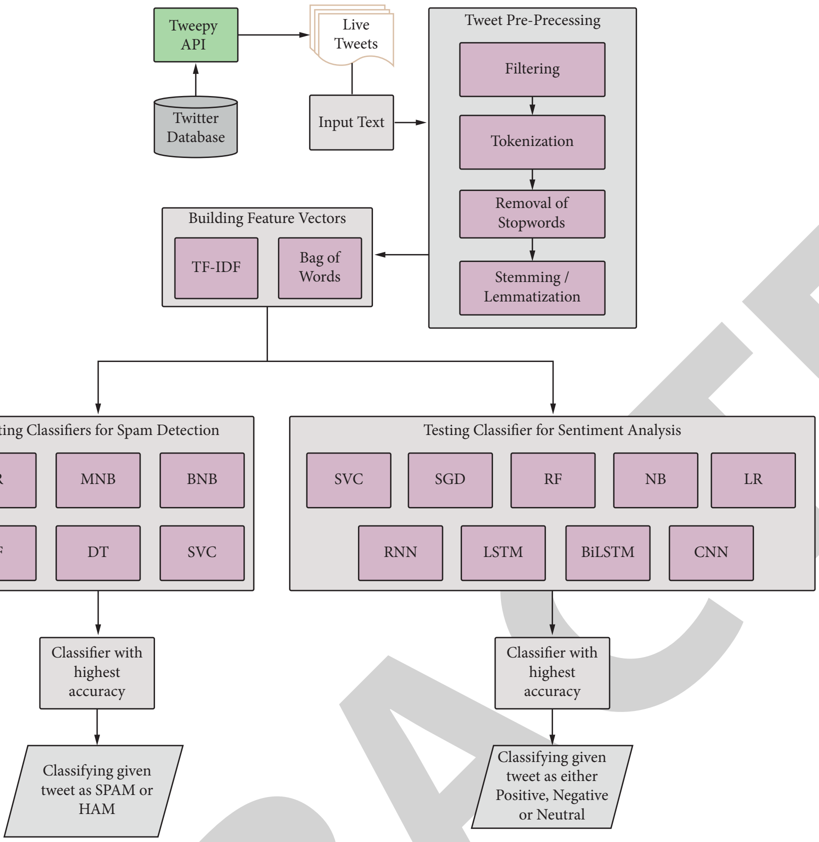

Tweepy API
Live Tweets
Tweet Pre-Processing
Filtering
Twitter Database
Input Text
Tokenization
Removal of Stopwords
Building Feature Vectors
TF-IDF
Bag of Words
Stemming / Lemmatization
Testing Classifiers for Spam Detection
LR
MNB
BNB
RF
DT
SVC
Testing Classifier for Sentiment Analysis
SVC
SGD
RF
NB
LR
RNN
LSTM
BiLSTM
CNN
Classifier with highest accuracy
Classifier with highest accuracy
Classifying given tweet as SPAM or HAM
Classifying given tweet as either Positive, Negative or Neutral

Figure 1: The architecture of Live Tweet analysis.

rapidly the probability changes when X is changed by a single unit.

3.2.3. Multinomial Naïve Bayes. Multinomial Naïve Bayes is used for features that reflect counts or count rates since the multinomial distribution describes the chance of detecting counts among a number of categories. Text classification, where the features are connected to word counts or frequencies inside the documents to be categorized, is one area where multinomial Naïve Bayes is frequently utilized.

Samples (feature vectors) in a multinomial event model describe the frequencies with which specific events have been created by a multinomial $(p_1 \ldots\ldots p_n)$, where {\displaystyle p_{i}} p_i is the chance that event i happens. A feature vector {\displaystyle \mathbf {x} =(x_{1},\dots,x_{n})}$X = (x_1 \ldots\ldots x_n)$ is then a histogram, with x_i representing the number of times event i was seen in a given instance. This is the most common event model for document classification. The likelihood of observing a histogram x is given as follows:

$$p(X|C_k) = \frac{\left(\sum_{i=1}^{n} x_i\right)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} Pk_i^{x_i}. \tag{3}$$

3.2.4. Support Vector Machine. Each data is represented as a point in n-dimensional space (with n being the number of features), with each feature's value becoming the SVM algorithm's value for a specific coordinate. SVMs have supervised machine learning models that address two-group classification problems using classification techniques. By providing labeled training data for each category, SVM models are capable of categorizing new texts. They have two major advantages over modern methods, such as neural networks: they are faster and perform better with fewer data (in the thousands). This makes the method particularly well suited to text classification problems, where just a few thousand tagged examples are often available. A technique called kernel trick is used by the SVM algorithm, by which it converts low input dimensions to higher input dimensions using complex data transformations. This is how the SVM converts a nonseparable problem into a separable one.

3.2.5. Random Forest. Random Forest is a supervised learning approach that can be employed for regression and classification purposes, with the algorithm being highly adjustable and user-friendly. Random Forests create decision trees from data samples picked at random, get predictions from each tree, and then vote on the best option. The feature's worth can also be evaluated reliably. It is given by the following formula:

$$ni_j = w_j C_j - W_{\text{left}(j)} C_{\text{left}(j)} - W_{\text{right}(j)} C_{\text{right}(j)}, \tag{4}$$

where ni_j is the importance of node j, w_j is the weighted number of samples reaching node j, C_j is the impurity value of node j, left(j) is the child node from left split on node j, and right(j) is the child node from right split on node j.

3.2.6. Bernoulli Naïve Bayes. The Boolean variables are similar to multinomial Naïve Bayes variables and act as predictors. The parameters used to forecast the class variables only accept binary replies, for instance, if a word occurs in the text or not. If x_i is a Boolean expressing the presence or absence of the ith phrase from the lexicon, then the likelihood of a document given a class {\displaystyle C_{k}} C_k is given by the following:

$$p(X|C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i} (1 - pk_i)^{(1-x_i)}. \tag{5}$$

3.2.7. Stochastic Gradient Descent. Stochastic Gradient Descent is a machine learning optimization technique for identifying model parameters that best match expected and actual outcomes. It is a clumsy but efficient technique. It is efficient because rather than calculating the cost of multiple data points, we just consider one data point and the accompanying gradient descent, after which the weights are updated. The update step is shown in the following:

$$w_j := w_j - \alpha \frac{\partial J_i}{\partial w_j}, \tag{6}$$

where J_i is the cost of ith training example.

3.2.8. Deep Learning Methods Used for Twitter Spam Detection Sentiment Analysis. Deep learning is a branch of machine learning whose methods are based on the form and composition of ANNs. The proposed work used four deep learning models for Twitter sentiment analysis, namely, Simple RNN, LSTM, BiLSTM, and 1D CNN model.

3.2.9. Simple RNN Model. A RNN is an ANN in which nodes are connected in a directed graph in a temporal order. This allows it to respond in a time-dependent manner. RNNs, which are created from feedforward neural networks, can process variable-length sequences of inputs by using their internal state. To add new information, the model alters the existing data by applying a function. As a result, the entire information is altered; i.e., there is no distinction between ‘important’ and ‘not so important information.

3.2.10. Long Short-Term Memory (LSTM) Model. Long short-term memory is a prominent RNN architecture that was developed to deal with the issue of long-term dependence and solve the vanishing gradient problem. The RNN model may be unable to forecast the present state well if the previous state influencing the current prediction is not recent. LSTMs have three gates in the deep levels of the neural network: an input gate, an output gate, and a forget gate. These gates control the flow of data needed to forecast the network’s output.

3.2.11. Bidirectional Long Short-Term Memory (BiLSTM) Model. A bidirectional LSTM is a sequence processing model that comprises two LSTMs: one that forwards the input and the other that reverses it. BiLSTM effectively improves the amount of data available to the network, providing a richer context for the algorithm.

3.2.12. 1D Convolutional Neural Network (CNN) Model. A CNN is effective in detecting simple patterns in data, which are subsequently utilized to create more sophisticated patterns in the upper layers. When we want to extract valuable features from small (fixed-length) chunks of the whole dataset and the location of the feature inside the segment is not important, a 1D CNN is quite useful. This holds good for analysis and retrospection of time sequences of sensor data (such as proximity or barometer data) and the study of any type of signal data over a set time frame (like audio signals). A convolution neural network comprises 3 layers: input, output, and hidden layer. The middle layers act as a feedforward neural network. These layers are considered hidden as both the activation function and the final convolution are concealed from their inputs and outputs. The hidden layers also include convolutional layers. The dot product of the convolution kernel with the input matrix of the layer is performed here. ReLU and the Frobenius inner product act as the activation functions. A feature map is generated by the convolution operation as the convolution kernel slides along the input matrix for the layer, later contributing to the input of the following layer. Pooling layers, fully connected layers, and normalization layers are added soon after to improve functionality.

After having trained various models, we tested these classifiers with live tweets from Twitter and this task is accomplished through the TweepyAPI. Tweepy is a python module that makes it possible to use the Twitter API. The TweepyAPI has many ways inbuilt through which it can relay the necessary information in JSON format. We used the oath method to communicate with the API. This involved using the existing Twitter account to create a developer account. After the developer account is created, Twitter provides us with four keys of which two are private keys. We have to use these keys to access the JSON data. These JSON data contain a lot of information about every tweet we wish to analyze, including its timestamp, the text, user, and device used.

We analyze these tweets for both spam detection and sentiment analysis separately. For spam detection, we found that due to Twitter’s strict policies on account creation, there are not a lot of accounts that run bots that constantly tweet spam content. Thus, analyzing live spam tweets was a difficult proposition. Hence, we used an SMS dataset that had spam and nonspam classification for our training purposes. The SMS and tweet formats are very similar in format and thus could be used for our training purposes. After the preprocessing steps are applied, we turn the texts in the dataset to feature vectors, and then they are used for training our models. After the classification models have been trained with sufficient accuracy, we use the classifiers on actual live

RETRACTED

RETRACTED

tweets that appear on our account's feed. Finally, we classify these tweets as whether they are spam or not.

For sentiment analysis, we performed multi-classification on whether a given tweet's sentiment is positive, negative, or neutral. We obtained a large dataset from Kaggle that was used for our training purposes. After performing the preprocessing steps, we created the feature vectors to be used for training our models. After obtaining sufficient accuracy, we used these classifiers to detect various real-world trends. For us to do that, we created a program in the Jupyter Notebook that can take in a keyword or hashtag that we need to analyze along with the number of tweets that we would like to take into consideration. Since obtaining tweets in this manner also means that we might be able to get a significant number of tweets in various languages, we used the Text Blob package to change tweets from other languages into English. TextBlob library is a very useful library to work on various languages; we can use it to detect various languages and also translate from one language to another. We gather several tweets on relevant topics in JSON format and we need to convert them into a pandas.DataFrame. We used various classifiers to determine the sentiment of these tweets and observed how accurate our classifiers are for real-world texts.

The various evaluation metrics used in the proposed work include accuracy, recall, negative recall, precision, and F1-score.

Accuracy is computed as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \tag{7}$$

The accuracy measure gives how many data values are correctly predicted.

Sensitivity (or Recall) computes how many test case samples are predicted correctly among all the positive classes. It is computed as follows:

$$\text{Sensitivity} = \frac{\text{Number of True Positives}}{\text{Number of true Positives} + \text{Number of False negatives}}. \tag{8}$$

Specificity (or Negative Recall) computes how many test case samples are predicted correctly among all the negative classes. It is computed as follows:

$$\text{Specificity} = \frac{\text{Number of True Negatives}}{\text{Number of True Negatives} + \text{Number of False Positives}}. \tag{9}$$

Precision measure computes the number of actually positive samples among all the predicted positive class samples as follows:

$$\text{Precision} = \frac{\text{Number of True Positives}}{\text{Number of True Positives} + \text{Number of False Positives}}. \tag{10}$$

F1-score is the harmonic mean of Precision and Sensitivity. It is also known as the Sorensen–Dice Coefficient or Dice Similarity Coefficient. The perfect value is 1. F1-score is computed as shown in the following:

$$F1 - \text{score} = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}. \tag{11}$$

4. Results and Discussion

The results section is divided into two sections, Twitter spam detection and sentiment analysis using machine learning and deep learning techniques.

4.1. Machine Learning Techniques for Twitter Spam Detection.

The dataset used for the spam detection has a size of 5572, in which 4825 ham and 747 spam contents are present. The training data and testing data are split up at 70 : 30. Using WordCloud, we examined the word frequencies in Spam tweets. The WordCloud results for spam tweets are shown in Figure 2. According to the analysis, the English word "Free" was the most frequently occurring of all the words in the spam tweet data. As a result, the word takes up a large portion of the WordCloud image. In terms of frequency of occurrence, this word is closely followed by "Call" and thus occupies a similarly large portion of the WordCloud. Simply put, more frequent words take up a larger portion of the WordCloudthan less frequent words.

The proposed work used multinomial NB (MNB), Bernoulli NB (BNB), support vector machine (SVM), decision tree (DT), RF, and logistic regression (LG) classifiers to detect whether the Twitter data is spam or not. The proposed work used both TF-IDF and Bag of Words vectorizer before applying machine learning and deep learning. Table 2 gives various performance measures (in percentage) obtained for spam detection after applying the TF-IDF vectorizer.

Table 3 gives various performance measures (in percentage) obtained for spam detection after applying the Bag of Words vectorizer.

The analysis is further continued after selecting the Bag of Words and TF-IDF model to perform the vectorization of the tweet dataset, with the help of different stemming algorithms, which help reduce the features in its word stem. Before applying the various stemming algorithms, normalization is applied to the tweets along with preprocessing. The main steps implemented in the normalization process include the following: cleaning URLs, emojis, and hashtags; making tweets into lowercase; removing whitespaces; removing punctuations; autocorrect; tokenizing the tweet; removing stopwords. Table 4 gives the comparison of accuracy between normal analysis (without using any stemmers and lemmatizer), different stemmers, and lemmatizer with Bag of Words using different machine learning classifiers.

Table 5 gives the comparison of accuracy between normal analysis (without using any stemmers and lemmatizer), different stemmers, and lemmatizer with TF-IDF model using different machine learning classifiers.

The average of the evaluation parameter values was obtained using normal analysis, different stemmers, and a

FIGURE 2: Spam WordCloud.

TABLE 2: Performance Measures (in percentage) for spam detection after applying TF-IDF vectorizer.

Performance measures	Multinomial NB	Bernoulli NB	SVM	Decision tree classifier	Random forest classifier	Logistic regression
Classification accuracy	98.21	96.77	96.59	95.75	97.19	94.92
Precision	96.89	96.38	97.13	90.94	98.42	96.61
Recall	95.52	89.81	88.44	91.39	89.87	82.04
F1-score	96.19	92.74	92.16	91.16	93.56	87.47
Negative recall	99.24	99.44	99.72	97.43	100	99.86

TABLE 3: Performance measures (in percentage) for spam detection after applying the Bag of Words vectorizer.

Performance measures	Multinomial NB	Bernoulli NB	SVM	Decision tree classifier	Random forest classifier	Logistic regression
Classification accuracy	97.37	96.77	96.95	95.39	97.19	97.85
Precision	92.89	96.38	97.81	91.98	97.95	98.78
Recall	96.85	89.81	89.37	88.11	90.23	92.24
F1-score	94.74	92.74	93.01	89.90	93.61	95.18
Negative recall	97.57	99.44	99.86	98.19	99.86	100

TABLE 4: Accuracy measure (in percentage) for different stemmers and lemmatizer using BoW model.

Classifier	Normal analysis (BoW)	Porter stemmer	Snowball stemmer	Lancaster stemmer	Lemmatizer
Multinomial NB	97.37	97.07	97.19	97.49	97.19
Bernoulli NB	96.77	96.83	96.77	97.13	96.83
SVM	96.95	97.13	97.13	97.67	97.13
Decision tree	95.39	96.65	96.47	96.23	96.59
Random forest	97.19	97.43	97.37	97.43	97.25
Logistic regression	97.85	97.91	97.91	98.15	97.85

lemmatizer. The average values of the evaluation parameters observed for each classifier are shown in Figure 3.

4.2. Deep Learning Techniques for Twitter Spam Detection. The proposed work used four deep learning models for Twitter spam detection, namely, Simple RNN, LSTM, BiLSTM, and 1D CNN model. Table 6 gives the validation accuracy, validation loss, test accuracy, and test loss obtained for Twitter spam detection using various deep learning models.

Figure 4 shows the validation accuracy graph for the above-mentioned deep learning techniques over 70 epochs. Figure 5 shows the validation loss for the above-mentioned deep learning techniques over 70 epochs.

We have selected logistic regression (LR) for further real-time tweet spam detection. Table 7 gives the confusion matrix for predicting the real-time Twitter as spam or ham.

The sample live tweet fetched from Twitter is classified as spam and not spam (ham) according to the logistic regression classifier, as shown in Figures 6 and 7, respectively.

4.3. Machine Learning Techniques for Twitter Sentiment Analysis. The experiment used a dataset of tweets that were categorized as positive, negative, or neutral. The number of tweets used for the experiment is 31015, of which 12548 are labeled neutral, 9685 are labeled positive, and 8782 are labeled negative. These tweets are preprocessed by removing @user, removing HTTP and URLs, and removing

RETRACTED

RETRACTED

Table 5: Accuracy measure (In percentage) for different stemmers and lemmatizer using TF-IDF model.

Classifier	Normal Analysis (TF-IDF)	Porter stemmer	Snowball stemmer	Lancaster stemmer	Lemmatizer
Multinomial NB	98.21	97.85	97.85	97.85	97.97
Bernoulli NB	96.77	96.83	96.77	97.13	96.83
SVM	96.59	96.77	96.77	97.19	96.71
Decision tree	95.75	96.65	96.29	94.68	96.83
Random forest	97.19	97.31	97.55	97.07	97.31
Logistic regression	94.92	94.68	94.68	95.22	94.92

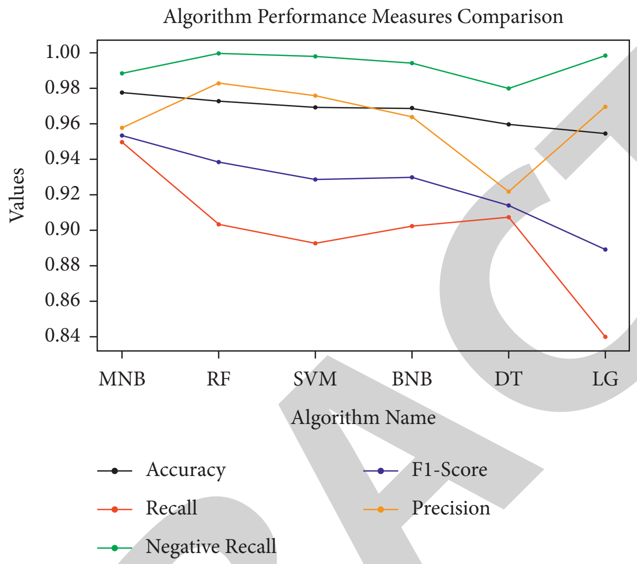


Figure 3: Comparison of average performance measures.

Table 6: Evaluation parameter values obtained for Twitter spam detection using deep learning models.

Deep learning models	Validation accuracy	Validation loss	Test accuracy	Test loss
Simple RNN	0.98684	0.0537	0.973	0.309
LSTM	0.98744	0.0524	0.974	0.200
Bidirectional LSTM	0.98445	0.0736	0.975	0.205
1D CNN	0.9797	0.1041	0.9743	0.110

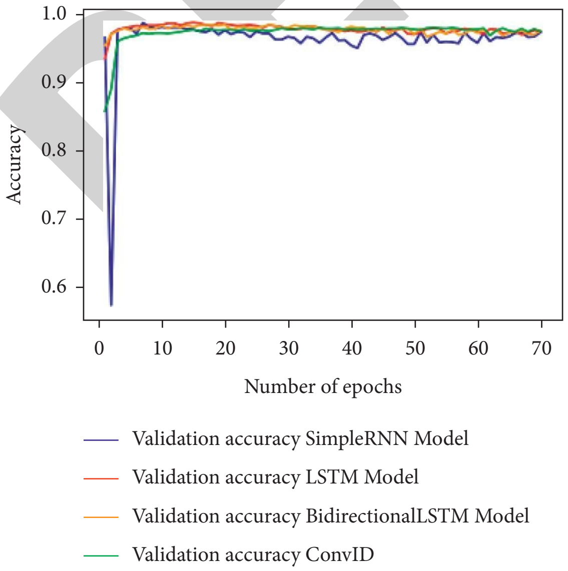


Figure 4: Validation accuracy for deep learning models.

special characters, numbers, and punctuation. The preprocessing step is followed by a tokenizer and Porter stemmer has been applied to these tokens. Then the tweets are reframed by combining the tokens. The count vectorizer (Bag of Words) technique is used to extract the features. The dataset is divided into 75% for training and 25% for testing.

WordCloud is used to analyze the word frequencies in the sentiment tweets. Figures 8–10 show the WordCloud results for positive, neutral, and negative tweets, respectively.

Table 8 gives the results for tweet sentiment classification giving evaluation parameters for SVM, Stochastic Gradient Descent (SGD), RF, LR, and multinomial naïve Bayes (MNB) classifier. Among the classifiers, the SVM has the highest accuracy of 70.56 percent for the Twitter dataset used in the experiment.

Figure 11 is a graphical representation of the data in Table 8. The *Y*-axis represents the values of the performance measures discovered during the tests, while the *X*-axis

RETRACTED

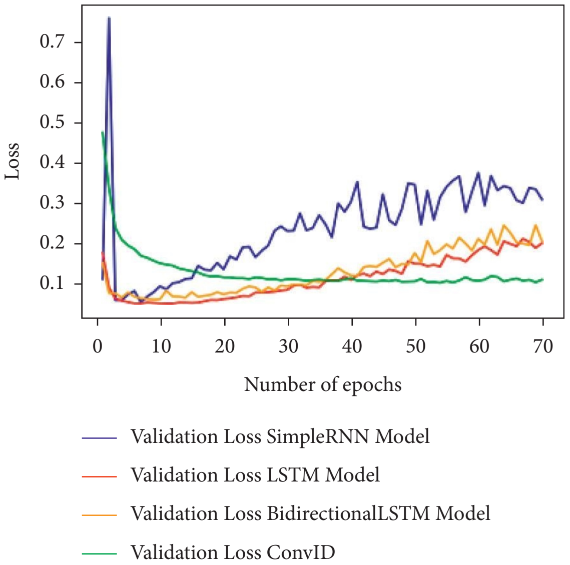
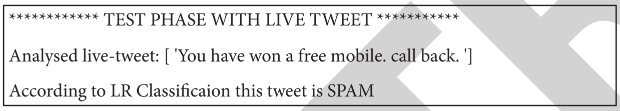


Figure 5: Validation loss for deep learning models.

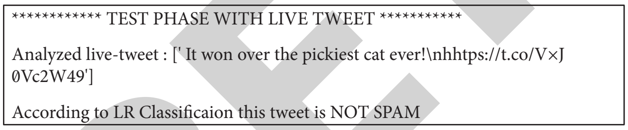
Table 7: Confusion matrix obtained for predicting real-time tweets as spam or ham.

Actual	Predicted	
	Ham	Spam
Ham	1918	4
Spam	44	263



```
************ TEST PHASE WITH LIVE TWEET ***********
Analysed live-tweet: [ 'You have won a free mobile. call back. ']
According to LR Classificaion this tweet is SPAM
```

Figure 6: Live tweet spam prediction test.



```
************ TEST PHASE WITH LIVE TWEET ***********
Analyzed live-tweet : [' It won over the pickiest cat ever!\nhhttps://t.co/V×J
0Vc2W49']
According to LR Classificaion this tweet is NOT SPAM
```

Figure 7: Live tweet not spam (ham) prediction test.

represents the Classifier names. To differentiate between the various performance measures, the lines plotted depicting the comparison are color-coded.

4.4. Deep Learning Techniques for Twitter Sentiment Analysis. The proposed work used four deep learning models for Twitter sentiment analysis, namely, Simple RNN, LSTM, BiLSTM, and 1D CNN model. Table 9 gives the validation accuracy, validation loss, test accuracy, and test loss obtained for Twitter sentiment analysis using various deep learning models.

Figure 12 shows the validation accuracy graph for the above-mentioned deep learning techniques for Twitter sentiment analysis over 70 epochs. Figure 13 shows the validation loss for the above-mentioned deep learning techniques for Twitter sentiment analysis over 70 epochs.

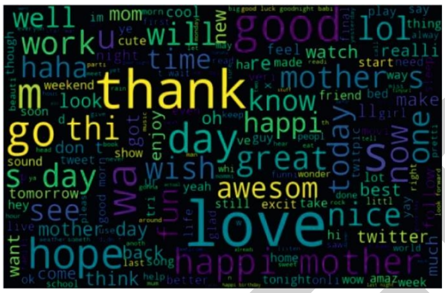


Figure 8: Positive sentiment WordCloud.

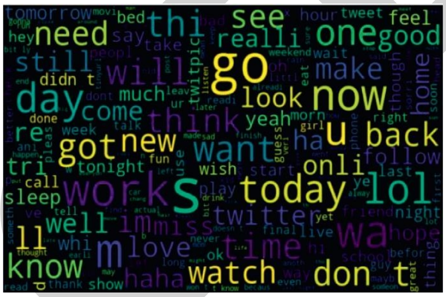


Figure 9: Neutral sentiment WordCloud.

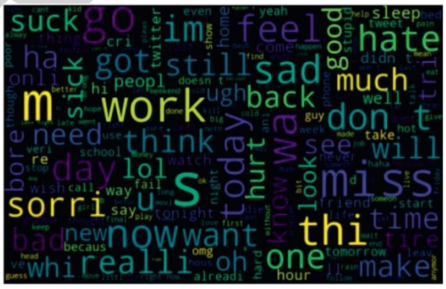


Figure 10: Negative sentiment WordCloud.

Table 8: Performance measures (in percentage) for different tweet sentiment classification models.

Performance measures	SVM	SGD	RF	LR	MNB
Classification accuracy	70.56	69.91	69.76	69.16	65.39
Sensitivity (recall)	69.23	69.38	69.32	68.61	64.91
F1-score	70.23	69.87	69.62	69.17	65.48
Precision	72.99	70.78	70.28	70.16	66.55

To demonstrate the live tweet sentiment analysis, the proposed system extracted 39 tweets for a request of a maximum of 50 tweets on the topic of India for analysis, as shown in Figure 14.

The extracted tweets were subjected to preprocessing steps and then each tweet was analyzed for sentiment using SVM as our classifier and then the sentiment generated was saved in a new data frame. The sample sentiment values for five live tweets are displayed and shown in Figure 15.

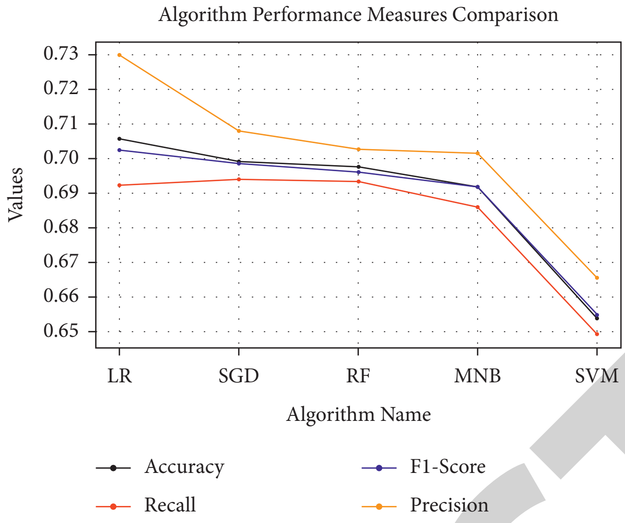


Figure 11: Comparison of performance measures for different tweet sentiment classification models.

Table 9: Evaluation parameter values obtained for Twitter sentiment analysis using deep learning models.

Deep learning models	Validation accuracy	Validation loss	Test accuracy	Test loss
Simple RNN	0.5761	1.77	0.576	1.771
LSTM	0.7381	0.6974	0.728	0.696
Bidirectional LSTM	0.7374	0.6845	0.73	0.718
1D CNN	0.3968	1.01	0.397	1.01

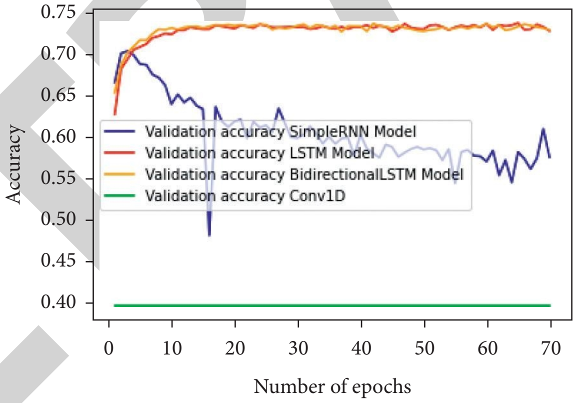


Figure 12: Validation accuracy for deep learning models.

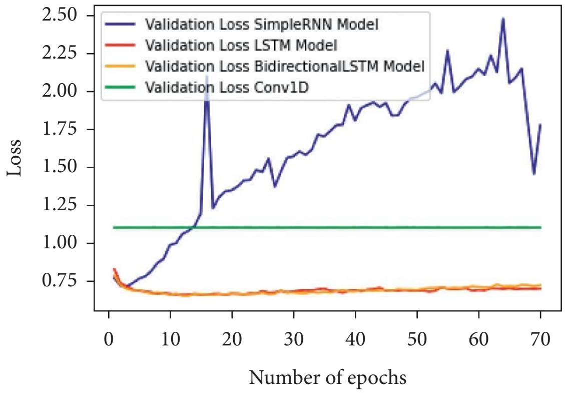
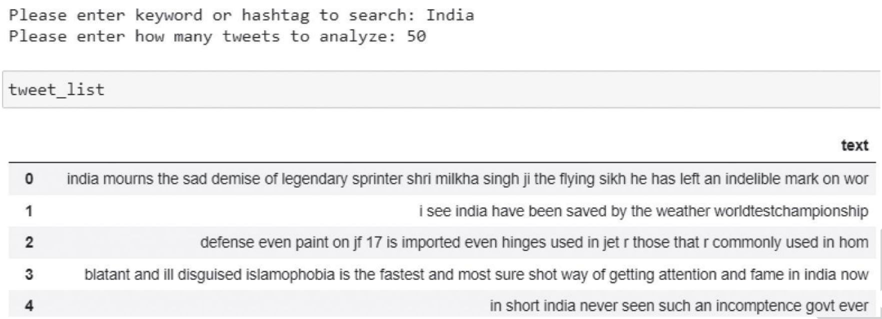


Figure 13: Validation loss for deep learning models.

RETRACTED

RETRACTED

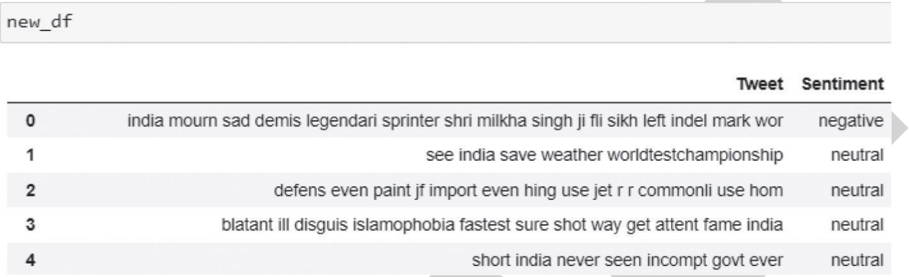


```
Please enter keyword or hashtag to search: India
Please enter how many tweets to analyze: 50

tweet_list

                                                                                                   text
0  india mourns the sad demise of legendary sprinter shri milkha singh ji the flying sikh he has left an indelible mark on wor
1                                                                  i see india have been saved by the weather worldtestchampionship
2                                          defense even paint on jf 17 is imported even hinges used in jet r those that r commonly used in hom
3                          blatant and ill disguised islamophobia is the fastest and most sure shot way of getting attention and fame in india now
4                                                                           in short india never seen such an incomptence govt ever
```

Figure 14: Sample output obtained for extraction of live tweets for sentiment analysis.



```
new_df

                                                                     Tweet  Sentiment
0  india mourn sad demis legendari sprinter shri milkha singh ji fli sikh left indel mark wor   negative
1                                          see india save weather worldtestchampionship    neutral
2                                 defens even paint jf import even hing use jet r r commonli use hom    neutral
3                         blatant ill disguis islamophobia fastest sure shot way get attent fame india    neutral
4                                                   short india never seen incompt govt ever    neutral
```

Figure 15: Sentiment values for a sample of five live tweets.

Table 10: Live tweet sentiment classification details.

Sentiment class	Total live tweets	Percentage
Neutral	27	69.23
Negative	9	23.08
Positive	3	7.69

The number of positive, neutral, and negative tweets found in our extracted tweets are presented in Table 10.

5. Conclusion and Future Work

This research article focuses on detecting real-time Twitter spam tweets and performing sentiment analysis on stored tweets and real-time live tweets. The proposed methodology has used two different datasets, one for spam detection and the other for sentiment analysis. We have applied different vectorization techniques and compared the results. This will enable the researchers to choose the best vectorization technique based on the dataset available. The spam detection and sentiment analysis on the static dataset and real-time live tweets is performed by applying various machine learning and deep learning algorithms. The multinomial naïve Bayes classifier achieved a classification accuracy of 97.78% and the deep learning model, namely, LSTM, achieved a validation accuracy of 98.74% for the Twitter spam classification. The classification process demonstrated that the features retrieved from tweets can be utilized to reliably determine whether a tweet is spam or not. The classification results revealed that the features retrieved from tweets can be used to accurately determine the Sentiment Value of tweets. The SVM classifier achieved a classification accuracy of 70.56% and the deep learning model, namely, LSTM, achieved a validation accuracy of 73.81% for the Twitter sentiment analysis.

Our future work will mainly dwell on the connection between accounts and their tendency to give out spam tweets. When we classify a tweet as spam, we can also analyze the tweets from the same account and find out how likely the given account writes out spam tweets. Another clue on whether a given account is spam can be found by analyzing the followers to following ratio. If they have a low number of followers to their following numbers, they can also reasonably be classified as spam accounts. Since spam tweets are mostly neutral and have no relevance to any of the key topics. We also would find insight into determining the sentiments of spam tweets.

Data Availability

We obtained the dataset from Kaggle that was used for our training purposes

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. K. Rawat and S. Sharma, “A real time spam classification of twitter data with comparative analysis of classifiers,” *IJSTE - International Journal of Science Technology & Engineering*, vol. 2, p. 12, 2016.

[2] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, “A framework for real-time spam detection in Twitter,” in *Proceedings of the 2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, pp. 380–383, IEEE, Bengaluru, India, 3-7 Jan. 2018.