Applied Bionics and Biomechanics

Advances in Computing and Engineering for Bionics and Medical Applications

Lead Guest Editor: Fahd Abd Algalil
Guest Editors: Mohammed Yahya Alzahrani, Cinzia Amici, and Alberto Borboni

Advances in Computing and Engineering for Bionics and Medical Applications

Applied Bionics and Biomechanics

Advances in Computing and Engineering for Bionics and Medical Applications

Lead Guest Editor: Fahd Abd Algalil
Guest Editors: Mohammed Yahya Alzahrani, Cinzia Amici, and Alberto Borboni

Contents

Retracted: Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)
Applied Bionics and Biomechanics
Retraction (1 page), Article ID 9759506, Volume 2023 (2023)

Retracted: Adaptive Anomaly Detection Framework Model Objects in Cyberspace
Applied Bionics and Biomechanics
Retraction (1 page), Article ID 9819236, Volume 2023 (2023)

Retracted: An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning
Applied Bionics and Biomechanics
Retraction (1 page), Article ID 9808790, Volume 2023 (2023)

Retracted: Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets
Applied Bionics and Biomechanics
Retraction (1 page), Article ID 9816851, Volume 2023 (2023)

[Retracted] Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model
Suzan Cansel Dogru and Yunus Ziya Arslan
Research Article (9 pages), Article ID 5593037, Volume 2021 (2021)

[Retracted] Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances
Jyotindra Narayan and Santosha K. Dwivedy
Research Article (20 pages), Article ID 5573041, Volume 2021 (2021)

[Retracted] Development of High Accuracy Classifier for the Speaker Recognition System
Raghad Tariq Al-Hassani, Dogu Cagdas Atilla, and Çağatay Aydin
Research Article (10 pages), Article ID 5559616, Volume 2021 (2021)

[Retracted] Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets
Saleh Nagi Alsubari, Sachin N. Deshmukh, Mosleh Hmoud Al-Adhaileh, Fawaz Waselalla Alsaade, and Theyazn H. H. Aldhyani
Research Article (11 pages), Article ID 5522574, Volume 2021 (2021)

[Retracted] Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models
Changwei Zhou, Lili Zhang, Yuanbo Wu, Xiaojun Zhang, Di Wu, and Zhi Tao
Research Article (11 pages), Article ID 6662625, Volume 2021 (2021)

Contents

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9856462, 1 page
https://doi.org/10.1155/2023/9856462

Retraction

Retracted: Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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] J. Narayan and S. K. Dwivedy, "Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5573041, 20 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9798010, 1 page
https://doi.org/10.1155/2023/9798010

Retraction

Retracted: Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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] C. Zhou, L. Zhang, Y. Wu, X. Zhang, D. Wu, and Z. Tao, "Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6662625, 11 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9764916, 1 page
https://doi.org/10.1155/2023/9764916

Retraction

Retracted: Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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. C. Dogru and Y. Z. Arslan, "Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5593037, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9848196, 1 page
https://doi.org/10.1155/2023/9848196

Retraction

Retracted: Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier

Applied Bionics and Biomechanics

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

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] Z. N. Ali, I. Askerzade, and S. Abdulwahab, "Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6670316, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9846843, 1 page
https://doi.org/10.1155/2023/9846843

Retraction

Retracted: Classification of Lactate Level Using Resting-State EEG Measurements

Applied Bionics and Biomechanics

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

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. A. Shaban, O. N. Ucan, and A. D. Duru, "Classification of Lactate Level Using Resting-State EEG Measurements," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6662074, 8 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9796728, 1 page
https://doi.org/10.1155/2023/9796728

Retraction

Retracted: The Use of Hellinger Distance Undersampling Model to Improve the Classification of Disease Class in Imbalanced Medical Datasets

Applied Bionics and Biomechanics

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

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] Z. Z. R. Al-Shamaa, S. Kurnaz, A. D. Duru, N. Peppa, A. H. Mirnezami, and Z. Z. R. Hamady, "The Use of Hellinger Distance Undersampling Model to Improve the Classification of Disease Class in Imbalanced Medical Datasets," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8824625, 10 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9782627, 1 page
https://doi.org/10.1155/2023/9782627

Retraction

Retracted: Development of High Accuracy Classifier for the Speaker Recognition System

Applied Bionics and Biomechanics

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

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] R. T. Al-Hassani, D. C. Atilla, and Ç. Aydin, "Development of High Accuracy Classifier for the Speaker Recognition System," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5559616, 10 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9762945, 1 page
https://doi.org/10.1155/2023/9762945

Retraction

Retracted: Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network

Applied Bionics and Biomechanics

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

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] F. E. K. Al-Khuzaie, O. Bayat, and A. D. Duru, "Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6690539, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9762024, 1 page
https://doi.org/10.1155/2023/9762024

Retraction

Retracted: The Optimal Adaptive-Based Neurofuzzy Control of the 3-DOF Musculoskeletal System of Human Arm in a 2D Plane

Applied Bionics and Biomechanics

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

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. Valizadeh and A. A. Akbari, "The Optimal Adaptive-Based Neurofuzzy Control of the 3-DOF Musculoskeletal System of Human Arm in a 2D Plane," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5514693, 10 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9761657, 1 page
https://doi.org/10.1155/2023/9761657

Retraction

Retracted: Water Quality Prediction Using Artificial Intelligence Algorithms

Applied Bionics and Biomechanics

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

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] T. H. H. Aldhyani, M. Al-Yaari, H. Alkahtani, and M. Maashi, "Water Quality Prediction Using Artificial Intelligence Algorithms," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 6659314, 12 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9759506, 1 page
https://doi.org/10.1155/2023/9759506

Retraction

Retracted: Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)

Applied Bionics and Biomechanics

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

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] A. M. A. Mohamed, O. N. Uçan, O. Bayat, and A. D. Duru, "Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8853238, 10 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9819236, 1 page
https://doi.org/10.1155/2023/9819236

Retraction

Retracted: Adaptive Anomaly Detection Framework Model Objects in Cyberspace

Applied Bionics and Biomechanics

Received 15 August 2023; Accepted 15 August 2023; Published 16 August 2023

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] H. Alkahtani, T. H. H. Aldhyani, and M. Al-Yaari, "Adaptive Anomaly Detection Framework Model Objects in Cyberspace," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 6660489, 14 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9808790, 1 page
https://doi.org/10.1155/2023/9808790

Retraction

Retracted: An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning

Applied Bionics and Biomechanics

Received 15 August 2023; Accepted 15 August 2023; Published 16 August 2023

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. M. A. Alkadhmi, O. N. Uçan, and M. Ilyas, "An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8888904, 13 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9816851, 1 page
https://doi.org/10.1155/2023/9816851

Retraction

Retracted: Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets

Applied Bionics and Biomechanics

Received 31 January 2023; Accepted 31 January 2023; Published 5 February 2023

Applied Bionics and Biomechanics has retracted the article titled "Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets" [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process and the article is being retracted with the agreement of the Chief Editor.

The authors do not agree to the retraction.

References

[1] S. N. Alsubari, S. N. Deshmukh, M. H. Al-Adhaileh, F. W. Alsaade, and T. H. 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.

[2] L. Ferguson, "Advancing Research Integrity Collaboratively and with Vigour," 2022, https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9764916, 1 page
https://doi.org/10.1155/2023/9764916

Retraction

Retracted: Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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. C. Dogru and Y. Z. Arslan, "Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5593037, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 5593037, 9 pages
https://doi.org/10.1155/2021/5593037

Research Article

Effect of Model Parameters on the Biomechanical Behavior of the Finite Element Cervical Spine Model

Suzan Cansel Dogru [1] **and Yunus Ziya Arslan** [2]

[1] *Department of Mechanical Engineering, Faculty of Engineering, Istanbul University-Cerrahpasa, Turkey*
[2] *Department of Robotics and Intelligent Systems, Institute of Graduate Studies in Science and Engineering, Turkish-German University, Turkey*

Correspondence should be addressed to Yunus Ziya Arslan; yunus.arslan@tau.edu.tr

Received 4 February 2021; Revised 11 May 2021; Accepted 14 June 2021; Published 28 June 2021

Academic Editor: Fahd Abd Algalil

Finite element (FE) models have frequently been used to analyze spine biomechanics. Material parameters assigned to FE spine models are generally uncertain, and their effect on the characterization of the spinal components is not clear. In this study, we aimed to analyze the effect of model parameters on the range of motion, stress, and strain responses of a FE cervical spine model. To do so, we created a computed tomography-based FE model that consisted of C2-C3 vertebrae, intervertebral disc, facet joints, and ligaments. A total of 32 FE analyses were carried out for two different elastic modulus equations and four different bone layer numbers under four different loading conditions. We evaluated the effects of elastic modulus equations and layer number on the biomechanical behavior of the FE spine model by taking the range of angular motion, stress, and strain responses into account. We found that the angular motions of the one- and two-layer models had a greater variation than those in the models with four and eight layers. The angular motions obtained for the four- and eight-layer models were almost the same, indicating that the use of a four-layer model would be sufficient to achieve a stress value converging to a certain level as the number of layers increases. We also observed that the equation proposed by Gupta and Dan (2004) agreed well with the experimental angular motion data. The outcomes of this study are expected to contribute to the determination of the model parameters used in FE spine models.

1. Introduction

Due to the ethical concerns and the requirement of invasive methods, determining the in vivo stress and strain values that occur on the vertebrae under different loading conditions is challenging [1]. The finite element-based computational modeling and simulation approach provide a practical and efficient solution to this problem. By using finite element (FE) analysis, it is possible to simulate the biomechanical behavior of the spinal components and calculate various biomechanical parameters such as stress, strain, and angular motion noninvasively [2–4]. Most of the FE models of the spine are based on computed tomography (CT) data [5]. In the literature, the relationships between the Hounsfield Unit (HU), which is a dimensionless unit used in CT, density, and modulus of elasticity of the anatomical structures were defined through various empirical equations [6–9]. In many FE-based studies, the values of the elastic modulus were calculated by using these equations [10–12], while very few of them focused on the spinal region [9, 13, 14]. The reliability of these equations is still controversial, and a consensus on the use of these equations has not yet been reached [15].

In the literature, the effect of assigned material properties of the ligament, intervertebral disc, and bone tissue on FE analysis results was investigated, but the effect of the number of layers of bone tissues with different material parameters based on HU level was not investigated [16–18]. Kumaresan et al. [17] analyzed the sensitivity of the output of the FE analysis of cervical spinal components including the intervertebral disc, posterior elements, endplates, ligaments, and cortical and cancellous bones to variations in the assigned material properties. They considered the angular motion,

intervertebral disc stress, endplate stress, and vertebrae stress as the output of the analysis. They concluded that the effect of changes in material properties of the soft tissues was more determinant than that in the material properties of the bone [17]. However, bone tissue is the main load-bearing structure in the human body, and the mechanical composition of bone tissue should be accurately modeled in a FE analysis to obtain reliable stress and strain levels [19]. In the literature, models were typically separated into three or four layers that are representing vertebral body structures such as the endplates and cortical and cancellous bones [17, 20]. In these studies, the distinction between these tissues was not made considering the level of HU. Rayudu et al. [21] predicted the elastic modulus by taking into account the level of HU, and a vertebra model was built up with nine layers. There is still little knowledge about the required number of layers to be defined for a FE vertebrae model to reflect the actual biomechanical behavior of the bone tissue. Moreover, the question of how sensitive the stress and strain results obtained from FE analyses are to variations in layer numbers and assigned material parameters of the vertebrae has not been answered yet.

In this study, we aimed to analyze the effect of model parameters on the range of motion, stress, and strain responses of a FE cervical spine model. More specifically, we focused on (i) how many layers would be required to obtain an accurate model and (ii) which of the two widely used HU-elastic modulus relationships in the literature would provide a more accurate result in terms of joint range of motion. The FE spinal unit included C2 and C3 vertebrae, intervertebral discs, ligaments, and facets at the same level.

2. Materials and Methods

2.1. Modeling of the Spinal Unit. CT data of a 55-year-old male cadaver was processed to create the vertebrae model. CT images of the nonpathological cervical spine were obtained from the archive of the Istanbul University-Cerrahpasa, Turkey. The pixel size and slice thickness of the tomography data were 0.49 mm and 0.63 mm, respectively. The spinal functional unit was modeled between the C2 and C3 vertebrae, including the intervertebral disc, ligaments, and facets at the same level. Four different FE vertebrae models were created separately, which were formed from one, two, four, and eight layers depending on HU values (Table 1) [22]. The vertebrae model comprising one layer was defined as a homogenized bone. In the two-layer model, one bone layer was defined for each of the cancellous and cortical bones. In the four-layer model, two bone layers were defined for each of the cancellous and cortical bones. In the eight-layer model, the first three layers were represented as the cancellous bone and the remaining layers were represented as the cortical bone [23]. Bonemat software (Bonemat, BO, Italy) was used to build up the models depending on CT data.

All vertebrae models were assumed to have linearly elastic and isotropic behavior, and their mechanical properties were described by elastic modulus and Poisson's ratio [24]. The material properties for each layer were calculated by taking HU and related density values into account. To do so, the two most common empirical equations for relating HU-elastic modulus in the literature were used to obtain the elastic modulus of the modeled bone tissues [25, 26]. In both studies, CT data were taken from bone samples and compressive force was applied to them. The stress-strain curves were plotted to calculate the elastic modulus. In addition, tissue density measurements were measured. The empirical equations were established between HU, density, and elastic modulus.

Table 1: Generated finite element vertebrae models consisted of different layer numbers depending on HU values of the cortical and cancellous bone.

Range of HU values defined for the corresponding finite element vertebrae models			
One-layer model	Two-layer model	Four-layer model	Eight-layer model
148-1988	148-661 662-1988	148-300 301-661 662-1300 1301-1988	148-300 301-500 501-661 662-900 901-1100 1101-1300 1301-1600 1601-1988

Equation (1) was used for the relationships between HU and density (ρ) expressed in kg/m^3 [25, 26]:

$$\rho = 0.44 \times \mathrm{HU} + 527. \tag{1}$$

Gupta and Dan [25] proposed the following equation set (equation (2)) to establish the relationships between density (ρ) (kg/m^3) and elastic modulus (E) (MPa):

$$\begin{aligned} E &= 3 \times 10^{-6} \times \rho^3 \quad \text{for } 350 < \rho \leq 1800, \\ E &= 1049.25 \times 10^{-6} \times \rho^2 \quad \text{for } \rho \leq 350. \end{aligned} \tag{2}$$

Morgan et al. [26] reported the following equation for the relationship between ρ (kg/m^3) and E (MPa):

$$E = 4730 \times \left(\frac{\rho}{1000}\right)^{1.56}. \tag{3}$$

Average HU values for all bone layers were specified from CT image data, and the related elastic modulus values were calculated from the abovementioned equations. Poisson's ratio was 0.3 for all models.

The intervertebral disc was modeled to fill the space between the endplates of the vertebrae. The intervertebral disc located between the C2 and C3 vertebrae consisted of the annulus fibrosus and nucleus pulposus layers. The Mooney-Rivlin hyperelastic elements were used to model the annulus ground substance [27]. For the nucleus pulposus, the elastic modulus and Poisson's ratio were 1 MPa and 0.49, respectively, which were reported by Ruberte et al. [27] based on the experimental data available in the literature [28–30].

Anterior longitudinal, posterior longitudinal, facet capsular, supraspinous, interspinous ligaments, and ligamentum

Table 2: The force of ligaments relative to the displacement [31].

Anterior longitudinal		Posterior longitudinal		Facet capsular		Interspinous ligaments/supraspinous		Ligamentum flavum	
Force (N)	Displacement (mm)	Force (N)	Displacement (mm)	Force (N)	Displacement (mm)	Force (N)	Displacement (mm)	Force (N)	Displacement (mm)
32.5	1.24	26.8	1.02	59.5	2.02	8.6	1.38	29.2	1.71
60.8	2.46	49.5	2.12	122.8	4.00	16.9	2.74	54.9	3.37
82.4	3.63	65.0	3.13	170.2	5.92	22.7	4.12	71.9	5.10
100.3	4.78	79.8	4.23	206.5	7.99	28.8	5.55	94.5	6.68

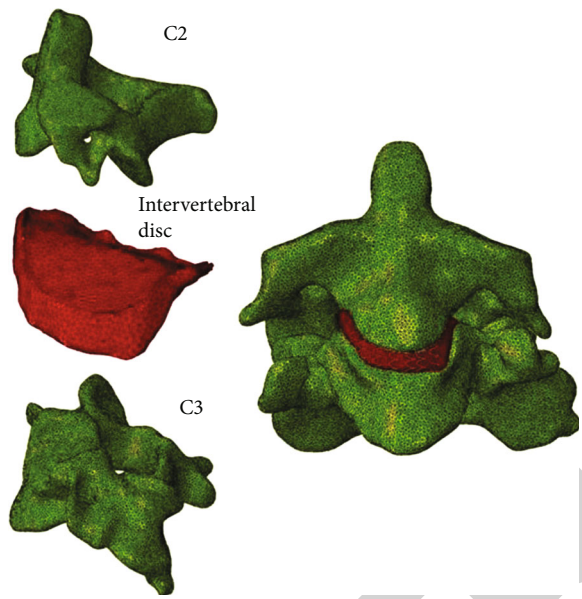


Figure 1: Finite element model of the C2 and C3 vertebrae and intervertebral disc.

flavum were represented by tension-only spring-like connectors with nonlinear material properties (Table 2). The material property of the ligaments was defined in terms of stiffness. The experimentally obtained nonlinear stiffness property was drawn from the literature [31]. Yoganandan et al. [31] measured the tensile force-displacement of all ligaments at different levels of the cervical region.

2.2. The Meshing of the Model. The hybrid quadratic tetrahedral element was chosen for the vertebrae and intervertebral disc. The criteria for creating elements influence the number of elements [32]. The geometric criteria of the hybrid quadratic tetrahedral element have been determined such that the lowest volume was 0.3 mm^3, the lowest internal angle was 10°, and the highest internal angle was 130° [33]. Sizing iterations were carried out up to the longest and shortest side length ratio of 5 and the largest and lowest volume ratio of 2 [33]. All models had 120000 elements (Figure 1).

2.3. Loading and Boundary Conditions. Flexion, extension, lateral bending, and axial rotation moments, all of which were 1.5 Nm, were applied to the models [34]. The moment value of 1.5 Nm was assumed to be sufficient to produce motion, but small enough to not injure the tissues [34]. The upper surface of the C2 vertebra was connected to the reference point, which was created in line with the adjacent vertebral body. The moment applied to the reference point was thus distributed over the upper surface of the C2 vertebra. The C3 inferior surface was fixed in all directions. The contact between the intervertebral disc and the endplate was determined to be bounded (slip and clearance were not allowed). The facet joints were coupled between the C2 and C3 vertebrae as a continuum distributing type. For all cases, the loading conditions and analysis were assumed static.

The effects of layer number and elastic modulus on the angular motion, stress, and strain values obtained from the FE analysis of the vertebrae models were compared for four different loading conditions. As a result, a total of 32 analyses (two different elastic modulus equations × four different layer numbers × four different loading conditions) were performed. All analyses were performed by using Ansys software (Ansys, Inc., Canonsburg, PA, USA).

3. Results

Angular motion results at the C2/C3 segment were given for four different layer numbers (one, two, four, and eight), two different equations, and four different loading conditions in Figure 2. The model-predicted angular motions were also compared with those obtained from the literature [35]. White and Panjabi [35] analyzed various experimental results from the literature to describe the range of angular motion of the C2-C3 vertebrae. The degree of motion of the C2-C3 vertebrae was experimentally obtained during the same loading conditions that we applied to the models [35]. It can be deduced from Figure 2 that the model-predicted angular motions under flexion, extension, and axial rotation moments were consistent with the experimental data [35]. However, the results obtained for the lateral bending moment were not in agreement with the experimental data. Angular motions obtained by using the equations by Gupta and Dan [25] and Morgan et al. [26] were found similar to each other, especially under the lateral bending moment. The number of layers was found as an effective parameter in the calculation of the angular motion, while the major differences in terms of angular motion were found between one- and two-layer models.

Maximum von Mises stress and strain values that occurred on the C2/C3 intervertebral disc were given in Figure 3. To determine whether the stress and strain values converge to a certain value as the number of layers increases,

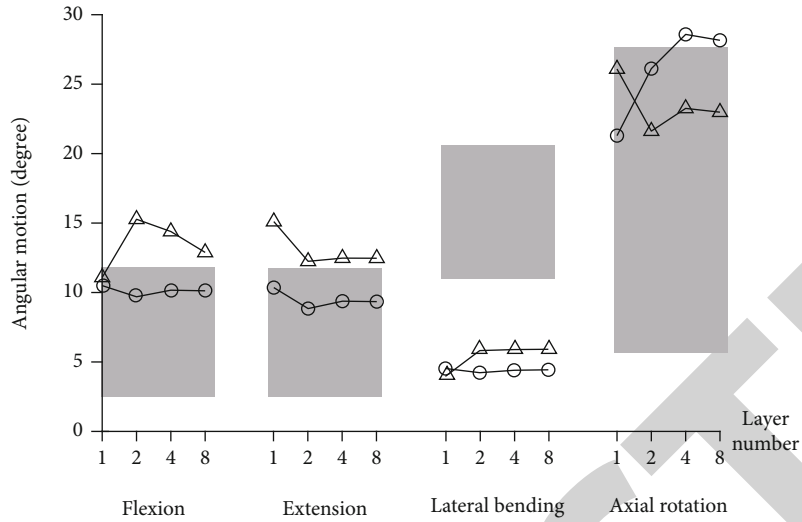


Figure 2: Angular motion at the C2/C3 joint for four different layer numbers (one-, two-, four-, and eight-layer models), two different equations (equation (2) and equation (3)), and four different loading conditions (flexion, extension, lateral bending, and axial rotation moments). The results based on equation (2) were represented by a solid line with a circle [25] and those based on equation (3) by a solid line with a triangle [26]. Experimentally obtained angular motions were represented by grey zones [35].

the stress and strain values obtained from one-, two-, and four-layer models were normalized to those obtained from eight-layer models (Figures 3(a) and 3(b)). When the equation proposed by Gupta and Dan [25] was taken into account, it was seen that the variations in the number of layers led to a 10% change in the stress and 30% in the strain values. As for the equation proposed by Morgan et al. [26], the variations in the number of layers caused a 62% change in the stress and 30% in the strain values. It was also observed that the stress and strain results obtained from four-layer models converged to those of the eight-layer models for both equations.

Table 3 and Table 4 indicate the maximum von Mises stress and strain values that occurred on the vertebrae, respectively. The results were given for each layer and all loading conditions. In Table 3 and Table 4, as the number of layers increases in the first row, the level of HU value for the corresponding layer increases. The first and second values in each cell were based on equation (1) [25] and equation (2) [26], respectively. It was observed from Table 3 that as the layer number in the vertebrae increased, the stress values also increased. In terms of strain values, an opposite trend was observed such that the strain values decreased as the HU value of the layer increased (Table 4). The difference in stress and strain values between layers decreased as the number of layers increased, indicating that more homogenous stress and strain distributions occurred over the vertebrae as the number of layers increased. The use of equation (2) and equation (3) did not result in significant differences between the maximum stress values.

4. Discussion

To deepen our understanding of the effect of various surgical interventions on the spinal components, in silico analysis of the spine provides a practical and efficient complimentary solution. Finite element-based in silico analysis has become widespread in the assessment of the spine over the last two decades [15]. The accuracy of such an analysis is critical to obtain clinically meaningful outcomes. In particular, model parameters play an important role in obtaining reliable biomechanical results from spinal FE modeling and simulation studies. The material properties of each element can be assigned depending on the HU value. On the other hand, such a methodology would lead to a high computational cost and errors due to discontinuities in the internal structures of the bone tissue. Therefore, there are many studies in the literature defining the vertebral bone into different layers, namely cortical and trabecular bone layers [36, 37]. Accordingly, we aimed to analyze the effects of variations in elastic modulus and layer number on the model-predicted angular motion, stress, and strain values that occurred at the C2/C3 level of the FE spine model under various loading modes. We also compared the model-predicted angular motion with the experimentally obtained data. The strain and stress values on the intervertebral disc and vertebrae were separately evaluated.

Angular motions under flexion and extension moments occurred in a similar range (Figure 2), and they agreed with the experimental data [35], indicating that the mechanical properties of the disc structure were well defined. The model-predicted angular motion under the lateral bending moment was lower than the experimental data, which indicates that the stiffness value assigned to the FE model in the direction of the lateral bending movement was quite high. The facet joint influences the lateral bending motion considerably [38], and hence the low level of model-predicted motion may be attributed to the uncertainty of the assigned mechanical properties of the facet joint. Under the axial rotation moment, the calculated angular motion from the FE analysis was within the range of the experimentally obtained data. In the literature, it was reported that the level of angular

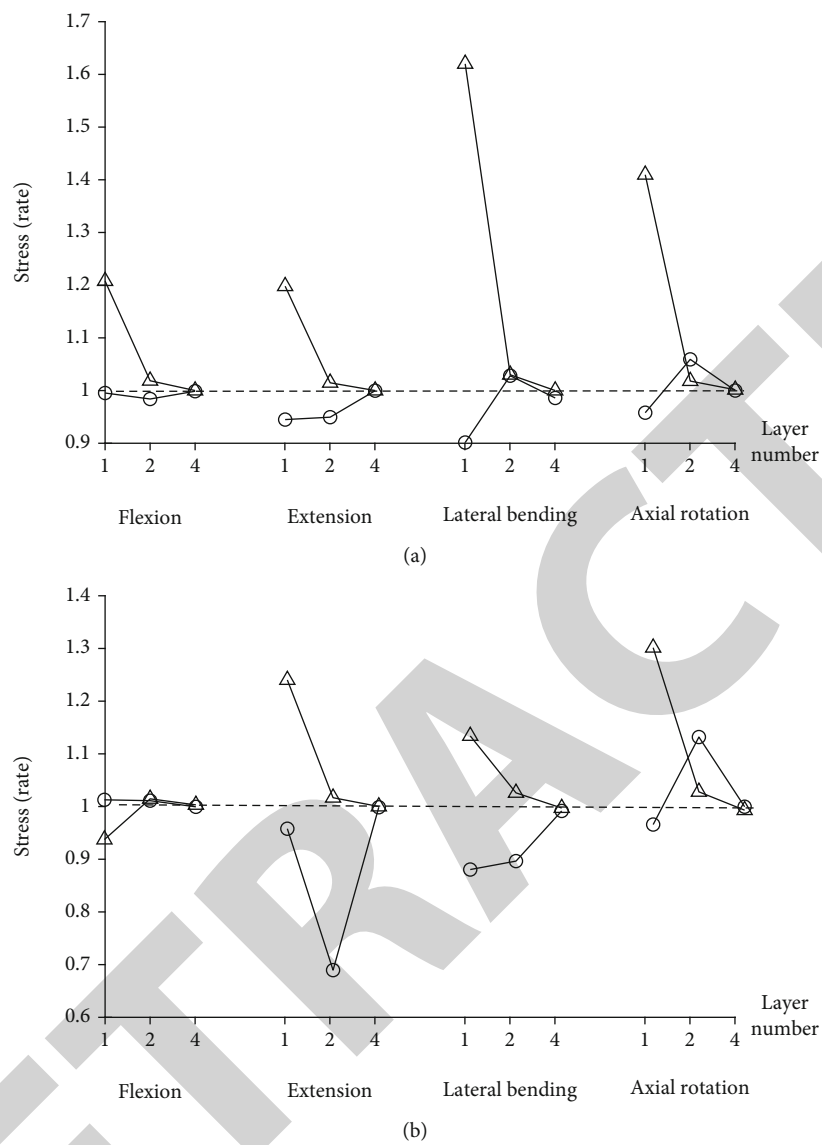


Figure 3: Maximum normalized stress (a) and strain (b) values on the C2/C3 intervertebral disc. The stress and strain values obtained from one-, two-, and four-layer models were normalized to that obtained from eight-layer models. The results based on equation (2) were represented by a solid line with a circle [25] and those based on equation (3) by a solid line with a triangle [26].

motion is highly associated with the material properties of the soft tissues [39]. The results of our study showed that the soft tissues except facets were well defined in the FE model.

Angular motions of the one- and two-layer models had a greater variation than those in the models with four and eight layers (Figure 2). This result indicates that defining the vertebrae with the cancellous bone caused a big effect on the angular motion results. Unlike the one-layer model, the two-layer model characterized the cancellous bone. It was observed that the angular motion level converged to a certain degree as the number of layers increased. The angular motions obtained for the four- and eight-layer models were almost the same, indicating that the use of a four-layer model would be sufficient to achieve the stress value converging to a certain level as the number of layers increases.

Figure 3 illustrates how many numbers of layers would be adequate to define the vertebrae. It was revealed that the change in the material properties of the vertebrae plays a decisive role in the stress and strain values over the intervertebral disc. When the stress and strain on the intervertebral disc structure are examined in Figures 3(a) and 3(b), it was observed that the results of the model using both equations with four layers converged to the results of the model with eight layers.

The difference in stress and strain distributions between the layers decreased as the number of layers increased (Table 3 and Table 4). The stress and strain results obtained by using equation (2) and equation (3) on the one-layer vertebrae model were quite similar. However, the variation in the results was greater between equation (2) and equation (3) when the number of layers increased. When compared

Table 3: The maximum von Mises stress values on the vertebrae. The first and second values in each cell are based on equation (1) [25] and equation (2) [26], respectively.

Number of layer		Maximum von Mises stress value (MPa) 1	2	3	4	5	6	7	8
Flexion	One-layer model	17.8/19.0							
	Two-layer model	12.1/20.0	27.2/25.0						
	Four-layer model	4.9/7.0	15.5/23.8	19.4/20.2	26.7/26.0				
	Eight-layer model	5.8/9.0	7.0/9.7	7.9/14.1	13.3/25.8	16.3/18.6	24.0/23.6	23.7/23.9	17.8/18.7
Extension	One-layer model	17.8/20.0							
	Two-layer model	12.1/21.0	26.7/25.4						
	Four-layer model	4.9/7.0	16.5/24.8	20.4/20.9	33.4/26.9				
	Eight-layer model	5.8/7.1	7.0/9.7	8.5/14.7	14.1/26.9	17.1/19.7	25.5/24.5	28.9/24.8	17.7/19.8
Lateral bending	One-layer model	13.5/13.8							
	Two-layer model	10.3/5.8	24.4/0.4						
	Four-layer model	4.4/7.4	13.5/15.0	15.4/12.1	24.7/23.1				
	Eight-layer model	2.1/1.9	5.3/8.1	11.3/13.4	18.3/18.1	14.2/12.0	16.3/12.9	27.5/18.9	18.8/23.1
Axial rotation	One-layer model	9.3/9.3							
	Two-layer model	5.4/10.1	12.4/11.1						
	Four-layer model	3.7/5.7	7.6/8.3	9.5/9.0	14.2/10.2				
	Eight-layer model	2.4/2.5	4.2/6.4	5.6/7.3	8.1/9.9	9.6/11.3	10.3/13.6	12.2/14.2	12.7/10.7

Table 4: The maximum strain (%) values on the vertebrae. The first and second values in each cell are based on equation (1) [25] and equation (2) [26], respectively.

Number of layer		Maximum strain value (%) 1	2	3	4	5	6	7	8
Flexion	One-layer model	1.1/1.2							
	Two-layer model	1.9/1.1	1.0/0.8						
	Four-layer model	1.1/0.4	1.6/0.9	1.0/0.6	0.3/0.5				
	Eight-layer model	0.8/0.3	1.0/0.4	0.7/0.7	0.6/0.7	0.6/0.6	0.4/0.6	0.4/0.5	0.2/0.2
Extension	One-layer model	0.8/1.0							
	Two-layer model	1.2/0.7	0.7/0.6						
	Four-layer model	1.6/0.6	1.0/0.6	0.6/0.5	0.4/0.5				
	Eight-layer model	1.1/0.7	0.6/0.4	0.9/0.6	0.9 0.6	0.7/0.6	0.6/0.5	0.4/0.5	0.4/0.2
Lateral bending	One-layer model	0.8/0.8							
	Two-layer model	1.5/0.6	0.8/0.4						
	Four-layer model	1.0/0.4	1.2/0.4	0.7/0.3	0.4/0.2				
	Eight-layer model	0.4/0.1	0.9/0.4	1.2/0.5	1.4/0.6	0.8/0.3	0.4/0.2	0.3/0.2	0.3/0.2
Axial rotation	One-layer model	0.5/0.5							
	Two-layer model	0.7/0.3	0.3/0.2						
	Four-layer model	0.7/0.3	0.6/0.3	0.3/0.2	0.3/0.2				
	Eight-layer model	0.5/0.2	0.6/0.3	0.6/0.2	0.5/0.3	0.4/0.2	0.3/0.2	0.3/0.1	0.2/0.1

to the cortical bone, the experimental mechanical testing on the vertebrae specimens showed that the strain level in the cancellous bone was higher and the stress level in the cancellous bone was lower [40, 41]. It was observed from the stress results of our study that equation (2) agreed more with the experimental data than equation (3). The fact that Gupta and Dan [25] created separate sets of equations for the cancellous and cortical bone may have led to results that are more consistent with experimental data. When the stress values of the vertebra are examined, it was observed that the stress levels in some layers decreased as the elastic modulus of the layer increased, which is not compatible with the literature [41].

In 2001, Morgan and Keaveny reported in their experimental study on cadavers that the yield stress of the cancellous bone during compression was 2.02 MPa, the yield

strain was 0.77%, the yield stress was 1.72 MPa, and the yield strain during tensile was 0.70% [40]. In addition, the cortical bone with a modulus of elasticity of 18 GPa had a yield stress of 70 MPa and a yield strain of 0.55% [41]. When these experimental values are taken into account, it was seen that the stress and strain values obtained from the models with one and two layers exceeded the experimentally obtained stress and strain values over the cancellous bone. On the other hand, the eight-layer model provided more comparable results to the experimental data. When the model-predicted stress results from the cortical bone were considered, it was seen that all stress values remained within the specified range defined by the experiments [41]. In terms of strain value, it was observed that the models with four and eight layers provided more accurate results.

Studies on the investigation of the density and elastic modulus relationship for different bone structures are very few [25, 26]. The relationship between elastic modulus and apparent density does depend on the anatomic site, which was also experimentally proven by Morgan et al. [26]. Also, the empirical relations used for defining elastic modulus are not only anatomical site specific, they can also vary between patients and computed tomography (CT) scan machine. Although these limitations are present in creating the finite element bone models, these models are most often derived from CT data in the literature [42]. Equation (2) was empirically obtained based on the CT data of the scapula [25]. On the other hand, Morgan et al. [26] carried out their experimental study on different bone structures including vertebrae. The limitation in [26] is that the authors did not provide separate equations for the cancellous and cortical bones depending on the individual bone density, but rather a single equation was developed to reflect the mechanical properties of the entire bone structure.

There are some limitations to our study. First, we only focused on the C2/C3 segment of the spine in our study since well-defined and comparable experimental data were available for this segment. However, the anatomical and functional differences available in the vertebrae can influence the obtained results. Moreover, the spine model used in our study would not represent various pathological bone conditions such as osteoporotic bone. Therefore, more FE models characterizing different bone conditions are needed to be investigated. Second, to reduce the calculation burden and provide condensed and experimentally compared results, we did not perform dynamic analysis. However, the assigned mechanical properties of the vertebrae would play a critical role under the dynamic loading of the tissue. Third, due to the lack of well-defined experimental data, all vertebrae models were assumed to have linearly elastic and isotropic behavior, which would not perfectly reflect the mechanical characteristics of the tissues. Fourth, the cadaver samples from which the experimental data were obtained and the CT data from which the FE models were created were different. However, the experimental and modeling/simulation works should be ideally carried out on the same cadaveric specimen, but that was not possible in our study due to technical limitations in terms of obtaining a fresh cadaver. And, lastly, since -FE analyses were implemented to only one sample of the vertebral body, caution should be taken when interpreting and generalizing the results obtained from our study.

5. Conclusion

We aimed to analyze the effect of model parameters on the range of motion, stress, and strain responses of the FE cervical spine model. We concluded that the angular motions of the one- and two-layer models had a greater variation than those in the models with four and eight layers. The angular motions obtained for the four- and eight-layer models were almost the same, indicating that the use of a four-layer model would be sufficient to achieve the stress value converging to a certain level as the number of layers increases. We also observed that the equation proposed by Gupta and Dan in 2004 agreed well with the experimental data because the cortical and cancellous bones were modeled separately. In the next step, we plan to investigate the effects of assigned mechanical parameters on the response of the entire spine model under dynamic loading conditions.

Data Availability

No data is available.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] M. Xu, J. Yang, I. H. Lieberman, and R. Haddas, "Lumbar spine finite element model for healthy subjects: development and validation," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 20, no. 1, pp. 1–15, 2017.

[2] M. Dreischarf, T. Zander, A. Shirazi-Adl et al., "Comparison of eight published static finite element models of the intact lumbar spine: predictive power of models improves when combined together," *Journal of Biomechanics*, vol. 47, no. 8, pp. 1757–1766, 2014.

[3] A. Shirazi-Adl, "Analysis of large compression loads on lumbar spine in flexion and in torsion using a novel wrapping element," *Journal of Biomechanics*, vol. 39, no. 2, pp. 267–275, 2006.

[4] N. Yoganandan, S. Kumaresan, and F. A. Pintar, "Biomechanics of the cervical spine. Part 2. Cervical spine soft tissue responses and biomechanical modeling," *Clinical biomechanics*, vol. 16, no. 1, pp. 1–27, 2001.

[5] B. T. Allaire, D. Lu, F. Johannesdottir et al., "Prediction of incident vertebral fracture using CT-based finite element analysis," *Osteoporosis International*, vol. 30, no. 2, pp. 323–331, 2019.

[6] S. C. Dogru, E. Cansiz, and Y. Z. Arslan, "A review of finite element applications in oral and maxillofacial biomechanics," *Journal of Mechanics in Medicine and Biology*, vol. 18, no. 2, 2018.

[7] T. S. Keller, Z. Mao, and D. M. Spengler, "Young's modulus, bending strength, and tissue physical properties of human compact bone," *Journal of Orthopaedic Research*, vol. 8, no. 4, pp. 592–603, 1990.

[8] J. Y. Rho, M. C. Hobatho, and R. B. Ashman, "Relations of mechanical properties to density and CT numbers in human bone," *Medical Engineering & Physics*, vol. 17, no. 5, pp. 347–355, 1995.

[9] J. Homminga, R. Huiskes, B. Van Rietbergen, P. Rüegsegger, and H. Weinans, "Introduction and evaluation of a gray-value voxel conversion technique," *Journal of Biomechanics*, vol. 34, no. 4, pp. 513–517, 2001.

[10] F. Taddei, E. Schileo, B. Helgason, L. Cristofolini, and M. Viceconti, "The material mapping strategy influences the accuracy of CT-based finite element models of bones: an evaluation against experimental measurements," *Medical Engineering & Physics*, vol. 29, no. 9, pp. 973–979, 2007.

[11] E. Schileo, J. Pitocchi, C. Falcinelli, and F. Taddei, "Cortical bone mapping improves finite element strain prediction accuracy at the proximal femur," *Bone*, vol. 136, 2020.

[12] W. R. Taylor, E. Roland, H. Ploeg et al., "Determination of orthotropic bone elastic constants using FEA and modal analysis," *Journal of Biomechanics*, vol. 35, no. 6, pp. 767–773, 2002.

[13] J. Homminga, H. Weinans, W. Gowin, D. Felsenberg, and R. Huiskes, "Osteoporosis changes the amount of vertebral trabecular bone at risk of fracture but not the vertebral load distribution," *Spine*, vol. 26, no. 14, pp. 1555–1560, 2001.

[14] S. C. Dogru, "Nonlinear finite element analysis of intervertebral disc: a comparative study," *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergis*, vol. 22, no. 5, pp. 1282–1287, 2018.

[15] A. C. Jones and R. K. Wilcox, "Finite element analysis of the spine: towards a framework of verification, validation and sensitivity analysis," *Medical Engineering & Physics*, vol. 30, no. 10, pp. 1287–1304, 2008.

[16] T. Zander, M. Dreischarf, A. K. Timm, W. W. Baumann, and H. Schmidt, "Impact of material and morphological parameters on the mechanical response of the lumbar spine—a finite element sensitivity study," *Journal of Biomechanics*, vol. 53, pp. 185–190, 2017.

[17] S. Kumaresan, N. Yoganandan, and F. A. Pintar, "Finite element analysis of the cervical spine: a material property sensitivity study," *Clinical biomechanics*, vol. 14, no. 1, pp. 41–53, 1999.

[18] J. D. John, G. Saravana Kumar, and N. Yoganandan, "Cervical spine morphology and ligament property variations: a finite element study of their influence on sagittal bending characteristics," *Journal of Biomechanics*, vol. 85, pp. 18–26, 2019.

[19] Y. Chevalier, D. Pahr, and P. K. Zysset, "The role of cortical shell and trabecular fabric in finite element analysis of the human vertebral body," *Journal of Biomechanical Engineering*, vol. 131, no. 11, 2009.

[20] V. K. Goel and J. D. Clausen, "Prediction of load sharing among spinal components of a C5-C6 motion segment using the finite element approach," *Spine*, vol. 23, no. 6, pp. 684–691, 1998.

[21] N. M. Rayudu, K. Subburaj, K. Mei et al., "Finite element analysis-based vertebral bone strength prediction using MDCT data: how low can we go?," *Frontiers in Endocrinology*, vol. 11, pp. 442–452, 2020.

[22] S. L. Schoell, K. M. Beavers, D. P. Beavers et al., "Prediction of lumbar vertebral body compressive strength of overweight and obese older adults using morphed subject-specific finite-element models to evaluate the effects of weight loss," *Aging Clinical and Experimental Research*, vol. 31, no. 4, pp. 491–501, 2019.

[23] M. A. Pérez, P. Fornells, M. Doblaré, and J. M. García-Aznar, "Comparative analysis of bone remodelling models with respect to computerised tomography-based finite element models of bone," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 13, no. 1, pp. 71–80, 2010.

[24] Y. Guan, N. Yoganandan, J. Zhang et al., "Validation of a clinical finite element model of the human lumbosacral spine," *Medical & Biological Engineering & Computing*, vol. 44, no. 8, pp. 633–641, 2006.

[25] S. Gupta and P. Dan, "Bone geometry and mechanical properties of the human scapula using computed tomography data," *Biomaterials and Artificial Organs*, vol. 17, no. 2, pp. 61–70, 2004.

[26] E. F. Morgan, H. H. Bayraktar, and T. M. Keaveny, "Trabecular bone modulus-density relationships depend on anatomic site," *Journal of Biomechanics*, vol. 36, no. 7, pp. 897–904, 2003.

[27] L. M. Ruberté, R. N. Natarajan, and G. B. Andersson, "Influence of single-level lumbar degenerative disc disease on the behavior of the adjacent segments—a finite element model study," *Journal of Biomechanics*, vol. 42, no. 3, pp. 341–348, 2009.

[28] S. Umehara, S. Tadano, K. Abumi, K. Katagiri, K. Kaneda, and T. Ukai, "Effects of degeneration on the elastic modulus distribution in the lumbar intervertebral disc," *Spine*, vol. 21, no. 7, pp. 811–819, 1996.

[29] J. C. Iatridis, L. A. Setton, M. Weidenbaum, and V. C. Mow, "Alterations in the mechanical behavior of the human lumbar nucleus pulposus with degeneration and aging," *Journal of Orthopaedic Research*, vol. 15, no. 2, pp. 318–322, 1997.

[30] D. M. Elliott and L. A. Setton, "Anisotropic and inhomogeneous tensile behavior of the human anulus fibrosus: experimental measurement and material model predictions," *Journal of Biomechanical Engineering*, vol. 123, no. 3, pp. 256–263, 2001.

[31] N. Yoganandan, F. A. Pintar, S. Kumaresan, and A. Elhagediab, "Biomechanical assessment of human cervical spine ligaments," *SAE Transactions*, vol. 107, no. 6, pp. 2852–2861, 1998.

[32] R. H. Macneal and R. L. Harder, "A proposed standard set of problems to test finite element accuracy," *Finite Elements in Analysis and Design*, vol. 1, no. 1, pp. 3–20, 1985.

[33] J. R. Shewchuk, "What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures (preprint)," *University of California at Berkeley*, vol. 2002, 2002.

[34] M. Panjabi, J. Dvorak, J. Duranceau et al., "Three-dimensional movements of the upper cervical spine," *Spine*, vol. 13, no. 7, pp. 726–730, 1988.

[35] A. A. White III and M. M. Panjabi, "The basic kinematics of the human Spine," *Spine*, vol. 3, no. 1, pp. 12–20, 1978.

[36] Y. Purushothaman, N. Yoganandan, D. Jebaseelan, H. Choi, and J. Baisden, "External and internal responses of cervical disc arthroplasty and anterior cervical discectomy and fusion: a finite element modeling study," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 106, p. 103735, 2020.

[37] L. X. Guo and W. J. Li, "Finite element modeling and static/dynamic validation of thoracolumbar-pelvic segment," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 23, no. 2, pp. 69–80, 2020.

[38] M. Kurutz, "Finite element modelling of human lumbar spine," in *Finite Element Analysis*, D. Moratal, Ed., pp. 210–236, InTechOpen, United Kingdom, 1st ed edition, 2010, Sciyo, 2010, Chapter 9.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9856462, 1 page
https://doi.org/10.1155/2023/9856462

Retraction

Retracted: Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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] J. Narayan and S. K. Dwivedy, "Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5573041, 20 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 5573041, 20 pages
https://doi.org/10.1155/2021/5573041

Research Article

Robust LQR-Based Neural-Fuzzy Tracking Control for a Lower Limb Exoskeleton System with Parametric Uncertainties and External Disturbances

Jyotindra Narayan and Santosha K. Dwivedy

Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India

Correspondence should be addressed to Jyotindra Narayan; n.jyotindra@gmail.com

Received 23 February 2021; Revised 14 April 2021; Accepted 15 May 2021; Published 12 June 2021

Academic Editor: Fahd Abd Algalil

The design of an accurate control scheme for a lower limb exoskeleton system has few challenges due to the uncertain dynamics and the unintended subject's reflexes during gait rehabilitation. In this work, a robust linear quadratic regulator-(LQR-) based neural-fuzzy (NF) control scheme is proposed to address the effect of payload uncertainties and external disturbances during passive-assist gait training. Initially, the Euler-Lagrange principle-based nonlinear dynamic relations are established for the coupled system. The input-output feedback linearization approach is used to transform the nonlinear relations into a linearized state-space form. The architecture of the adaptive neuro-fuzzy inference system (ANFIS) and used membership function are briefly explained. While varying mass parameters up to 20%, three robust neural-fuzzy datasets are formulated offline with the joint error vector and LQR control input. Thereafter, to deal with external interferences, an error dynamics with a disturbance estimator is presented using an online adaptation of the firing strength matrix. The Lyapunov theory is carried out to ensure the asymptotic stability of the coupled human-exoskeleton system in view of the proposed controller. The gait tracking results for the proposed control scheme (RLQR-NF) are presented and compared with the exponential reaching law-based sliding mode (ERL-SM) controller. Furthermore, to investigate the robustness of the proposed control over LQR control, a comparative performance analysis is presented for two cases of parametric uncertainties and external disturbances. The first case considers the 20% raise in mass values with a trigonometric form of disturbances, and the second case includes the effect of the 30% increment in mass values with a random form of disturbances. The simulation runs have shown the promising gait tracking aspects of the designed controller for passive-assist gait training.

1. Introduction

Over the last two decades, an increasing number of neurological disorders such as stroke, spinal cord injury, and Parkinson's disease have been observed in different age groups. The World Health Organization (WHO) reported "stroke" as one of the principal reasons for nearly 5 million people's fatality through 2000-2016 and the third pioneering source of debility throughout the world [1]. To address the concerns of motor functionality in the lower body caused by neurological disorders, researchers have developed many robot-based lower limb exoskeleton devices to produce therapeutic effects during walking [2, 3]. In a recent work by Kalita et al. [4], a systematic yet comprehensive review has been carried out on the state-of-the-art developments of such multijoint and single-joint exoskeleton devices for gait rehabilitation, mobility aid, and strength amplification.

A well-known treadmill-oriented exoskeleton, LOPES [5], has been developed with a 2D translatable pelvis segment, two active hip joints, and an active knee joint for lower limb rehabilitation. The system was controlled to supervise or follow the subjects using "robot-in-charge" and "subject-in-charge" modes. Bortole et al. [6] designed a 6-DOF lower limb exoskeleton for overground training of stroke subjects with a body height of 1.50-1.95 m and a body mass of 100 kg. Hsieh et al. [7] proposed a soft exoskeleton design

for preswing gait training of subjects with weak muscles, where a single actuator with a pulley-slider arrangement is used to drive the lower limb joints. The prototype is developed and clinically investigated with seven subjects. In a study on a parallel mechanism-based lower limb rehabilitation, Rastegarpanah et al. [8] investigated the performance of a 6-DOF robot by executing foot trajectories of 20 healthy subjects. Furthermore, the same prototype was tested for eight poststroke patients while carrying out three exercises, i.e., hip flexion/extension, ankle dorsiflexion/plantarflexion, and marching [9]. Aggogeri et al. [10] presented a modular and reconfigurable mechanism for rehabilitating ankle joints of different subjects. Cestari et al. [11] introduced the ATLAS exoskeleton to assist the children during flexion/extension of the hip, knee, and ankle joints. At the preliminary level, a dummy with body features of a 10-year-old child was used to test the exoskeleton system. Patané et al. [12] proposed a multijoint exoskeleton, WAKE-up, to rehabilitate the knee, ankle, and foot of the pediatric subjects with neurological disorders. The device was tested with four healthy children and three children with cerebral palsy. To amplify human endurance while carrying heavy loads, BLEEX [13] (7-DOF/limb) was developed with intelligent and adaptable robot-based strategies where linear hydraulic actuators were used for the actuation of 4-DOF. Recently, Ji et al. [14] introduced a wearable exoskeleton, SIAT-WEXv2, to support the user's waist and bones while lifting heavy objects in construction and logistic industries by providing an assistive output of 28 N.

To augment the rehabilitation devices' performance, the appropriate control schemes are designed by the researchers for executing repetitive gait movements. The control architecture required for the exoskeleton systems poses extra complexity over the conventional robotic arm control due to the sophisticated mechanical configuration, complex motion trajectory, and human involvement. The researchers, in the literature, have regarded the predefined gait tracking control as the basis of every control scheme for exoskeleton systems, where the joint movements of the lower limb could be estimated using gait analysis experiments [15, 16, 17]. Although the exoskeleton systems exploit the gait of healthy humans to replicate the same using predefined trajectory control schemes, however, in practice, they are unable to attain the proper gait trajectory because of the parametric uncertainties and external disturbances (PUEDs). Therefore, various robust control strategies have been designed to deal with the limitations of classical trajectory tracking control in lower limb exoskeleton systems [18, 19, 20, 21, 22, 23]. Ajayi et al. [18] proposed a bounded control scheme for the rehabilitation of the knee ankle joint of a user in a sitting position. The stability of the control law and convergence analysis of the gain observer is validated with the Lyapunov theory. The simulation results are presented without and with the effect of the human interaction torque. Yang et al. [19] presented a sliding mode control (SMC) scheme where a second-order command filter-aided backstepping is incorporated to avert the "explosion of complexity." Moreover, the fuzzy logic is exploited to counter the chattering issues of the control scheme during the estimation of structured and unstructured uncertainties. In another work on robust control, Long et al. [20] presented a hybrid strategy where SMC is augmented with a cerebellar model articulation controller (CMAC) to predict the motion intent of the subject. The optimized sliding surface of the SMC is estimated using the genetic algorithm to improve the effectiveness of the proposed control scheme. Liu et al. [21] introduced an event-triggered SMC for effective tracking of the reference trajectory using a lower limb exoskeleton system with PUEDs. In another work to address the model uncertainties and the unintended subject's response, Wu et al. [22] proposed an adaptive control scheme for a 3-DOF lower extremity rehabilitation device. Working on the decoupled control strategy, Sun et al. [23] designed a reduced-order adaptive fuzzy approach and implemented it on a two-link exoskeleton system for lower limb rehabilitation.

Furthermore, in recent times, robust intelligent control schemes have gain popularity to address the adverse effects of PUEDs with effective approximation features. A neural network (NN) along with a time-delay evaluation-based control scheme is proposed by Zhang et al. [24] to realize the desired gait trajectory for a simulated model of a 10-DOF exoskeleton. The performance of the designed control scheme is investigated by comparing the classical PD control scheme. Narayan and Dwivedy [25] proposed a neuro-fuzzy compensator for PID control to deal with the system's known and unknown uncertainties during passive gait rehabilitation of a human child. The controller is found to be more robust towards external disturbances over payload uncertainties. Chen et al. [26] proposed a disturbance estimator-based subject-cooperative control for a weight-reinforced active-assist rehabilitation device. They computed the interaction torques using a backpropagation neural network-aided disturbance observer and proved the stability using the Lyapunov theory. In a recent work by Han et al. [27], time-delay estimator-aided computed torque control is designed to deal with PUEDs of a lower limb exoskeleton system. Moreover, an adaptive radial basis function neural network (RBFNN) is utilized to compensate for the time-delay error.

On the other hand, few researchers have explored the optimal control, especially the linear quadratic regulator (LQR), to realize the natural gait [28, 29, 30, 31]. The LQR scheme with full-state feedback yields control measures concerning the whole body compared to PD control for every independent joint [28]. In addition to that, the relative prominence of curtailing the tracking error and minimalizing the control torque can be regulated by computing optimal values of time-varying gain based on the design parameters of a single controller. Furthermore, LQR as a linear control scheme might be exploited for nonlinear system dynamics by approximating the linear time-varying form and significantly mitigating the computational complexity involved in several nonlinear controllers. Ajjanaromvat and Parnichkun [29] proposed an iterative online learning-based LQR control scheme for a treadmill-appended exoskeleton to investigate the robustness analysis. Moreover, the proposed control scheme is aided with an adaptive iterative learning control to address tracking errors. Gupta et al. [30] presented the LQR control for lower limb exoskeleton systems by considering the 4-DOF human gait model in the Single Support Phase

(SSP). They exploited the nondominated sorting genetic algorithm to find out the optimal weighing matrix. However, the formulation work has not considered the uncertain factors in system dynamics. Castro et al. [31] proposed an integral-aided LQR (LQRi) and unknown input disturbance observer (UIO) to address external interferences of the lower limb exoskeleton system. The results of the proposed control are compared with proportional-derivative control and found to be more effective.

Although the hybrid form of sliding mode control can be considered a highly robust control strategy, chattering always affects the performance of exoskeleton systems. On the other hand, the LQR is the most optimal control scheme and lacks to resolve uncertain exoskeleton dynamics. Therefore, in this work, a new robust LQR-based neural-fuzzy control scheme is designed for the lower limb exoskeleton system with parametric uncertainties and external disturbances during passive gait rehabilitation training. The key highlights of the present work are as follows:

(i) The input-output feedback linearization approach is represented to linearize the nonlinear dynamics of the lower limb exoskeleton system

(ii) A robust offline LQR-based neural-fuzzy control scheme is designed to deal with payload uncertainties

(iii) A disturbance estimator is proposed using an online adaptation of firing strength in offline designed LQR-NF architecture

(iv) The simulation results are carried out for the RLQR-NF control scheme and compared with an exponential reaching law-based sliding mode control (ERL-SM) to track the desired gait trajectory during passive therapeutic training

(v) The robustness performance of the proposed control scheme (RLQR-NF) is investigated by varying payload parameters and inducing different forms of external disturbances

The rest of the paperwork is structured as follows. The mechanical description of the lower limb exoskeleton system and the estimation of control input parameters are presented in Section 2. In Section 3, the nonlinear dynamic relation is formulated using the Euler-Lagrange principle, and thereafter, input-output linearization of the nonlinear form is explained. Section 4 presents the concept of ANFIS architecture with the selected membership function. Section 5 describes a detailed design procedure of the proposed control strategy. In Section 6, the Lyapunov theory of stability is presented. The control results are simulated and discussed in Section 7. The complete paperwork is concluded in Section 8.

2. Mechanical Configuration of the Lower Limb Exoskeleton System

The main criteria for the mechanical design of a lower limb exoskeleton system are to ensure its strength and stability of the subject's safety. Moreover, the adaptability of the system with different heights of the subjects augments the feature of cost-effectiveness. Considering the subject's physiological safety, all possible degrees of freedom should be avoided at the initial phases of rehabilitation training. Invoking the design features, authors have designed a low-cost stand-alone module-aided lower limb exoskeleton system for pediatric rehabilitation in their previous work [32]. The CAD model of the designed exoskeleton system is shown in Figures 1(a) and 1(b). A 3-DOF multilink mechanism for each leg was intended to carry out hip flexion/extension, knee flexion/extension, and ankle dorsiflexion/plantarflexion motions. The placements of the joint actuators were made to avoid any physical interference with the subject's body. To serve subjects of different heights, a telescopic link joint arrangement was designed around the knee joint of the exoskeleton system. Moreover, a detailed structural analysis of the stand-alone module was carried for maximum loading conditions at the hip joint [32].

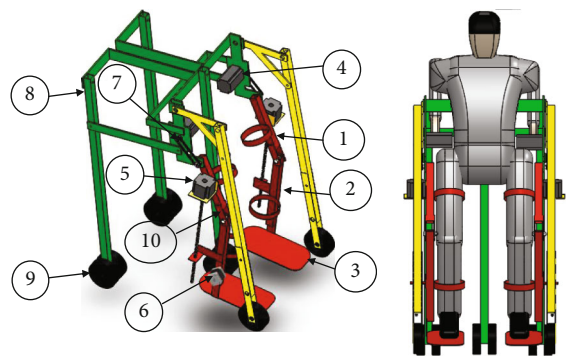
The mechanical configuration of the exoskeleton system is intended for children of 8-12 years of age, 25-40 kg weight, and 115-125 cm height. The possible range of motion (ROM) for three joints of the exoskeleton system in the sagittal plane is as follows: 30°/-12° (hip-f/e), 60°/-10° (knee-f/e), and 13°/-20° (ankle-d/p). To avoid any undesirable actions beyond the ROM, an emergency stop option is provisioned at the software interface during simulation runs. In this work, an eight-year-old male subject's anthropometric and kinematic parameters (body mass: 30 kg and body height: 1.22 m) are considered input parameters to the control architecture. The breakdown of input parameters for the lower limb exoskeleton and subject is shown in Table 1, where the length of the thigh and shank link is kept constant at 0.27 m and 0.30 m, respectively.

Furthermore, an affordable wireless Labview-aided Kinect setup was established to conduct the experimental gait analysis. With necessary approval, the child subject was asked to follow an inclined path over the ground in front of the experimental setup for 1.6-2.0 seconds. The angle estimation algorithm comprehended the information about the lower limb joint angles from the skeleton model in Labview. The angle estimation algorithm exploits the relation between joint triples using vector algebra.

The detailed procedure of performing gait analysis, as shown in Figure 2(a), is based on the work by Narayan et al. [33]. The skeleton form of the subject during the gait analysis is illustrated in Figure 2(b). The desired lower limb joint angles attained from the experiment are presented in Figure 3(a) and the corresponding trajectory in Figure 3(b). The ROM for the hip, knee, and ankle joints are recorded as 22.16° to -8.98°, 58.26° to 1.21°, and 5.84° to -7.94° for an eight-year-old child, respectively.

3. Dynamic Model of the Coupled Human-Exoskeleton System

In this section, the Euler-Lagrange principle is used to formulate the nonlinear dynamics of the coupled human-exoskeleton system. Thereafter, the input-output feedback linearization approach is exploited to linearize the nonlinear



(a) (b)

Figure 1: CAD model of (a) LLES (labels: (1) thigh link, (2) shank link, (3) foot link, (4) hybrid stepper motor, (5) lead screw actuator, (6) stepper motor, (7) timing belt, (8) support module, (9) wheels, and (10) telescopic link joint connector) and (b) LEES with a human dummy [32].

Table 1: Specifications of the lower limb exoskeleton system and child dummy.

Part	Mass (kg)	Length (m)	COM (m)
Lower limb exoskeleton system			
Thigh link	$m_1^e = 4.75$	$l_1^e = 0.25\text{-}0.30$	$l_{c1}^e = 0.12\text{-}0.15$
Shank link	$m_2^e = 1.60$	$l_2^e = 0.30\text{-}0.35$	$l_{c2}^e = 0.14\text{-}0.17$
Foot link	$m_3^e = 0.85$	$l_3^e = 0.05$	$l_{c3}^e = 0.02$
Child (age 8 years, body weight 30 kg, and body height 122 cm)			
Thigh	$m_1^h = 3.50$	$l_1^h = 0.27$	$l_{c1}^h = 0.13$
Shank	$m_2^h = 2.25$	$l_2^h = 0.30$	$l_{c2}^h = 0.15$
Foot	$m_3^h = 0.65$	$l_3^h = 0.04$	$l_{c3}^h = 0.02$

behavior of the dynamical system. The transformed linear state-space relation is established for the dynamics of the lower limb exoskeleton system.

3.1. Nonlinear Dynamic Formulation. Among various methods for expressing applied joint torques and angular acceleration, the Euler-Lagrange principle is well appreciated by the research communities [34]. Invoking the Euler-Lagrange principle, which employs kinetic and potential energy, the nonlinear representation of the coupled human-exoskeleton dynamics is obtained. A multilink structure of the coupled system with a collaboration effect is shown in Figures 4(a) and 4(b).

A generalized formulation to estimate the joint torques using the Lagrangian L is as follows:

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i}, \tag{1}$$

$$L = K - P, \tag{2}$$

where

$$K = \sum_{i=1}^{3}\left(\frac{1}{2} m_i \dot{s}_i^T \dot{s}_i + \frac{1}{2}\dot{\theta}_i^T I_i \dot{\theta}_i\right), \tag{3}$$

$$P = \sum_{i=1}^{3}(m_i g h_{ci}). \tag{4}$$

In the abovementioned relations, θ_i represents the generalized coordinate of the human-exoskeleton system. The kinetic and potential energy about the i-link is denoted by K and P, respectively. In Equation (3), $\dot{\theta}_i$, $\dot{s}_i$, m_i, and I_i signify the angular velocity, speed of the center of mass in translational direction, mass, and inertia corresponding to the i-link. The acceleration due to the gravitational effects is referred by g, and the distance between the i-link's center point forming the gravitational vector and the origin is denoted by h_{ci} as illustrated in Equation (4).

Referring to Equations (2)–(4) to solve Equation (1), the nonlinear dynamics of the coupled dynamical system can be articulated as follows:

$$\tau = M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + G(\theta), \tag{5}$$

where

$$\begin{cases} \tau = \tau_a + \tau_{eh} + \tau_{he}, \\ M(\theta) = M^e(\theta) + M^h(\theta), \\ C\left(\theta, \dot{\theta}\right) = C^e\left(\theta, \dot{\theta}\right) + C^h\left(\theta, \dot{\theta}\right), \\ G(\theta) = G^e(\theta) + G^h(\theta), \end{cases} \tag{6}$$

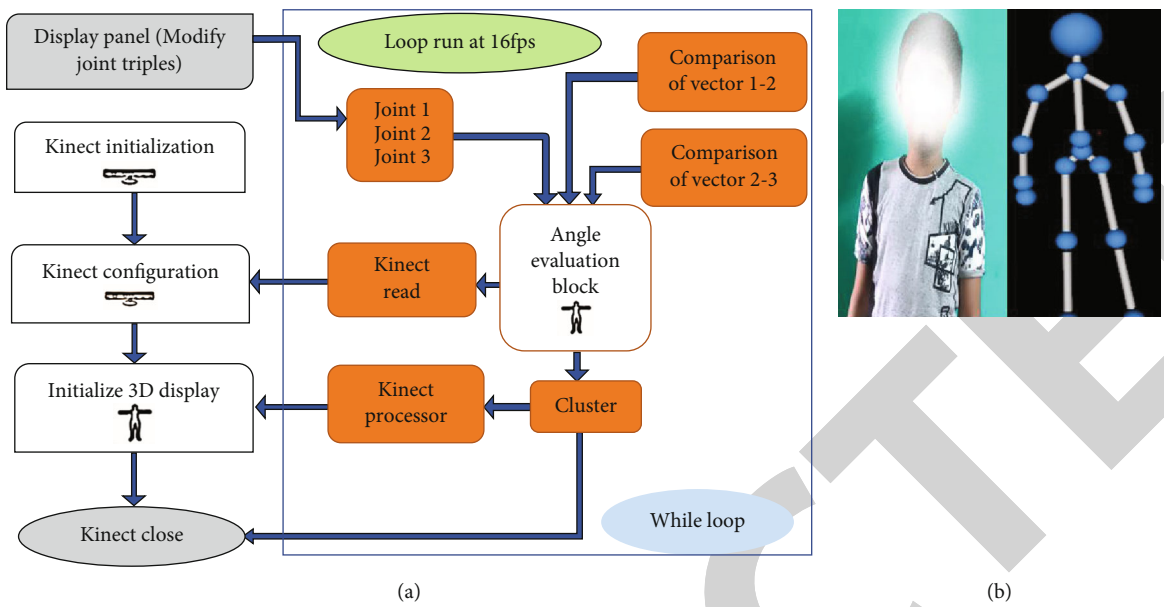


FIGURE 2: Gait analysis experiment. (a) Schematic diagram of the detailed procedure. (b) A child subject with the skeleton model during the experiment.

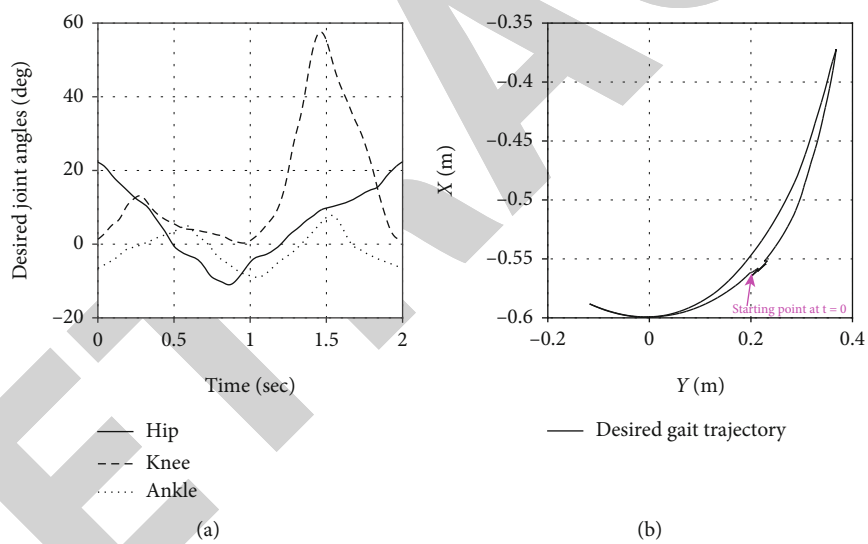


FIGURE 3: Experimental gait data. (a) Desired joint angular trajectory. (b) Desired gait trajectory.

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix},$$
$$C\left(\theta, \dot{\theta}\right) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}, \quad (7)$$
$$G(\theta) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix},$$

The matrix form of inertial, Coriolis-centrifugal, and gravity effects of the coupled dynamical system is signified by $M(\theta)$, $C(\theta, \dot{\theta})$, and $G(\theta)$, respectively. In Equation (6), $M^{e}(\theta)$ and $M^{h}(\theta)$ represent the inertial dynamics of the exoskeleton and human leg in the matrix form, respectively. The Coriolis-centrifugal matrix of the exoskeleton and human leg is represented by $C^{e}(\theta, \dot{\theta})$, and $C^{h}(\theta, \dot{\theta})$, respectively. The gravity matrix of the exoskeleton and human leg is referred by $G^{e}(\theta)$ and $G^{h}(\theta)$, respectively. τ_{a} implies the actuator torque while driving the joint of a human's lower limb. The collaboration torque is indicated by τ_{eh} and τ_{he} for collaboration of exoskeletons with humans and vice versa, respectively.

During exoskeleton-human interaction, splints are exploited to keep the exoskeleton link and human leg

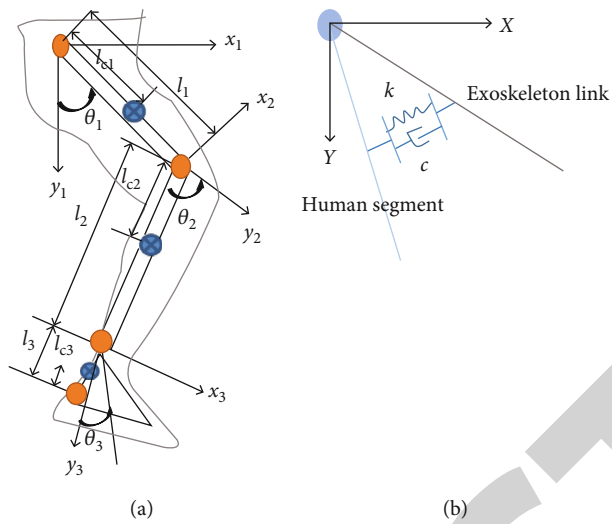


Figure 4: Coupled human-exoskeleton configuration. (a) A simplified linkage model. (b) Interaction dynamics of the coupled human-exoskeleton system.

attached, which induces the collaboration torques (τ_{eh} and τ_{he}). However, in passive gait rehabilitation, as considered in the present work, these collaborations are withdrawn by assuming rigid connections and matching joint angles for exoskeletons and humans. As illustrated in Figure 4(b), the interaction dynamics is formulated as below:

$$\tau_{eh} = -\tau_{he} = J^T f_{co} = J^T(k\Delta x + c\Delta\dot{x}) = k(\theta_h - \theta_e) + c\left(\dot{\theta}_h - \dot{\theta}_e\right), \tag{8}$$

where f_{co} represents the collaboration force between the exoskeleton and the human, k and c signify the mechanical stiffness and damping factors of the used splints, Δx denotes the Cartesian coordinate disparity between the human leg and the exoskeleton link, and $(\theta_h - \theta_e)$ refers to the joint angular disparity between the human leg and the exoskeleton link.

Furthermore, in the presence of parametric uncertainties and external disturbances, Equation (5) can be rewritten as

$$\check{\tau} = \check{M}(\theta)\ddot{\theta} + \check{C}\left(\theta, \dot{\theta}\right)\dot{\theta} + \check{G}(\theta), \tag{9}$$

where

$$\begin{cases} \check{\tau} = \eta\tau + D, \\ \check{M}(\theta) = M(\theta) + \eta M(\theta), \\ \check{C}\left(\theta, \dot{\theta}\right) = C\left(\theta, \dot{\theta}\right) + \eta C\left(\theta, \dot{\theta}\right), \\ \check{G}(\theta) = G(\theta) + \eta G(\theta), \end{cases} \tag{10}$$

where D denotes the external disturbances applied by the subjects to the system; η is the uncertain scaling factor when considering the same amount of variation in dynamic parameters.

After considering joint torques (τ) equivalent to joint actuator torques (τ_a) in case of passive rehabilitation training, the actuator dynamics can be formulated using Kirchhoff's law to obtain the control voltage ($\check{\mathbf{U}}_m$) as follows:

$$\check{\mathbf{U}}_m = \frac{\check{\tau} R_m}{S_m} + \dot{i}_m L_m + \Xi_e \dot{\theta}, \tag{11}$$

where R_m, S_m, $\dot{i}_m$, L_m, and Ξ_e denote the armature resistance, torque sensitivity, current, armature inductance, and back EMF constant of the DC motor. These parameters are selected from the specification sheet provided by Bholanath Precision Engineering Private Limited [35].

To imitate the realistic cases, the actuator saturation should be considered in the design of the control law to avoid the hysteresis cycle and maintain the linearity of the actuator. Moreover, this ensures closed-loop stability by limiting the large control signals. Based on the saturation theory, the control signal ($\check{\mathbf{U}}_m$) from Equation (11) can be further defined as below:

$$\check{\mathbf{U}}_m = \begin{cases} \mho_m & \left(\check{\mathbf{U}}_m > \mho_m\right), \\ \check{\mathbf{U}}_m & \left(\left|\check{\mathbf{U}}_m\right| \le \mho_m\right), \\ -\mho_m & \left(\check{\mathbf{U}}_m < -\mho_m\right), \end{cases} \tag{12}$$

where $\mho \in \mathbb{R}^3$ denotes a vector with positive elements. The upper and lower saturation bound is denoted by $\mho_m$ and $-\mho_m$, respectively.

3.2. Input-Output Feedback Linearization. The main objective of the feedback linearization is to correctly linearize the nonlinear dynamics with suitable modifications in state-space coordinates using an inner loop control [34]. Thereafter, an outer loop control with a new set of coordinates can be formed to establish a linear relationship between the output vector ($\check{\boldsymbol{y}}$) and the input vector ($\check{\boldsymbol{u}}$) and validate the cost

functions of the control design. Consider the nonlinear multiple-input and single-output (MISO) dynamic relation with n as the order and p as the total number of inputs as well as outputs, defined in the affine state:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{\Psi}(\boldsymbol{x}(t)) + \sum_{i=1}^{p} \Pi_i(\boldsymbol{x}(t))\check{\mathbf{U}}_{m_i}(t), \\ \boldsymbol{y}_i(t) = \Lambda_i(\boldsymbol{x}(t)), \end{cases} \tag{13}$$

where $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n$ denotes the state vector, $\check{\mathbf{U}}_{m_i} = [\check{\mathbf{U}}_{m_1}, \check{\mathbf{U}}_{m_2}, \cdots, \check{\mathbf{U}}_{m_p}]^T \in \mathbb{R}^p$ signifies the control input vector, and $\boldsymbol{y} = [y_1, y_2, \cdots, y_p]^T \in \mathbb{R}^p$ indicates the output vector.

Theorem 1. *Suppose $\boldsymbol{\Psi} : \mathbb{R}^n \Rightarrow \mathbb{R}^n$ signifies a smooth vector field on $\mathbb{R}^n$ and $\Lambda : \mathbb{R}^n \Rightarrow \mathbb{R}^n$ denotes a scalar function. Then, the Lie derivative of Λ to $\boldsymbol{\Psi}$, referred as $L_{\boldsymbol{\Psi}}\Lambda$, is expressed as follows [36, 37]:*

$$L_{\boldsymbol{\Psi}}\Lambda = \frac{\partial \Lambda}{\partial \boldsymbol{x}}\boldsymbol{\Psi}(\boldsymbol{x}) = \sum_{i=1}^{p} \frac{\partial \Lambda}{\partial x_i}\boldsymbol{\Psi}(\boldsymbol{x}). \tag{14}$$

Similarly, the Lie derivative of $L_{\boldsymbol{\Psi}}\Lambda$ with respect to $\boldsymbol{\Psi}$ is defined as

$$L_{\boldsymbol{\Psi}}^2 \Lambda = L_{\boldsymbol{\Psi}}(L_{\boldsymbol{\Psi}}\Lambda). \tag{15}$$

In general,

$$L_{\boldsymbol{\Psi}}^{\gamma} \Lambda = L_{\boldsymbol{\Psi}}\left(L_{\boldsymbol{\Psi}}^{\gamma-1}\Lambda\right) \quad \text{with } L_{\boldsymbol{\Psi}}^{0}\Lambda = \Lambda, \forall \gamma = 1, \cdots, p. \tag{16}$$

Theorem 2. *The function $\boldsymbol{\Phi} : \mathbb{R}^n \Rightarrow \mathbb{R}^n$, specified in a region $Y \subset \mathbb{R}^n$, is termed as diffeomorphism if the function $\boldsymbol{\Phi}$ along with the inverse $\boldsymbol{\Phi}^{-1}$ (if it exists) is smooth, i.e., differentiable everywhere [36, 37].*

However, as the global diffeomorphism is rare, one should check for local diffeomorphisms, i.e., transformations defined in a limited neighborhood of a specified point [37]. With the concept of diffeomorphism, we transform a nonlinear system into another one by changing the variables in the following form:

$$\boldsymbol{z} = [z_1, z_2, \cdots, z_n]^T = \boldsymbol{\Phi}(\boldsymbol{x}), \tag{17}$$

where $\boldsymbol{\Phi}(\boldsymbol{x})$ characterizes n variables as

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \Lambda_1 & L_{\boldsymbol{\Psi}}\Lambda_1 & \cdots & L_{\boldsymbol{\Psi}}^{r_1-1}\Lambda_1 \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} \Lambda_p & L_{\boldsymbol{\Psi}}\Lambda_p & \cdots & L_{\boldsymbol{\Psi}}^{r_p-1}\Lambda_p \end{bmatrix}^T \end{bmatrix}, \tag{18}$$

with $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$.

Now, a linear relationship between inputs and outputs is to be established by performing the differentiation of the outputs (y_i) till the input terms appear in the formulation. Considering r_j is the smallest integer, $(y_j^{(r_j)})$ can be evaluated with a complete term of inputs as follows:

$$y_j^{(r_j)} = L_{\boldsymbol{\Psi}}^{r_j}\Lambda_j(\boldsymbol{x}) + \sum_{i=1}^{p} L_{\boldsymbol{\Pi}_i}\left(L_{\boldsymbol{\Psi}}^{r_{j-1}}\Lambda_j(\boldsymbol{x})\right)\check{\mathbf{U}}_{m_i}, \quad i, j = 1, 2, \cdots, p, \tag{19}$$

where $L_{\boldsymbol{\Psi}}^{i}\Lambda_j$ and $L_{\boldsymbol{\Pi}}^{i}\Lambda_j$ signify the ith Lie derivatives of $\Lambda_j(\boldsymbol{x})$ in the direction of $\boldsymbol{\Psi}$ and $\boldsymbol{\Pi}$, respectively. In Equation (19), r_j denotes the relative degree for the output y_j which provides information about the number of derivatives required to carry out at least one of the inputs in the formulation [37, 38]. The sum of every relative degree from Equation (19) constitutes the total relative degree (r) which needs to be less than or equal to the system's order.

$$r = \sum_{j=1}^{n} r_j \le n. \tag{20}$$

Furthermore, rewriting Equation (19) and expressing the nonlinear control law $\check{\mathbf{U}}_m$ to form the linear relationship between the input and the output as follows, one can get

$$\left[y_1^{r_1}, \cdots, y_p^{r_p}\right]^T = \delta(\boldsymbol{x}) + \sigma(\boldsymbol{x}) \cdot \check{\mathbf{U}}_m, \tag{21}$$

$$\check{\boldsymbol{u}} = \left[\check{u}_1, \check{u}_2, \cdots, \check{u}_p\right]^T = \left[y_1^{r_1}, \cdots, y_p^{r_p}\right]^T, \tag{22}$$

where

$$\begin{cases} \delta(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{\Psi}}^{r_1}\Lambda_1(\boldsymbol{x}) \\ \vdots \\ \vdots \\ L_{\boldsymbol{\Psi}}^{r_p}\Lambda_p(\boldsymbol{x}) \end{bmatrix}, \\ \sigma(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{\Pi}_1}\left(L_{\boldsymbol{\Psi}}^{(r_{1-1})}\Lambda_1(\boldsymbol{x})\right) L_{\boldsymbol{\Pi}_2}\left(L_{\boldsymbol{\Psi}}^{(r_1-1)}\Lambda_1(\boldsymbol{x})\right) \cdots L_{\boldsymbol{\Pi}_p}\left(L_{\boldsymbol{\Psi}}^{(r_1-1)}\Lambda_1(\boldsymbol{x})\right) \\ L_{\boldsymbol{\Pi}_1}\left(L_{\boldsymbol{\Psi}}^{(r_2-1)}\Lambda_2(\boldsymbol{x})\right) L_{\boldsymbol{\Pi}_2}\left(L_{\boldsymbol{\Psi}}^{(r_2-1)}\Lambda_2(\boldsymbol{x})\right) \cdots L_{\boldsymbol{\Pi}_p}\left(L_{\boldsymbol{\Psi}}^{(r_2-1)}\Lambda_2(\boldsymbol{x})\right) \\ \vdots \quad \vdots \quad \vdots \\ L_{\boldsymbol{\Pi}_1}\left(L_{\boldsymbol{\Psi}}^{(r_p-1)}\Lambda_p(\boldsymbol{x})\right) L_{\boldsymbol{\Pi}_2}\left(L_{\boldsymbol{\Psi}}^{(r_{p-1})}\Lambda_p(\boldsymbol{x})\right) \cdots L_{\boldsymbol{\Pi}_p}\left(L_{\boldsymbol{\Psi}}^{(r_p-1)}\Lambda_p(\boldsymbol{x})\right) \end{bmatrix}. \end{cases} \tag{23}$$

Assuming $\sigma(\boldsymbol{x})$ is not singular, the input transferred form, i.e., the nonlinear control law, can be possibly defined as

$$\check{\mathbf{U}}_m = \sigma(\boldsymbol{x})^{-1}(-\delta(\boldsymbol{x}) + \check{\boldsymbol{u}}), \tag{24}$$

where $\check{\boldsymbol{u}} = [\check{u}_1, \check{u}_2, \cdots, \check{u}_p]^T$ and $\check{\mathbf{U}}_m = [\check{\mathbf{U}}_{m_1}, \check{\mathbf{U}}_{m_2}, \cdots, \check{\mathbf{U}}_{m_p}]^T$.

In Equation (24), $\check{\boldsymbol{u}}$ denotes the new input vector, $\check{\mathbf{U}}_m$ refers to the decoupling control law, $\sigma(\boldsymbol{x})$ signifies an

invertible matrix of order $p \times p$, and $\delta(\boldsymbol{x})$ represents a decoupling matrix of the system.

Furthermore, the linearizing law from Equation (24) is applied for transforming the nonlinear dynamics of the coupled human-exoskeleton system (Equation (9)) into the linear state-space representation as follows:

$$\begin{cases} \dot{\boldsymbol{z}} = \check{\mathcal{A}}\boldsymbol{z} + \check{\mathcal{B}}\check{\boldsymbol{u}}, \\ \boldsymbol{y} = \mathcal{C}\boldsymbol{z}, \end{cases} \tag{25}$$

where $\check{\mathcal{A}} = \mathcal{A} + \Delta\mathcal{A}$, $\check{\mathcal{B}} = \mathcal{B} + \Delta\mathcal{B}$, and $\check{\boldsymbol{u}} = (\Delta + 1)\boldsymbol{u} + d$.

$\check{\mathcal{A}}$ and $\check{\mathcal{B}}$ are the state-weight factor matrices with the effect of parametric uncertainties; $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ denote the state-weight factor matrices evaluated with nominal system parameters, i.e., in the absence of parametric uncertainties; Δ is the uncertain scaling factor in the linearized state-space form related to state-weight factor matrices and can be expressed in terms of dynamic parameters of the given system [39]. The input vector $\check{\boldsymbol{u}}$ carries the effect of parameter variations and external disturbances. The input vector in the absence of PUEDs is denoted by $\boldsymbol{u}$. The disturbance applied to the system after feedback linearization is denoted by d, being analogous to D from Equation (9). The linearized dynamics in Equation (25) stands valid with the following assumptions:

A1: the $(\check{\mathcal{A}}, \check{\mathcal{B}})$ is in the controllable form.

A2: the disturbance d is in the bounded form.

In the present work, the state vector and output vector related to the hip, knee, and ankle joints of the coupled dynamical system is considered $\boldsymbol{z} = [\theta_H \quad \theta_K \quad \theta_A \quad \dot{\theta}_H \quad \dot{\theta}_K \quad \dot{\theta}_A]^T$ and $\boldsymbol{y} = [\theta_H \quad \theta_K \quad \theta_A]^T$, respectively. The input vector with nominal system parameters is defined as $\boldsymbol{u} = [\boldsymbol{u}_H \quad \boldsymbol{u}_K \quad \boldsymbol{u}_A]^T$. Moreover, the respective state-weight factor matrices ($\mathcal{A} \in \mathbb{R}^{6\times6}$, $\mathcal{B} \in \mathbb{R}^{6\times3}$, and $\mathcal{C} \in \mathbb{R}^{3\times6}$) can be formulated as below. The entries of these matrices are extensively presented in the appendix.

$$\begin{cases} \mathcal{A} = \begin{pmatrix} [0]_{3\times3} & I_3 \\ \begin{matrix} \frac{\alpha_{41}}{\bar{\alpha}_{41}} & \frac{\alpha_{42}}{\bar{\alpha}_{42}} & \frac{\alpha_{43}}{\bar{\alpha}_{43}} \\ \frac{\alpha_{51}}{\bar{\alpha}_{51}} - \frac{\alpha_{51}'}{\bar{\alpha}_{51}'} + \frac{\alpha_{51}''}{\bar{\alpha}_{51}''} & \frac{\alpha_{52}}{\bar{\alpha}_{52}} - \frac{\alpha_{52}'}{\bar{\alpha}_{52}'} + \frac{\alpha_{52}''}{\bar{\alpha}_{52}''} & \frac{\alpha_{53}}{\bar{\alpha}_{53}} + \frac{\alpha_{53}'}{\bar{\alpha}_{53}'} - \frac{\alpha_{53}''}{\bar{\alpha}_{53}''} \\ \frac{\alpha_{61}}{\bar{\alpha}_{61}} - \frac{\alpha_{61}'}{\bar{\alpha}_{61}'} - \frac{\alpha_{61}''}{\bar{\alpha}_{61}''} & \frac{\alpha_{62}}{\bar{\alpha}_{62}} - \frac{\alpha_{62}'}{\bar{\alpha}_{62}'} - \frac{\alpha_{62}''}{\bar{\alpha}_{62}''} & \frac{\alpha_{63}}{\bar{\alpha}_{63}} - \frac{\alpha_{63}'}{\bar{\alpha}_{63}'} - \frac{\alpha_{63}''}{\bar{\alpha}_{63}''} \end{matrix} & [0]_{3\times3} \end{pmatrix}_{6\times6}, \\ \mathcal{B} = \begin{pmatrix} [0]_{3\times3} \\ \frac{\beta_{41}}{\bar{\beta}_{41}} \quad \frac{\beta_{42}}{\bar{\beta}_{42}} \quad \frac{\beta_{43}}{\bar{\beta}_{43}} \\ \frac{\beta_{51}}{\bar{\beta}_{51}} \quad \frac{\beta_{52}}{\bar{\beta}_{52}} \quad \frac{\beta_{53}}{\bar{\beta}_{53}} \\ \frac{\beta_{61}}{\bar{\beta}_{61}} \quad \frac{\beta_{62}}{\bar{\beta}_{62}} \quad \frac{\beta_{63}}{\bar{\beta}_{63}} \end{pmatrix}_{6\times3}, \\ \mathcal{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}_{3\times6}. \end{cases} \tag{26}$$

4. Adaptive Neural-Fuzzy Inference System

The adaptive neural-fuzzy inference system (ANFIS), colloquially known as the neural-fuzzy or neuro-fuzzy (NF) system, was proposed by Jang and Sun [40] by augmenting the benefits of adaptive neural networks and fuzzy reasoning. In the NF system, IF-THEN-based fuzzy logic inferences are constructed to form the learning rules with a defined input-output dataset and reproduce the output vector with zero error tolerance. A neural-fuzzy system exploits the fuzzy input variables and input-dependent nonfuzzy output variables given by Takagi and Sugeno [41]. For instance,

If the acceleration of the robot's end – effector is *high*, then

$$f(t) = \mathbb{c} \times \left(\text{acceleration of the robot's end} - \text{effector}\right)^2. \tag{27}$$

In Equation (27), *high* signifies a fuzzy label with the membership function (MF), representing the acceleration of the robot's end – effector in the rule proposition. The rule subsequent with nonfuzzy behavior is formulated according to the input variable of the premise step, acceleration of the robot's end – effector.

4.1. ANFIS Architecture. A neural-fuzzy system inherently exploits the five layers. The primary network architecture with two input vectors $(\varkappa : x, y)$ and one output vector $(f(\varkappa))$ is considered to show the generalized process of ANFIS. As shown in Figure 5, two kinds of nodes are employed in the architecture: first, a square node for adaptation of the parameters, and second, a circular node that behaves as a fixed node with no parameter. The layer-by-layer development of the ANFIS structure is given below [42, 43].

Layer 1. This layer acts as a conversion function for the crisp value of the input vector into an appropriate MF-based fuzzy language, depicted as follows:

$$\begin{aligned} \mathcal{O}_j^1 &= \Omega_{A_j}(x), \\ \mathcal{O}_j^1 &= \Omega_{B_j}(y), \end{aligned} \tag{28}$$

where $\mathcal{O}_j^1$ represents the jth node output for the first layer. $\Omega_{A_j}(x)$ and $\Omega_{B_j}(y)$ denote the membership weightage of respective input variables, defined for A_j- and B_j-type MF. In this work, the generalized bell membership function is chosen to replicate a proper probability distribution behavior and expressed as follows:

$$\text{bell}(\varkappa : a_j, b_j, c_j) = \Omega_{A_j}(x) = \Omega_{B_j}(y) = \frac{1}{1 + \left[\left((\varkappa - c_j)/a_j\right)^2\right]^{b_j}}. \tag{29}$$

In Equation (29), "a_j" and "b_j" signify the width and shape parameters of the membership function. The value of "b_j" is generally positive; however, it can be considered negative in case of inverted shape. "c_j" indicates the center

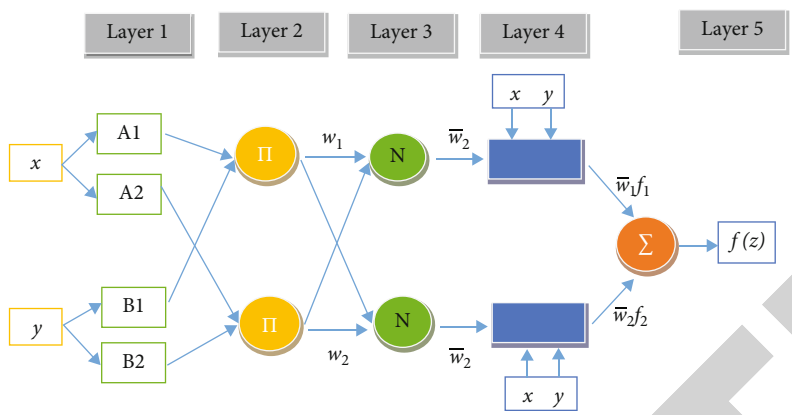


Figure 5: ANFIS architecture.

position of the membership function. Having an extra parameter compared to Gaussian MF, the generalized bell has the added advantage of tuning the steepness at cross-over positions.

Layer 2. This layer evaluates the firing strength for every rule using a product of incoming signals from each circular node. It is designated by notation **Π** in the ANFIS architecture. The following expression is used to estimate the firing strength (w_j) as follows:

$$\mathcal{O}_j^2 = w_j = \Omega_{A_j}(x) = \Omega_{B_j}(y), \quad j = 1, 2. \tag{30}$$

Layer 3. In this layer, the normalization of the node's firing strength is carried out by dividing the jth rule firing strength to all rules' total firing strength. This layer is designated by notation N in the ANFIS architecture. The firing strength ($\bar{w}_j$) is normalized as follows:

$$\mathcal{O}_j^3 = \bar{w}_j = \frac{w_j}{w_1 + w_2}, \quad j = 1, 2. \tag{31}$$

Layer 4. This layer, having the square nodes, is used to estimate the rule's involvement by defuzzification of input variables and produce the respective output as follows:

$$\mathcal{O}_j^4 = \bar{w}_j \angle_j = \bar{w}_j \left(p_j x + q_j y + r_j\right), \tag{32}$$

where $\bar{w}_j$ indicates the normalized firing strength and p_j, q_j, and r_j signify the subsequent limits.

Layer 5. This layer, having circular shape nodes with the designation $\sum$, processes the final output using the summation of all incoming signals from the preceding layer. Mathematically, it can be expressed as follows:

$$\mathcal{O}_j^5 = \sum_j \bar{w}_j \angle_j = \frac{\sum_j w_j \angle_j}{\sum_j w_j}. \tag{33}$$

5. Robust Design of the LQR-Based Neural-Fuzzy Control

The design procedure of RLQR-NF control is organized into two parts: first, the offline training of a robust LQR-based ANFIS training dataset to deal with parametric uncertainties, and second, the online training of the LQR-based ANFIS architecture using the adaptive law of weights to compensate for the external disturbances. In both parts, the effects of parametric uncertainties and external disturbances are explicitly considered. Thereafter, the stability of the proposed control strategy in the presence of PUEDs is addressed by the Lyapunov theory in the next section.

5.1. Offline Training of the RLQR-NF Dataset for Parametric Uncertainties. As shown in Figure 6, this subsection is further presented into two stages: first, the formulation of a robust LQR-NF dataset by varying mass parameters, and second, the stepwise layout of offline training of a dataset using ANFIS parameters to design the robust control strategy.

5.1.1. Stage I: Formulation of the RLQR-NF Training Dataset. The training dataset, having multiple-input and single-output (MISO), is formed by employing the concepts of the LQR control strategy as shown in Equation (25). The LQR cost function is considered a minimization problem while applying the optimality conditions and is expressed as follows [44]:

$$\mathcal{J} = \frac{1}{2}\int_0^\infty \left(\mathbf{z}^T \mathbf{Q} \mathbf{z} + \check{\mathbf{u}}^T \mathbf{R} \check{\mathbf{u}}\right) dt, \tag{34}$$

where $\mathbf{Q}$ and $\mathbf{R}$ denote the user-defined state-weight matrix and control cost matrix, respectively. An appropriate selection of both matrices directly influences the performance characteristics of the controller.

The generalized input ($\check{\mathbf{u}}$) to the control system is articulated by regulating the error vector (e) as follows:

$$\check{\mathbf{u}} = \mathcal{K} e = \mathcal{K}\left(\mathbf{z}_{\text{des}} - \mathbf{z}\right), \tag{35}$$

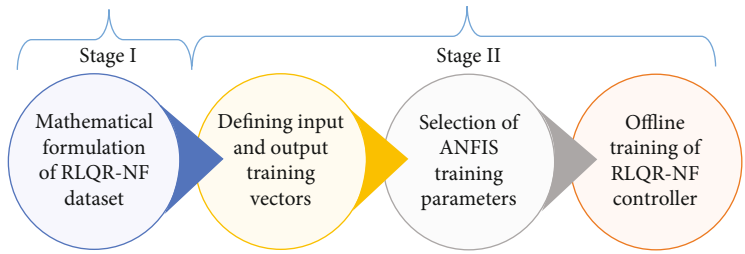


Figure 6: Flowchart representation of the stage-wise design procedure.

where $\mathscr{z}_{\text{des}}$ and $\mathscr{z}$ represent the desired and actual state vectors, respectively. The optimal state gain matrix ($\mathscr{K}$) can be expressed in terms of the control cost matrix ($\mathbf{R}$) and the state-weight factor matrix ($\check{\mathscr{B}}$) as follows:

$$\mathscr{K} = \mathbf{R}^{-1}\check{\mathscr{B}}^{T}\mathscr{P}, \tag{36}$$

where $\mathscr{P}$ is the answer for the algebraic form of the Riccati equation expressed in the form of state-weight factor matrices ($\check{\mathscr{A}}$ and $\check{\mathscr{B}}$), state-weight matrix ($\mathscr{Q}$), and control cost matrix ($\mathbf{R}$) as given below [44, 45]:

$$\mathscr{P}\check{\mathscr{A}} + \check{\mathscr{A}}^{T}\mathscr{P} - \mathscr{P}\check{\mathscr{B}}\mathbf{R}^{-1}\check{\mathscr{B}}^{T}\mathscr{P} + \mathscr{Q} = \mathbf{0}. \tag{37}$$

It is truly evident from Equation (36) that the state gain matrix ($\mathscr{K}$) is regulated by $\check{\mathscr{A}}$, $\check{\mathscr{B}}$, $\mathscr{Q}$, and $\mathbf{R}$ matrices where $\check{\mathscr{A}}$ and $\check{\mathscr{B}}$ are reliant on the mechanical arrangement and dynamic parameters of the requisite system.

Exploiting the controller's gain ($\mathscr{K}$), a generalized dataset (S_d) with the error vector (e) and respective input ($\check{u}$) to the control system can be created in the following form [44]:

$$S_d = \left[e^{T}\ \check{u}\right]. \tag{38}$$

In this work, the dataset (S_d) is expanded into a robust form by evaluating the controller's gain ($\mathscr{K}_{m_t,m_c,m_{hf}}$) for a bounded variation of the coupled thigh (m_t), calf (m_c), and heel-foot (m_{hf}) masses of the human-exoskeleton system. Thereafter, the controller input ($\check{u}_{m_t,m_c,m_{hf}}$) for the hip, knee, and ankle joints is formulated as follows:

$$\check{u}_{m_t,m_c,m_{hf}} = \mathscr{K}_{m_t,m_c,m_{hf}}\, e, \tag{39}$$

where

$$\check{u}_{m_t,m_c,m_{hf}} = \left[\check{u}_{H_{m_t,m_c,m_{hf}}}\ \check{u}_{K_{m_t,m_c,m_{hf}}}\ \check{u}_{A_{m_t,m_c,m_{hf}}}\right]^{T}, \tag{40}$$

$$e = \left[e_{\theta_H}\ e_{\theta_K}\ e_{\theta_A}\ e_{\dot{\theta}_H}\ e_{\dot{\theta}_K}\ e_{\dot{\theta}_A}\right]^{T}. \tag{41}$$

In Equation (40), $\check{u}_{H_{m_t,m_c,m_{hf}}}$, $\check{u}_{K_{m_t,m_c,m_{hf}}}$, and $\check{u}_{A_{m_t,m_c,m_{hf}}}$ denote the controller output for the hip, knee, and ankle joints of the exoskeleton device, respectively. In Equation (41), e_{θ_H}, e_{θ_K}, and e_{θ_A} signify the hip, knee, and ankle joint angular errors, respectively. $e_{\dot{\theta}_H}$, $e_{\dot{\theta}_K}$, and $e_{\dot{\theta}_A}$ represent the respective errors of the hip, knee, and ankle joint angular velocities.

The expanded structure of the robust dataset (S_{rd}) is finally depicted as below:

$$S_{rd} = \left[e^{T}\ \check{u}_{m_t,m_c,m_{hf}}\right], \tag{42}$$

where

$$S_{rd} = \left[S(1)_{rd}\ S(2)_{rd}\ S(3)_{rd}\right]^{T}. \tag{43}$$

5.1.2. Stage II: Stepwise Layout of Offline Training of the RLQR-NF Dataset. The layout and execution of the proposed control strategy for the exoskeleton device are presented in the following steps.

Step 1. The state gain matrix ($\mathscr{K}_{m_t,m_c,m_{hf}}$) is evaluated by solving Equations (36) and (37) for a different set of coupled thigh (m_t), calf (m_c), and heel-foot (m_{hf}) masses, as shown in Table 2. The parametric variation is incorporated by increasing the nominal mass values up to 20%, with an increment of 0.3, 0.15, and 0.06 kg for the thigh, calf, and heel-foot. After performing several numerical experiments, the state-weight matrix and control cost matrix are selected as $\mathscr{Q} = \text{diag}\,(400000, 4000, 8000, 800, 8000, 800)$ and $\mathbf{R} = \text{eye}(3, 3)$.

Step 2. Apply Equation (39) to compute the controller input ($\check{u}_{m_t,m_c,m_{hf}}$) for the operating range of state variables in the error vector as shown in Table 3. The structure of three robust datasets is formed by exploiting Equations (42) and (43).

Step 3. The training of robust datasets is carried out using the ANFIS approach. The first six columns of every dataset are inherently considered the input set. The last column of every dataset is regarded as the output set. The input set comprises the error vector (e), and the output set contains the controller input vector ($\check{u}_{m_t,m_c,m_{hf}}$). The three ANFIS architectures are formed, trained, and saved as *anfis1.fis*, *anfis2.fis*, and *anfis3.fis* for three controller inputs. Several simulation runs are

Table 2: Variation in lower limb mass parameters of the coupled human-exoskeleton system.

Thigh, m_t (kg)	Calf, m_c (kg)	Heel-foot, m_{hf} (kg)
8.25	3.85	1.50
8.58	4.00	1.56
8.91	4.16	1.62
9.24	4.31	1.68
9.57	4.47	1.74
9.90	4.62	1.80

Table 3: Operating range of the error in state variables.

Variables in the error vector	Minimum value	Maximum value	Units
e_{θ_H}	-60	60	Degree
e_{θ_K}	-60	60	Degree
e_{θ_A}	-30	30	Degree
$e_{\dot{\theta}_H}$	-90	90	Degree/sec
$e_{\dot{\theta}_K}$	-90	90	Degree/sec
$e_{\dot{\theta}_A}$	-60	60	Degree/sec

performed by varying the number of MF from 1 to 50 and epochs from 1 to 30. Thereafter, the training parameters are selected based on the zero error tolerance between the desired and predicted output vectors. In general, ANFIS utilizes two optimization methods: backpropagation and hybrid, to establish the learning between the input and output vectors. A gradient descent model is employed to evaluate the node error in the backpropagation method. In contrast, a least square algorithm along with the gradient descent model is exploited to regulate the errors in the hybrid method. In this work, the hybrid method is used with all three datasets for training the neural-fuzzy networks. The complete details of training parameters are given in Table 4. After generating training files (*anfis1.fis*, *anfis2.fis*, and *anfis3.fis*) as the desired robust LQR-based neural-fuzzy controller, the respective signals are inputted to the nonlinear dynamics of the coupled human-exoskeleton system.

5.2. Online Training of the RLQR-NF Dataset for External Disturbances. Considering assumption A1, the control input with uncertain parameters ($\hat{\boldsymbol{u}}$) can be further expressed in terms of the state feedback ($\mathscr{K}\boldsymbol{z}$) and disturbance observer ($\hat{d}$) as

$$\hat{\boldsymbol{u}} = (\Delta + 1)\boldsymbol{u} = \hat{d} - \mathscr{K}\boldsymbol{z}, \tag{44}$$

where $\hat{d}$ denotes the estimated disturbance by ANFIS architecture, and $\mathscr{K}$ signifies the state feedback matrix.

Table 4: Training parameters of ANFIS architectures.

Training parameters	*anfis1.fis*	*anfis2.fis*	*anfis3.fis*
MF type	Gaussian	Gaussian	Gaussian
MF number	5	12	3
Error tolerance	0.00001	0.0001	0.001
Epochs	10	15	5
Learning model	Hybrid	Hybrid	Hybrid

Substituting Equation (44) into Equation (22), the linearized dynamic model can be written as

$$\dot{\boldsymbol{z}} = \left(\check{\mathscr{A}} - \check{\mathscr{B}}\mathscr{K}\right)\boldsymbol{z} + \check{\mathscr{B}}\left(\hat{d} + d\right). \tag{45}$$

Now, the error dynamics of the control design in the time domain can be expressed as below:

$$\dot{e}(t) = \left(\check{\mathscr{A}} - \check{\mathscr{B}}\mathscr{K}\right)e(t) + \check{\mathscr{B}}\left(\hat{d}(t) - f(t)\right), \tag{46}$$

where

$$\check{\mathscr{B}}f(t) = \dot{\boldsymbol{z}}_{\text{des}}(t) - \check{\mathscr{B}}d(t) - \left(\check{\mathscr{A}} - \check{\mathscr{B}}\mathscr{K}\right)\boldsymbol{z}_{\text{des}}(t). \tag{47}$$

Employing assumption A2, the $f(t)$ can be considered a function with an upper limit. Therefore, the effectiveness of the coupled human-exoskeleton system can be augmented by estimating the $f(t)$ from $\hat{d}(t)$. Using the ANFIS architecture mentioned in Section 4, the $f(t)$ is approximated as follows [46, 47]:

$$f(\varkappa) = \bar{W}^T\boldsymbol{\Omega}(\varkappa) + \bar{\varrho}(\varkappa), \tag{48}$$

where $\bar{W}$ denotes the ideal normalized firing strength matrix; $\boldsymbol{\Omega}(\varkappa)$ signifies the membership function vector; and $\bar{\varrho}(\varkappa)$ represents an error of approximation with the condition $\bar{\varrho}(\varkappa) \le \zeta$, where ζ is a constant factor.

Now utilizing Equations (46) and (48), the error dynamics can be rewritten as

$$\dot{e}(t) = \left(\check{\mathscr{A}} - \check{\mathscr{B}}\mathscr{K}\right)e(t) + \check{\mathscr{B}}\left(\hat{d}(t) - \bar{W}^T\boldsymbol{\Omega}(\varkappa) - \bar{\varrho}(\varkappa)\right). \tag{49}$$

From the above equation, the disturbance estimator $\hat{d}(t)$ is defined as follows to design the control law [47]:

$$\hat{d}_i = \hat{W}_{ji}^T\boldsymbol{\Omega}_j(\varkappa) - \zeta\ \text{sign}\left(e^T\mathscr{P}\check{\mathscr{B}}_i\right), \quad i = 1, 2, \cdots, p \text{ and } j = 1, 2, \cdots, l, \tag{50}$$

where p and l denote the number of inputs and network nodes in the hidden layer.

In the above expression, the estimated strength matrix $(\widehat{W}_{ji})$ is updated according to the following law:

$$\dot{\widehat{W}}_{ji} = -\Gamma_i \mathbf{\Omega}_j(\varkappa) e^T \mathscr{P} \check{\mathscr{B}}_i, \tag{51}$$

where Γ_i denotes a positive definite matrix with the symmetric property.

6. Stability Analysis of the LQR-Based Neural-Fuzzy Control

In this section, the Lyapunov function is presented to analyze the global stability of the proposed control scheme under the effect of parametric uncertainties and external disturbances. Consider the Lyapunov candidate function as follows:

$$V = \frac{1}{2} e^T \mathscr{P} e + \sum_{i=1}^{p} \left(\tilde{W}_{ji}^T \Gamma_i^{-1} \tilde{W}_{ji} \right), \tag{52}$$

with

$$\tilde{W}_{ji} = \widehat{W}_{ji} - \bar{W}_{ji}, \tag{53}$$

where $\tilde{W}_{ji}$ denotes the estimation error between the estimated strength matrix and the ideal constant strength matrix.

Differentiating Equation (52) and employing Equations (49) and (37), one can obtain

$$\begin{aligned} \dot{V} = &-e^T \left(\mathscr{Q} + \mathscr{K}^T \mathbf{R} \mathscr{K} \right) e + 2 e^T \mathscr{P} \check{\mathscr{B}}_i \\ &\times \left(\widehat{d}_i - \bar{W}_{ji}^T \mathbf{\Omega}_j(\varkappa) - \bar{\mathrm{Q}}_i(\varkappa) \right) + 2 \sum_{i=1}^{p} \left(\tilde{W}_{ji}^T \Gamma_i^{-1} \dot{\tilde{W}}_{ji} \right). \end{aligned} \tag{54}$$

Now utilizing Equation (53), the error difference between the actual value and the desired value by the designed neural-fuzzy network can be expressed as

$$\tilde{W}_{ji} \mathbf{\Omega}_j(\varkappa) = \widehat{W}_{ji} \mathbf{\Omega}_j(\varkappa) - \bar{W}_{ji} \mathbf{\Omega}_j(\varkappa). \tag{55}$$

Moreover, as the firing strength matrix $(\bar{W}_{ji})$ is a constant matrix, Equation (53) holds the following relation after differentiation:

$$\dot{\tilde{W}}_{ji} = \dot{\widehat{W}}_{ji}. \tag{56}$$

Reconstituting Equation (54) using Equations (55) and (56), one can obtain

$$\begin{aligned} \dot{V} = &-e^T \left(\mathscr{Q} + \mathscr{K}^T \mathbf{R} \mathscr{K} \right) e + 2 e^T \mathscr{P} \check{\mathscr{B}}_i \\ &\times \left(\widehat{d}_i - \widehat{W}_{ji} \mathbf{\Omega}_j(\varkappa) + \tilde{W}_{ji} \mathbf{\Omega}_j(\varkappa) - \bar{\mathrm{Q}}_i(\varkappa) \right) \\ &+ 2 \sum_{i=1}^{p} \left(\tilde{W}_{ji}^T \Gamma_i^{-1} \dot{\widehat{W}}_{ji} \right), \end{aligned} \tag{57}$$

$$\begin{aligned} \dot{V} = &-e^T \left(\mathscr{Q} + \mathscr{K}^T \mathbf{R} \mathscr{K} \right) e + 2 \sum_{i=1}^{p} e^T \mathscr{P} \check{\mathscr{B}}_i \\ &\times \left(\widehat{d}_i - \widehat{W}_{ji} \mathbf{\Omega}_j(\varkappa) + \tilde{W}_{ji} \mathbf{\Omega}_j(\varkappa) - \bar{\mathrm{Q}}_i(\varkappa) \right) \\ &+ 2 \sum_{i=1}^{p} \left(\tilde{W}_{ji}^T \Gamma_i^{-1} \dot{\widehat{W}}_{ji} \right). \end{aligned} \tag{58}$$

Now substituting Equations (50) and (51) into Equation (58), one can obtain

$$\begin{aligned} \dot{V} &= -e^T \left(\mathscr{Q} + \mathscr{K}^T \mathbf{R} \mathscr{K} \right) e + 2 \sum_{i=1}^{p} e^T \mathscr{P} \check{\mathscr{B}}_i \\ &\quad \times \left(\widehat{d}_i - \widehat{W}_{ji} \mathbf{\Omega}_j(\varkappa) - \bar{\mathrm{Q}}_i(\varkappa) \right) \\ &\leq -e^T \left(\mathscr{Q} + \mathscr{K}^T \mathbf{R} \mathscr{K} \right) e \\ &\leq 0. \end{aligned} \tag{59}$$

Invoking the above equation, it can be concluded that the error dynamics of the coupled human-exoskeleton system is asymptotically stable. Therefore, the proposed controller carries out the asymptotic tracking with error $e \longrightarrow 0$ (as $t \longrightarrow \infty$) following the disturbance estimator (Equation (50)) and firing strength adaptation law (Equation (51)). Furthermore, the local stability of the proposed controller can be analyzed using the pole placement theory, where the convergence rate can be investigated by keeping the poles on the left side of the s-plane [45].

7. Results and Discussion

In this section, the simulation results and analyses are presented to evaluate the effectiveness of the proposed control strategy (RLQR-NF) for an exoskeleton device during passive gait rehabilitation measures. The block representation for the proposed control strategy is schematically shown in Figure 7. At first, the performance of the proposed control strategy is compared with a contrast control strategy without parametric uncertainties and external disturbances. In this work, the exponential reaching law-based sliding mode (ERL-SM) control is used as a contrast control strategy [48]. Thereafter, two cases are contemplated to demonstrate the controller's robustness: the first case, increasing the coupled segment masses by 20% with a trigonometric form of disturbances, and the second case, increasing the coupled segment masses by 30% with a random form of disturbances.

7.1. Simulation Results without Parametric Uncertainties and External Disturbances. In this subsection, a comparative analysis between the RLQR-NF and ERL-SM control strategies is presented for desired gait tracking during passive gait rehabilitation measures. In the absence of parametric uncertainties and external disturbances, i.e., $\Delta = 0$ and $d = 0$ in Equation (25), the nominal mass values of lower limb segments $(m_{\mathrm{t}} = 8.25, m_{\mathrm{c}} = 3.85, \text{and } m_{\mathrm{hf}} = 1.5)$ are taken into account for drawing the state of comparison. Using Equation

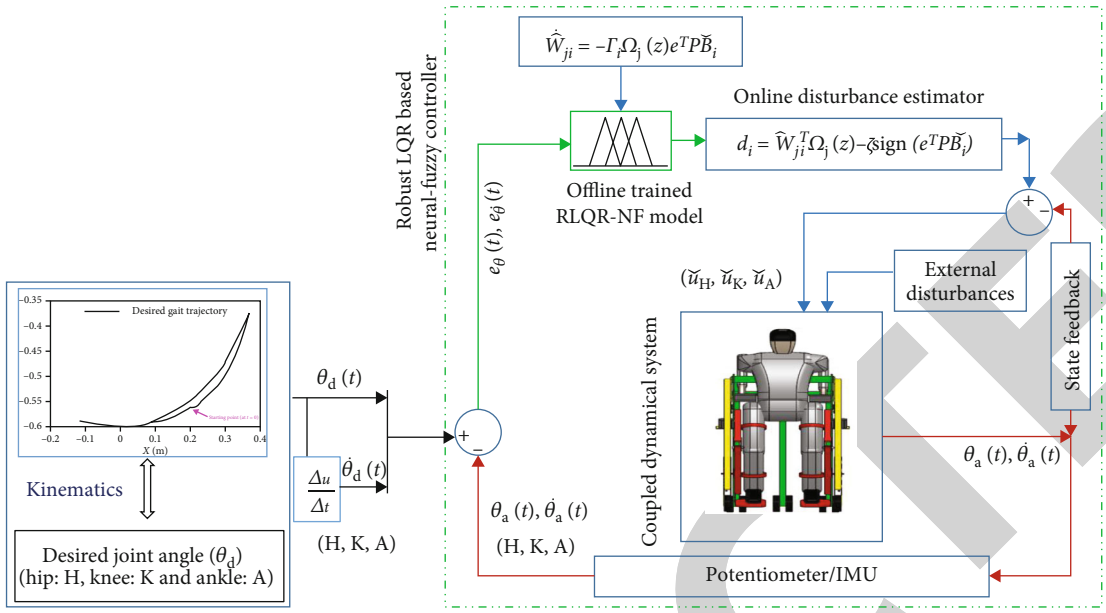


Figure 7: Schematic representation of the implemented proposed RLQR-NF controller.

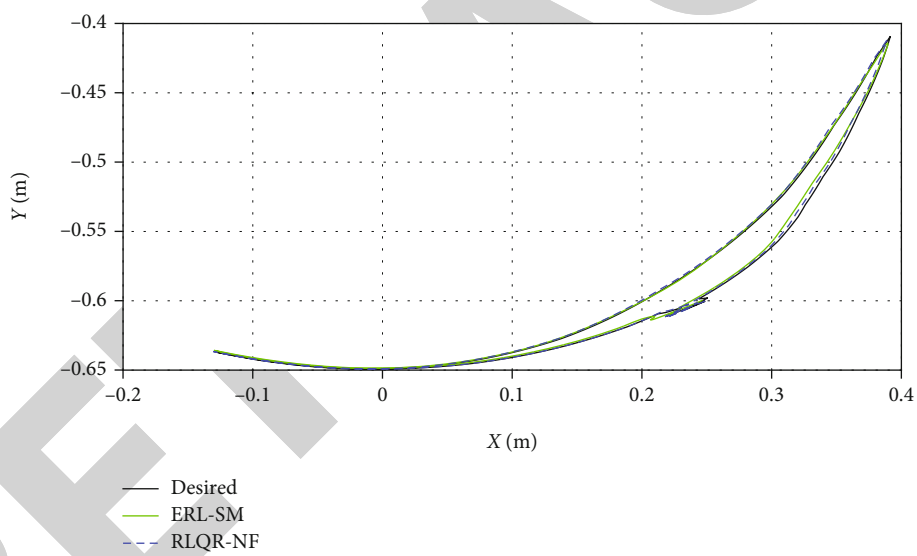


Figure 8: Desired trajectory tracking for ERL-SM and RLQR-NF control schemes with nominal mass values.

(36), the optimal state gain matrix for the nominal mass parameters ($\mathscr{K}_{8.25,3.85,1.5}$) is calculated as below:

$$\mathscr{K}_{8.25,3.85,1.5} = \begin{pmatrix} 448.63 & -2.72 & -0.57 & 17.84 & -0.31 & -0.127 \\ 7.65 & 12.40 & 2.49 & 3.20 & 20.27 & 0.82 \\ 8.72 & -1.92 & 20.17 & 0.33 & -2.56 & 6.40 \end{pmatrix}. \tag{60}$$

In the ERL-SM control strategy [48], the control law parameters are used as $c = \mathrm{diag}\,(50, 50, 50)$, $\varepsilon = 0.5 \times I_3$, and $k_e = I_3$. As shown in Figure 8, the healthy gait trajectory is tracked by incorporating the kinematic and dynamic parameters into the proposed control (RLQR-NF) and ERL-SM control strategy. A time period of 2 seconds is considered to complete one gait cycle. The starting Cartesian position (X, Y: 0.25 m, -0.60 m) of the desired gait trajectory is illustrated in black color. The actual trajectories in the Cartesian coordinate frame are presented by the dashed blue line (RLQR-NF) and the green line (ERL-SM).

Figures 9(a) and 9(b) depict the tracking error in both directions, i.e., X- and Y-directions (e_x and e_y). The maximum absolute deviation in the X-direction ($|e_x|_{\max}$) for the ERL-SM and RLQR-NF control schemes is 0.013 m and 0.008 m, respectively. In the Y-direction, the respective deviation ($|e_y|_{\max}$) is observed to be 0.009 m and 0.006 m for the ERL-SM and RLQR-NF control strategies.

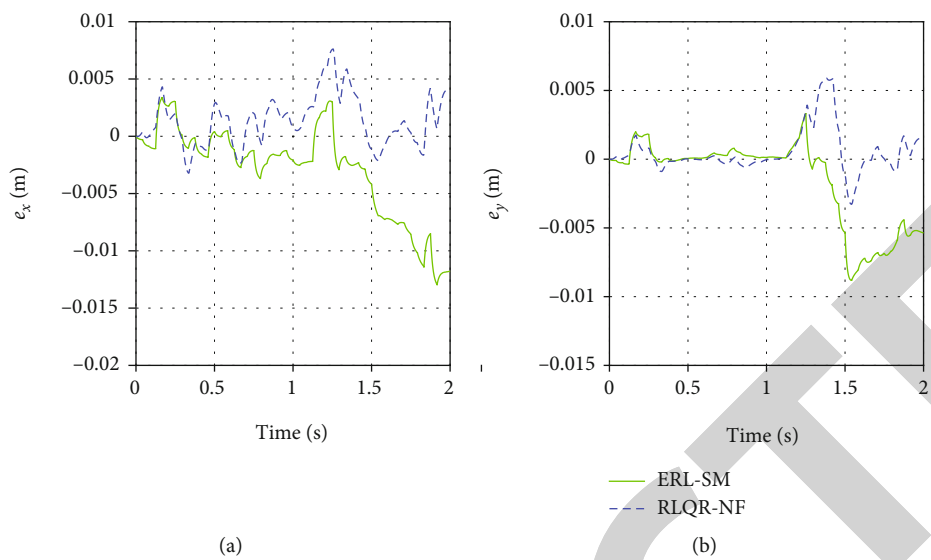


Figure 9: Position tracking error for ERL-SM and RLQR-NF control schemes with nominal mass values. (a) X-direction. (b) Y-direction.

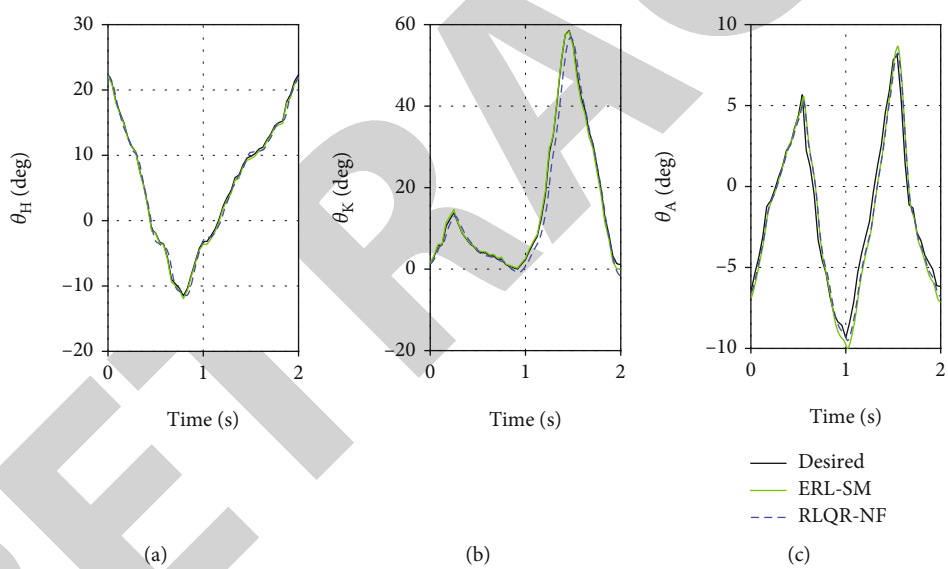


Figure 10: Joint angle tracking for ERL-SM and RLQR-NF control schemes with nominal mass values. (a) Hip joint. (b) Knee joint. (c) Ankle joint.

The tracking of desired joint angles with applied control strategies is illustrated in Figures 10(a)–10(c). The angular deviations (e_{θ_H}, e_{θ_K}, and e_{θ_A}) from desired joint trajectories are shown in Figures 11(a)–11(c). Considering the hip joint, it is observed that the maximum absolute deviation ($|e_{\theta_H}|_{max}$) for respective controllers is 0.78° (ERL-SM) and 0.51° (RLQR-NF). In the knee joint, the deviation ($|e_{\theta_K}|_{max}$) is found to be 1.15° and 1.16° for the system with the ERL-SM and RLQR-NF control strategies, respectively. For the ankle joint, the respective deviations ($|e_{\theta_A}|_{max}$) are estimated as 0.81° (ERL-SM) and 0.32° (RLQR-NF).

Figures 12(a)–12(c) demonstrate the generated control signals ($\boldsymbol{u}_H$, $\boldsymbol{u}_K$, and $\boldsymbol{u}_A$) following the desired trajectory through repetitive gait rehabilitation exercises. With the ERL-SM control scheme, the peak values of the hip, knee, and ankle signals are estimated as 32.98 V, 19.88 V, and 1.8 V. On the other hand, with the RLQR-NF control scheme, the respective values of control signals are found to be 30.25 V, 18.1 V, and 1.25 V. It is evident from the results that the proposed control strategy (RLQR-NF) outperforms the contrast control strategy (ERL-SM) to track the desired gait trajectory, however, with a marginal difference. Therefore, to demonstrate the effectiveness of the proposed control when dealing with PUEDs, variations in mass parameters and the form of disturbances are considered further.

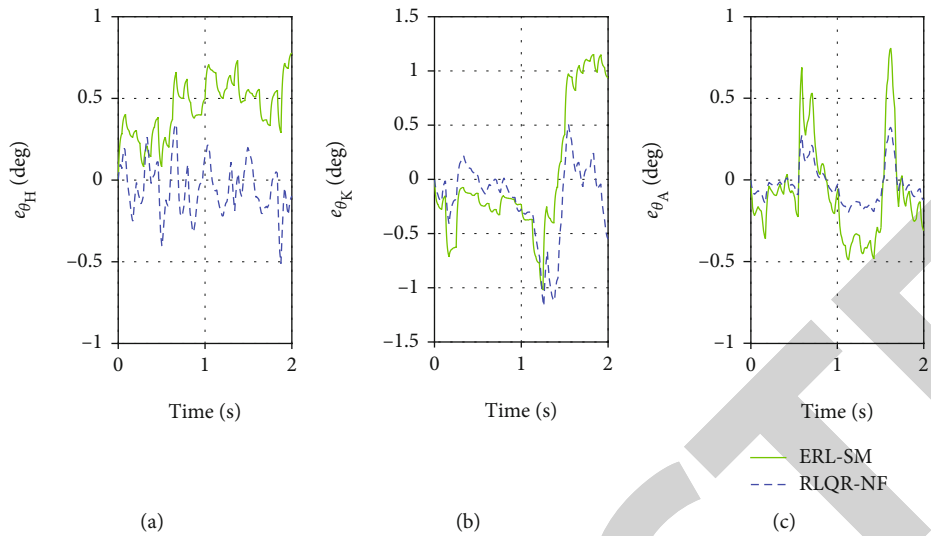


Figure 11: Joint tracking error for ERL-SM and RLQR-NF control schemes with nominal mass values. (a) Hip joint. (b) Knee joint. (c) Ankle joint.

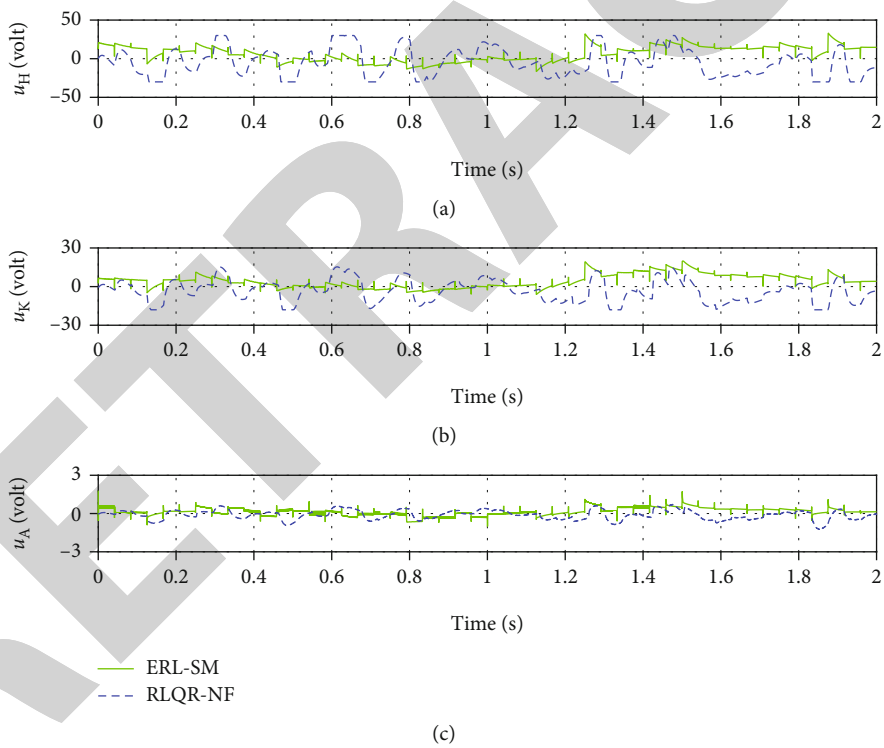


Figure 12: Control signals for ERL-SM and RLQR-NF control schemes with nominal mass values. (a) Hip joint. (b) Knee joint. (c) Ankle joint.

7.2. Simulation Results for Parametric Variations and External Disturbances. In order to realize the robustness of the proposed control strategy (RLQR-NF), the root mean square error (RMSE) is computed and compared with the ERL-SM control strategy for parametric uncertainties and external disturbances. Moreover, based on RMSE values, the performance index (PI) is calculated to analyze the improvement of the proposed control scheme over the contrast control scheme.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{a=1}^{N} \|e_a\|^2},$$
$$\mathrm{PI} = \frac{\mathrm{RMSE}_{\text{ERL-SM}} - \mathrm{RMSE}_{\text{RLQR-NF}}}{\mathrm{RMSE}_{\text{ERL-SM}}} \times 100\%, \tag{61}$$

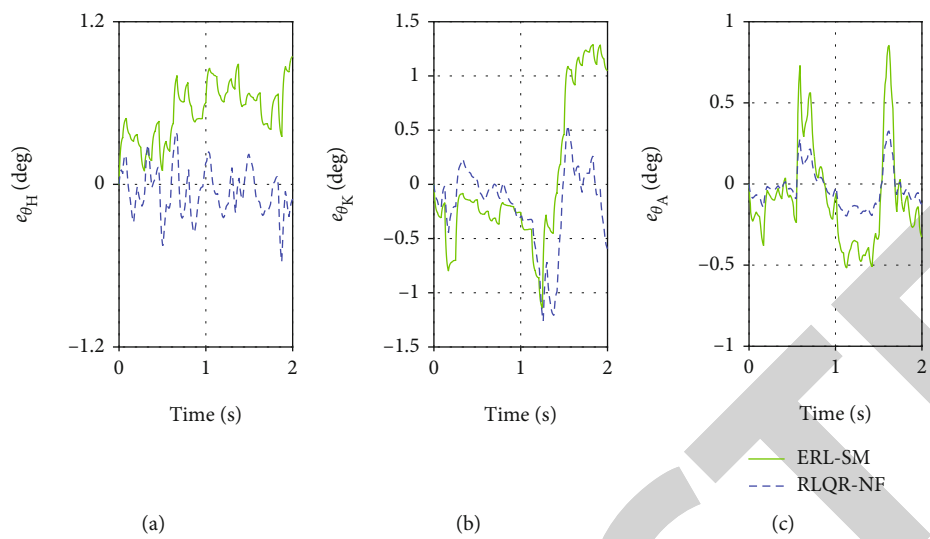


Figure 13: Joint tracking error for ERL-SM and RLQR-NF control schemes with the first case of PUEDs. (a) Hip joint. (b) Knee joint. (c) Ankle joint.

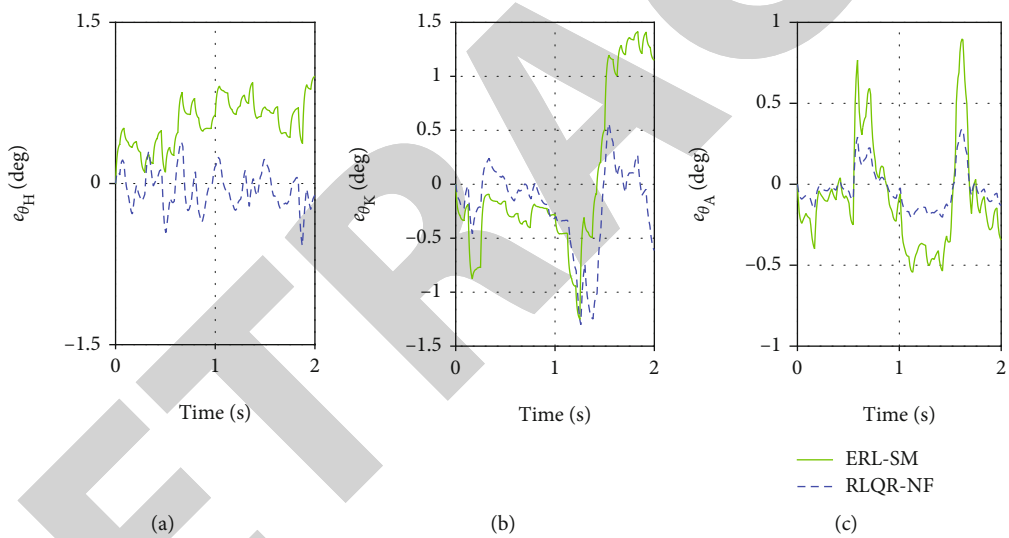


Figure 14: Joint tracking error for ERL-SM and RLQR-NF control schemes with the second case of PUEDs. (a) Hip joint. (b) Knee joint. (c) Ankle joint.

where $e_a : e_\theta$ represents the error between the desired and actual joint angles, and N is the size of the error vector. $\text{RMSE}_{\text{ERL-SM}}$ and $\text{RMSE}_{\text{RLQR-NF}}$ signify the root mean square errors related to ERL-SM and RLQR-NF control strategies.

Considering the first case of PUEDs (Case I), where the system masses are increased by 20% ($m_t = 9.90$, $m_c = 4.62$, and $m_{hf} = 1.80$) along with a trigonometric form of disturbances ($\mathscr{D}_1 = (6 \sin(4\pi t))$, $\mathscr{D}_2 = (5 \sin(3\pi t))$, and $\mathscr{D}_3 = (3 \sin(2\pi t))$), the joint angular errors (e_{θ_H}, e_{θ_K}, and e_{θ_A}) for the applied control strategies are shown in Figures 13(a)–13(c). For the hip joint, $\text{RMSE}_{\text{ERL-SM}}$ and $\text{RMSE}_{\text{RLQR-NF}}$ are estimated as 0.578° and 0.283°, respectively. The respective RMSE values for the knee joint are found to be 0.672° and 0.42°. In ankle joint tracking, the RMSE values for the control strategies are recorded as 0.321° and 0.224°.

Considering the second case of PUEDs (Case II), where the system masses are increased by 30% ($m_t = 10.73$, $m_c = 5.00$, and $m_{hf} = 1.95$) along with a random form of disturbances ($D_1 = (5 \times \text{random}(1))$, $D_2 = (3 \times \text{random}(1))$, and $D_3 = (2 \times \text{random}(1))$), the joint angular errors (e_{θ_H}, e_{θ_K}, and e_{θ_A}) for the proposed and contrast control strategies are shown in Figures 14(a)–14(c). For the hip joint, the values of $\text{RMSE}_{\text{ERL-SM}}$ and $\text{RMSE}_{\text{RLQR-NF}}$ are found to be 0.613° and 0.287°, respectively. The respective RMSE values related to the knee joint tracking are obtained as 0.742° and 0.434°. The following RMSE values for the ankle joint are found to be 0.334° and 0.228°.

Table 5: Comparative performance analysis of the proposed control over the contrast control.

Joint name	RMSE (ERL-SM) (deg)	RMSE (RLQR-NF) (deg)	PI (%)
Case I: with a 20% increment in mass parameters and a trigonometric form of external disturbances ($m_t = 9.90$, $m_c = 4.62$, and $m_{hf} = 1.80$) ($\mathscr{D}_1 = (6 \times \sin(4\pi t))$, $\mathscr{D}_2 = (5 \times \sin(3\pi t))$, and $\mathscr{D}_3 = (3 \times \sin(2\pi t))$)			
Hip	0.578	0.283	51.04
Knee	0.672	0.420	37.50
Ankle	0.321	0.224	30.21
Case II: with a 30% increment in mass parameters and a random form of external disturbances ($m_t = 10.73$, $m_c = 5.00$, and $m_{hf} = 1.95$) ($\mathscr{D}_1 = (6 \times \text{random}(4))$, $\mathscr{D}_2 = (5 \times \text{random}(3))$, and $\mathscr{D}_3 = (3 \times \text{random}(2))$)			
Hip	0.613	0.287	53.19
Knee	0.742	0.434	41.51
Ankle	0.334	0.228	31.73

Table 6: Settling time of ERL-SM and RLQR-NF control schemes for convergence analysis.

Control scheme	Lower limb joint	Settling time (sec)		
		Nominal system mass	Case I of PUEDs	Case II of PUEDs
ERL-SM	Hip	1.971	1.986	1.993
	Knee	1.962	1.974	1.988
	Ankle	1.927	1.943	1.969
RLQR-NF	Hip	1.951	1.967	1.991
	Knee	1.946	1.959	1.987
	Ankle	1.914	1.933	1.958

Table 5 presents the performance index (PI) of the proposed control over the contrast control. The proposed control is promising in desired gait tracking compared to the contrast control, subjected to PUEDs. Moreover, as observed from Table 5, the performance index (PI) is improved by 2.15%, 4.01%, and 1.52% in Case II as compared to Case I. During rehabilitation exercises, this performance investigation allows the lower limb exoskeleton system to carry out repetitive movements with greater accuracy under the presence of PUEDs.

The convergence of both control schemes is investigated by evaluating the settling time, i.e., the time lapsed for the error to drop within 2% of the final value. The settling time for the error in the hip, knee, and ankle joints for every set of mass values is presented in Table 6. The low values of settling time indicate the faster convergence of the proposed controller (RLQR-NF) over the contrast controller (ERL-SM) before achieving the full stable state.

8. Conclusions

In this work, a robust LQR-based neural-fuzzy control has been proposed to follow the natural gait trajectory using an exoskeleton system during passive rehabilitation measures. Primarily, a linearized state-space form of the nonlinear human-exoskeleton has been established via the input-output feedback linearization method. Employing the LQR state gain matrix, robust ANFIS training datasets have been formed with a variation of system parameters. The operating range of the error vector and control responses have been regarded as the training input and output vectors. The ANFIS architectures have been trained offline to deal with the effect of parametric uncertainties. Thereafter, the online adaptation law of firing strength in ANFIS architectures has been incorporated to deal with external disturbances. The asymptotic stability of the coupled dynamics while applying the proposed control has been ensured using the Lyapunov theory. Finally, the effectiveness of the proposed controller has been investigated by comparing it to the exponential reaching law-based sliding mode control. The robustness analysis has been carried out by varying mass parameters and inducing different forms of external disturbances. The simulation results have shown the potential of the proposed robust tracking control for passive gait rehabilitation using an exoskeleton system. In the future, the effect of human involvement will be considered to design an "assist-as-needed" control strategy during active rehabilitation.

Appendix

The elements of the state-weight factor matrices ($\mathscr{A}, \mathscr{B}$) can be evaluated using the following derived formulations.

For the $\mathscr{A}$ matrix,

$$\alpha_{41} = -12 l_1 l_2 l_3 g\left(2m_2^2 + 4m_1 m_2 + 3m_1 m_3 + 2m_2 m_3\right),$$

$$\tilde{\alpha}_{41} = 2 l_2 l_3 \left(12 l_1^2 m_2^2 + 12 l_1^2 m_2 m_3 + 16 l_1^2 m_1 m_2 + 12 l_1^2 m_1 m_3 - 48 l_2 l_3 m_2 m_3 - 36 l_2 l_3 m_3^2\right),$$

$$\alpha_{42} = 72 l_1 l_2 l_3 g\left(m_2^2 + m_3^2 + 2m_2 m_3\right),$$

$$\tilde{\alpha}_{42} = \tilde{\alpha}_{41},$$

$$\alpha_{43} = -18 g l_1 m_2 m_3,$$

$$\tilde{\alpha}_{43} = \frac{\tilde{\alpha}_{41}}{2 l_2 l_3},$$

$$\alpha_{51} = 12 g\left(6 l_1 m_2 + 3 l_1 m_3 + 4 l_2 m_2 + 3 l_2 m_3\right) \cdot \left(l_1 m_2 + l_1 m_3 + l_2 m_3 + \frac{l_1 m_1}{2} + \frac{l_2 m_2}{2} + \frac{l_3 m_3}{2}\right),$$

$$\tilde{\alpha}_{51} = \tilde{\alpha}_{51''} = \tilde{\alpha}_{52'} = \tilde{\alpha}_{52''} = \tilde{\alpha}_{53} = \tilde{\alpha}_{53''} = \tilde{\alpha}_{63} = \tilde{\alpha}_{52} = \tilde{\alpha}_{53'} = \tilde{\alpha}_{63''} = \frac{\tilde{\alpha}_{41}}{2l_3},$$

$$\alpha_{51'} = 6g(l_2m_2 + 2l_2m_3 + l_3m_3) \cdot \left(4l_1^2m_1 + 12l_1^2m_2 + 3l_1^2m_3 + 4l_2^2m_2 + 3l_2^2m_3 + 12l_1l_2m_2 + 6l_1l_2m_3 - 12l_2l_3m_3\right),$$

$$\tilde{\alpha}_{51'} = \frac{l_2\tilde{\alpha}_{41}}{2l_3},$$

$$\alpha_{51''} = 6gm_3\left(6l_1^2l_2m_1 + 3l_2^2l_1m_2 + 9l_1^2l_2m_2 + 4l_1^2l_3m_1 + 12l_1^2l_3m_2 + 3l_1^2l_3m_3 - 12l_3^2l_2m_3 - 18l_2^2l_3m_3 + 6l_1l_2l_3m_2 + 3l_1l_2l_3m_3\right),$$

$$\alpha_{52} = 6g(l_2m_2 + 2l_2m_3 + l_3m_3) \cdot (6l_1m_2 + 3l_1m_3 + 4l_2m_2 + 3l_2m_3),$$

$$\alpha_{52'} = 6g(l_2m_2 + 2l_2m_3 + l_3m_3) \cdot \left(4l_1^2m_1 + 12l_1^2m_2 + 3l_1^2m_3 + 4l_2^2m_2 + 3l_2^2m_3 + 12l_1l_2m_2 + 6l_1l_2m_3 - 12l_2l_3m_3\right),$$

$$\alpha_{52''} = \alpha_{51''} = \alpha_{53} = \alpha_{63},$$

$$\alpha_{53'} = 6gl_3m_3(6l_1m_2 + 3l_1m_3 + 4l_2m_2 + 3l_2m_3),$$

$$\alpha_{53''} = 6gl_3m_3\left(4l_1^2m_1 + 12l_1^2m_2 + 3l_1^2m_3 + 4l_2^2m_2 + 3l_2^2m_3 + 12l_1l_2m_2 + 6l_1l_2m_3 - 12l_2l_3m_3\right),$$

$$\alpha_{61} = 6g(l_2m_2 + 2l_2m_3 + l_3m_3)\left(6l_1^2l_2m_1 + 3l_2^2l_1m_2 + 9l_1^2l_2m_2 + 4l_1^2l_3m_1 + 12l_1^2l_3m_2 + 3l_1^2l_3m_3 - 12l_3^2l_2m_3 - 18l_2^2l_3m_3 + 6l_1l_2l_3m_2 + 3l_1l_2l_3m_3\right),$$

$$\tilde{\alpha}_{61} = \tilde{\alpha}_{61'} = \tilde{\alpha}_{62} = \tilde{\alpha}_{62'} = \tilde{\alpha}_{63'} = \frac{l_2\tilde{\alpha}_{41}}{2},$$

$$\alpha_{61'} = 6g\left(3l_1^2l_2^2m_2^2 + 12l_1^2l_2^2m_2m_3 + 4l_1^2l_2^2m_1m_2 + 12l_1^2l_2^2m_1m_3 + 18l_1^2l_2l_3m_2m_3 + 12l_1^2l_2l_3m_1m_3 + 12l_1^2l_3^2m_2m_3 + 3l_1^2l_3^2m_3^2 + 4l_1^2l_3^2m_1m_3 - 12l_2^3l_3m_2m_3 - 36l_2^3l_3m_3^2 - 36l_2^2l_3^2m_3^2 - 12l_3^3l_2m_3^2\right),$$

$$\alpha_{61''} = 36gl_1(l_2m_2 + 2l_3m_2 + l_3m_3) \cdot \left(l_1m_2 + l_1m_3 + l_2m_3 + \frac{l_1m_1}{2} + \frac{l_2m_2}{2} + \frac{l_3m_3}{2}\right),$$

$$\tilde{\alpha}_{61''} = \tilde{\alpha}_{62''} = \frac{\tilde{\alpha}_{41}}{2},$$

$$\alpha_{62} = \alpha_{61},$$

$$\alpha_{62'} = \alpha_{61'} = \alpha_{63'},$$

$$\alpha_{62''} = 36gl_1(l_2m_2 + 2l_3m_2 + l_3m_3)\left(l_2m_3 + \frac{l_2m_2}{2} + \frac{l_3m_3}{2}\right),$$

$$\alpha_{63''} = 18gl_1m_3(l_2m_2 + 2l_3m_2 + l_3m_3). \quad (A.1)$$

For the $\mathscr{B}$ matrix,

$$\beta_{41} = 96l_2l_3m_2 + 72l_2l_3m_3,$$

$$\tilde{\beta}_{41} = \tilde{\alpha}_{41},$$

$$\beta_{42} = -(144l_1l_3m_2 + 72l_1l_3m_3 + 96l_2l_3m_2 + 72l_2l_3m_3),$$

$$\tilde{\beta}_{42} = \tilde{\beta}_{43} = \tilde{\beta}_{41},$$

$$\beta_{43} = 72l_1l_2m_2 + 144l_1l_3m_2 + 72l_1l_3m_3,$$

$$\beta_{51} = -(72l_1m_2 + 36l_1m_3 + 48l_2m_2 + 36l_2m_3),$$

$$\tilde{\beta}_{51} = \frac{\tilde{\beta}_{41}}{2l_3},$$

$$\beta_{52} = 48l_1^2m_1 + 144l_1^2m_2 + 36l_1^2m_3 + 48l_2^2m_2 + 36l_2^2m_3 + 144l_1l_2m_2 + 72l_1l_2m_3 - 144l_2l_3m_3,$$

$$\tilde{\beta}_{52} = \frac{l_2\tilde{\beta}_{41}}{2l_3},$$

$$\beta_{53} = -\left(72l_1^2l_2m_1 + 36l_2^2l_1m_2 + 108l_1^2l_2m_2 + 48l_1^2l_3m_1 + 144l_1^2l_3m_2 + 36l_1^2l_3m_3 - 144l_3^2l_2m_3 - 216l_2^2l_3m_3 + 72l_1l_2l_3m_2 + 36l_1l_2l_3m_3\right),$$

$$\tilde{\beta}_{53} = \frac{l_2\tilde{\beta}_{41}}{2},$$

$$\beta_{61} = 36l_1(l_2m_2 + 2l_3m_2 + l_3m_3),$$

$$\tilde{\beta}_{61} = \frac{\tilde{\beta}_{41}}{2},$$

$$\beta_{62} = \beta_{53},$$

$$\tilde{\beta}_{62} = \tilde{\beta}_{53},$$

$$\beta_{63} = 36l_1^2l_2^2m_2^2 + 144l_1^2l_2^2m_2m_3 + 48l_1^2l_2^2m_1m_2 + 144l_1^2l_2^2m_1m_3 + 216l_1^2l_2l_3m_2m_3 + 144l_1^2l_2l_3m_1m_3 + 144l_1^2l_3^2m_2m_3 + 36l_1^2l_3^2m_3^2 + 48l_1^2l_3^2m_1m_3 - 144l_2^3l_3m_2m_3 - 432l_2^3l_3m_3^2 - 432l_2^2l_3^2m_3^2 - 144l_3^3l_2m_3^2,$$

$$\tilde{\beta}_{63} = \frac{l_2l_3m_3\tilde{\beta}_{41}}{2}. \quad (A.2)$$

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the Department of Scientific and Industrial Research, India, for establishing the PRISM (Promoting Innovations in Individuals, Start-ups, and MSMEs) scheme under which this project work is carried out. The authors are also grateful for the amiable support of Mechatronics and Robotics Laboratory, IIT Guwahati, in performing the experiments.

References

[1] World Health Organization, "Global health estimates 2016: deaths by cause, age, sex, by country and by region, 2000-2016," WHO, Geneva, 2018.

[2] P. K. Jamwal, S. Hussain, and M. H. Ghayesh, "Robotic orthoses for gait rehabilitation: an overview of mechanical design and control strategies," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 234, no. 5, pp. 444–457, 2020.

[3] A. Plaza, M. Hernandez, G. Puyuelo, E. Garces, and E. Garcia, "Wearable rehabilitation exoskeletons of the lower limb: analysis of versatility and adaptability," *Disability and Rehabilitation: Assistive Technology*, pp. 1–15, 2020.

[4] B. Kalita, J. Narayan, and S. K. Dwivedy, "Development of active lower limb robotic-based orthosis and exoskeleton devices: a systematic review," *International Journal of Social Robotics*, pp. 1–19, 2020.

[5] J. F. Veneman, R. Kruidhof, E. E. Hekman, R. Ekkelenkamp, E. H. Van Asseldonk, and H. Van Der Kooij, "Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 379–386, 2007.

[6] M. Bortole, A. Venkatakrishnan, F. Zhu et al., "The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," *Journal of Neuroengineering and Rehabilitation*, vol. 12, no. 1, pp. 1–14, 2015.

[7] M. H. Hsieh, Y. H. Huang, C. L. Chao, C. H. Liu, W. L. Hsu, and W. P. Shih, "Single-actuator-based lower-limb soft exoskeleton for preswing gait assistance," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 5927657, 12 pages, 2020.

[8] A. Rastegarpanah, M. Saadat, and A. Borboni, "Parallel robot for lower limb rehabilitation exercises," *Applied Bionics and Biomechanics*, vol. 2016, Article ID 8584735, 10 pages, 2016.

[9] A. Rastegarpanah, M. Saadat, A. Borboni, and R. Stolkin, "Application of a parallel robot in lower limb rehabilitation: a brief capability study," in *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, pp. 1–6, Amritapuri, India, 2016.

[10] F. Aggogeri, A. Borboni, N. Pellegrini, and R. Adamini, "Design and development of a mechanism for lower limb movement," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 6, 2019.

[11] M. Cestari, D. Sanz-Merodio, J. C. Arevalo, and E. Garcia, "ARES, a variable stiffness actuator with embedded force sensor for the ATLAS exoskeleton," *Industrial Robot: An International Journal*, vol. 41, no. 6, pp. 518–526, 2014.

[12] F. Patané, S. Rossi, F. Del Sette, J. Taborri, and P. Cappa, "WAKE-up exoskeleton to assist children with cerebral palsy: design and preliminary evaluation in level walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 906–916, 2017.

[13] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the Berkeley lower extremity exoskeleton (BLEEX)," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[14] X. Ji, D. Wang, P. Li, L. Zheng, J. Sun, and X. Wu, "SIAT-WEXv2: a wearable exoskeleton for reducing lumbar load during lifting tasks," *Complexity*, vol. 2020, Article ID 8849427, 12 pages, 2020.

[15] A. Esquenazi, M. Talaty, A. Packel, and M. Saulino, "The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury," *American Journal of Physical Medicine & Rehabilitation*, vol. 91, no. 11, pp. 911–921, 2012.

[16] M. S. Amiri, R. Ramli, and M. F. Ibrahim, "Hybrid design of PID controller for four DoF lower limb exoskeleton," *Applied Mathematical Modelling*, vol. 72, pp. 17–27, 2019.

[17] J. Narayan, A. Kalani, and S. K. Dwivedy, "Reference trajectory based Jacobian transpose control of a novel lower limb exoskeleton system for children," in *2019 5th international conference on signal processing, computing and control (ISPCC)*, pp. 102–107, Solan, India, 2019.

[18] M. O. Ajayi, K. Djouani, and Y. Hamam, "Bounded control of an actuated lower-limb exoskeleton," *Journal of Robotics*, vol. 2017, 20 pages, 2017.

[19] P. Yang, G. Zhang, J. Wang, X. Wang, L. Zhang, and L. Chen, "Command filter backstepping sliding model control for lower-limb exoskeleton," *Mathematical Problems in Engineering*, vol. 2017, Article ID 1064535, 10 pages, 2017.

[20] Y. Long, Z. J. Du, W. D. Wang, and W. Dong, "Robust sliding mode control based on GA optimization and CMAC compensation for lower limb exoskeleton," *Applied bionics and biomechanics*, vol. 2016, Article ID 5017381, 13 pages, 2016.

[21] J. Liu, J. Wang, and G. Zhang, "Event-triggered sliding mode controller design for lower limb exoskeleton," in *2020 39th Chinese Control Conference (CCC)*, pp. 593–598, Shenyang, China, 2020.

[22] J. Wu, J. Gao, R. Song, R. Li, Y. Li, and L. Jiang, "The design and control of a 3DOF lower limb rehabilitation robot," *Mechatronics*, vol. 33, pp. 13–22, 2016.

[23] W. Sun, J. W. Lin, S. F. Su, N. Wang, and M. J. Er, "Reduced adaptive fuzzy decoupling control for lower limb exoskeleton," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1099–1109, 2021.

[24] X. Zhang, H. Wang, Y. Tian, L. Peyrodie, and X. Wang, "Model-free based neural network control with time-delay estimation for lower extremity exoskeleton," *Neurocomputing*, vol. 272, pp. 178–188, 2018.

[25] J. Narayan and S. K. Dwivedy, "Towards neuro-fuzzy compensated PID control of lower extremity exoskeleton system for passive gait rehabilitation," *IETE Journal of Research*, pp. 1–18, 2020.

[26] C. Chen, S. Zhang, X. Zhu, J. Shen, and Z. Xu, "Disturbance observer-based patient-cooperative control of a lower extremity rehabilitation exoskeleton," *International Journal of Precision Engineering and Manufacturing*, vol. 21, no. 5, pp. 957–968, 2020.

[27] S. Han, H. Wang, Y. Tian, and N. Christov, "Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton," *ISA Transactions*, vol. 97, pp. 171–181, 2020.

[28] R. Nataraj and A. J. van den Bogert, "Simulation analysis of linear quadratic regulator control of sagittal-plane human walking—implications for exoskeletons," *Journal of Biomechanical Engineering*, vol. 139, no. 10, 2017.

[29] N. Ajjanaromvat and M. Parnichkun, "Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation," *Mechatronics*, vol. 51, pp. 85–96, 2018.

[30] J. Gupta, R. Datta, A. K. Sharma, A. Segev, and B. Bhattacharya, "Evolutionary computation for optimal LQR weighting matrices for lower limb exoskeleton feedback control," in *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 24–29, New York, NY, USA, 2019.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9782627, 1 page
https://doi.org/10.1155/2023/9782627

Retraction

Retracted: Development of High Accuracy Classifier for the Speaker Recognition System

Applied Bionics and Biomechanics

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

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] R. T. Al-Hassani, D. C. Atilla, and Ç. Aydin, "Development of High Accuracy Classifier for the Speaker Recognition System," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5559616, 10 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 5559616, 10 pages
https://doi.org/10.1155/2021/5559616

Research Article

Development of High Accuracy Classifier for the Speaker Recognition System

Raghad Tariq Al-Hassani,[1,2] **Dogu Cagdas Atilla**,[1] **and Çağatay Aydin**[1]

[1]*Faculty of Engineering, Altinbas University, Istanbul 34676, Turkey*
[2]*Ministry of Higher Education and Scientific Research in Iraq, Minister Office, Baghdad, Iraq*

Correspondence should be addressed to Raghad Tariq Al-Hassani; eng_raghadtariq@yahoo.com

Received 17 January 2021; Revised 9 March 2021; Accepted 5 April 2021; Published 20 May 2021

Academic Editor: Mohammed Yahya Alzahrani

Speech signal is enriched with plenty of features used for biometrical recognition and other applications like gender and emotional recognition. Channel conditions manifested by background noise and reverberation are the main challenges causing feature shifts in the test and training data. In this paper, a hybrid speaker identification model for consistent speech features and high recognition accuracy is made. Features using Mel frequency spectrum coefficients (MFCC) have been improved by incorporating a pitch frequency coefficient from speech time domain analysis. In order to enhance noise immunity, we proposed a single hidden layer feed-forward neural network (FFNN) tuned by an optimized particle swarm optimization (OPSO) algorithm. The proposed model is tested using 10-fold cross-validation over different levels of Adaptive White Gaussian Noise (AWGN) (0-50 dB). A recognition accuracy of 97.83% was obtained from the proposed model in clean voice environments. However, a noisy channel is realized with lesser impact on the proposed model as compared with other baseline classifiers such as plain-FFNN, random forest (RF), K-nearest neighbour (KNN), and support vector machine (SVM).

1. Introduction

Voice is the oldest method of communication reported in human history on earth. It was enforced by the fact that humans continuously need to share their feelings and requirements for surviving. Voice communication is inspired by the fact that an enormous amount of information can be exchanged which makes it the optimum medium of communication over other possible alternatives such as writing and even modern day communication facilities such as electronic texting [1]. The concept of language was invented due to the diversity of human descent exhibiting various geographical areas. Languages vary in accordance with the surroundings and nature of the place exhibited by the human. On these bases, different speaking tongues and accents are recognized in today's world [2]. Speaker recognition systems are further implemented as an electronic solution for supporting security and privacy enforcement systems. It is preferred to be used by service providers for securing personal data and for preventing any autonomous attack. A voice recognition system is based on the fact that the voice generation system is uniquely structured in every human [3]. A vocal track is the main contributor in the voice creation process; it works as air blows over a set of vocal cords making it vibrate, and hence, vibration produces the tone of voice. Voices flow henceforth through the vocal track propagating through the throat and mouth. Speech tone is directly impacted by the shape of the vocal tacks and by the objects available inside the mouth such as the number of teeth [4]. A speech system is encountered for various numbers of challenges that are vital to the modelling process of the speaker. The main obstacle of speaker identification system is the random nature of voice signals. Those signals are termed by their randomness nature, which can be realized from the electrical property fluctuation throughout time. The information of the spectrum in a speech signal can vary within the time period, so

it is difficult to rely on frequency information for modulating the process of the voice track [5, 6]. The speaker recognition process involves two different phases of processing. The first phase of processing is called a text-dependent speaker identification system which is simply depending on the exact (being told) voice imprint in both testing and training. Text-dependent speaker recognition is implementable using the time domain analysis, and the drawback of this method is that complete matching between the test and train data is required which is practically not possible [7, 8]. On the other hand, in text-independent speaker identification, speakers can be recognized based on their signal frequency analysis. This is commonly done using the frequency domain analysis such as Fourier transform. The main drawback of this method in such a domain is the inconsistency with the practical reality of voice nature. As the voice signal is time variant, signal frequency (spectrum) information is changed by time [9]. With respect to the above scenarios, the traditional models seem unable to accommodate the varying nature of voice signals. However, traditional models can use the Fourier transformation as an essential method to analyse the frequency. Other methods such as zero-crossing, convolution, and correlation are commonly used as a time domain method to analyse the speech signal. Conventional voice recognition models depending on the aforementioned approaches are not consistence for overcoming the time-varying nature of voice signals [10]. From the popular acoustic feature extraction methods, LPCC is used in [11] for modelling the speech production process. LPCC is abled for producing linear prediction coefficients sensitive to the spectral smoothness and spectral bias. In [12, 13], LPCC features are fused with MFCC features with efforts to improve the later features while implementing a Gaussian mixture universal background model (GMM-UBM). In order to improve the MFCC features, hamming windows have been replaced by multiple windows in order to achieve smoother spectrum results. In [14], bottleneck features are extracted from speech signals by using a deep neural network; however, the same is concatenated with MFCC features for speaker identification improvement. Feature sections from the fused sets of features are performed using kernel-based learning (e.g., support vector machine (SVM)) and using the reduced features of the SR model [15]. Another approach is illustrated in [16] for SR performance enhancement using speech data from different channels for constructing the acoustic features. Dimensionality reduction techniques are being proposed in [17] by adopting a PCA algorithm. Computational reduction is the key solution for performance enhancement; one of the efficient approaches for the same is frame rate reduction of the speech signal [18]. Dimensionality reduction is performed in [19] by manipulating the speaker or utterance layer by reducing the channel noise impact (channel scoring) [20]. The machine learning algorithms, namely, naïve Bayes (NB), support vector machine (SVM), and K-nearest neighbours (KNN), are presented in classification algorithms to predicates [21, 22].

In this paper, we are developing a smart voice recognition system using a deep learning approach for predicting the speakers. The deep learning model performs the recognition tasks as the model is trained with the features of the speech signal. A feed-forward neural network is used in an optimized version for serving the required recognition purpose.

2. Voice Processor

Preprocessing of a voice signal refers to all the changes that apply to the signal before it is actually passed to the analyser. However, preprocessing is proceeded by sampling where the signal is converted to a set of samples for efficient analysis. Herein, as the speaker recognition system might deal with a large number of speakers, the data set preparation is an important step in preprocessing [23]. Hereinafter, points are noteworthy and set to be covered while preprocessing. The data set involves 250 voice clips recorded from many speakers, and the same clips need to be ordered and named in numerical or alphabetical form in order to feed them into the processing system easily. An index is associated with the data set that enlists all the speech signals' names. If the same is not available on the data set, it needs to be created. Such an index can be formed as character strings more likely as index = [voice1, voice2, voice3, voice4, voice5,−−−−,voice − n]; in case the index is available by default along with the data set, index verification is to be started for matching the index with the voice clips in the database, as, in many cases, the index may lose some voice clips and that will create an error on the further process. Figure 1 depicts the process of data set preprocessing.

3. Hybrid Speech Features

3.1. Time Domain. Fundamental frequency is one of the interesting features in speech signal; it can be produced in time domain analysis using the cross-correlation approach. The aim of this feature is to identify the fundamental frequency in the speech signal [24]. The fundamental frequency is also called pitch frequency and is calculated using the pitch period. This period lies on the cross-correlation signal and represents the time between the minimum local maxima and the maximum local maxima on the signal corpus. Assuming that the sampled speech signal is represented by $S[n]$, let the $S'[n] = S(n-1)$ be the time-shifted copy of the same signal. Cross-correlation ($C[n]$) can be given in

$$C[n] = \sum_{n=1}^{N} S[n] \cdot S'[n]. \tag{1}$$

Figures 2 and 3 depict the resultant of cross-correlation between a speech signal and the same copy of it in shifted samples (phase). The next step is to evaluate the peaks of the resultant signal; those peaks are named as maximum local maximum as in Figure 4 and minimum local maxima as in Figure 2.

3.2. Mel Domain. Mel scale is a popular terminology in speech context; it simulates the value of human ear sensation to the speech signal. Mel frequency is different from the local frequency of the signal, and Mel spectrum coefficients formulate the Mel set which represents the amount of ear sensitivity of the human ear to a particular voice signal. Therefore, each

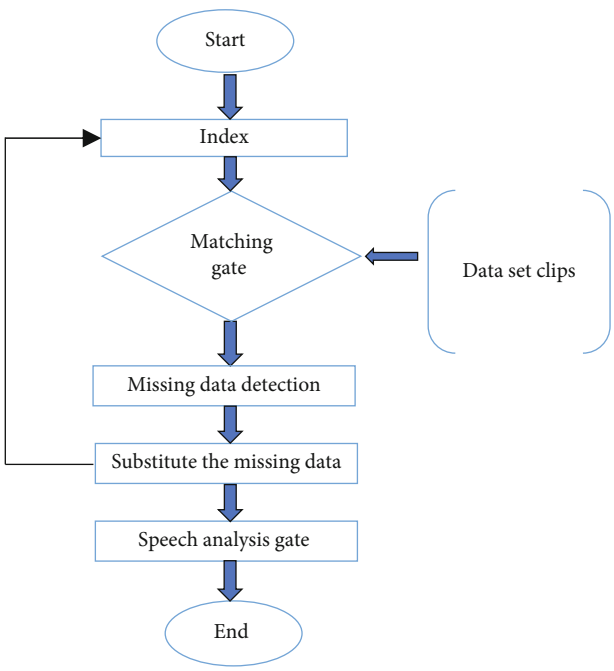


Figure 1: Data set preparation action plan.

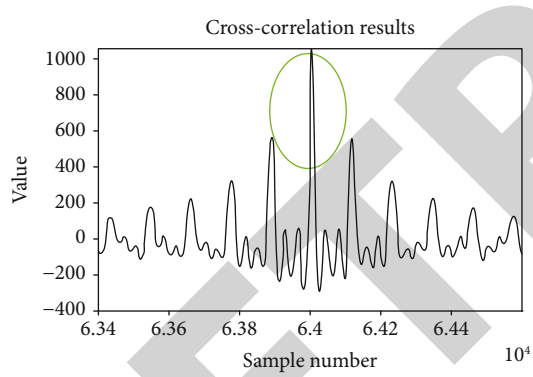


Figure 2: Peak (maximum) local maxima.

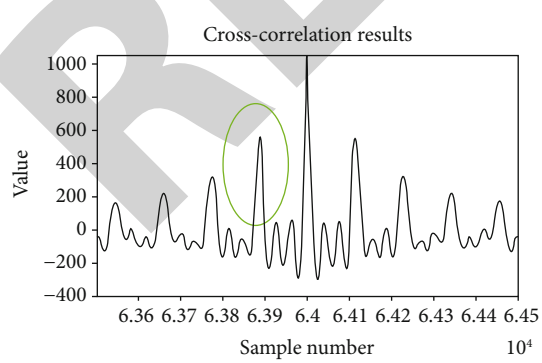


Figure 3: Bottom (minimum) local maxima.

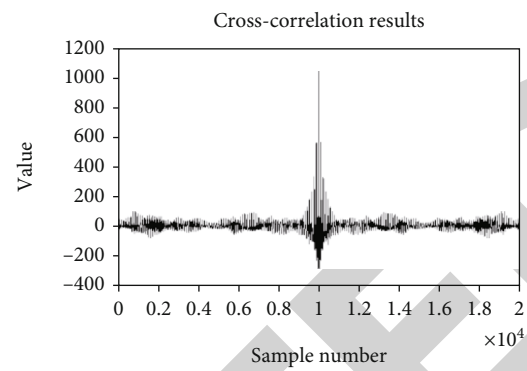


Figure 4: Cross-correlation resultant signal.

voice signal has different effects on the ear, and Mel frequency spectrum coefficients represent the ear response in the form of a vector of eight values. The Mel frequency spectrum coefficient vector can be represented in the following equation [25]. In order to derive the Mel frequency spectrum coefficients which represent the ear response to the voice firstly, the voice signal is passed through the preemphasis filter as an attempt to amplify the low power samples. This process is important for the reason that the voice may include low frequency segments resulting in voice waveforms (samples) due to whispering or a none loud voice [26]. However, the preemphasis filter will take the voice signal and attempt to unify the power so that the power can be distributed uniformly among the frequencies as demonstrated in Figure 5. Signal passing from the preemphasis filter can result in a new version of the signal with an enhanced signal to noise ratio. As low power frequencies are more susceptible to noise impact, the preemphasis filter produces a signal with higher power for those slots, and hence, the ratio of the signal power to the noise power will be larger [27]. As soon as a signal results with good SNR (signal to noise ratio) from the preemphasis filter, signal framing is the next process in the Mel frequency spectrum coefficient algorithm. However, since the speech signal is a time variant signal which means that the frequency keeps changing with time and a nonfixed frequency repose can be ensured, researchers agreed to the fact that the speech signal remains stationary in a very short time frame more likely within 25 milliseconds. For this purpose and in order to determine the signal properties as a time invariant signal, framing of the signal is a must. This window is called as the hamming window $T[n]$ and can be presented by the following equation (2); the samples and Fourier transform of the hamming window is depicted in Figure 6.

$$\begin{aligned} M_f &= [m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8], \\ T[n] &= \left\{46 \times 10^{-2} \times \cos\left[(2n\pi)(N-1)^{-1}\right]\right\} + 0.56, \end{aligned} \tag{2}$$

where $T[n]$ represents the hamming window and N is the total number of samples in the speech signal.

In further steps, each hamming window is converted from samples into a spectrum using the fast Fourier transform (FFT) as given in

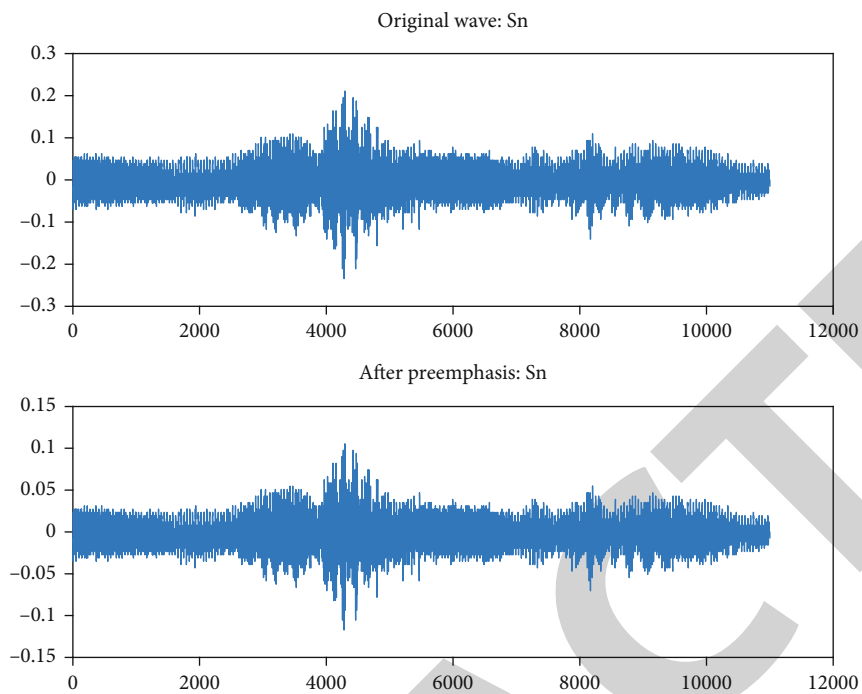


Figure 5: Preemphasis filter input and output signals.

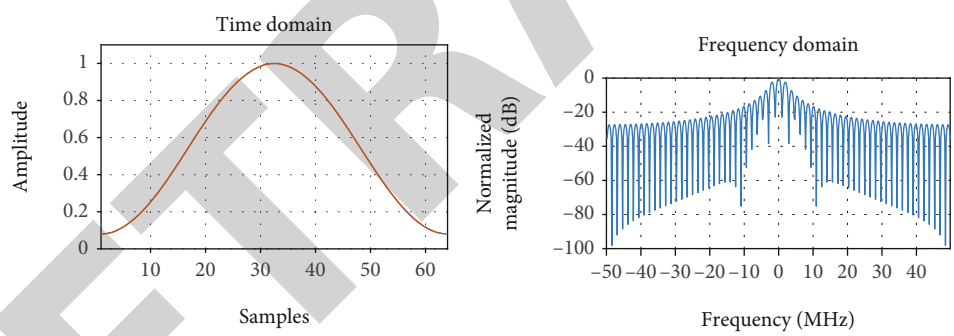


Figure 6: Hamming window (single window) in both sample and frequency domains.

$$S[n] = \sum_{n=1}^{N} s[n]\, e^{2n\pi}, \tag{3}$$

where $S[n]$ is the fast Fourier transform of the sampled signal $s[n]$; furthermore, Mel frequency is derived from the above components using the Mel conversion equation as in (3).

$$f_m = 2595 \log\left(1 + \frac{f_n}{700}\right), \tag{4}$$

where f_m is frequency of speech signal in the Mel scale and f_n is the Hertz scale speech frequency.

The last step in the Mel frequency spectrum algorithm is to simulate the human ear perception to the voice signal. Therefore, the filter bank in used to perform the same. The filter bank with the transfer functions given below is implemented to produce the human ear voice perception. The filter bank response to the input can be demonstrated in Figure 7. The same is representing the Mel scale of the spectrum according to the ear which is usually responding to the voice signal in low and high frequencies, so the ear as in the figure can respond with a narrow response to low frequencies and give a wide response to high frequencies, and accordingly, for each voice signal, there will be a different response.

4. Feature Mapping

Features of speech signal are generated from both the Mel frequency spectrum coefficient (MFCC) method and the fundamental frequency method (pitch frequency). The Mel scale

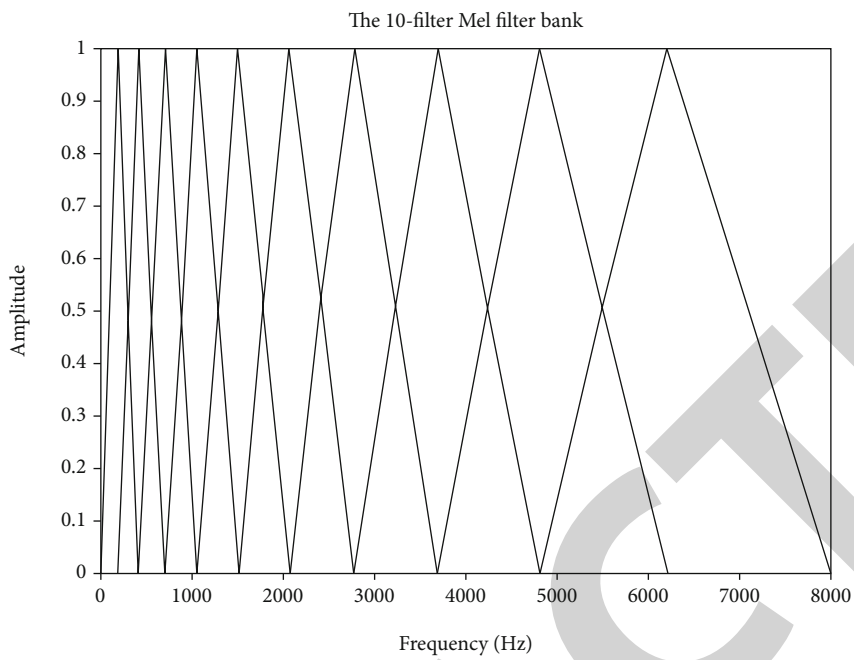


Figure 7: Filter bank response on the Mel frequency scale of voice signal.

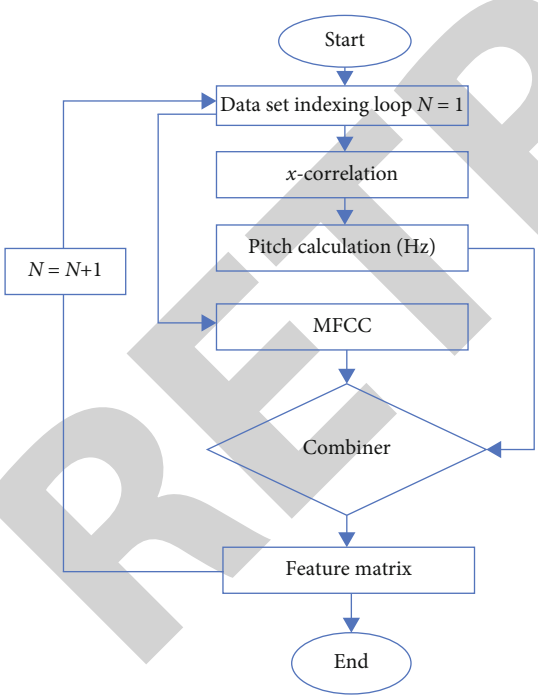


Figure 8: Feature formation of the empirical model.

of the speech signal is obtained from the MFCC method which represents the human ear response to speech, and it basically divides the speech signal into a set of windows using a triangular filter bank and returns different window sizes for the higher and lower frequencies. In other words, MFCC may segregate the speech signal according to the frequency range presented on it and depends on the Mel scale (human ear perception). Furthermore, the pitch frequency is also obtained from the speech signal; it produces a single value in hertz; the fundamental frequency is vital in speaker recognition because it presents the minimum frequency of vocal chord vibration [16, 17]. The combination between the two aforementioned methods is performed as the pitch frequency may be affected by the noise association, and hence, it might not return the exact character of the speech signal. Accordingly, features from both the pitch frequency method and the Mel frequency spectrum coefficient method are obtained and used for recognition work. For 250 speech signals and nine features for each signal, a total of 2250 features (elements) are generated from the speaker model. Figure 8 depicts the process of feature combination.

Table 1: FFNN first experiment parameter table.

Term	Values
Total layers	Three
Node distribution	Thirty, twenty, and one
Learning algorithm	Built-in LM
MSE goal	1*e*-200
Repetitions	100

5. Model Optimizer

5.1. Plain FFNN. The feed-forward neural network model is used in this project to predict the speaker characters.

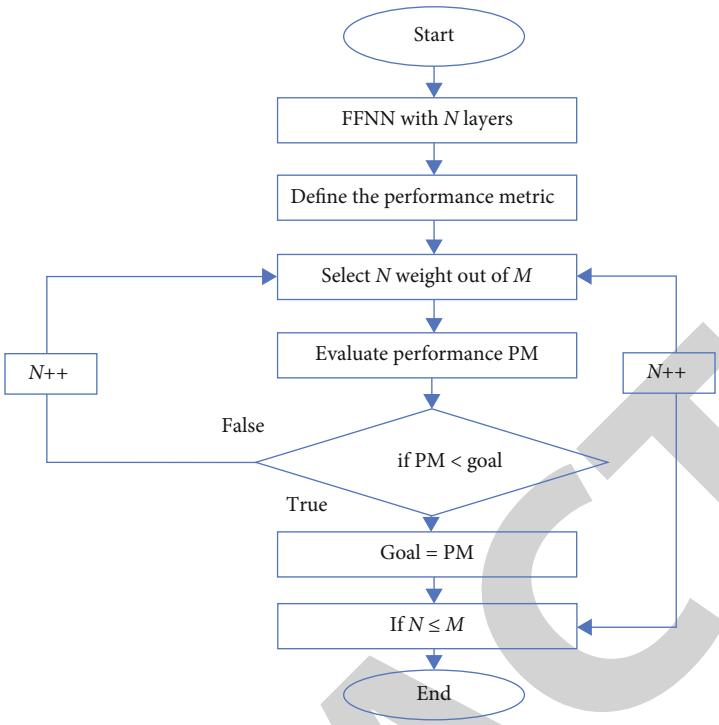


Figure 9: Weight freezing program flow diagram.

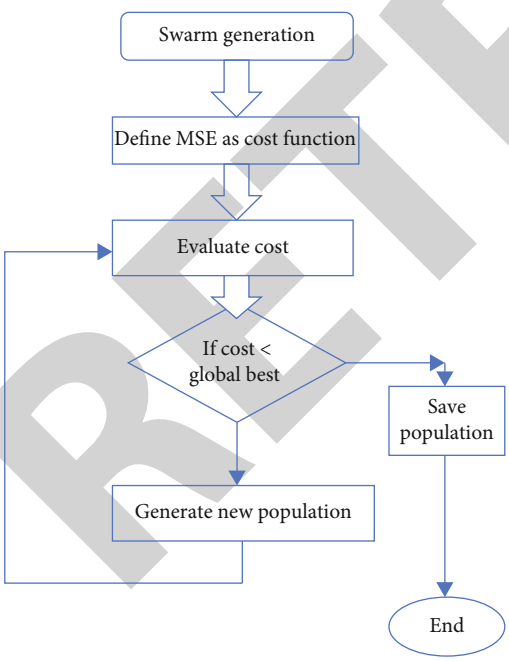


Figure 10: PSO-based feed-forward neural network optimization.

However, the features of each speaker are applied to the model for training; model is established according to the parameters in Table 1. A three-layer FFNN with 30, 20, and 1 nodes are made; the reason for selecting this number of nodes is to reduce the delay taken by the model at the training and testing stages. According to Figure 9, the LM algorithm is used to train the model, and the target performance (mean square error) is made equal to 1*e*-29. Three experiments are made as the time line of enhancement to the model; in each experiment, the model is upgraded for the sake of performance enhancement. Therefore, the first experiment relied on the parameters given in Table 1. During the training stage of this experiment, it was noticed that results were varying every time the model is restarted since the LM algorithm is allotting the weight values randomly, and it repeats the same whenever the model is used. In order to monitor the model performance and to tackle this random nature of the results, the experiment is repeated for 100 times and the results are recorded, and then, the average results are used for examining the model performance [28, 29].

Table 2: Results of performance metrics for all algorithms.

Tool	Accuracy	MSE	Time	Epochs	RMSE
FFNN	78.59	5.72	2.52	12.00	2.39
MFFNN	89.25	2.11	2.10	10.00	1.45
PSO-FFNN	97.83	1.77	0.97	14.00	1.33

5.2. Model Freezing. A second experiment was done according to the results monitored from the first experience; the performance of the neural network is realized for all 100 repetitions; hence, the weight of every repetition is recorded. However, weight freezing technology involves presetting of

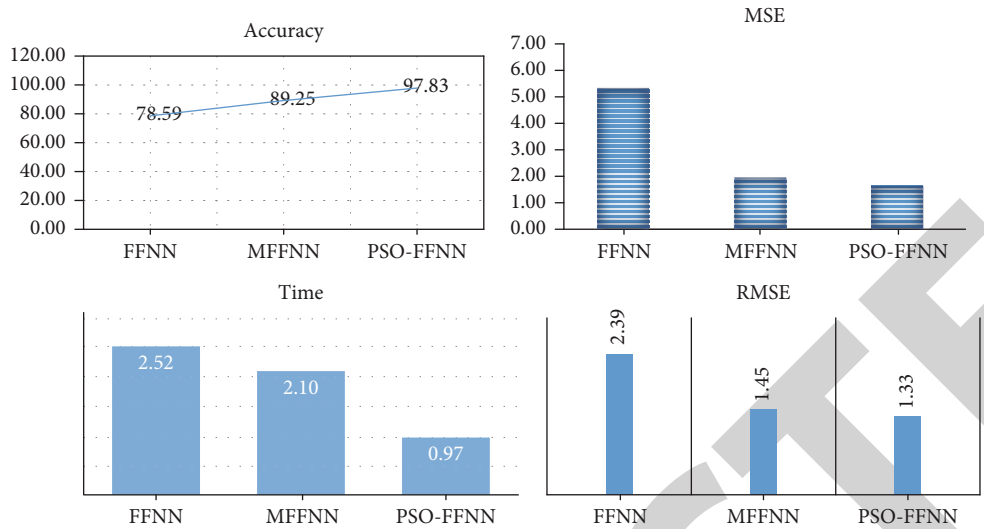


Figure 11: Graphical representation of the performance metrics.

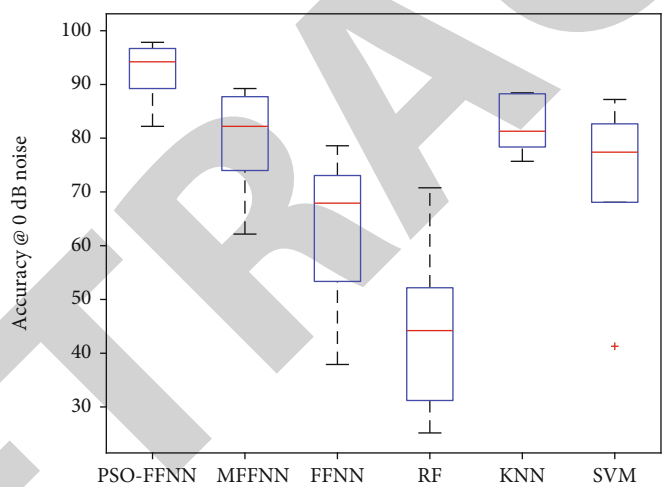


Figure 12: Accuracy scores for all proposed classifiers under clean voice channel.

the weight values of the FFNN model to those weight values that return the best cost. The technology of freezing dispenses with the need for a training algorithm as ready-made weights can be fed into the model with predetermined performance. The selection of proper weight values depends totally on the previous experiment which involves the record of weights and their cost values. Figure 9 demonstrates the process of model freezing. The figure shows that the program may be established to test all the weights and to select the weight that yields the best cost.

The third experiment is made as another attempt to enhance the performance of prediction, so the new algorithm is used as a training algorithm. Particle swarm optimization algorithm is proven to have noticeable performance in optimizing the feed-forward neural network. Figure 10 shows the flow diagram of the PSO-FFNN algorithm. The PSO algorithm is made to produce the weight values that yield an enhanced performance; the following steps are taken to execute the algorithm.

6. Results and Discussion

As discussed in the previous sections, the feed-forward neural network is examined under several performance metrics in order to identify the best model that is capable of predicting the speaker identity. Three models are used, namely, the plain feed-forward neural network, the weight freezing-based feed-forward neural network, and ultimately the particle swarm optimization-based feed-forward neural network. Results of those models' performances are listed in Table 2. It is observed that the accuracy of speaker prediction is optimum at FFNN while PSO is used for performance optimization; hence, a 97.83% accuracy is recorded from the aforementioned model. In other models, namely, plain

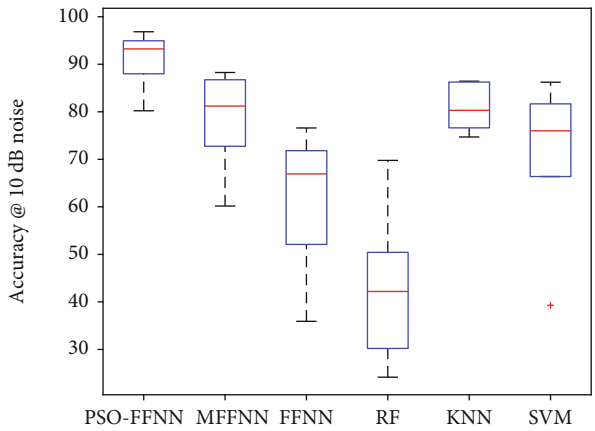


Figure 13: Accuracy scores for all proposed classifiers under 10 dB noise conditions.

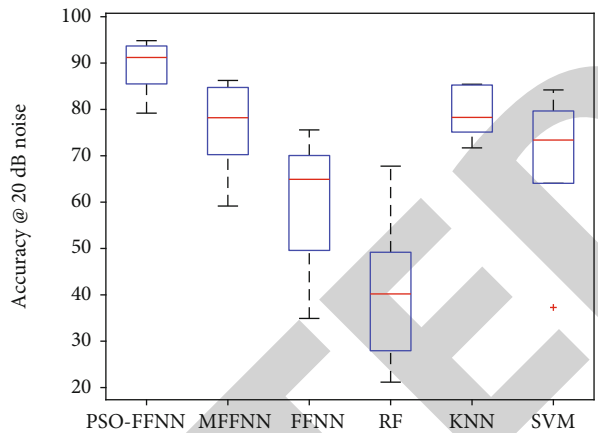


Figure 15: Accuracy scores for all proposed classifiers under 20 dB noise conditions.

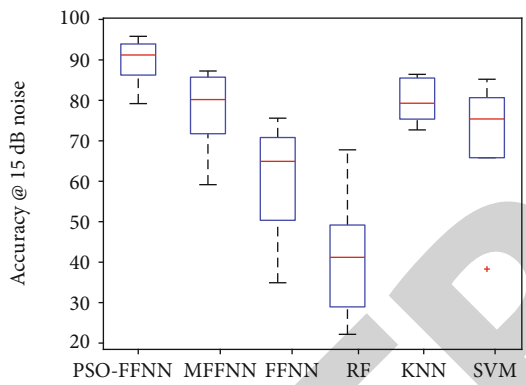


Figure 14: Accuracy scores for all proposed classifiers under 15 dB noise conditions.

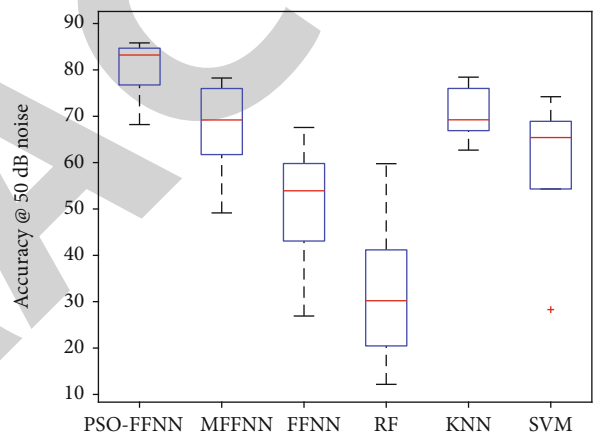


Figure 16: Accuracy scores for all proposed classifiers under 50 dB noise conditions.

FFNN and modified FFNN (MFFNN), the recorded accuracies of speaker recognition are, respectively, 78.59% and 89.25%. The optimum accuracies were detected in the PSO-FFNN model. The noble method adopted in the PSO algorithm for tuning up FFNN weight coefficients has produced the coefficients of weights resulting in the least error in the prediction. By using the plain FFNN and weight freezing method in MFFNN, the results of the prediction of both models have been analysed and used over the PSO swarm (weight) generator for building the seed of generated random swarms. On the other hand, it was realized that the time taken for predicting the speaker identity using the PSO-FFNN model was 0.97, which means that the proposed model was capable of performing the required tasks in minimum time compared to the other models. The rapid operation performance of the PSO-FFNN model has been reported since the FFNN model will be totally relying on the PSO algorithm for producing the weight coefficients without the need of performing standalone (internal) weight generation. Eventually, the mean square error (MSE) and root mean square error (RMSE) metrics are also found to be minimal [30–32]. MSE and RMSE metrics imply lesser error existence on the proposed model predictions. The results are graphically demonstrated in Figure 11.

Furthermore, different classifiers were used for predicting the speaker identity such as the random forest (RF) algorithm, K-nearest neighbour (KNN), and support vector machine (SVM). In order to evaluate the performance of all proposed tools, k-means validation is used where various input styles were tested. The accuracy scores of our proposed model as well as the other algorithms are demonstrated 10-folds in various noise conditions as in Figures 12–16. The accuracy measures under clear voice environments are depicted in Figure 12. The accuracy measures under 10 dB AWGN voice environments are depicted in Figure 13. The accuracy measures under 15 dB AWGN voice environments are depicted in Figure 14. The accuracy measures under 20 dB AWGN voice environments are depicted in Figure 15. The accuracy measures under 50 dB AWGN voice environments are depicted in Figure 16.

The proposed classifier e.g. PSO-FNN had yielded best accuracy score during all noise conditions as demonstrated in Table 3.

Table 3: Mean accuracy measure for all proposed classifiers at different noise levels.

Algorithm/accuracy	PSO-FFNN	MFFNN	FFNN	RF	KNN	SVM
Noise @ 0 dB	92.4280	79.7520	62.8260	43.8660	82.5720	72.8080
Noise @ 10 dB	91.0280	78.5520	61.4260	42.4660	80.9720	71.4080
Noise @ 15 dB	89.6280	77.5520	60.0260	41.0660	79.9720	70.6080
Noise @ 20 dB	89.2280	76.3520	59.6260	40.4660	79.3720	69.2080
Noise @ 50 dB	80.2280	67.5520	50.8260	32.0660	70.7720	59.6080

7. Conclusions

Speaker recognition is a vital stage in many personal authentication and security systems; it builds the logic of person verification using their biometrical features, more specifically, voice features. The entity of the speaker recognition system involves the major two stages called feature extraction and speaker classification. However, these processes may begin with voice preprocessing involving the preparation of voice signals and set them together in the data set. Speech features include time domain and frequency domain processing; each is an integral part of speech processing and can be used to form a final recognition system. Speech signal preprocessing is about signal enhancement by reducing the noise level and removing other unnecessary information such as background noise and other associates. It might involve silence removal, which deletes the samples of low power that represent the silence in the uttered sentence (breaks while speaking). These processes are important to enhance the signal quality, which makes the signal more readable by the further process (stages). However, preprocessing is important to reduce the extra computation power that might utilize the capacity of the processor and distort the performance of the entire system. On the other hand, several approaches are knocked out to perform feature extraction of the speech signal. The fundamental frequency and Mel frequency cepstrum coefficients are the main approaches employed over this system, whether deep learning approaches are however employed for speaker classification tasks (mapping the features to particular speakers). FFNN is used for mapping the features to their perspective speaker, and the results have shown that PSO-FFNN outperformed the other techniques used in this paper.

Data Availability

The used data is public and available online freely.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Abou-Zleikha, Z.-H. Tan, M. G. Christensen, and S. H. Jensen, "A discriminative approach for speaker selection in speaker de-identification systems," in *23rd European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015.

[2] X. Fan and J. H. L. Hansen, "Speaker identification with whispered speech based on modified LFCC parameters and feature mapping," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, 2009.

[3] B. Wang, J. Zhao, X. Peng, and B.-c. Li, "A novel speaker clustering algorithm in speaker recognition system," in *IEEE Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian*, Dalian, China, 2016.

[4] B. G. Nagaraja and H. S. Jayanna, "Efficient window for monolingual and crosslingual speaker identification using MFCC," in *IEEE International Conference on Advanced Computing and Communication Systems*, Coimbatore, India, 2015.

[5] E. B. Tazi and N. El Makhfi, "An hybrid front-end for robust speaker identification under noisy conditions," in *2017 Intelligent Systems Conference (IntelliSys)*, London, UK, 2017.

[6] R. Martsyshyn, M. Medykovskyy, L. Sikora, Y. Miyushkovych, N. Lysa, and B. Yakymchuk, "Technology of speaker recognition of multimodal interfaces automated systems under stress," in *2013 12th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 447-448, Lviv, UKraine, 2013.

[7] V. M. Sardar and S. D. Shrbahadurkar, "Speaker identification using whispered speech," in *IEEE International Conference on Communication Systems and Network Technologies*, Pune, India, 2015.

[8] A. Maazouzi, N. Aqili, A. Aamoud, M. Raji, and A. Hammouch, "MFCC and similarity measurements for speaker identification systems," in *2017 International Conference on Electrical and Information Technologies (ICEIT)*, Rabat, Morocco, 2017.

[9] K. Daqrouq, W. Al-Sawalmeh, A.-R. Al-Qawasmi, and I. N. Abu-Isbeih, "Speaker identification wavelet transform based method," in *2008 5th International Multi-Conference on Systems, Signals and Devices*, Amman, Jordan, 2015.

[10] R. S. Mohsen Bazyar, "A new speaker change detection method in a speaker identification system for two-speakers segmentation," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang, Malaysia, 2014.

[11] M. J. Alam, T. Kinnunen, P. Kenny, P. Ouellet, and D. O'Shaughnessy, "Multitaper MFCC and PLP features for speaker verification using i-vectors," *Speech Communication*, vol. 55, no. 2, pp. 237–251, 2013.

[12] N. M. Omar and M. E. El-Hawary, "Feature fusion techniques based training MLP for speaker identification system," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, Canada, 2017.

[13] B. Dautrich, L. Rabiner, and T. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 793–807, 1983.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9816851, 1 page
https://doi.org/10.1155/2023/9816851

Retraction

Retracted: Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets

Applied Bionics and Biomechanics

Received 31 January 2023; Accepted 31 January 2023; Published 5 February 2023

Applied Bionics and Biomechanics has retracted the article titled "Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets" [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process and the article is being retracted with the agreement of the Chief Editor.

The authors agree to the retraction.

References

[1] S. N. Alsubari, S. N. Deshmukh, M. H. Al-Adhaileh, F. W. Alsaade, and T. H. 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.

[2] L. Ferguson, "Advancing Research Integrity Collaboratively and with Vigour," 2022, https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 5522574, 11 pages
https://doi.org/10.1155/2021/5522574

Research Article

Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets

Saleh Nagi Alsubari,[1] **Sachin N. Deshmukh**,[1] **Mosleh Hmoud Al-Adhaileh**,[2] **Fawaz Waselalla Alsaade**,[3] **and Theyazn H. H. Aldhyani**[4]

[1]*Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India*
[2]*Deanship of E-Learning and Distance Education King Faisal University Saudi Arabia, Al-Ahsa, Saudi Arabia*
[3]*College of Computer Sciences and Information Technology, King Faisal University, Hofuf, Saudi Arabia*
[4]*Community College of Abqaiq, King Faisal University, P.O. Box 400, Al-Ahsa, Saudi Arabia*

Correspondence should be addressed to Saleh Nagi Alsubari; salehalsubri2018@gmail.com

Received 28 February 2021; Revised 20 March 2021; Accepted 5 April 2021; Published 15 April 2021

Academic Editor: Fahd Abd Algalil

Online product reviews play a major role in the success or failure of an E-commerce business. Before procuring products or services, the shoppers usually go through the online reviews posted by previous customers to get recommendations of the details of products and make purchasing decisions. Nevertheless, it is possible to enhance or hamper specific E-business products by posting fake reviews, which can be written by persons called fraudsters. These reviews can cause financial loss to E-commerce businesses and misguide consumers to take the wrong decision to search for alternative products. Thus, developing a fake review detection system is ultimately required for E-commerce business. The proposed methodology has used four standard fake review datasets of multidomains include hotels, restaurants, Yelp, and Amazon. Further, preprocessing methods such as stopword removal, punctuation removal, and tokenization have performed as well as padding sequence method for making the input sequence has fixed length during training, validation, and testing the model. As this methodology uses different sizes of datasets, various input word-embedding matrices of n-gram features of the review's text are developed and created with help of word-embedding layer that is one component of the proposed model. Convolutional and max-pooling layers of the CNN technique are implemented for dimensionality reduction and feature extraction, respectively. Based on gate mechanisms, the LSTM layer is combined with the CNN technique for learning and handling the contextual information of n-gram features of the review's text. Finally, a sigmoid activation function as the last layer of the proposed model receives the input sequences from the previous layer and performs binary classification task of review text into fake or truthful. In this paper, the proposed CNN-LSTM model was evaluated in two types of experiments, in-domain and cross-domain experiments. For an in-domain experiment, the model is applied on each dataset individually, while in the case of a cross-domain experiment, all datasets are gathered and put into a single data frame and evaluated entirely. The testing results of the model in-domain experiment datasets were 77%, 85%, 86%, and 87% in the terms of accuracy for restaurant, hotel, Yelp, and Amazon datasets, respectively. Concerning the cross-domain experiment, the proposed model has attained 89% accuracy. Furthermore, comparative analysis of the results of in-domain experiments with existing approaches has been done based on accuracy metric and, it is observed that the proposed model outperformed the compared methods.

1. Introduction

The development of Web 4.0 has increased the activity of internet shopping through E-commerce platforms. Online reviews posted on E-commerce sites represent the opinions of customers, and now these opinions play a significant role in E-businesses because they could potentially influence customer-buying decisions. Business owners use online

customer reviews to detect product issues and to discover business intelligence knowledge about their opponents [1]. Fraudsters post fake comments termed misleading reviews to affect business by manipulating potential reputation of product brands. Fake reviews are divided into 3 types: (1) untrusted (fake) reviews, (2) review on product name only, and (3) nonreviews. The fake reviews are posted deliberately to mislead and deceive buyers and consumers. These reviews contain unjust positive reviews for particular desired products to promote them and provide unfavorable reviews to worthy products for deprecating. Hyperactive fake reviews are linked to this type of review. Reviews on products brand only are the second version of fake reviews that can be created to manipulate the brands of products. Nonreviews are composed of two subsets, namely, (a) advertisement and (b) unrelated reviews [2]. Larger amounts of positive reviews lead to making the shoppers and customers buy products and enhance companies' financial benefits, whereas negative reviews can make customers to search for substitute products that way resulting in revenue loss. However, a significant number of review comments are generated across social media applications, adding complications for extracting views and difficulty in obtaining accurate findings. In addition, there is no monitoring on the reliability of digital content generated on the E-commerce websites, and this encourages the creation of several low-quality reviews possible. Various companies hire persons to write fake reviews for rising the purchasing of their online products and services. Such persons are known as fake reviewers or spammers, and the activities they perform are called review faking [3]. Therefore, the existence of fake and spam reviews makes the issue more considerable to be handled because they affect the possible changing of buying decision to customers and shoppers. A huge amount of positive reviews enable a consumer to purchase a product and improve the manufacture's financial profits, whereas negative reviews encourage consumers to search for substitutes and therefore causing financial losses [3, 4]. Consumer-generated reviews can get a huge influence on the reputation of products and brands, and hence, E-business companies would be motivated to produce positive fake reviews over their products and negative deceptive reviews over their competitors' products [5–7]. Electronic commerce sites have numerous ways of spamming with spam reviews, for instance, hiring expert persons who are specialized in generating fraud reviews, utilizing crowdsourcing websites to utilize review fraudsters, and using automation tool bots for feedback [8, 9]. The capability of vendors to produce misleading opinions as a way of either promoting their products or defame the reputation of their competitors is indeed worth remembering. Fake reviews have a tremendous influence on consumer satisfaction. For example, when a consumer is tricked or mislead via a fake review, a consumer will not utilize that E-commerce website again for purchasing. Ott et al. [10] reported that about 57% is the total average of testing accuracy of human judges for distinguishing fake reviews from truthful ones; therefore, further research is required in identifying misleading (fake) reviews. The limitations of existing studies of fake/deceptive/spam review detection are proposing automated methods for detecting and discriminating between fake and truthful reviews in online E-commerce websites. In order to mitigate the problems of online review mining systems, it is necessary for developing a model to detect and eliminate online fake reviews due to their effect on customers and E-commerce companies.

2. Literature Review

This section sheds light on methods and datasets used in previous studies for fake/spam review detection. Online product reviews are defined as guidelines that are widely used by a potential customer to make online purchasing that involves choosing or not to purchase a particular product, identifying the problems of manufacturing companies' products, and gaining intelligent information of their competitors in marketing research. Recently media news from the New York Times and the BBC have reported that counterfeit reviews are very widespread on E-commerce, for example, a photography company has recently been targeted by fake reviews of thousands of fraudulent [11]. Over the last two decades, fake/spam review detection has become a popular topic of study. Since fake reviews have such a significant effect on E-commerce and customers, several researchers have conducted several types of research on spam/fake review analysis.

2.1. Fake Review Detection Based on Machine Learning Methods. Jindal et al. [2] have presented first research towards spam review detection. The authors dealt with duplicate or near-duplicate in Amazon product reviews as fake reviews that were comprised attributes regarding the review text and reviewer. It has been applied the logistic regression technique for classifying reviews into truthful or fake with reaching 78% in the terms of accuracy.

Ott et al. [10] have utilized the crowdsourcing website (Amazon Mechanical Turk) to create a dataset, and the natural language processing tool was also used to obtain linguistic features from the review contents. They trained and compared several types of supervised machine learning techniques. However, the obtained results on real market datasets have not been very good. Lau et al. [11] have presented model for fake opinion identification using an LDA algorithm, namely, Topic Spam that can categorize the text of the review by calculating the likelihood of spam index to the little dissimilarity between the distribution of the keywords of the spam and the nonspam reviews.

Shojaee et al. [12] have proposed syntactic grammar and lexical-based attributes named stylometric attributes. These attributes are utilized to distinguish fake reviews from online hotel reviews. Using lexical features, the authors implemented SMO (sequential minimal optimization) and Naive Bayes methods for classifying the reviews into fake or truthful and the obtained results were 81% and 70% in the terms of F1-score, respectively. However, then, they have enhanced the performance of the model by merging lexical and syntactic features, and the SMO technique attained 84% F1-score. Xu and Zhao [13] suggested a parsing tree-based model for detecting and classifying fake reviews. They used textual features of the review text that were taken out from the parsing

tree by using syntactic analysis and implemented them to the model for identifying fake reviews. They just concentrated on textual features and ignored behavioral features. Allahbakhsh et al. [14] have examined the involvement of reviewers who place prejudiced score reviews on online rating classification systems collected through some attributes that can assist to point out a set of spammers. In their model, they utilized the Amazon log (AMZLog) with its dataset for carrying out the experiments. Duan and Yang [15] explored fake review identification based on reviews of the hotels. Through their method, they measured and used three features of the review text for detecting spam actions, general score, subscore, and review content. Feng et al. [16] have concentrated on dissemination footprints of reviewers and giving an association between distribution abnormalities and spammer's actions. Using the Markov model, they assessed the product review dataset collected from the Trip Advisor website. Barbado et al. [17] have proposed framework of significant features for deceptive review detection. Based on online Yelp product reviews, they carried out experiments using different supervised machine learning techniques. In terms of features, reviewer (personal, social, review activity, and trust) and review features (sentiment score) were used. Their experimental result showed that the AdaBoost algorithm provided the best performance with obtained 82% accuracy. Noekhah et al. [18] have presented a novel approach-based graph for detecting opinion spam in Amazon product reviews. First, they calculated an average value for review and reviewer features individually. Then, they asked three experts for assigning weight for every feature. Finally, they are multiplying the weight of the feature with its average value for calculating the spamicity for the review text and reviewer. Their approach achieved 93% accuracy. Alsubari et al. [3] have proposed different models based on supervised machine learning algorithms such as Random Forest, AdaBoost, and Decision tree. They used the standard Yelp product review dataset. The information gain method was applied as feature selection. From their experimental results, it is observed that the AdaBoost algorithm has provided the best performance by recording 97% accuracy.

2.2. Fake Review Detection Based on Deep Learning Methods. The use of deep learning neural network models for fake review identification has three key points. The first point is that deep learning models utilize real-valued hidden layers for automated feature compositions that can catch complicated global semantic data, which is difficult by utilizing typical specific handcrafted features. This provides an effective way in solving the shortcomings of different traditional models aforementioned above. The second point is that neural networks consider clustered word embedding as inputs that can be learned from raw text, hence mitigating the shortage of labeled data. The third point is that neural models can learn consistent text structure instantaneously. Based on Amazon electronic product review dataset, Hajek et al. [19] have implemented two neural network methods that were Deep Feed-Forward Neural Network and convolution neural network. Then, they extracted features from the review text set such as word emotions and n-grams. Their methodology results were 82% and 81% in terms of accuracy for DFFN and CNN methods, respectively. Goswami et al. [20, 21] have proposed Artificial Neural Network model to investigate the influences of social relations of reviewers for deception recognition at online customer reviews, and in their experiment, Yelp's review dataset was gathered and preprocessed. Then, they mined behavioral and social relation features of customers and applied the backpropagation neural network algorithm for classifying reviews into genuine and fake with a detection rate of 95% accuracy. Ren and Ji [22] have proposed a hyper deep learning model that is consisted of a gated recurrent neural network and convolutional neural network (GRNN-CNN) for detecting deceptive opinion spam on in-domain and in-cross domain datasets. They used doctors, restaurants, and hotels with a size of 432, 720, and 1280 reviews, respectively. By combining all these datasets, they applied their proposed method for classification of the reviews into spam and nonspam reviews. The best classification result obtained was 83% in terms of accuracy. Using the same datasets used in [22], Zeng et al. [23] have proposed a recurrent neural network-bidirectional long-short technique for deceptive review detection. They divided the review text into three parts: a first sentence, middle context, and last sentence. The best-achieved results of their method were 85% in terms of accuracy.

3. Methodology

Figure 1 shows the proposed methodology for fake review identification system that is consisted of six modules, namely, datasets, preprocessing, CNN-LSTM method, data splitting, evaluation metrics, and results. The details of the framework are discussed below.

3.1. Datasets. This module presents the datasets used in these experiments that are performed for the identification of deceptive/fake reviews. We have employed four standard fake review datasets: hotel, restaurant, Amazon, and Yelp.

3.1.1. Amazon-Based Dataset. This dataset is standard fake Amazon product reviews consists of 21,000 reviews (10500 truthful and 10500 fake), and each review has metafeature such as product Id, product name, reviewer name, verified purchase (no or yes), and rating value as well as a class label, while in the statistical analysis of the dataset, we found that the average rating value of the reviews was 4.13, and 55.7% of the data was recognized as verified purchases. The reviews of this dataset are equally distributed through 30 discrete product classifications (e.g., wireless, PC, health, etc.). Each product has 700 reviews (350 fake and 350 truthful reviews). Furthermore, the reference for labeling this dataset is the Amazon filtering algorithm that is employed by the Amazon website [20, 21, 24].

3.1.2. Yelp-Based Dataset. This dataset is standard fake electronic products reviews combined from four USA cities (Los Angeles, Miami, NY, and San Francisco) by Barbado et al. [17]. A reference for labeling this dataset is the Yelp filtering algorithm utilized by the http://Yelp.com/ website [25]. The dataset includes 9461 reviews and reviewers with

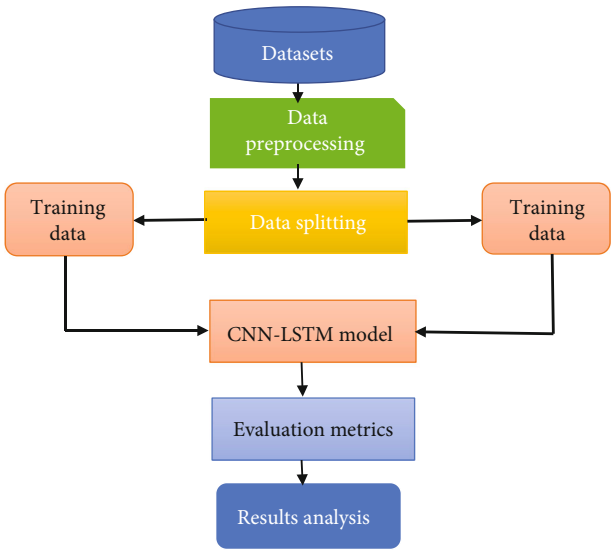


Figure 1: A Framework for the proposed methodology.

features such as rating value, reviewer name, verified purchase (yes or no), reviewer Id, product Id, review title, and review text as well as the class label.

3.1.3. Restaurant-Based Dataset. This dataset is fake restaurant reviews developed by Abri et al. [26, 27]. It includes 110 reviews belong to three local Indian restaurants and has organized a way to have an equivalent distribution of fake and real reviews (55 fake and 55 truthful). The metafeatures of the dataset are sentiment polarity that means positive or negative review, review text, reviewer Id, restaurant name, and a class label.

3.1.4. Hotel-Based Dataset. This is a publicly available standard dataset developed by Ott et al. [10, 28, 29]. It contains 1600 hotel reviews (800 truthful and 800 fake) collected from one of the popular hotel booking websites, that is, a Trip advisor. The authors of this dataset have refined all 5- and 3-star rated reviews from 20 hotels in Chicago city. The features of the dataset consist of review text, reviewer name, hotel name, sentiment polarity, and class label.

3.2. Data Preprocessing. The aim of preprocessing is applied to make the data clean and easy to process. For this purpose, the following preprocessing techniques are implemented on whole datasets.

3.2.1. Lowercase. It is the process of converting whole words of the review text into lowercase words.

3.2.2. Stopword Removal. Stopwords are a collection of widely utilized words in a language, as these words do not carry any significant information for the model; they have been removed from the contents of the review. Instances of stopwords are "the," "a," "an," "is," "are," etc.

3.2.3. Punctuation Removal. This process is aimed at removing all punctuation marks in the review text.

3.2.4. Removing Contractions. This process is aimed at removing a word that has been written with the short form and replaces it with full form. Example "when've" will become "when have."

3.2.5. Tokenization. This process can be defined as dividing each textual review sentence into small pieces of words or tokens.

3.2.6. Padding Sequences. The deep learning algorithms require input sequences in text classification to have the same length; therefore, for this purpose, we have used the padding sequence method and set the maximum length of the review text to 500 words.

3.3. Data Splitting. This subsection introduces the details of dividing the multidomain datasets that are evaluated in our experiments. Each used dataset has divided into 70% as a training set, 10% as a validation set, and 20% as testing set. Then, we have adopted a hyperneural network model that is consisting of a convolutional neural network integrated with long short-term memory (CNN-LSTM) for detecting and classifying the review text into a fake or truthful review. Table 1 summarizes the splitting of each dataset individually.

3.4. CNN-LSTM-Based Fake Review Identification. The suggested method applies and assists the performance of integrated convolution neural network with long short-term memory (CNN-LSTM) to detect and identify the review text comprising content with fake linguistic clues. For this purpose, we train the deep learning-based neural network model for classifying the input review text of different domain datasets. Figure 1 illustrates the structure of the CNN-LSTM model.

Figure 2 presents the structure of the proposed model used in this research work for identifying the fake reviews in different domain datasets. The components of the CNN-LSTM model are discussed in detail as follows.

(A) *Word Embedding.* The embedding layer is an initial layer of the proposed CNN-LSTM model that is used for the transformation of each word presented in training data into an actual-valued vector representation that means a set of words as features of the dataset are constructed and transformed into numerical form. This process is named word embedding. The word embedding is inputted as a matrix of sequences to the following layer. An embedding layer used in this model has made of three components that are the vocabulary size (maximum features), embedding dimension, and input sequence length. Maximum features which can keep the most frequent and topwords represent the size of the vocabulary. Embedding dimension demonstrates the dimensions of each word that is transformed and by using the embedding layer into real-valued vector representations. Further, the input sequence length defines the maximum length of the input sequence of the review text. The sentences of the review text contain a sequence of words that can be

Table 1: Splitting of datasets used in the experiments.

Dataset name	Total of samples	Training set (70%)	Validation set (10%)	Testing set (20%)	Total of deceptive reviews	Total of truthful reviews
Amazon	21,000	15120	1680	4200	10500	10500
Yelp	9460	6622	946	1892	4730	4730
Restaurants	110	80	8	22	55	55
Hotels	1600	1152	128	320	800	800

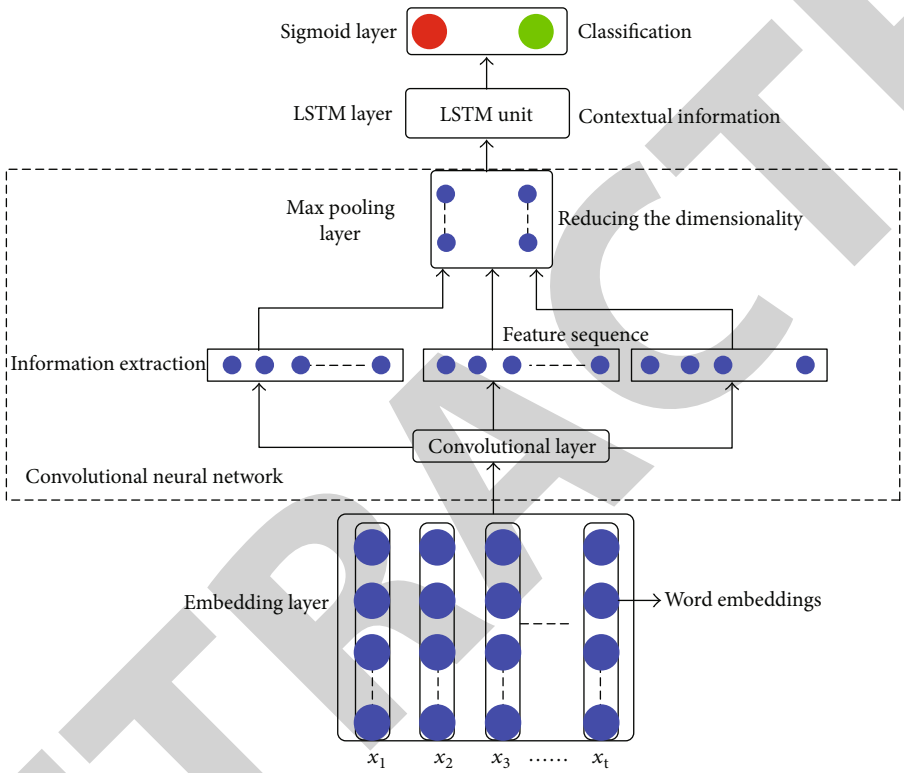


Figure 2: The structure of the CNN-LSTM model.

annotated as X_1, X_2, X_3,....X_t as shown in Figure 2 cited above section, and each word is assigned a specific index integer number. The embedding layer converts the indices of each word into D dimensional word vector representation. In our proposed model, we have used dissimilar domain datasets and for each dataset, we have created different embedding matrix sizes $[V \times D]$ where V represents the vocabulary size and D is the dimension vector representations of each word in V. For input sequence length, we assigned a fixed sequence length for all datasets with 500 words. The embedding matrix can be symbolized as $E \in R^{V \times D}$.

(B) *Convolution Layer.* In the CNN-LSTM model, the convolution layer is a second layer and performing a mathematical operation that is applied on two objective functions, which produces a third function. The convolutional operation is calculated on the dimension vectors of various matrices such as input matrix (I), filter matrix (F), and output matrix (O). These matrices can be expressed in equations (1), (2), and (3) that are given below.

$$P = R^{lxw}, \tag{1}$$

$$F = R^{l \times m}, \tag{2}$$

$$O = R^{l \times d}, \tag{3}$$

where P, F, and O indicate the input, filter, and output matrices, respectively, R is representing entirely real numbers, l is the sequence length, and w denotes the width of the input matrix that is presented as $R^{30000 \times 100}$ for Amazon and Yelp datasets and $R^{10000 \times 100}$ for restaurant and hotel datasets. M is the width of the filter matrix, and d is the width of the output matrix. A convolutional layer is utilized to mine the sequence knowledge and decrease the dimensions of the input sequences [30–32]. It has parameters such as filters with window size. Here, we set the window size to 2×2 and the number of filters to 100, which passes over the input

matrix to extract the features. The formula for convolutional operation is given as follows.

$$t_{i,j} = \sum_{l=1}^{n} \sum_{w=1}^{m} f_{l,w} \otimes P_i + l - 1, j + w - 1, \tag{4}$$

where $\otimes$ represents element-wise cross multiplication, $t_{i,j} \in R^{l \times d}$ is indicating tth element of output matrix, $f_{l,w} \in R^{n \times m}$ denotes the elements of the weight matrix, $P_i + l - 1, j + w - 1 \in R^{lxw}$ is represented pth elements of the input matrix.

(C) *LSTM Layer.* Long short-term memory network (LSTM) is one type of recurrent neural network (RNN) that has the capability for learning long-term dependence and contextual information of the input sequence. We have utilized LSTM as one layer of the CNN-LSTM model and assigned it with different values which include 50 cells in the case in-domain experiment and 100 cells in the cross-domain experiment. LSTM cell executes precalculations for input sequence before giving an output to the last layer of the network. Figure 3 depicts the structure for the LSTM cell.

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 equations for these gates are introduced as follows [31].

$$\begin{aligned} f_t &= \mathrm{sig}\left(Wf_{xt} + Uf_{h_t} - 1 + b_f\right), \\ i_t &= \mathrm{sig}\left(Wi_{xt} + Ui_{h_t} - 1 + b_i\right), \\ O_t &= \mathrm{sig}\left(Wo_{xt} + Uo_{h_t} - 1 + b_o\right), \\ c \sim t &= \tanh\left(wc_{xt} + Uc_{h_t} - 1 + bc\right), \\ C_t &= \left(f_{\mathrm{to}} ct - 1 + i_{\mathrm{to}} c \sim t\right., \\ h_t &= O_{\mathrm{to}} * \tanh\left(C_t\right), \\ \tanh\left(x\right) &= \frac{1 - e^{2x}}{1 - e^{2x}}, \end{aligned} \tag{5}$$

where sig and tanh are sigmoid and tangent activation functions, respectively. X is the input data. W and b represent the weight and bias factors, respectively. C_t is cell state, $c \sim t$ is candidate gate, and h_t refers to the output of the LSTM cell.

(D) Dense Layer. The dense layer (fully connected layer) is one of the hidden layers in the CNN-LSTM model. It consists of N artificial connected neurons and is used to connect all neurons of the network [33]. The function applied to this layer is Rectified Linear Unit (RLU). This function is used to speed up the training process of the model. It has the following equation.

$$f(x) = \max\left(0, x\right). \tag{6}$$

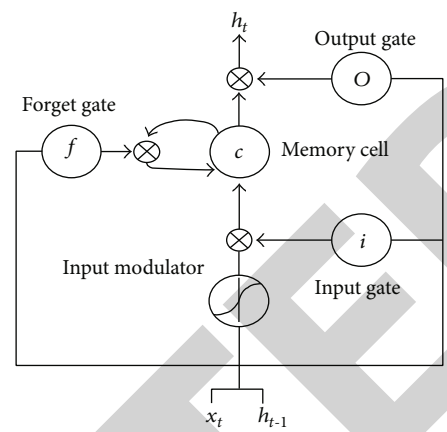


Figure 3: The structure of LSTM Unit.

(E) A sigmoid activation function is the last layer of the model that is applied to detect and classify output classes (fake or truthful review). The equation for a sigmoid function is given as follows

$$\sigma = \frac{1}{1 - e^{2x}}. \tag{7}$$

3.5. Evaluation Matrices. This subsection presents an evaluation of how proficiently the proposed model can classify and distinguish between fake and truthful review text in terms of false-positive and false-negative rates. For measurement of the performance of the classification capability of the CNN-LSTM model, we employed dissimilar performance metrics as follows.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \times 100, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100, \\ \text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100, \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100, \\ \text{F1} - \text{score} &= 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \times 100. \end{aligned} \tag{8}$$

3.6. Experimental Results and Analysis. We assessed the proposed CNN-LSTM model in two different types of experiments (in-domain and cross-domain) based on four standard fake review datasets (Amazon, Yelp, restaurant, and hotel). We also analyze the performance of the model on each dataset and across datasets.

3.6.1. In-Domain Experiment. In this section, we introduce the results of the experiments executed to assess the efficiency of the proposed integrated CNN-LSTM model on the four publicly available fake review datasets individually. We have split each dataset as 70% training, 10 as validation, and 20% as testing. Based on the learning of n-grams of the review

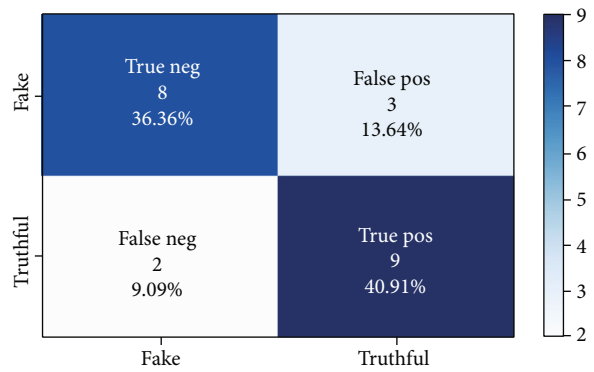


Figure 4: Confusion matrix for restaurant dataset.

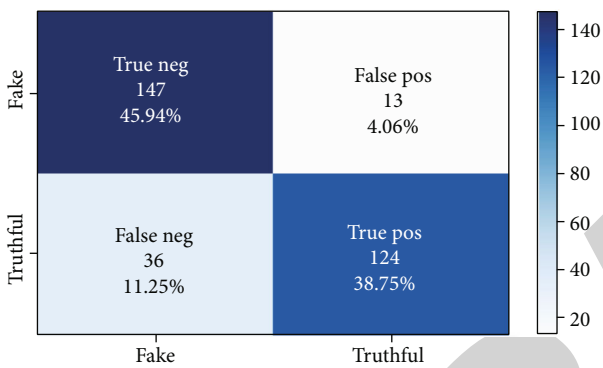


Figure 5: Confusion matrix for hotel dataset.

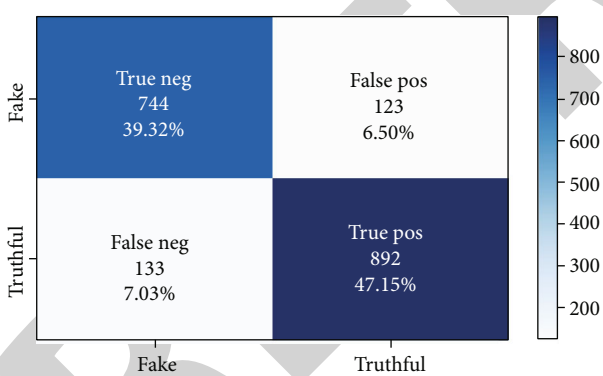


Figure 6: Confusion matrix for Yelp dataset.

text, we create a specific word-embedding matrix for every dataset using a hidden neural network-embedding layer, which is one component of the proposed CNN-LSTM model. In this experiment, we create different embedding matrices of size $V \times D$, where V is the vocabulary size (number of the topwords selected as features from the dataset) and D refers to an embedding dimension. For example, the restaurant and hotel datasets have an input embedding matrix of size 10000×100, the Yelp dataset has 20000×100, and the Amazon dataset has 30000×100. Further, convolutional and max-pooling layers of CNN technique are applied to extract and select the features of input sequences. The

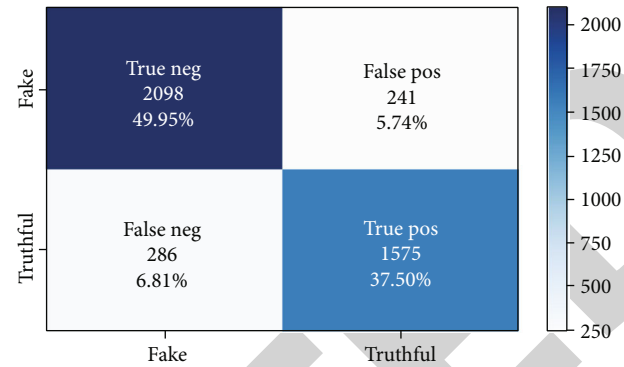


Figure 7: Confusion matrix for Amazon dataset.

Table 2: Classification results for in-domain experiment.

In-domain datasets	Sensitivity (%)	Specificity (%)	Precision (%)	F1-score (%)	Accuracy (%)
Restaurant	82	72	75	78	77
Hotel	77.5	92	90	83	85
Yelp	87	86	88	87	86
Amazon	85	90	87	86	87

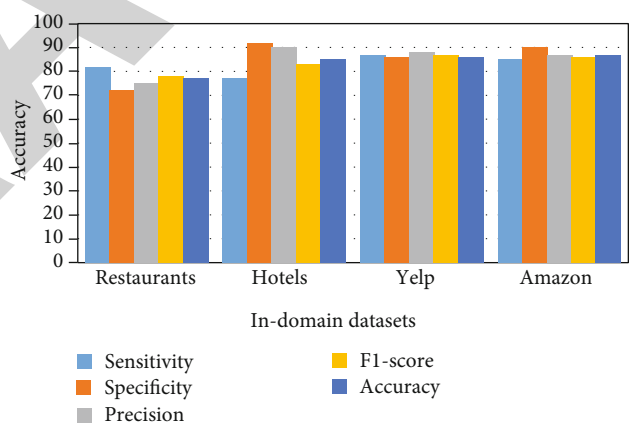


Figure 8: Visualization of the classification results for in-domain experiment.

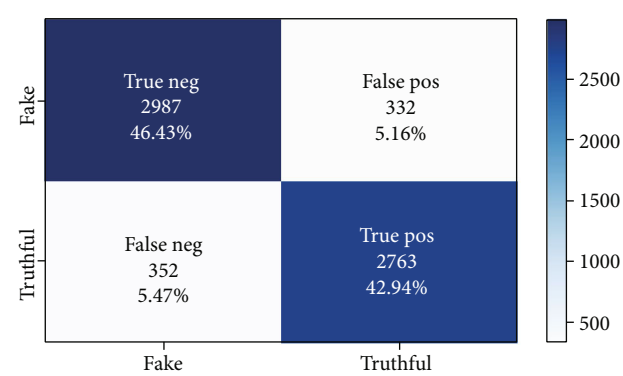


Figure 9: Confusion matrix for cross-domain datasets.

Table 3: Classification results for cross-domain experiment.

In-cross domain datasets	Sensitivity (%)	Specificity (%)	Precision (%)	F1-score (%)	Accuracy (%)
Restaurant+hotel+Yelp+Amazon	89	90	90	89	89

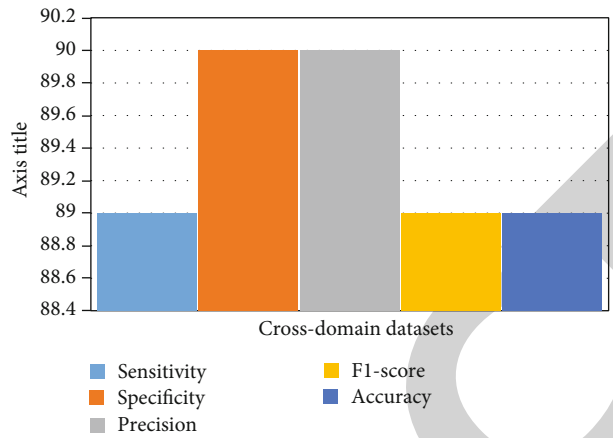


Figure 10: Visualization of the classification results for cross-domain experiment.

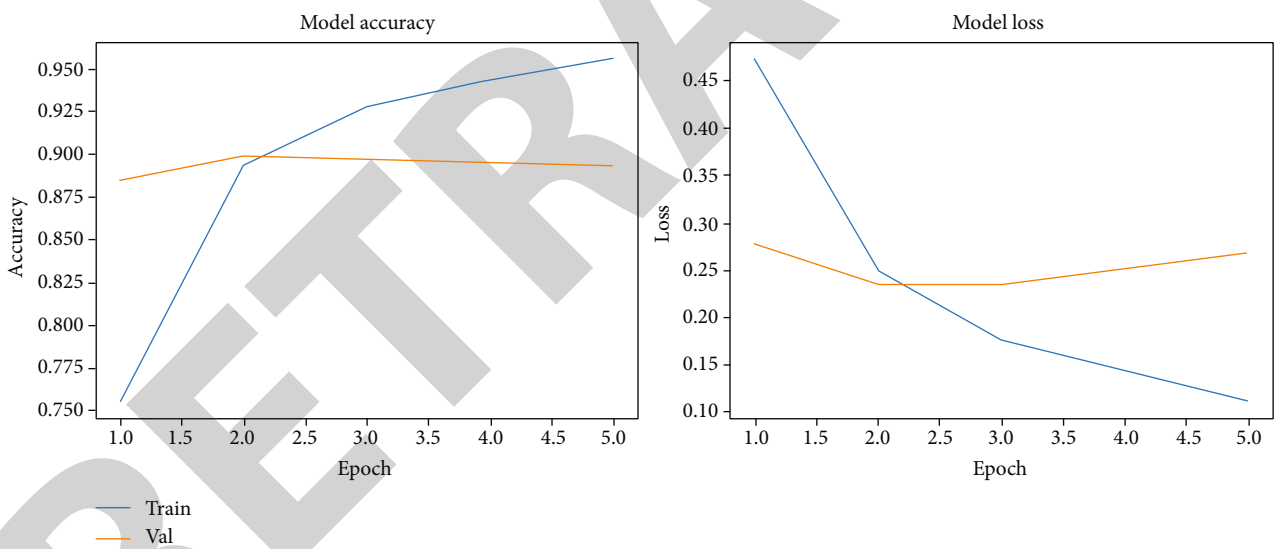


Figure 11: The performance and loss of the CNN-LSTM model on cross-domain datasets.

LSTM layer with sigmoid function is used for learning and classifying an input sequences into fake or truthful reviews. Figures 4–7 show the confusion matrices for restaurant, hotels, Yelp, and Amazon datasets.

In confusion matrices depicted in above Figures 4–7, true negative (TN) represents the total numbers of samples that the model successfully predicted as fake reviews. False negative denotes the total number of samples that the model incorrectly predicted as truthful reviews. True positive denotes the total number of samples that the model successfully predicted as truthful reviews. FP represents the total number of samples that the model incorrectly predicted as fake reviews. Table 2 and Figure 8 summarize and visualize the results for the in-domain experiments.

3.6.2. Cross-Domain Experiment. In this experiment, we have gathered all domain datasets into a single data frame for discovering features that are more robust. The size of this dataset is 32170 review text distributed as 21,000 different Amazon product reviews, 9460 Yelp electronic product reviews, 110 restaurant reviews, and 1600 hotel reviews. We have split the datasets into 70% as a training set, 10% as a validation set, and 20% as a testing set. Based on word embedding of n-gram features of the review text, we have

Table 4: Comparing the results of an in-domain datasets with existing work.

Paper id	Domain dataset	Features used	Method	Accuracy
Faranak Abri et al. [27]	Restaurant	Linguistic features from review content	MLP	73%
Ren Y et al. [22]	Hotel	Review content and pretrained word embedding (bag of word)	CNN	84%
Barushka et al. [33]	Hotel	Review content with pretrained word embedding (skip-gram)	DFNN	83%
Garcia L. [24]	Amazon	Review content with TF-IDF	SVM	63%
Hajek et al. [19]	Amazon	Review content with pretrained word embedding (skip-gram)	DFFNN	82%
			CNN	81%
Barbado et al. [17]	Yelp	Review content with TF-IDF	AdaBoost	82%
This study	Restaurant	n-grams of the review content with word-embedding matrix using embedding layer	CNN-LSTM	77%
	Hotel			85%
	Yelp			86%
	Amazon			87%

created an input embedding matrix that has the size of $V \times D$ (V is vocabulary size of the dataset, and D is embedding dimensions of each word in V) which is equal to 50000×100. Further, the convolutional and max-pooling layers of CNN are utilized for sliding over an input matrix and extract the feature maps from input sequences. Then, LSTM layer receives the output from the max-pooling layer and performs the processing task for handling of contextual information of the sequences based on gate mechanism. Finally, last layer is the sigmoid function that is applied for classification of the input sequence into truthful or fake. The experimental results show that CNN-LSTM model provides better performance in cross-domain than an in-domain datasets. Figure 9 below presents the confusion matrix for cross-domain datasets.

From the experimental results carried out in this research work, we conclude that a large number of n-gram features lead to better accuracy with deep learning neural network techniques. Table 3 and Figure 10 show the classification and visualization of results in cross-domain experiment.

In the above Figure 11 and on the left plot, the X-axis represents the training and validation accuracy and Y is the number of epochs, which indicate the number of iterations that the CNN-LSTM model has trained and tested on the dataset. The right plot shows the model loss.

4. Comparative Analysis

In this section, we compare the results of in-domain experiments performed by the proposed model (CNN-LSTM) with the existing works based on accuracy metric. Table 4 reports the comparative analysis using the accuracy metric.

According to the literature review of fake review detection, there is no research work has used the same datasets in a cross-domain experiment. Thus, we are unable to make comparative analyses for cross-domain datasets.

5. Conclusion

This paper presents a hyperneural network model comprising of convolutional neural network along with long short-term memory (CNN-LSTM) techniques for detecting and classifying the review text into fake or truthful. In the proposed methodology, two different experiments that are in-domain and cross-domain have been carried out on four standard fake review datasets (hotel, restaurant, Yelp, and Amazon). Preprocessing methods such as lowercase, removing of stopword and punctuation, and tokenization have been conducted for the dataset cleaning as well as padding sequence method was used to make a fixed length for all input sequences. Further, an embedding layer as one component of the proposed model was applied to create different types of word-embedding matrices of size $V * D$ (V is the vocabulary size of the dataset, and D is an embedding dimension of each word in V) for in-domain and cross-domain experiments. Convolutional and max-pooling layers of the CNN technique perform the feature extraction and selection. Further, the LSTM technique is combined with the CNN for contextual information processing of input sequences that are based on gate mechanisms and forward the output to the last layer. A sigmoid function as last layer of the proposed model is used to classify the review text sequences into fake or truthful. For in-domain experiments, the proposed model is applied to each dataset individually for fake review detection. Further, a cross-domain experiment was performing on mixed data of restaurants, hotels, Yelp, and Amazon reviews. From experimental results, we conclude that a large number of features lead to better accuracy while using deep learning neural network (DLNN) algorithms. Outstandingly, the proposed model surpassed existing baseline and state-of-the-art fake review identification techniques in terms of accuracy and F1-score measures for in-domain experiment. The experimental results also revealed that the proposed model provides better performance in a cross-domain experiment than an in-domain experiment because the first one is implemented to a large-size dataset with more features. According to the literature review of fake review detection methods, there is no research work has used the same datasets in a cross-domain experiment. Thus, we are unable to make comparative analyses with cross-domain datasets.

Data Availability

The data are available in the following links: https://www.kaggle.com/rtatman/deceptive-opinion-spam-corpus; https://github.com/asiamina/FakeReviews-RestaurantDataset; https://github.com/aayush210789/Deception-Detection-on-Amazon-reviews-dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. U. Vidanagama, T. P. Silva, and A. S. Karunananda, "Deceptive consumer review detection: a survey," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1323–1352, 2020.

[2] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 international conference on web search and data mining*, pp. 219–230, Palo Alto, California, USA, 2008.

[3] S. N. Alsubari, M. B. Shelke, and S. N. Deshmukh, "Fake reviews identification based on deep computational linguistic," *International Journal of Advanced Science and Technology*, vol. 29, pp. 3846–3856, 2020.

[4] S. Rayana and L. Akoglu, "Collective opinion spam detection: bridging review networks and metadata," in *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 985–994, Sydney, NSW, Australia, 2015.

[5] C. Miller, *Company settles case of reviews it faked*, New York Times, 2009.

[6] Y. Ren, D. Ji, and H. Zhang, "Positive unlabeled learning for deceptive reviews detection," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 488–498, Doha, Qatar, 2014.

[7] D. Streitfeld, *For $2 a star, an online retailer gets 5-star product reviews*, vol. 26, New York Times, 2012.

[8] A. Heydari, M. Ali Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: a survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015.

[9] M. Arjun, V. Vivek, L. Bing, and G. Natalie, "What yelp fake review filter might be doing," in *Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013)*, Massachusetts USA, 2013.

[10] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Oregon, USA, 2011.

[11] R. Y. Lau, S. Y. Liao, R. C. Kwok, K. Xu, Y. Xia, and Y. Li, "Text mining and probabilistic language modeling for online review spam detection," *ACM Transactions on Management Information Systems (TMIS)*, vol. 2, no. 4, pp. 1–30, 2011.

[12] S. Shojaee, M. A. A. Murad, A. B. Azman, N. M. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *2013 13th international conference on intelligent systems design and applications (ISDA)*, pp. 53–58, Salangor, Malaysia, 2013.

[13] Q. Xu and H. Zhao, "Using deep linguistic features for finding deceptive opinion spam," *Proceedings of COLING 2012: Posters*, pp. 1341–1350, 2012.

[14] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S. M. R. Beheshti, N. Foo, and E. Bertino, "Detecting, representing and querying collusion in online rating systems," 2012, https://arxiv.org/abs/1211.0963.

[15] H. Duan and P. Yang, "Building robust reputation systems for travel-related services," in *Proceedings of the 10th Annual Conference on Privacy, Security and Trust (PST 2012)*, Paris, France, 2012.

[16] S. Feng, "Distributional footprints of deceptive product reviews," in *Sixth International AAAI Conference on Weblogs and Social Media*, Dublin, Ireland, 2012.

[17] R. Barbado, O. Araque, and C. A. Iglesias, "A framework for fake review detection in online consumer electronics retailers," *Information Processing & Management*, vol. 56, no. 4, pp. 1234–1244, 2019.

[18] S. Noekhah, E. Fouladfar, N. Salim, S. H. Ghorashi, and A. A. Hozhabri, "A novel approach for opinion spam detection in e-commerce," in *Proceedings of the 8th IEEE international conference on E-commerce with focus on E-trust*, Mashhad, Iran, 2014.

[19] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17259–17274, 2020.

[20] K. Goswami, Y. Park, and C. Song, "Impact of reviewer social interaction on online consumer review fraud detection," *Journal of Big Data*, vol. 4, no. 1, pp. 1–9, 2017.

[21] M. Young, *The Technical Writer's Handbook*, University Science, Mill Valley, CA, 1989.

[22] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: an empirical study," *Information Sciences*, vol. 385, pp. 213–224, 2017.

[23] Z. Y. Zeng, J. J. Lin, M. S. Chen, M. H. Chen, Y. Q. Lan, and J. L. Liu, "A review structure based ensemble model for deceptive review spam," *Information*, vol. 10, no. 7, p. 243, 2019.

[24] L. Garcia, *Deception on Amazon on an NL exploration*, 2018, https://medium.com//@lievgarcial/deception-on-amazon-cle30d977cfd.

[25] S. Kim, H. Chang, S. Lee, M. Yu, and J. Kang, "Deep semantic frame-based deceptive opinion spam analysis," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1131–1140, Melbourne, Australia, 2015.

[26] L. Gutierrez-Espinoza, F. Abri, A. S. Namin, K. S. Jones, and D. R. Sears, "Ensemble learning for detecting fake reviews," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1320–1325, Madrid, Spain, 2020.

[27] F. Abri, L. F. Gutierrez, A. S. Namin, K. S. Jones, and D. R. Sears, "Fake reviews detection through analysis of linguistic features," 2020, https://arxiv.org/abs/2010.04260.

[28] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proceedings of the 21st international conference on World Wide Web*, pp. 201–210, Lyon, France, 2012.

[29] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 497–501, Atlanta, Georgia, 2013.

[30] S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and I. Awan, "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 24, 2019.

[31] *Understanding LSTM cells using C#*https://msdn.microsoft.com/en-us/magazine/mt846470.aspx.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9798010, 1 page
https://doi.org/10.1155/2023/9798010

Retraction

Retracted: Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models

Applied Bionics and Biomechanics

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret 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 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] C. Zhou, L. Zhang, Y. Wu, X. Zhang, D. Wu, and Z. Tao, "Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6662625, 11 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 6662625, 11 pages
https://doi.org/10.1155/2021/6662625

Research Article

Effects of Sulcus Vocalis Depth on Phonation in Three-Dimensional Fluid-Structure Interaction Laryngeal Models

Changwei Zhou, Lili Zhang, Yuanbo Wu, Xiaojun Zhang, Di Wu, and Zhi Tao

School of Optoelectronic Science and Engineering, Soochow University, Suzhou 215000, China

Correspondence should be addressed to Di Wu; wudi@suda.edu.cn and Zhi Tao; taoz@suda.edu.cn

Received 20 November 2020; Revised 19 March 2021; Accepted 29 March 2021; Published 9 April 2021

Academic Editor: Fahd Abd Algalil

Sulcus vocalis is an indentation parallel to the edge of vocal fold, which may extend into the cover and ligament layer of the vocal fold or deeper. The effects of sulcus vocalis depth d on phonation and the vocal cord vibrations are investigated in this study. The three-dimensional laryngeal models were established for healthy vocal folds (0 mm) and different types of sulcus vocalis with the typical depth of 1 mm, 2 mm, and 3 mm. These models with fluid-structure interaction (FSI) are computed numerically by sequential coupling method, which includes an immersed boundary method (IBM) for modelling the glottal airflow, a finite-element method (FEM) for modelling vocal fold tissue. The results show that a deeper sulcus vocalis in the cover layer decreases the vibrating frequency of vocal folds and expands the prephonatory glottal half-width which increases the phonation threshold pressure. The larger sulcus vocalis depth makes vocal folds difficult to vibrate and phonate. The effects of sulcus vocalis depth suggest that the feature such as phonation threshold pressure could assist in the detection of healthy vocal folds and different types of sulcus vocalis.

1. Introduction

Sulcus vocalis is related to the inhomogeneous damage of the cover and ligament with structural malformation of the vocal fold. This disease often leads to concomitant vocal fold disorders and has a significant influence on vocal fold vibratory function [1, 2]. According to clinical and histopathologic analysis, Ford et al. [3] have classified sulcus vocalis into three types: type I is a physiologic variant in cover of the vocal folds; type II and III are characterized by destruction of the intermediate and deep lamina propria, respectively. Different types of sulcus vocalis have different sulcus depth ranges.

To explore the pathophysiology, clinical characteristics of sulcus vocalis, open and endoscopic procedures were applied by Seth et al. [4]. Welham et al. [5] used the Voice Handicap Index (VHI) for characterizing the psychosocial effect of disorders on patients with sulcus vocalis. Combining subjective and objective methods, Soares et al. [6] investigated the vocal characteristics of individuals with sulcus vocalis, especially including asymptomatic subjects, using suspension microlaryngoscopy, voice self-assessment, and acoustic evaluation of the voice. However, considering the aerodynamic theory of phonation, these methods ignore the effects of airflow viscosity.

The features of vocal cord vibration and glottal jet dynamics are difficult to characterize with physical models and reduced-order vocal models. On the contrary, computational models do well in that. The distribution of pressure and airflow was obtained from studies of steady flow in laryngeal models [7]. For exploring the more complex biomechanical modelling of phonation, Alipour et al. [8] modelled an oscillating glottis to study pulsatile flow with a finite volume method. However, the forced oscillation model only characterized glottis dynamics, which ignored vortex dynamics and temporal flow variation, especially, significant asymmetric flow due to jet deflection. Thus, based on the combination of the two-mass vocal fold model and the Navier-Stokes equation, Xue et al. [9] focused on capturing asymmetric glottal jet deflection in an asymmetric larynx. In the past few decades, most of the scholars employed a two-dimensional laryngeal model [10, 11] for the sake of computational cost and complex laryngeal structure. However, experimental studies mostly have shown that the glottal flow

is highly three-dimensional [12–14]. What is more, three-dimensional vibration could have influence on laryngeal pathologies for anterior-posterior vibration in the three-dimensional model, such as sulcus vocalis and vocal fold paralysis [15–17].

Recently, the fluid-structure coupling method is widely applied in numerical simulation of the larynx model [18, 19]. Valasek et al. [20] coupled linear elastic problem with the incompressible Navier-Stokes equations in the arbitrary Lagrangian-Eulerian form in order to model fluid-structure interaction problem. Smith and Thomson [21] investigated the effect of subglottic stenosis on vocal fold vibration by establishing a fully coupled finite element model of the vocal folds as well. Luo et al. [22] devised a coupled solver with the immersed boundary method to model the flow-structure interaction of phonation. Tian et al. [23] combined the immersed boundary flow solver with a nonlinear finite-element solid-mechanics solver, handling boundaries of large displacements with simple mesh generation. Chang et al. [24] studied the effect of geometric nonlinearity on the vocal fold and the airflow, suggested that accurate simulations of vocal fold dynamic needs the large-displacement formulation in a computational model. Although there are some numerical simulation studies of sulcus vocalis with respect to sulcus patients' vocal fold vibration and hydrodynamic analysis, the effects of sulcus vocalis depth on phonation have not yet been performed. Sulcus vocalis that the depth is less than the cover of the vocal fold generally does not give rise to dysfunction, which is classified as type I. As the depth gets larger with sulcus extending into the intermediate and deep layers of vocal folds, these types of sulcus vocalis usually bring about moderate to severe anomaly.

In current research, the three-dimensional laryngeal models are developed to simulate and study vocal fold vibrations as well as the aerodynamics in the larynx model with different sulcus vocalis depth.

2. Computational Model

We established a computational model of the larynx utilized by Zheng et al. [25]. In fact, the human vocal folds have three layers, consisting of cover, ligament, and muscle, and can be assumed to be made of a viscoelastic material which is transversally isotropic [26].

For the reality of numerical simulation, the isotropic linear elastic material was applied in all layers with the density of 1070 kg/m^3, and the Poisson ratio to be 0.4. The elastic modulus of cover, ligament, and body was 10, 100, and 40 kPa. In Figure 1, the larynx is assumed to be a rectangular duct. Based on the real size of the human larynx, the total length of the duct is 120 mm, and the length of supraglottal and false vocal folds is 63 mm and 23 mm, respectively, for the easy observation of glottal jet dynamics. The glottal gap which is the distance between vocal folds is 1 mm primarily. And the thickness T and depth of vocal folds are 10 mm and 9 mm. For the diseased vocal folds, the sulcus vocalis is modelled by shallow longitudinal furrow to cover, and the depth d is 1 mm, which is type I sulcus. In addition, there are shallow longitudinal furrows with depth of 2 mm and 3 mm, which are type II sulcus and type III sulcus, respectively. All types of sulcus vocalis have the same length with 2 mm. The sulcus depth of the healthy vocal fold is considered to be 0 mm in this study.

The adduction of the vocal folds closes the glottis, thereby creating a barrier for the expulsion of air from the lungs. As air is forced from the lungs, the adducted vocal folds are pushed apart owing to air pressure, and the vocal folds are set into sustained flow-induced vibrations. Thus, the boundary conditions of vocal folds' surfaces are fluid-structure interaction conditions. It is noted that the air pressure from the lungs is between 0.5 kPa and 2.5 kPa. When the human speaks normally, the pressure of the lungs is often considered as 1 kPa. Therefore, the total pressure on the inlet boundary is set to 1 kPa. And the relative pressure of the outlet boundary is equal to 0 kPa.

3. Numerical Method

Four laryngeal models are established with the depth of 0 mm, 1 mm, 2 mm, and 3 mm. The model with a depth of 0 mm corresponds to the healthy vocal folds, while the others are different types of sulcus vocalis. Numerical analysis of flow fields in the larynx with sulcus vocalis needs to consider the highly complex interaction between vocal fold tissue and glottal airflow. In this research, a flow-structure interaction computational solver is applied, which includes an immersed boundary method (IBM) [27, 28] for modelling the glottal airflow, a finite-element method (FEM) for modelling vocal fold tissue, and a sequential coupling method for fluid-structure interaction during phonation. Each model with fluid-structure interaction has the same computational domain. And sequential coupling method was applied to solve fluid-structure interaction problem.

3.1. Fluid Mechanics. The Mach number of the flow in the glottis is generally assumed to be smaller than 0.3 [29], which could be considered that the airflow is incompressible. Based on the laws of momentum and mass conservation, the governing set of airflow is as follows:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right), \tag{2}$$

where $i, j = 1, 2, 3$, u_i are the velocity components, P is the pressure, and ρ and ν are the fluid density and kinematic viscosity.

Equation (2) is discretized in space using a cell-centre collocated finite difference scheme. The description of the naming convention and location of variables used in the spatial discretization is shown in Figure 2.

Equation (2) is integrated in time using the fractional-step method [31], including three substeps.

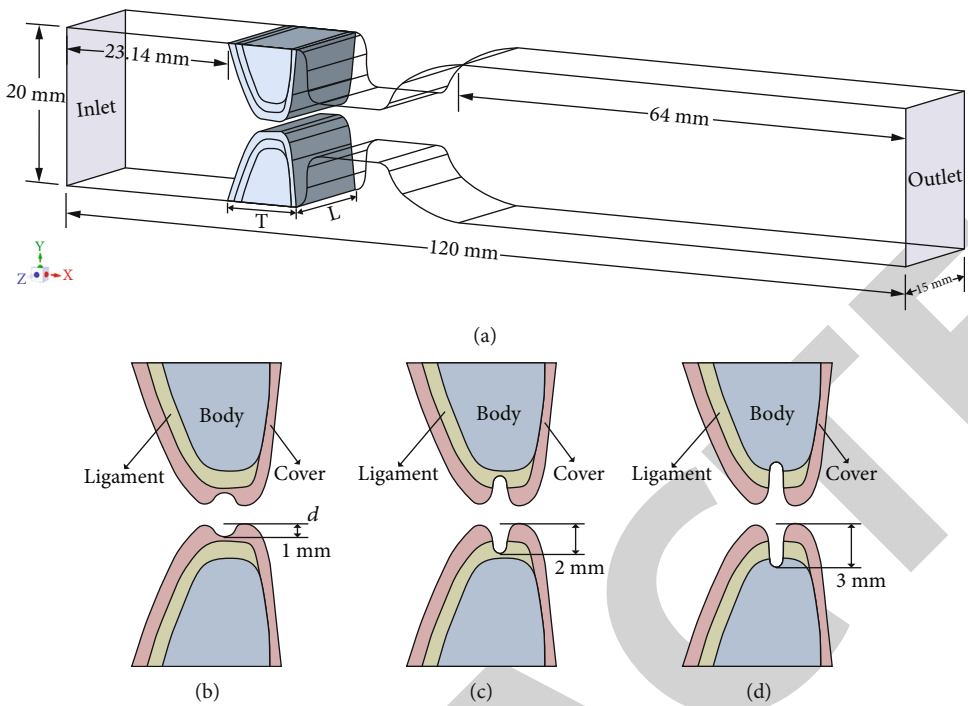


Figure 1: (a) The geometry of computational domain with true and false vocal folds. (b) Type I sulcus vocalis (with depth of 1 mm). (c) Type II (2 mm). (d) Type III (3 mm).

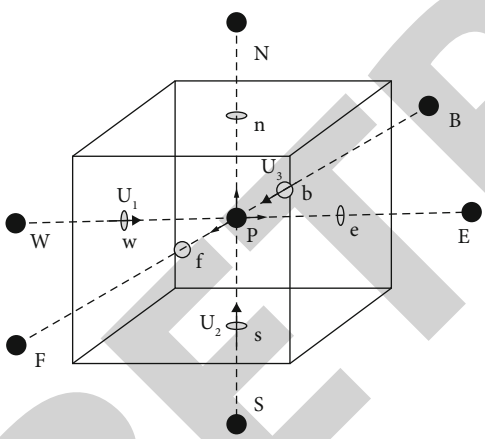


Figure 2: The description of the naming convention and location of variables used in the spatial discretization [30].

(1) A modified momentum equation with a previously computed pressure field is solved to calculate an intermediate velocity U^*.

$$\frac{U_i^* - u_i^n}{\Delta t} + \frac{1}{2}\left[3N_i^n - N_i^{n-1}\right] = -\frac{1}{\rho}\frac{\delta P^n}{\delta x_i} + \frac{1}{2}\left(D_i^* + D_i^n\right), \quad (3)$$

where $N_i = \delta(U_j u_i)/\delta x_i$, $D_i = \nu(\delta/\delta x_j)(\delta u_i/\delta x_j)$ are the convective and diffusion terms, respectively, and $\delta/\delta x$ denotes a second-order central difference. The intermediate face-centre velocity U^* is computed by averaging the neighboring cell-centre velocities u^*, and only the face velocity component normal to the cell-face is computed and stored. The averaging procedure is as follows:

$$\begin{aligned}
\tilde{U}_i &= U_i^* + \Delta t \frac{1}{\rho}\left(\frac{\delta P^n}{\delta x_i}\right)_{cc}, \\
\tilde{U}_1 &= \gamma_v \tilde{u}_{1P} + (1-\gamma_v)\tilde{u}_{1W}, \\
\tilde{U}_2 &= \gamma_s \tilde{u}_{2P} + (1-\gamma_s)\tilde{u}_{2S}, \\
\tilde{U}_3 &= \gamma_b \tilde{u}_{3P} + (1-\gamma_b)\tilde{u}_{3B}, \\
U_i^* &= \tilde{U}_i - \Delta t \frac{1}{\rho}\left(\frac{\delta P^n}{\delta x_i}\right)_{f_c},
\end{aligned} \quad (4)$$

where γ_w, γ_s, and γ_b are linear interpolation weights for the west, south, and back face velocity components, respectively. cc and fc represent cell-centre and face-centre.

(2) A pressure correction variable P' is computed from the pressure correction equation

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{1}{\rho}\frac{\delta P'}{\delta x_i}. \quad (5)$$

When the final velocity u^{n+1} is divergence free, equation (5) is integrated the equation over the single cell, which results in the pressure Poisson equation as follows:

$$\frac{1}{\rho}\frac{\delta}{\delta x_i}\left(\frac{\delta P'}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta U_i^*}{\delta x_i}. \quad (6)$$

A highly efficient multigrid method is applied to solve equation (6).

(3) The pressure and velocity field is corrected by using pressure correction variable P'

$$P^{n+1} = P^n + P',$$
$$u_i^{n+1} = u_i^* - \Delta t \frac{1}{\rho}\left(\frac{\delta P'}{\delta x_i}\right)_{cc}, \tag{7}$$
$$U_i^{n+1} = U_i^* - \Delta t \frac{1}{\rho}\left(\frac{\delta P'}{\delta x_i}\right)_{fc}.$$

In this research, vocal folds and airway surface are represented by the unstructured mesh with triangular elements. The boundary conditions are imposed on the nodes of the triangular elements through either the prescribed motion or the computed motion from flow-structure interaction.

3.2. Structural Mechanics. Vocal folds are composed of multilayered tissue, which are nonlinear, transversely isotropic, and viscoelastic. Considering the fact that the deformation of the vocal folds during normal phonation, it can be assumed that the relationship between stress and strain is linear. Thus, the Kelvin-Voigt model [32] for linear viscoelastic material is adopted for vocal fold tissue.

Based on the Kelvin-Voigt model, the constitutive law can be written as follows:

$$\sigma = E\varepsilon + C\dot{\varepsilon}, \tag{8}$$

where E and C are elasticity and damping matrix.

The governing equation of solid dynamics is given by

$$\rho_s \frac{\partial^2 u_t}{\partial t^2} = f_i + \frac{\partial \sigma_{ij}}{\partial x_j}, \tag{9}$$

where $i, j = 1, 2, 3, \sigma$ is the stress tensor, ρ_s is the tissue density, f is the body force, and u is vocal fold tissue displacement.

3.3. Fluid-Structure Coupling. In order to ensure that the fluid-structure coupling follows the conservation principle, the following equations should be applied on the fluid-structure coupling interface.

$$\begin{cases} \tau_f \cdot n_f = \tau_s \cdot n_s \\ d_f = d_s \end{cases}. \tag{10}$$

For the convenience of analysis, a general form of governing equations was established. And we solved unified equations with appropriate initial conditions and boundary conditions. This method solves the problem by coupling the fluid-structure control equations to the same equation matrix, which solving the fluid and structure governing equations in the same solver.

$$\begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix} \begin{bmatrix} \Delta X_f^t \\ \Delta X_s^t \end{bmatrix} = \begin{bmatrix} B_f \\ B_s \end{bmatrix}, \tag{11}$$

where t is the step size of time; A_{ff}, ΔX_f^t, and B_f denote system matrix of flow field, unsolved, and external force, respectively; and A_{sf} and A_{fs} represent the coupling matrix of fluid-structure.

The methods of FSI include direct coupling and sequential coupling method. In fact, the direct coupling method calculates the fluid and solid equations at the same time, and there is no lag of calculation time. However, the direct coupling method results in calculation divergence and a large amount of calculation. Thus, we applied the sequential coupling method to fluid-solid coupling. The computation of fluid and structural domain is separate. Different solvers are used to calculate their physical variables, and the common variables are updated asynchronously. When the fluid solver sends pressure to the solid solver, it receives displacement computed by the solid solver.

4. Results

4.1. Validation and Vocal Fold Vibration. The time history of the displacement of a healthy vocal fold and sulcus vocalis in Y direction is shown in Figures 3(a) and 3(b). Figure 3(a) suggests that the numerical simulation ran successfully for providing quasisteady cycles. The result from Vazifehdoost-saleh et al. [15] is included here for comparison. Figure 3 illustrates that the current results are in good agreement with previous simulation data. In addition, the displacement of the right side is greater than that of the left side in the sulcus vocalis, which also validates the credibility of the sulcus vocalis model.

More importance is often attached to the glottal waveform for voice quality assessment. Due to the incompressibility of flow, it can be measured by the equipment which cover subject's mouth and nose frequently. Several statistical parameters such as the fundamental frequency, the peak and mean glottal flow rate, and the open and skewing quotient [33] are usually extracted from this waveform to examine voice quality. The vibration frequency of the healthy vocal folds is estimated to be 107.5 Hz from Figure 3(a). The flow parameters for healthy vocal folds and sulcus vocalis (1 mm) were compared with other simulation and experimental data in Table 1; this simulation could appear to be the representation of real human phonation.

The vibration of the vocal folds is much more complex, which behaves partly like string and partly like spring. The fundamental frequency F_0 of vocal folds can be expressed by equating the expressions for a vibrating string and spring.

$$F_0 = \frac{1}{2L}\sqrt{\frac{\sigma}{\rho}} = \frac{1}{2\pi}\sqrt{\frac{k}{m}}, \tag{12}$$

where L is the length of vocal folds, σ is the stress, ρ is tissue density, k is the stiffness of vocal folds, and m is the mass of vocal folds.

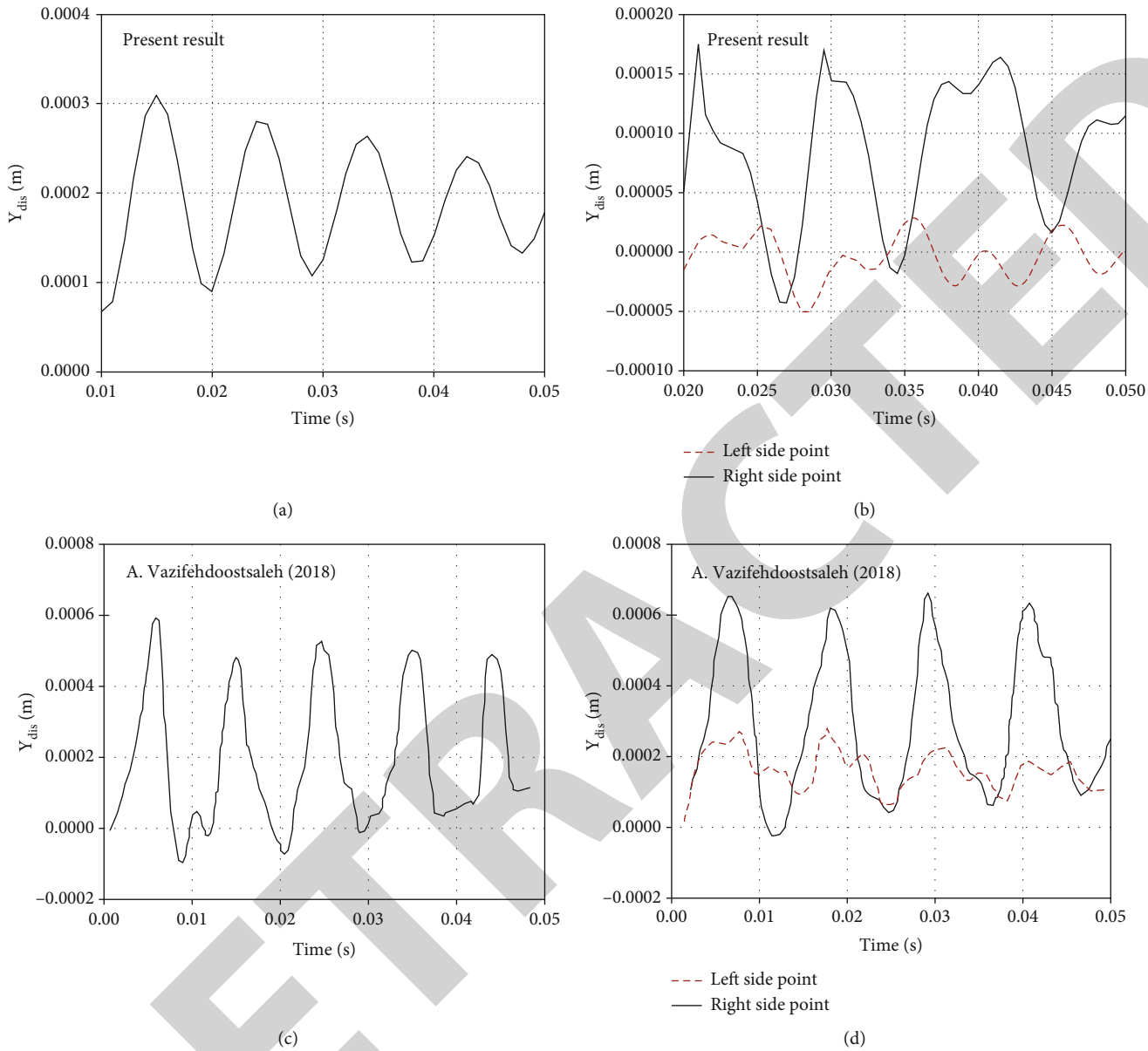


Figure 3: Variation of healthy vocal fold displacement versus time from this work (a) and A. Vazifehdoostsaleh's result (c). Variation of right and left side displacement of sulcus vocalis (1 mm) versus time from this work (b) and A. Vazifehdoostsaleh's result (d).

Table 1: Comparison of flow parameters for healthy vocal folds and sulcus vocalis (1 mm).

	Typical range for healthy vocal folds	A. Vazifehdoostsaleh (2018)		Present result	
		Healthy vocal folds	Sulcus vocalis	Healthy vocal folds	Sulcus vocalis
F_0 (Hz)	65-260 [34]	104.2	83.3	107.5	85.0
Q_{max} (ml/s)	200-580 [35]	345.1	390.0	359.2	407.5
Q_{mean} (ml/s)	110-220 [34]	198.2	226.2	186.5	241.4

The expression of the mass of vocal fold is $m = \rho LTD$, where ρ and T are the density and thickness of vocal folds and D is the depth of the cover layer in that only the pliable cover of vocal folds is vibrating during soft talking. The effective stiffness of vocal folds k can be derived from equation (11), which is $k = \pi^2 \sigma TD/L$. Then, the expression for stiffness of sulcus vocalis k_s is $(\pi^2/L)\sigma T(D-d)$. The stress σ is constant with the same elastic modulus and lung pressure. Thereby, an increase in sulcus depth d will produce a lower vibrating frequency of vocal folds.

The vibrating frequency of the sulcus vocalis is estimated to be about 85 Hz from Figure 4. The vibrating frequency of

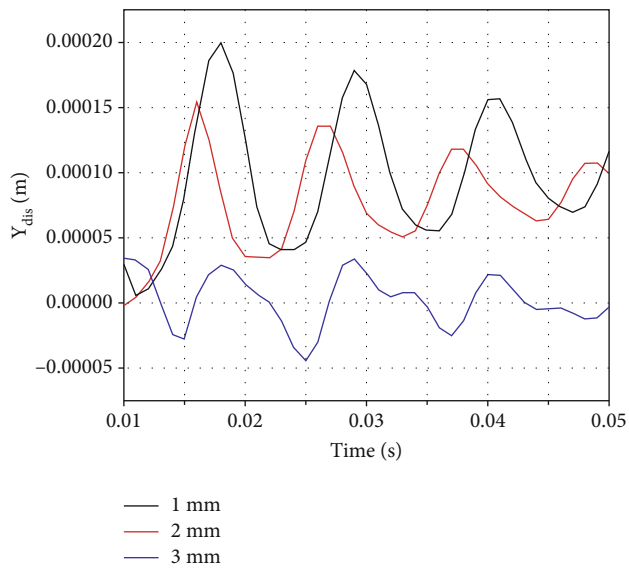


Figure 4: The right side displacement of vocal folds in Y direction of sulcus vocalis(1 mm, 2 mm, and 3 mm).

the sulcus vocalis (1 mm) is lower than that of the healthy vocal cord (0 mm). A deeper sulcus vocalis in the cover layer decreases the fundamental frequency and amplitude of the vibration. However, when the sulcus depth gets larger that extends into the ligament layer or deeper, the vibrating frequency of vocal folds has almost no change, which is identical to the expression of fundamental frequency.

4.2. Dynamics of the Glottal Jet. The glottal flow could exhibit a variety of phenomena such as jet deflection, flow transition, and instability of jet. These phenomena have a close relation with the quality of voice and phonation. And the spatial and temporal details of the glottal jet could be performed by numerical simulation with finite element analysis methods. Figure 5 illustrates the contours of the glottal jet for healthy vocal folds (0 mm) and sulcus vocalis with the typical depth of 1 mm, 2 mm, and 3 mm during one vibration cycle.

In Figure 5, time instants of one cycle were selected instead of specific time due to the different vibration frequency of healthy vocal fold and three types of the sulcus vocalis. At the time instant of 0.1 T, a glottal jet emanated from the glottal gap as the vocal folds were diverging. What is more, the glottal jet was symmetrical whether the computational model is healthy vocal fold or sulcus vocalis at the beginning of the cycle. However, when the glottis further opened, the front of the glottal jet was attaching to one side of the glottal tract, which is a notable phenomenon called the Coanda Effect. The existence of the glottal jet deflection is debated now. Comparing with other studies that performed the glottal jet deflection [36, 37], Xue et al. [33] found that the glottal jet was almost symmetric, which had no strong jet deflection for the sake of the application of more realistic geometric configuration.

In our research, the laryngeal models with simplified geometric configuration and isotropic material properties were applied. The flow deflection occurred in the laryngeal models with a depth of 0 mm, 1 mm, and 2 mm. It should be noticed that the glottal jet was nearly symmetrical during three quarters of a cycle in the model with a depth of 3 mm. In addition, the jet deflection was decreasing when the depth of sulcus vocalis increased.

Figure 6 exhibits a more quantitative comparison about total velocities between the laryngeal models with different sulcus depths. The velocity data were extracted along cross-lines that are located in the X-Y plane of different laryngeal models as Figure 6(a) shows. The max total velocity of 0 mm laryngeal model was larger than other types of laryngeal models on line 1 shown in Figure 6(a), which is in the centre of the glottis. It is also presented that the speed curve is symmetrical about $y = 10$ mm from line 1 to 3 in Figure 6(b). When the cross-line was in the region of inferior glottis, the location of the max velocity was gradually changing to up or down side in the Y direction. It can be presented that the glottal jet tended to be deflected into upper or lower walls of the glottal tract, which indicated that the direction of the jet deflection is stochastic.

4.3. Pressure Distribution and Phonation Threshold Pressure. Vibration of the vocal cord is governed primarily by the principles of aerodynamics, structural dynamics, and fluid-structure interactions. It could assist in revealing the effects of different sulcus vocalis depths on phonation by studying glottal pressure distribution. With FSI numerical method in a 3D computational model, phonation threshold pressure was estimated to study the effects further.

Figure 7 illustrates the variation of glottal pressure distributions for the line ($y = 10$ mm, $z = 7$ mm) for the laryngeal models with different sulcus depths at four different time instants during one vibration cycle, respectively. It is presented that glottal pressure has one pressure drop at $x = 30$ mm in Figure 7(a), which is the centre of the glottis. However, the glottal pressure occurs two pressure drops for sulcus vocalis; one is at the entrance of sulcus vocalis, and the other is at the exit of sulcus vocalis. Based on the inference of Bernoulli's principle, the airflow in the vocal tract accelerates due to the reduction of fluid area, which makes the pressure of airflow falls quickly. After entering the sulcus vocalis, the pressure of airflow rises slowly until the airflow arrives at the exit of sulcus vocalis. What is more, it should be noted that with sulcus vocalis depth increasing, the minimal glottal pressure gradually gets larger and the gradient of glottal pressure became smaller.

Phonation threshold pressure is associated with pathological voice frequently, which is the minimal glottal pressure needed to initiate and sustain vocal fold vibration [38], for the sake of the sensitivity in laryngeal biomechanics' changes. In this research, the validated computational model was applied to obtain phonation threshold pressure. Figure 8 shows the time evolution of the pressure at the entrance of the glottis.

According to Titze's derivations for phonation threshold pressure [39], phonation threshold pressure P_{th} for sulcus vocalis is $k_t Bc(\xi_0 + d)/T$, where k_t is a transglottal pressure coefficient, B is the viscous damping coefficient, c is the velocity of the mucosal wave, and ξ_0 is the prephonatory glottal

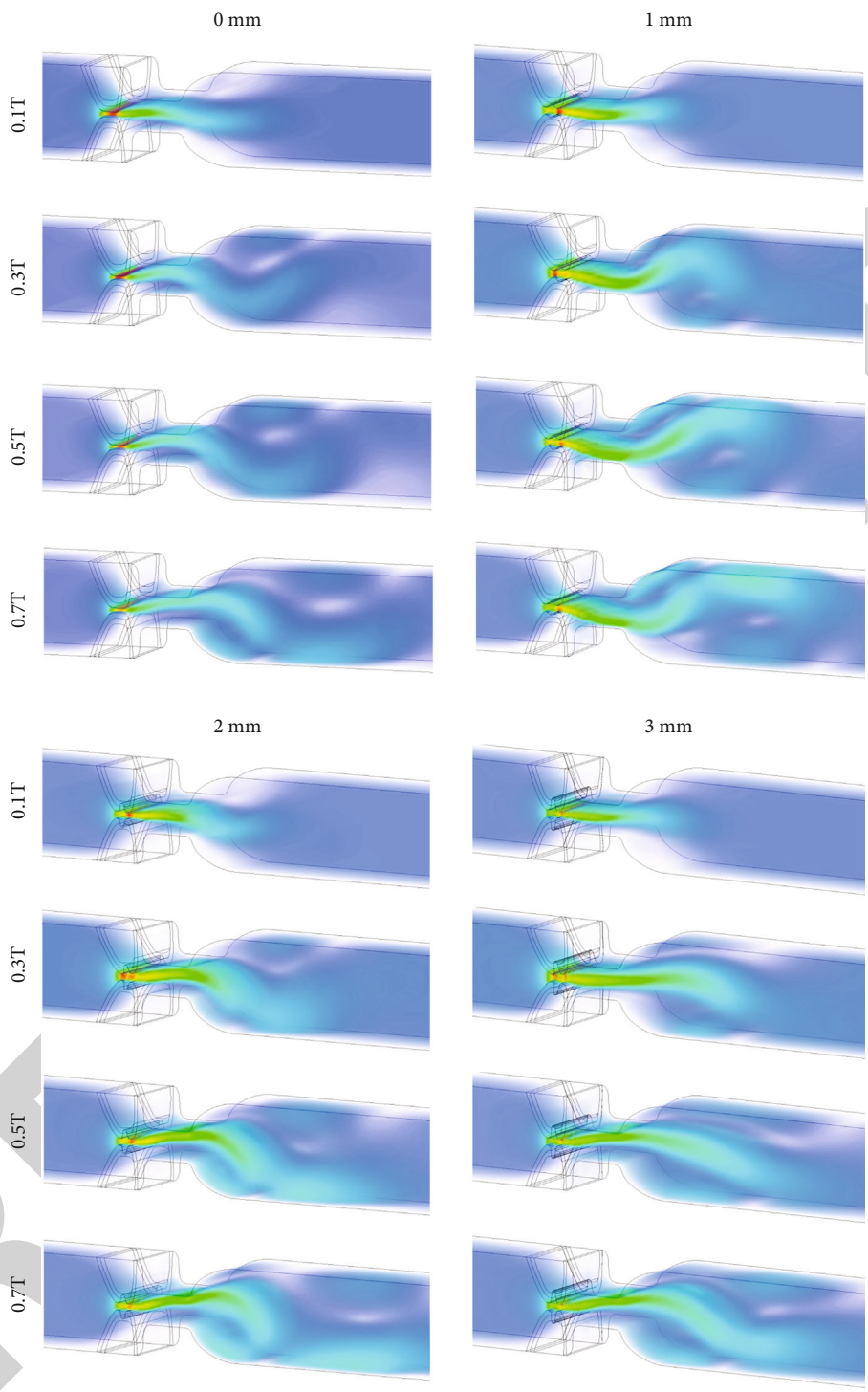


Figure 5: Contours of velocity at four different time instants in one cycle for healthy vocal fold (0 mm) and sulcus vocalis with different depths (1 mm, 2 mm, and 3 mm).

half-width. For the vocal fold model with sulcus depth of 0 mm, the vocal fold starts to vibrate when the pressure at the entrance of the glottis is up to 450 Pa. The general description for the variation of pressure with time is the following: (1) the pressures at the entrance of the glottis dropped to different suitable values to initiate and sustain vocal fold vibration, when the pressure on the inlet surface was 1000 Pa. As the vocal folds started to converge and diverge, the pressure curves were quasi-periodic waveforms based on phonation threshold pressure. (2) For sulcus depths of 0 mm, 1 mm, 2 mm, and 3 mm, the phonation threshold pressures in the sulcus vocalis were obviously larger than in

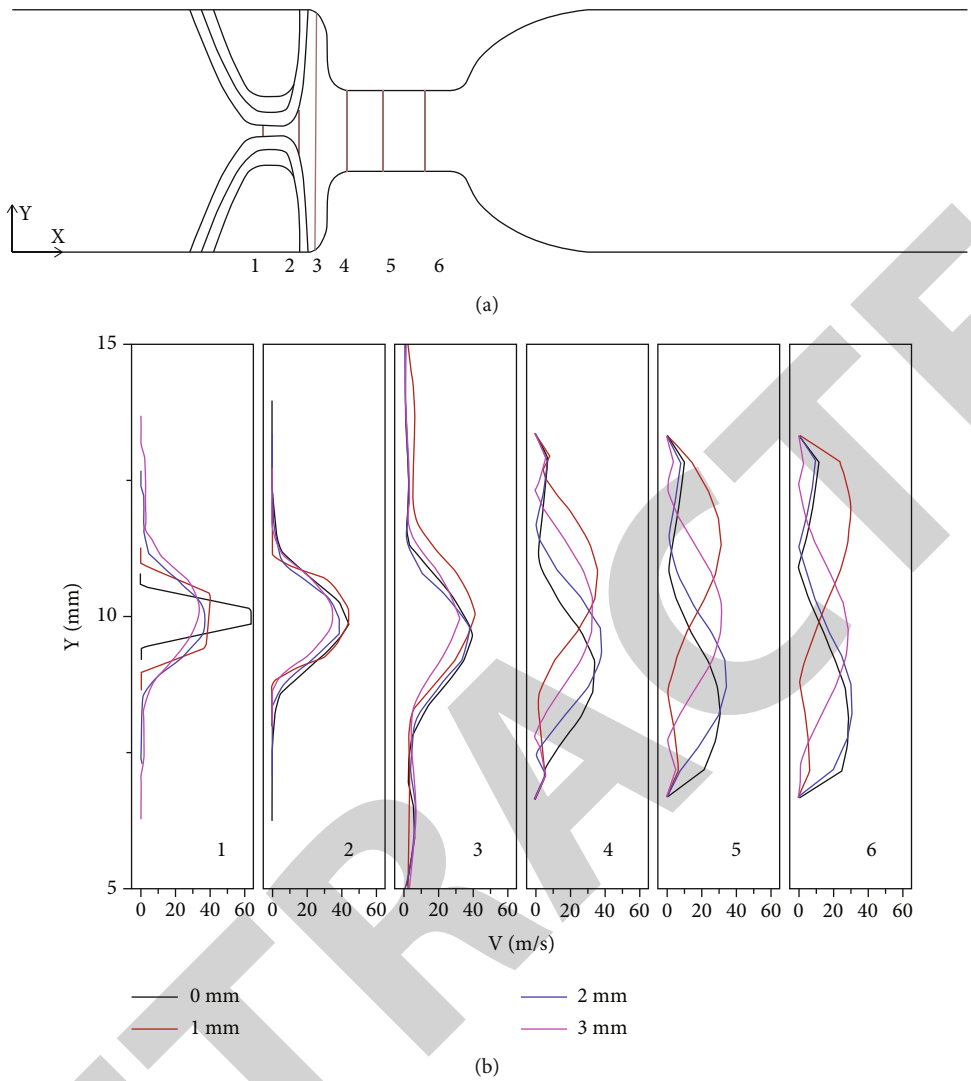


Figure 6: Total velocities, plotted along the six lines located in the X-Y plane.

the healthy vocal folds. The phonation threshold pressure raise at glottal entry became more prominent with increasing sulcus vocalis depth. The change of sulcus vocalis depth expands the prephonatory glottal half-width, which is a significant factor of phonation threshold pressure according to its expression. These results suggest that pressure distribution and phonation threshold pressure will be highly sensitive to sulcus vocalis with different depth. The deeper sulcus vocalis will increase phonation threshold pressure, making it difficult to vibrate and phonate.

5. Discussion

The increase in depth of sulcus vocalis changes the biomechanical properties of the vocal folds, which compromises the pliability of the vocal fold. The glottis closure is crippled by the indentation that extends into the ligament layer or deeper. Thus, the superior glottal angle becomes larger which leads to the lower velocity in the glottis as the sulcus vocalis depth increases from 1 mm to 3 mm. The lower velocity of the jet reduces the interactions of vortices with each other. When the sulcus vocalis depth gets biggest, the motion of vortices emerging downstream of the glottis is rarely influenced by the mutual induction. This hardly brings about asymmetric vortex structures, and there is almost no change in the flow direction. On the other hand, the increase of sulcus vocalis expands the geometry in the vertical direction suddenly. More airflow is gathered in the sulcus vocalis which could affect the supra-glottal vortex structure that is related to the glottal jet deflection.

The pressure in the glottis occurs unstable fluctuations for sulcus vocalis. When there is a sulcus vocalis, the pressure appears two short-term rebounds and returns to 0 Pa finally. Due to the discrepancy of sulcus vocalis depth, phonation threshold pressure reaches a different value. In the condition of the same lung pressure, phonation threshold pressure increases with deeper sulcus vocalis. The larger phonation threshold pressure becomes, the more energy to overcome

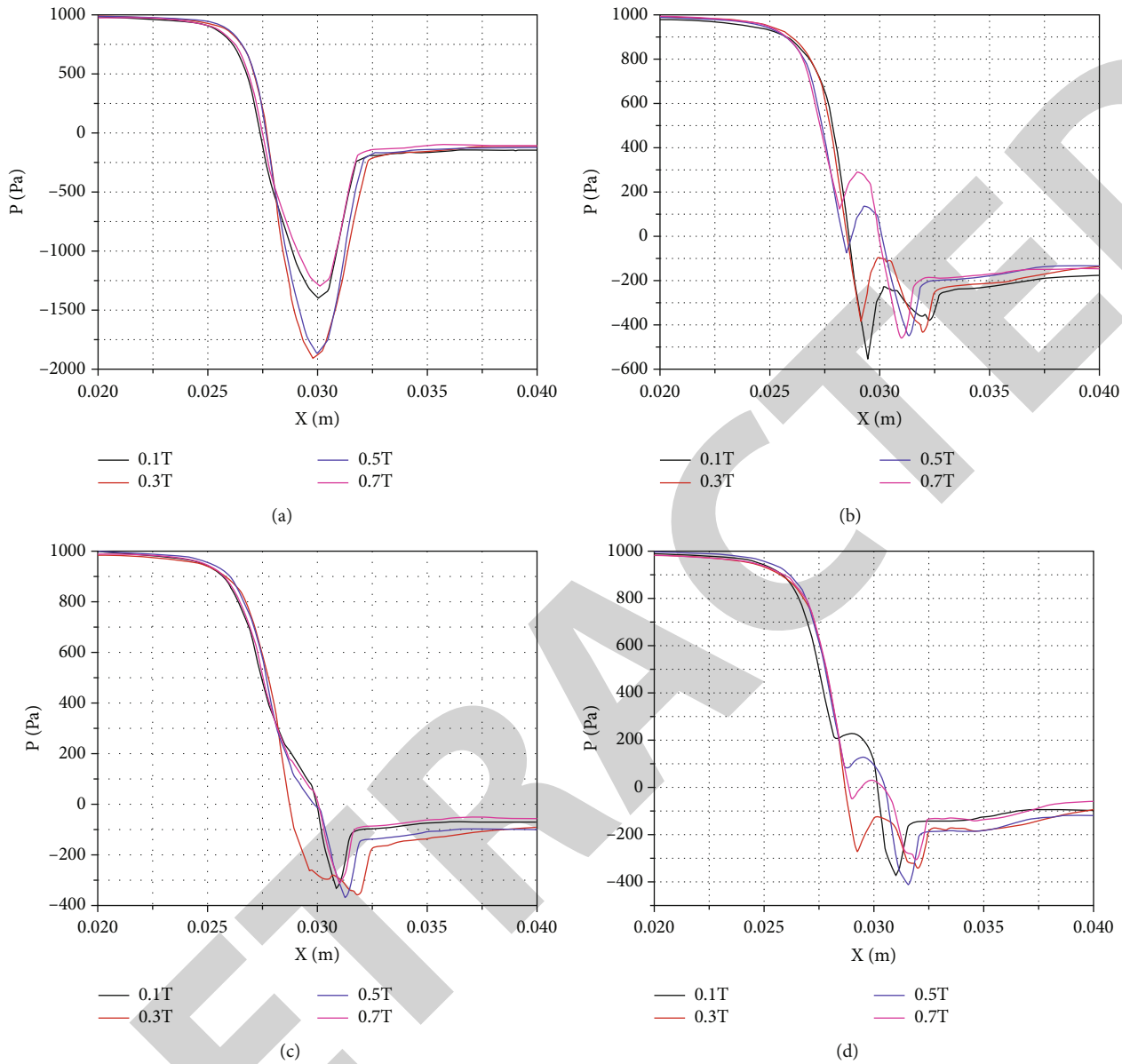


Figure 7: Glottal pressure distribution at different time-instant of one cycle. (a) 0 mm. (b) 1 mm. (c) 2 mm. (d) 3 mm.

the viscous resistance and friction of the airflow is required. Hence, the sulcus vocalis and increased sulcus vocalis depth will cause vocal fatigue and harsh vocal quality.

The sulcus vocalis models with different depths could be a tool to observe and connect the physical mechanism of the disorder and their causes to symptoms, furtherly conducting qualitative and quantitative comparison of healthy and sulcus vocalis. This method is noninvasive, which provides clinicians with guidelines about the treatment of sulcus vocalis.

6. Conclusion

Three-dimensional FSI laryngeal models that consist of fully coupled vocal folds have been applied to enhance the understanding of the effects of sulcus vocalis depth on phonation. The simulation results computed by the sequential fluid-structure coupling method for healthy vocal folds are verified to be within the normal physiological range. And then, three models with different sulcus vocalis are developed to study the vibration of vocal folds, the glottal jet dynamics, pressure distribution, and phonation threshold pressure. In addition, with increasing sulcus vocalis depth, the glottal jet deflection is not obvious and the glottal jet tends to be symmetrical. Finally, the glottal pressure has dropped twice in the direction of airflow for the sake of sulcus vocalis. As the sulcus vocalis depth increases, the phonation threshold pressures in the different types of laryngeal models get larger, which disturbs phonation and leads to poor voice quality.

The primary contributions of this study are the following: (1) under the promise of the same lung pressure, the deeper sulcus vocalis in the cover layer decreases the vibrating frequency of vocal folds. It also increases the phonation

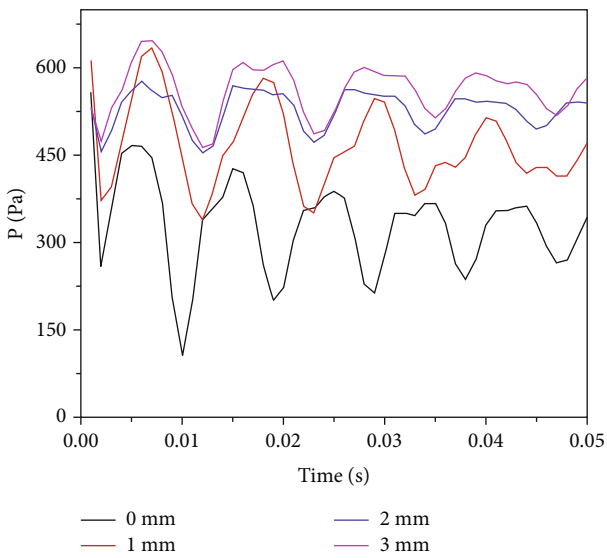


Figure 8: Variation of pressure at the entrance of the glottis with time.

threshold pressure and influences the vortex structures, which weakens the glottal jet deflection; (2) different sulcus vocalis depths correspond to three types of sulcus vocalis. The effects of sulcus depth on phonation suggest that aeroacoustic parameters like phonation threshold pressures could be used to detect different types of sulcus vocalis. It may help for the clinical diagnosis of three types of sulcus vocalis and healthy vocal folds.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.61271359 and Postgraduate Education Reform Project of Jiangsu Province No.JGLX19-141.

References

[1] A. V. Sünter, T. Kırgezen, Ö. Yiğit, and M. Çakır, "The association of sulcus vocalis and benign vocal cord lesions: intraoperative findings," *European Archives of Oto-Rhino-Laryngology*, vol. 276, no. 11, pp. 3165–3171, 2019.

[2] A. Yamauchi, H. Yokonishi, H. Imagawa et al., "Characterization of vocal fold vibration in sulcus Vocalis using high-speed digital imaging," *Journal of Speech, Language, and Hearing Research*, vol. 60, no. 1, pp. 24–37, 2017.

[3] C. N. Ford, K. Inagi, A. Khidr, D. M. Bless, and K. W. Gilchrist, "Sulcus vocalis: a rational analytical approach to diagnosis and management," *The Annals of Otology, Rhinology, and Laryngology*, vol. 105, no. 3, pp. 189–200, 1996.

[4] S. H. Dailey and C. N. Ford, "Surgical management of sulcus vocalis and vocal fold scarring," *Otolaryngologic Clinics of North America*, vol. 39, no. 1, pp. 23–42, 2006.

[5] N. V. Welham, S. H. Dailey, C. N. Ford, and D. M. Bless, "Voice handicap evaluation of patients with pathologic sulcus vocalis," *The Annals of Otology, Rhinology, and Laryngology*, vol. 116, no. 6, pp. 411–417, 2007.

[6] A. Soares, B. T. De Moares, A. N. B. . Araújo, N. G. de Biase, and J. A. Lucena, "Laryngeal and vocal characterization of asymptomatic adults with sulcus vocalis," *International Archives of Otorhinolaryngology*, vol. 23, no. 3, pp. e331–e337, 2019.

[7] C. Guo and R. C. Scherer, "Finite element simulation of glottal flow and pressure," *The Journal of the Acoustical Society of America*, vol. 94, no. 2, pp. 688–700, 1993.

[8] F. Alipour, C. Fan, and R. C. Scherer, "A numerical simulation of laryngeal flow in a forced-oscillation glottal model," *Computer Speech & Language*, vol. 10, no. 2, pp. 75–93, 1996.

[9] Q. Xue, R. Mittal, X. Zheng, and S. Bielamowicz, "A computational study of the effect of vocal-fold asymmetry on phonation," *The Journal of the Acoustical Society of America*, vol. 128, no. 2, pp. 818–827, 2010.

[10] L. P. Fulcher, R. C. Scherer, K. J. de Witt, P. Thapa, Y. Bo, and B. R. Kucinschi, "Pressure distributions in a static physical model of the hemilarynx: measurements and computations," *Journal of Voice*, vol. 24, no. 1, pp. 2–20, 2010.

[11] M. Larsson and B. Muller, "Numerical simulation of confined pulsating jets in human phonation," *Computers & Fluids*, vol. 38, no. 7, pp. 1375–1383, 2009.

[12] M. Triep and C. Brucker, "Three-dimensional nature of the glottal jet," *The Journal of the Acoustical Society of America*, vol. 127, no. 3, pp. 1537–1547, 2010.

[13] Q. Xue and X. Zheng, "The effect of false vocal folds on laryngeal flow resistance in a tubular three-dimensional computational laryngeal model," *Journal of Voice*, vol. 31, no. 3, pp. 275–281, 2017.

[14] Y. Jo, H. Ra, Y. J. Moon, and M. Döllinger, "Three-dimensional computation of flow and sound for human hemilarynx," *Computers & Fluids*, vol. 134-135, pp. 41–50, 2016.

[15] A. Vazifehdoostsaleh, N. Fatouraee, M. Navidbakhsh, and F. Izadi, "Three dimensional FSI modelling of sulcus vocalis disorders of vocal folds," *Journal of Mechanics*, vol. 34, no. 6, pp. 791–800, 2018.

[16] A. B. Sarvestani, E. G. Rad, and K. Iravani, "Numerical analysis and comparison of flow fields in normal larynx and larynx with unilateral vocal fold paralysis," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 21, no. 8, pp. 532–540, 2018.

[17] X. Zhang, L. Gu, W. Wei, D. Wu, Z. Tao, and H. Zhao, "Pathological voice source analysis system using a flow waveform-matched biomechanical model," *Applied Bionics and Biomechanics*, vol. 2018, Article ID 3158439, 13 pages, 2018.

[18] R. Mittal, B. D. Erath, and M. W. Plesniak, "Fluid dynamics of human phonation and speech," *Annual Review of Fluid Mechanics*, vol. 45, no. 1, pp. 437–467, 2013.

[19] S. Li, R. C. Scherer, L. P. Fulcher et al., "Effects of vertical glottal duct length on intraglottal pressures and phonation threshold pressure in the uniform glottis," *Journal of Voice*, vol. 32, no. 1, pp. 8–22, 2017.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9762024, 1 page
https://doi.org/10.1155/2023/9762024

Retraction

Retracted: The Optimal Adaptive-Based Neurofuzzy Control of the 3-DOF Musculoskeletal System of Human Arm in a 2D Plane

Applied Bionics and Biomechanics

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

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. Valizadeh and A. A. Akbari, "The Optimal Adaptive-Based Neurofuzzy Control of the 3-DOF Musculoskeletal System of Human Arm in a 2D Plane," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 5514693, 10 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 5514693, 10 pages
https://doi.org/10.1155/2021/5514693

Research Article

The Optimal Adaptive-Based Neurofuzzy Control of the 3-DOF Musculoskeletal System of Human Arm in a 2D Plane

Amin Valizadeh and Ali Akbar Akbari

Department of Mechanical Engineering, Ferdowsi University of Mashhad, Iran

Correspondence should be addressed to Ali Akbar Akbari; akbari@um.ac.ir

Received 7 February 2021; Revised 4 March 2021; Accepted 15 March 2021; Published 7 April 2021

Academic Editor: Fahd Abd Algalil

Each individual performs different daily activities such as reaching and lifting with his hand that shows the important role of robots designed to estimate the position of the objects or the muscle forces. Understanding the body's musculoskeletal system's learning control mechanism can lead us to develop a robust control technique that can be applied to rehabilitation robotics. The musculoskeletal model of the human arm used in this study is a 3-link robot coupled with 6 muscles which a neurofuzzy controller of TSK type along multicritic agents is used for training and learning fuzzy rules. The adaptive critic agents based on reinforcement learning oversees the controller's parameters and avoids overtraining. The simulation results show that in both states of with/without optimization, the controller can well track the desired trajectory smoothly and with acceptable accuracy. The magnitude of forces in the optimized model is significantly lower, implying the controller's correct operation. Also, links take the same trajectory with a lower overall displacement than that of the nonoptimized mode, which is consistent with the hand's natural motion, seeking the most optimum trajectory.

1. Introduction

In many countries, population aging leads to a decrease in productivity of useful work, and this will cause serious problems. Many robots are designed and employed for self-rehabilitation of elderly, disabled, damaged people in daily activities [1–11]. The hand is one part of the body that is frequently involved and employed in most individuals' daily activities. Each individual performs different daily activities such as reaching and lifting with his hand that shows the important role of robots designed in this field to estimate the position of the forces exerted by the hand. There is a growing trend worldwide for the application of handling machines, inspired by human arms, in all industrial sectors, to carry materials from one destination to the other under limited operating conditions. Advances in manipulators are manifested both in their high technical level and growing economy and safety [12]. In the robotic human arm, two links are usually used as the arm and forearm segments with two-degree-of-freedom (DOF), and at least four muscle elements are used for moving it in the 2D space. The inverse dynamic model is applied to generate joint torques in this robot [13]. The motion or force predetermined and designed by powerful controllers is used in rehabilitation applications. The training is an important factor for controlling the arm to achieve a static goal, and the body's musculoskeletal system gradually gains this capability through interaction with the surrounding environment. For example, a soccer player performs a series of random activities to deliver the ball to the gate, but the more professional he becomes in this field, the faster and more efficient he hits the ball [14, 15]. This is achieved by the gradual training of the muscles' kinematic, and the related information can be saved and used in the future [16]. Therefore, understanding the training mechanism of the musculoskeletal system of the body can lead us to employ a powerful controller for body rehabilitation robotics. Many researchers used the training controls, which will gradually train the arm controller [17–21]. Golkhou et al. [22] employed an improved Actor-Critic algorithm for the controller of a single-link musculoskeletal arm with two extensor and flexor muscles during vibrational motion. A CMAC controller was applied to the Critic section to

estimate the optimal activities and update the Actor section's coefficients. Zacharie et al. [23] applied an advanced logic-based neural network to a robotic hand. The logical function was determined based on the endpoint of the arm's arbitrary trajectory in space to compute the possible conditions of the neuron's activity to respond to the desired field. Bouganis and Shanahan [24] presented a neural network that could automatically learn to control a robotic hand with 5 degrees of freedom and the motor's initial time conditions. Kambara et al. [25] proposed a control model for motion training based on the inverse static model, direct dynamic model, and feedback control combined with Actor-Critic. Their model supported the trajectory prediction of a 2-DOF arm with six artificial muscles. Thomas et al. [26] applied an improved learning controller based on a proportional derivative control technique (PD) to control a robotic hand with four muscles for conducting the Reaching activity. Dong et al. [27, 28] implemented an adaptive sliding mode control strategy on a 2-DOF robotic hand with biarticular muscles so that the dynamic parameters were updated, which caused the input disturbances and stimulations of the system to be considered. Zadravec et al. [29] implemented an optimal controller, whose cost function was to minimize the joint torques, on a 2-DOF robotic hand. In this study, the authors could predict optimum trajectories along with the functional constraints of the muscles.

This model requires accurate dynamic parameters; however, accurately determine these parameters for different people is impractical. According to the literature review above, adaptability and optimality are the basic characteristics of the human brain, and the lack of a powerful controller that can implement the control strategy of the brain to some extent is very noticeable. In the present study, first, the equations governing the 3-link human arm's motion and the related dynamic equations are expressed in Section 2. An adaptive neurofuzzy controller is presented in the next section. The results obtained from the simulation of controllers with/without optimization are presented in Section 4. Finally, the concluded remarks of this study are described in Section 5.

2. The 3-DOF Human Arm Musculoskeletal Model

The multibody planar model of the human arm with 3-DOF is presented in Figure 1, in which the upper arm, forearms, and hand are considered three rigid links. This model considers the planar motion around three revolute joints at the shoulder, elbow, and wrist and neglects the gravitational effects. As shown in Figure 1, this model consists of six muscles that can only apply tensile forces so that each joint rotates by some of these related muscles. Muscles are assumed to be without weight and designed based on the Hill model, which are directly connected to links as [30]:

$$f_i = \bar{\omega} - \bar{\omega} b \dot{l} - \bar{\omega} c \Delta l, \bar{\omega} = \alpha . f_0 \tag{1}$$

f_i denotes the output force of ith muscle, f_0 is the maximum contractile muscle force, α expresses the activation level of controlled muscle, and $\dot{l}$ is the contractile muscle velocity. b, c is also the muscle damping coefficients and stiffness, respectively. Considering the number of six muscles, the matrix form of Eq. (1) is

$$f = [f_1 f_2 f_3 f_4 f_5 f_6]^T = A\left(U - B\dot{l} - C\Delta l\right)$$
$$\begin{cases} A = \text{diag}\,(\bar{\omega}_1, \bar{\omega}_2, \cdots, \bar{\omega}_6)\ \epsilon\ R^{6\times 6} \\ U = (1, 1, \cdots, 1)^T\ \epsilon\ R^{6\times 1} \\ B = \text{diag}\,(b_1, b_2, \cdots, b_6)\ \epsilon\ R^{6\times 6} \\ C = \text{diag}\,(c_1, c_2, \cdots, c_6)\ \epsilon\ R^{6\times 6} \\ \Delta l = \text{diag}\,(\Delta l_1, \Delta l_2, \cdots, \Delta l_6)^T\ \epsilon\ R^{6\times 1} \end{cases} . \tag{2}$$

The following equation expresses the relation between the position vector of the end effector of the arm and joint angles:

$$X = \begin{bmatrix} L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}, \tag{3}$$

where L_1, L_2, and L_3 represent the first, second, and third links, respectively. θ_1, θ_2, and θ_3 are also the relevant link's angle to the x-axis, the second and third link. The velocity at the end effector of the arm, which is dependent on angular velocities, are expressed as follows:

$$\begin{aligned} &\dot{X} = J\dot{\theta} \\ &J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \end{bmatrix} \\ &J_{11} = -L_1\, \boldsymbol{sin}(\theta_1) - L_2\, \boldsymbol{sin}(\theta_1 + \theta_2) - L_3\, \boldsymbol{sin}(\theta_1 + \theta_2 + \theta_3)\,; \\ &J_{12} = -L_2\, \boldsymbol{sin}(\theta_1 + \theta_2) - L_3\, \boldsymbol{sin}(\theta_1 + \theta_2 + \theta_3)\,; \\ &J_{13} = -L_2\, \boldsymbol{sin}(\theta_1 + \theta_2) - L_3\, \boldsymbol{sin}(\theta_1 + \theta_2 + \theta_3)\,; \\ &J_{21} = L_1\, \boldsymbol{cos}(\theta_1) - L_2\, \boldsymbol{cos}(\theta_1 + \theta_2) + L_3\, \boldsymbol{cos}(\theta_1 + \theta_2 + \theta_3)\,; \\ &J_{22} = L_2\, \boldsymbol{cos}(\theta_1 + \theta_2) + L_3\, \boldsymbol{cos}(\theta_1 + \theta_2 + \theta_3)\,; \\ &J_{23} = L_3\, \boldsymbol{cos}(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{4}$$

$J \in R^{2\times 3}$ is the Jacobian matrix that shows the relation between the arm's end effector's linear velocities and angular velocities. The length vectors of the muscles are defined as

$$l = [\, l_1 \quad l_2 \quad l_3 \quad l_4 \quad l_5 \quad l_6 \,]^T, \tag{5}$$

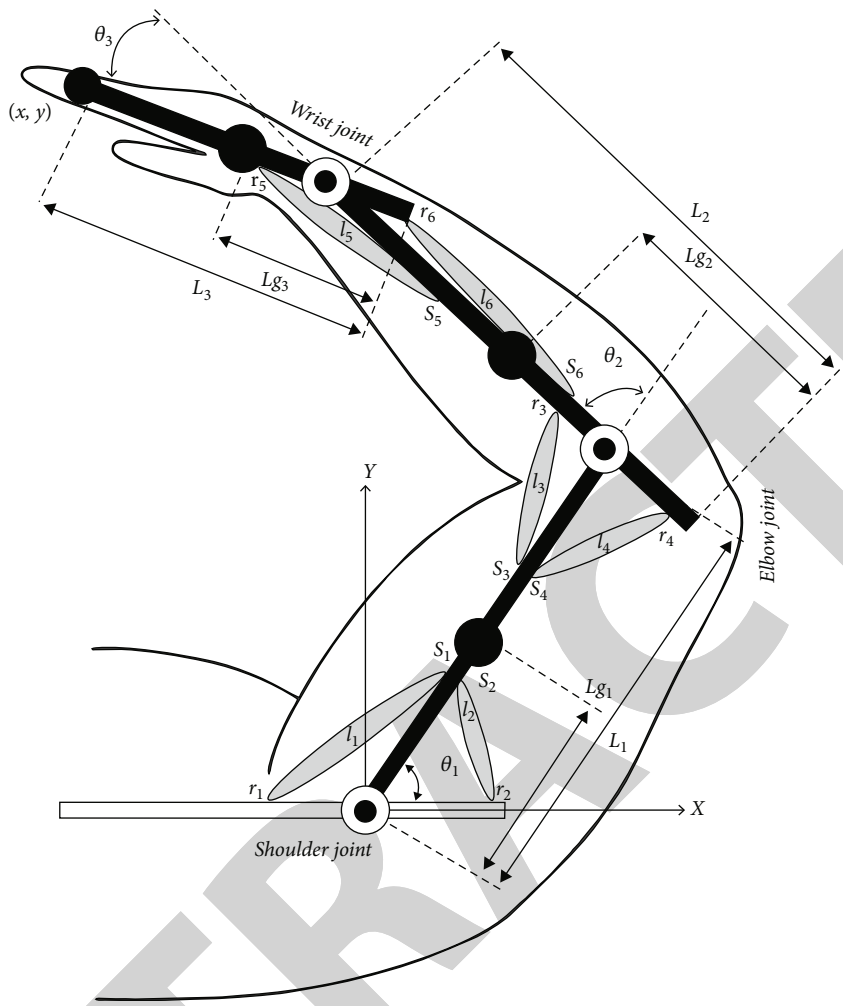


Figure 1: Schematic view of the 3-DOF musculoskeletal model.

and

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix} = \begin{bmatrix} r_1\left(\pi - \theta_1 - acos\left(\frac{r_1}{s_1}\right)\right) + \sqrt{s_1^2 - r_1^2} \\ r_2\left(\frac{\pi}{2} + \theta_1 - acos\left(\frac{r_2}{s_2}\right)\right) + \sqrt{s_2^2 - r_2^2} \\ r_3\left(\pi - \theta_2 - acos\left(\frac{r_3}{s_3}\right)\right) + \sqrt{s_3^2 - r_3^2} \\ r_4\left(\frac{\pi}{2} + \theta_2 - acos\left(\frac{r_4}{s_4}\right)\right) + \sqrt{s_4^2 - r_4^2} \\ r_5\left(\frac{\pi}{2} + \theta_3 - acos\left(\frac{r_5}{s_5}\right)\right) + \sqrt{s_5^2 - r_5^2} \\ r_6\left(\frac{\pi}{2} + \theta_3 - acos\left(\frac{r_6}{s_6}\right)\right) + \sqrt{s_6^2 - r_6^2} \end{bmatrix}, \quad (6)$$

where r_{1-6} and s_{1-6} represent the torque surfaces, as shown in Figure 1. The following equation is obtained by taking the time derivative of the above equation to time:

$$\dot{l} = W\dot{\theta}. \quad (7)$$

$W \in R^{6\times 3}$ is the Jacobian matrix, which relates the muscles' contractile rate to the joints' angular velocity, and $\dot{l} = \begin{bmatrix} \dot{l}_1 & \dot{l}_2 & \dot{l}_3 & \dot{l}_4 & \dot{l}_5 & \dot{l}_6 \end{bmatrix}^T$ represents the stretch rate of muscles. By applying the principle of virtual work, the work done by muscle torque is defined as follows:

$$\tau = -W^T f_m, \quad (8)$$

where $f_m = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}^T$ is the vector represents the tensile forces of muscles and $\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$ is the joint torque vector. As depicted in Table 1, by putting muscle parameters in Eq. (6), W is defined as follows:

$$W = \begin{bmatrix} -r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -r_3 & r_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -r_5 & r_6 \end{bmatrix}^T. \quad (9)$$

Table 1: Numerical values related to the muscle's geometrical shape.

Muscle	Value (m)	
l_1	$r_1 = 0.055$	$s_1 = 0.080$
l_2	$r_2 = 0.055$	$s_2 = 0.080$
l_3	$r_3 = 0.030$	$s_3 = 0.120$
l_4	$r_4 = 0.030$	$s_4 = 0.120$
l_5	$r_5 = 0.035$	$s_5 = 0.220$
l_6	$r_6 = 0.040$	$s_6 = 0.250$

Using Lagrange's equations [29]

$$H(\theta)\ddot{\theta} + \left\{\frac{1}{2}\dot{H}(\theta) + C\left(\dot{\theta}, \theta\right)\right\}\dot{\theta} = \tau. \tag{10}$$

H is a symmetric matrix representing the mass momentum, and C is a skew-symmetric matrix of Coriolis, centrifugal, and friction torques. By substituting Eq. (8) into the above equation, the dynamic equations of the musculoskeletal system are obtained as

$$H(\theta)\ddot{\theta} + \left\{\frac{1}{2}\dot{H}(\theta) + C\left(\dot{\theta}, \theta\right)\right\}\dot{\theta} = -W^T f_m. \tag{11}$$

3. Controller Design

The controller design's main purpose is to use appropriate motion commands for each muscle in the process of interacting with the environment and learning the kinematics of the arm in the movement toward a fixed target. Neurofuzzy systems are a combination of neural networks with fuzzy logic systems and utilized to simplify problems and apply the subjective, complex rules and concepts. To mimic the human brain's function in these systems, which consists of a set of artificial neurons, an artificial neural network is used with fuzzy logic rules. Ghanooni et al. [31] found that the adaptive multicritic neurofuzzy control framework can help identify the unknown systems and suggested that the computational load required for this controller's parameters compatibility is lower than the conventional neurofuzzy controllers, and this is one of the advantages of this controller in real-time applications. They also claimed that their controller would benefit from the reinforcement learning compared to supervisory learning in the online evaluation of the output, which led to the capability of controlling any uncertainty in the system.

A new structure of adaptive neurofuzzy control framework composed of several inputs and outputs based on reinforcement learning was investigated by Balaghi et al. [32]. Their study aimed to control the motion trajectory by optimizing a 2-DOF model of the human arm's contractile muscle forces. The "critic estimates the system's achievement," and the "actor" updates the controller parameters by generating the associated signal. They argued that the difficulty of determining the precise arm's biological specification values such as mass and inertia made them use this controller because it is independent of the model parameters. Moreover, this controller's generated inputs are optimum, which is significant in the musculoskeletal system due to the biological limitations of human muscle limitations. This controller is implemented for the existing 3-DOF model in this study because of the advantages mentioned above. The model's endpoint has to be directed on the arbitrary trajectory for all initial values in the X and Y direction by multiple muscle contractile forces. Hence, a multiple-input and multiple-output system (MIMO) consisting of muscle inputs and endpoint outputs should be considered.

3.1. Neurofuzzy Network. Fuzzy systems consist a fuzzification unit, a defuzzification unit, a fuzzy rule base, and an inference engine. The fuzzy system can be regarded as performing a real and nonlinear mapping from an input vector $x \in R^n$ to an output vector $y = f(x) \in R^m$, where m and n are the dimensions of the input and output vectors, respectively. The bitwise interfaces of the real and fuzzy worlds are fuzzifier and defuzzifier, respectively. The earlier addresses real inputs to the associated fuzzy sets, and the latter serves to address the fuzzy sets of output variables to the associated real outputs in the reverse direction.

Two types of fuzzy systems, called Takagi-Sugeno-Kang (TSK) and systems with fuzzifiers and defuzzifiers (Mamdani), are more common in the literature, and the TSK type is used in this study for adaptive neurofuzzy control framework. The multi-input single-output (MISO) neurofuzzy system—including N rules—is defined as follows:

$Rule_i$: if (u_1 is A_{i1}) and if (u_2 is A_{i2}) and … and if (u_n is A_{im})

then if $y = G_i(u_1, u_2, \cdots, u_m)$

where i is the rule number, u_m are the inputs with m number, A_{im} indicates the fuzzy set for inputs, and G_i which is the linear relation of inputs evaluated as a crisp function as

$$G_i = \omega_{i0} + \omega_{i1}u_1 + \cdots + \omega_{im}u_m. \tag{12}$$

Consequently, the TSK neurofuzzy output can be expressed as

$$y = \frac{\sum_{i=1}^{M}\mu_i G_i}{\sum_{i=1}^{M}\mu_i} = \sum_{i=1}^{M}\mu_i G_i, \mu_i(u_1, u_2, \cdots, u_m) = \prod_{j=1}^{P}\mu_{ij}\left(u_j\right). \tag{13}$$

In Eq. (12), M is the number of rules, and μ_i is the membership function for the ith rule.

The inputs of the adaptive critic-based neurofuzzy controller applied to the endpoint of the human arm model in this study are e_x, $\dot{e}_x$, e_y, and $\dot{e}_y$ as

$$\begin{cases} e_x = x_d - x \\ e_y = y_d - y \end{cases}, \begin{cases} \dot{e}_x = \dot{x}_d - \dot{x} \\ \dot{e}_y = \dot{y}_d - \dot{y} \end{cases}, \tag{14}$$

where (x_d, y_d) and (x, y) are the desired and real output of the system in the 2-D workspace, respectively. $(\dot{x}_d, \dot{y}_d)$ and

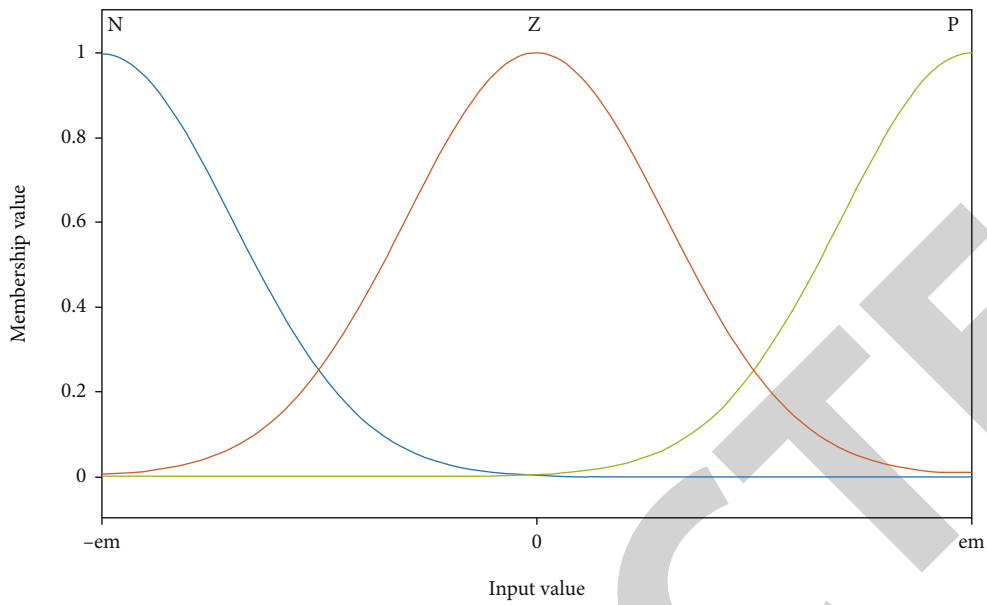


Figure 2: The input membership function of the TSK fuzzy system.

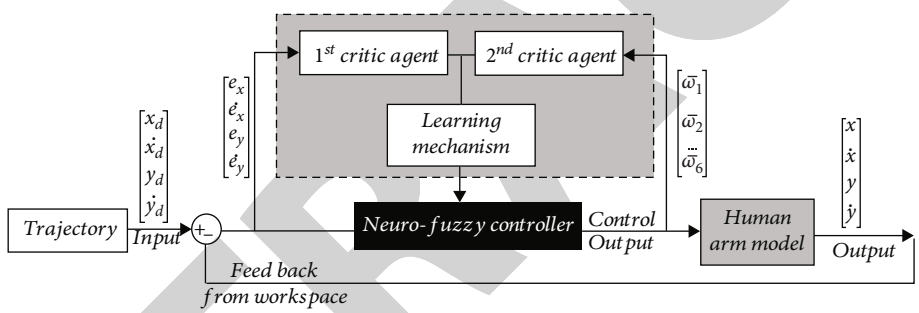


Figure 3: Controller block diagram and system critic rules.

$(\dot{x}, \dot{y})$ are also the desired and real velocity of the arm's end-point in the task space. As a consequence, the vector form of the TSK neurofuzzy output for M rules is calculated as

$$\bar{\omega} = \mu\omega X = [\mu_1 \quad \mu_2 \quad \cdots \quad \mu_M] \begin{bmatrix} \omega_{1,0} & \omega_{1,1} & \omega_{1,2} & \omega_{1,3} & \omega_{1,4} \\ & \cdot & \cdot & \cdot & \\ \omega_{j,0} & \omega_{j,1} & \omega_{j,2} & \omega_{j,3} & \omega_{j,4} \\ & \cdot & \cdot & \cdot & \\ \omega_{M,0} & \omega_{M,1} & \omega_{M,2} & \omega_{M,3} & \omega_{M,4} \end{bmatrix} \begin{bmatrix} 1 \\ e_x \\ \dot{e}_x \\ e_y \\ \dot{e}_y \end{bmatrix}. \tag{15}$$

The fuzzy system in an adaptive neural network is a standard TSK system, which leads to the formation of a four-layered network. In the first layer, all inputs are directed into the [-1, 1] scope of the membership function. Based on Figure 2, three membership functions were determined for each input and labeled using N, Z, and P, representing the negative, zero, and positive expression, respectively. Also, the fuzzification and defuzzification process is performing in the second and fourth layers, respectively. The third layer performs decision-making with Max-Product law. Therefore, there are 81 rules for each controller of the TSK system.

Table 2: Numerical values of the model.

	Length (m)	Mass (kg)	Inertial moment (kg·m^2)	CoM position (m)
1st link	0.31	1.93	0.0141	0.165
2nd link	0.27	1.32	0.0120	0.135
3nd link	0.15	0.35	0.0010	0.075

3.2. Adaptive Critic. The critic agent is the main part of any learning system. Each critic agent examines a system's state by evaluating its output and generates a critic signal called r. The signal r is a real number in the range of [-1, 1] and is implemented by the learning process to train and adjust the TSK fuzzy system's parameters to minimize the signal to reach zero value indicates that the system does not require more training. In multicritic systems, the evaluation of a system's performance is carried out by each agent separately. Accordingly, all critic signals should become zero, which

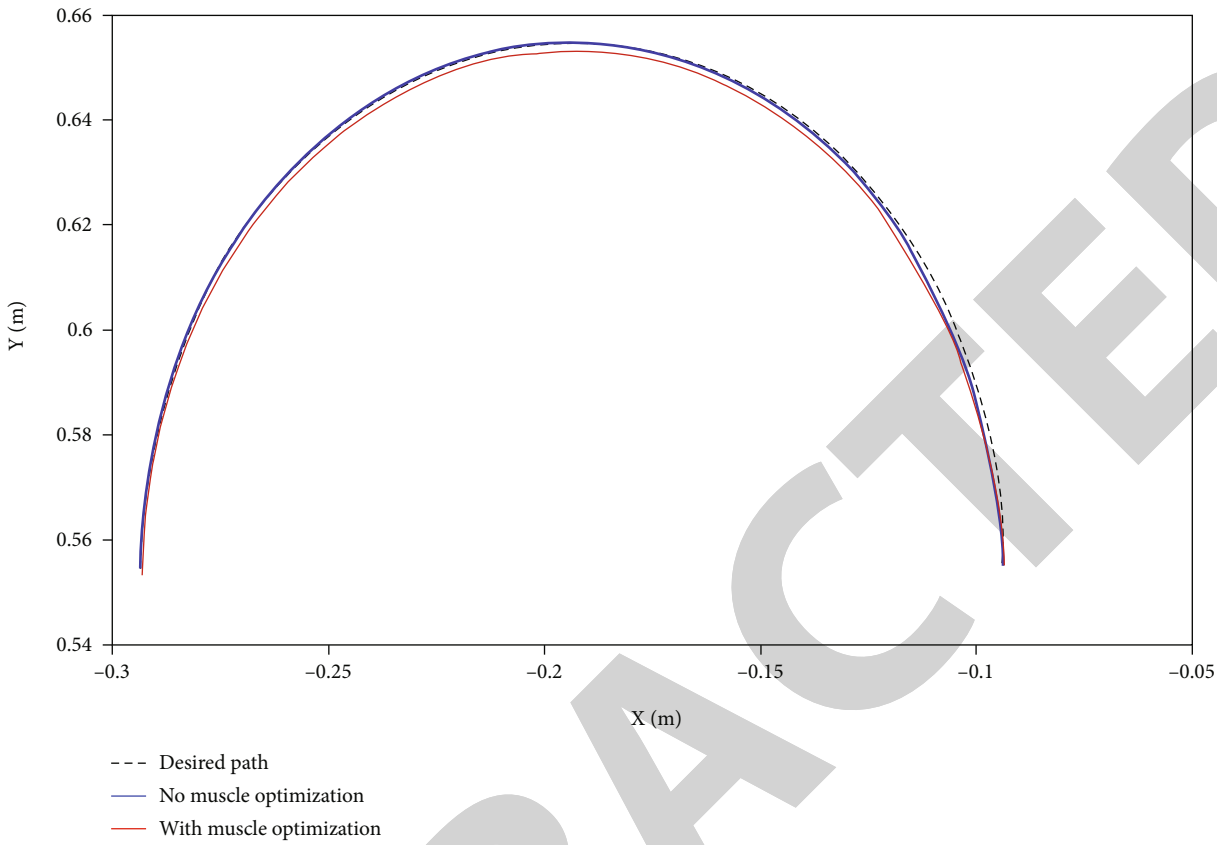


Figure 4: The moving trajectory of the model with/without optimizing the muscular forces.

indicates the critic is satisfied by the system's performance. Here, two cost functions are studied to gratified the critics by minimizing as [33]

$$E = E_e + E_f = \frac{1}{2}k_e(h_1 e + h_2 \dot{e})^2 + \frac{1}{2}k_f(abs(\alpha))^2, \tag{16}$$

where e and $\dot{e}$ are position and velocity tracing error of the arm's endpoint, k_e and k_f are the critics' weight, which indicates the component preferences in the cost function, h_1 and h_2 scale variables bring the items in [-1, 1], and α as mentioned before is the activation level of controlled muscle. In Eq. (16), the second term of the right-hand side is the TSK system's optimization, which minimizes the muscles' tensile forces. The reform of the above equation for the number of m muscles and s system's output is represented as

$$E = \frac{1}{2}\sum_{i=1}^{s} k_i \left(\frac{e_i + d_i \dot{e}_i}{(e_i + d_i \dot{e}_i)_{\max}}\right)^2 + \frac{1}{2}\sum_{j=1}^{m} k_j \left(\frac{f_j}{f_{j,\max}}\right)^2, \tag{17}$$

where f_j is the contractile force of jth muscle, $f_{j,\max}$ is the max amount of f_j, k_i and k_j are the critic weights, and d_i is an arbitrary positive number. As stated, the aim of controlling the musculoskeletal system is that the arm's endpoint reaches the desired position simultaneously with minimizing the contractile muscle forces; thus, in Eq. (17), $s = 2$ and $m = 6$. The block diagram and critic rules of the controller are shown in Figure 3.

3.3. Learning System. As previously described, the primary purpose of the learning mechanism is to minimize the error function's critic effects and satisfy all critic's criteria. In a learning system, updating neurofuzzy control parameters by critical signals is called emotional training. Therefore, emotional training aims to minimize the cost function E in Eq. (17). By using the Newton gradient descent method, the variation in critic weights should conform to the following rule:

$$\Delta\omega = -\eta \frac{\partial E}{\partial \omega}, \tag{18}$$

where η is the learning rate of the corresponding neurofuzzy controller and ω is the adjustable parameter of the controller. Substituting Eq. (17) and Eq. (18) in the above equation and using chain rule yields in

$$\begin{aligned}\Delta\omega_m &= -\eta \frac{\partial(E_e + E_f)}{\partial \omega_m} \\ &= -\eta\left(\frac{\partial E_e}{\partial e_x}\frac{\partial e_x}{\partial x}\left(\sum_{j=1}^{3}\frac{\partial x}{\partial \theta_j}\sum_{i=1}^{3}\frac{\partial \theta_i}{\partial \tau_i}\frac{\partial \tau_i}{\partial f_m}\right)\frac{\partial f_m}{\partial \omega_m}\right. \\ &\quad -\eta\left(\frac{\partial E_e}{\partial e_y}\frac{\partial e_y}{\partial y}\left(\sum_{j=1}^{3}\frac{\partial y}{\partial \theta_j}\sum_{i=1}^{3}\frac{\partial \theta_i}{\partial \tau_i}\frac{\partial \tau_i}{\partial f_m}\right)\frac{\partial f_m}{\partial \omega_m}\right. \\ &\quad -\eta\frac{\partial E_f}{\partial f_m}\frac{\partial f_m}{\partial \omega_m}, (m = 1, 2, \cdots, 6).\end{aligned} \tag{19}$$

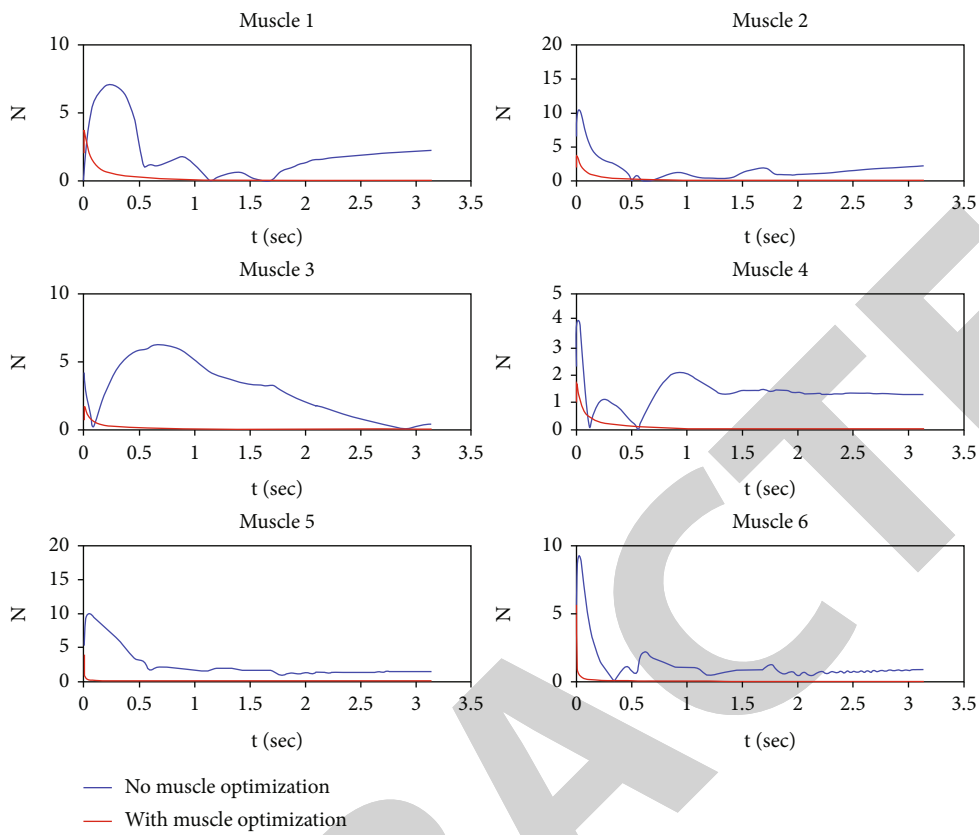


Figure 5: The input forces of each muscle during the motion.

m is the number of inputs to the model and the term, and $\partial\tau_i/\partial f_m$ is the Jacobian matrix in Eq. (7). According to the method in Ref. [33], which proposes a matrix by implementing a neural network, the Jacobian term $(\partial\theta_i/\partial\tau_i)$ is obtained as

$$\acute{J}_{3\times3} = \frac{\partial\theta_i}{\partial\tau_i} \approx H(\theta)^{-1}, \tag{20}$$

where $H(\theta)$ is the mass momentum in the dynamic equation of the system. Also, by taking Eq. (1), into account, the term $\partial f_m/\partial\omega_m$ is calculated as

$$\frac{\partial f_m}{\partial\omega_m} = \frac{\partial f_m}{\partial\bar{\omega}}\frac{\partial\bar{\omega}}{\partial\omega_m} = \left(1 - b\dot{l} - c\Delta l\right)\mu X. \tag{21}$$

Eq. (19) updates the coefficients of the TSK controller as the critic rule.

4. Results

In this section, the 3-DOF model, along with the neurofuzzy critic-based controller allocated individually to each of the muscles, is simulated numerically. First, the limits of membership functions e and $\dot{e}$ are determined for the TSK system. Simulation of the model by arbitrary shows that the values of $e_x = e_y = 0.2$, as well as $\dot{e}_x = \dot{e}_y = 0.4$, can be acceptable. In the next step, the initial values selected randomly in the range of $[-100,100]$ are assigned to matrix ω in Eq. (15), for six muscles. These coefficients are updated by Eq. (19) in each step to minimize the cost function value. The minimization is originally conducted by minimizing the error values of e and $\dot{e}$ that finally resulted in the appropriate system's performance. The parameters related to the controller are selected as

$$\begin{cases} d_i = 1, k_i = 10^{-4} (\text{for } i = 1, 2) \\ k_j = \text{ones}1.6 \times 10^{-5} \ (\text{for } j = 6)\,. \\ \eta = 10^4 \end{cases} \tag{22}$$

To indicate the controller's performance without considering the effect of muscle optimization, the process is performed also with $k_j = \text{zeros}1.6 \times 10^{-5}$ (for $j = 6$). The model parameters and the values associated with the joint types of muscles are listed in Table 2 and Table 1.

For evaluating the controller's performance, a semicircular trajectory is applied by the following equation in the workspace:

$$\begin{cases} x = 0.2 + 0.1\cos(t) \\ y = 0.55 + 0.1\sin(t) \end{cases}, (x_0, y_0) = (-0.1, 0.55). \tag{23}$$

The total simulation time is assumed to be $T = \pi\,(s)$, and during the aforementioned period, the model is expected to fully go through the trajectory. To show the controller's

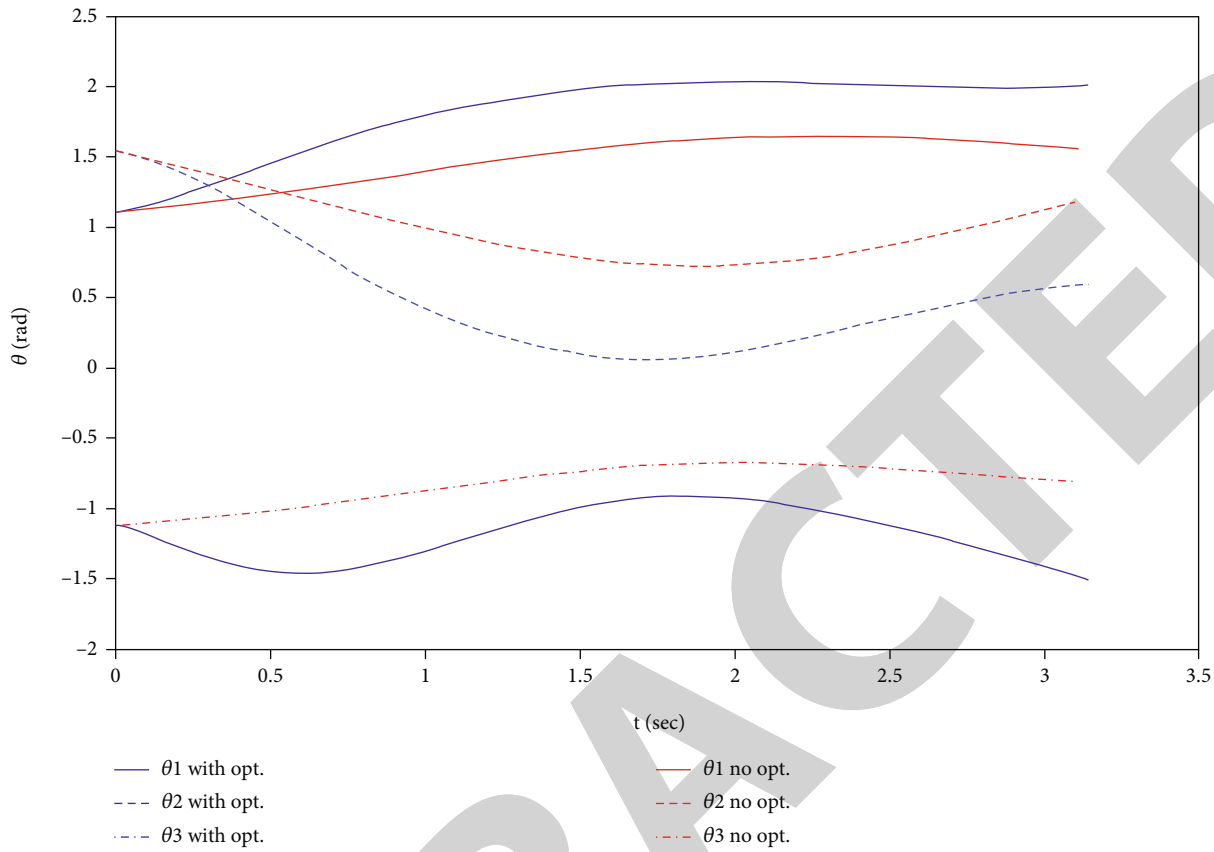


Figure 6: The displacement of each joint during motion.

robustness against the system uncertainties, a 10% diversion is considered for the values of the mass and inertia in the model. Figure 4 displays the arm model's motion trajectory in both cases with and without considering muscles' optimization. As it is depicted, both models can follow the desired trajectory with acceptable accuracy. It should be noted that the following error is a little less for nonoptimized mode. This is because the focus, in this case, is only on reducing the trajectory error, and the model is not seeking to optimize the muscle forces.

Figure 5 shows the magnitude of forces applied to each muscle during the motion. The muscle forces' values are significantly lower in the optimized mode, showing the controller's correct performance. The maximum values of the forces are also in the intended range and controlled properly. These limited values in muscle forces are one of the main features that resulted from applying optimal control to the model.

Finally, Figure 6 illustrates how each joint displaces during motion. The proposed values imply that the two cases have select completely different configurations to go through the trajectory. In the case of optimized muscles, the displacement of muscles is lower, i.e., links got through the trajectory with a lower overall displacement than the optimized muscles. The obtained results are in good agreement with the hand's natural motion, which is always sought the optimal trajectory of motion. This figure shows the advantage of the muscle optimization method.

5. Conclusion

The given trajectory was followed properly by controllers with/without muscle optimization. However, the tracking error was slightly lower in the absence of optimization, caused by the controller's focus to track the desired trajectory without minimizing the muscle forces. In conjunction with the optimal controller, the muscle forces were much lower than those of the nonoptimal controller, suggesting a significant role of muscle optimization in improving the controller's performance. The maximum values of muscle forces were also in the desired range and well-controlled. This limited force is one of the main features of the optimal control strategy applied to the model. In the case of optimized muscles, the joints displacement was lower, i.e., links go through the trajectory with a lower overall displacement compared to nonoptimized muscles case, and this shows the good agreement of results with the natural motion of the hand, which is always sought the optimal trajectory of motion. We intend to enable the movement of the arm exactly along complex trajectories as well as the compensation of dominant external disturbances [34, 35]. Moreover, future research will mainly aim to experimentally analyze the results obtained. The feasibility of the proposed neurofuzzy control system is proposed for future researches. The proposed neurofuzzy controller should contain essential features such as adaptivity and muscle force optimization. Moreover, other methods such as

machine learning and artificial intelligence can be applied to reach optimum results.

Data Availability

The data is extracted from the paper entitled "On Control of Reaching Movements for Musculo-Skeletal Redundant Arm Model".

Conflicts of Interest

The authors declare that they have no conflicts of interest in relation to this research.

Acknowledgments

The authors declare that they have no financial fund from any resource in relation to this research and its publication.

References

[1] E. Guizzo and H. Goldstein, "The rise of the body bots [robotic exoskeletons]," *IEEE Spectrum*, vol. 42, no. 10, pp. 50–56, 2005.

[2] B. Dellon and Y. Matsuoka, "Prosthetics, exoskeletons, and rehabilitation [grand challenges of robotics]," *IEEE robotics & automation magazine*, vol. 14, no. 1, pp. 30–34, 2007.

[3] M. A. Oskoei and H. Hu, "Myoelectric control systems–A survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.

[4] J. Rosen, M. Brand, M. B. Fuchs, and M. Arcan, "A myosignal-based powered exoskeleton system," *IEEE Transactions on systems, Man, and Cybernetics-part A: Systems and humans*, vol. 31, no. 3, pp. 210–222, 2001.

[5] K. Kiguchi, T. Tanaka, and T. Fukuda, "Neuro-fuzzy control of a robotic exoskeleton with EMG signals," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 481–490, 2004.

[6] K. Kiguchi, M. H. Rahman, M. Sasaki, and K. Teramoto, "Development of a 3DOF mobile exoskeleton robot for human upper-limb motion assist," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 678–691, 2008.

[7] D. Sasaki, "Development of pneumatic soft Robot hand for human friendly robot," *Journal of Robotics and Mechatronics*, vol. 15, no. 2, pp. 164–171, 2003.

[8] J. L. Pons, E. Rocon, A. F. Ruiz, and J. C. Moreno, "Upper-limb robotic rehabilitation exoskeleton: tremor suppression," *Rehabilitation Robotics*, vol. 1, pp. 453–470, 2007.

[9] N. G. Tsagarakis and D. G. Caldwell, "Development and control of a 'soft-actuated'exoskeleton for use in physiotherapy and training," *Autonomous Robots*, vol. 15, no. 1, pp. 21–33, 2003.

[10] K. Kong and D. Jeon, "Design and control of an exoskeleton for the elderly and patients," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 4, pp. 428–432, 2006.

[11] L. Lucas, M. DiCicco, and Y. Matsuoka, "An EMG-controlled hand exoskeleton for natural pinching," *Journal of Robotics and Mechatronics*, vol. 16, no. 5, pp. 482–488, 2004.

[12] M. Blatnický, J. Dižo, M. Sága, J. Gerlici, and E. Kuba, "Design of a mechanical part of an automated platform for oblique manipulation," *Applied Sciences*, vol. 10, no. 23, p. 8467, 2020.

[13] H. Kazerooni and R. Steger, "The Berkeley lower extremity exoskeleton," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 14–25, 2006.

[14] J. Konczak and J. Dichgans, "The development toward stereotypic arm kinematics during reaching in the first 3 years of life," *Experimental Brain Research*, vol. 117, no. 2, pp. 346–354, 1997.

[15] K. Tahara, S. Arimoto, M. Sekimoto, and Z. W. Luo, "On control of reaching movements for musculo-skeletal redundant arm model," *Applied Bionics and Biomechanics*, vol. 6, no. 1, pp. 11–26, 2009.

[16] A. R. Luft and M. M. Buitrago, "Stages of motor skill learning," *Molecular Neurobiology*, vol. 32, no. 3, pp. 205–216, 2005.

[17] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological Cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.

[18] P. Haggard, G. Leschziner, R. C. Miall, and J. F. Stein, "Local learning of inverse kinematics in human reaching movement," *Human Movement Science*, vol. 16, no. 1, pp. 133–147, 1997.

[19] M. G. Pandy, F. C. Anderson, and D. G. Hull, "A parameter optimization approach for the optimal control of large-scale musculoskeletal systems," *Journal of Biomechanical Engineering*, vol. 114, no. 4, pp. 450–460, 1992.

[20] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature Neuroscience*, vol. 5, no. 11, pp. 1226–1235, 2002.

[21] H. Miyamoto, E. Nakano, D. M. Wolpert, and M. Kawato, "TOPS (task optimization in the presence of signal-dependent noise) model," *Systems and Computers in Japan*, vol. 35, no. 11, pp. 48–58, 2004.

[22] V. GOLKHOU, C. LUCAS, and M. PARNIANPOUR, "Application of actor-critic reinforcement learning method for control of a sagittal arm during oscillatory movement," *Biomedical Engineering: Applications, Basis and Communications*, vol. 16, no. 6, pp. 305–312, 2004.

[23] M. Zacharie, "Advanced logistic belief neural network algorithm for robot arm control," *Journal of Computer Science*, vol. 8, no. 6, p. 965, 2012.

[24] A. Bouganis and M. Shanahan, "Training a spiking neural network to control a 4-dof robotic arm based on spike timing-dependent plasticity," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Barcelona, Spain, 2010, July.

[25] H. Kambara, K. Kim, D. Shin, M. Sato, and Y. Koike, "Motor control-learning model for reaching movements," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 555–562, IEEE, Vancouver, BC, Canada, 2006, July.

[26] K. M. Jagodnik, P. S. Thomas, A. J. van den Bogert, M. S. Branicky, and R. F. Kirsch, "Training an actor-critic reinforcement learning controller for arm movement using human-generated rewards," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1892–1905, 2017.

[27] H. Dong and N. Mavridis, "Adaptive biarticular muscle force control for humanoid robot arms," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 284–290, IEEE, Osaka, Japan, 2012, November.

[28] H. Dong, N. Figueroa, and A. El Saddik, "Muscle force control of a kinematically redundant bionic arm with real-time parameter update," in *2013 IEEE International Conference on*

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9848196, 1 page
https://doi.org/10.1155/2023/9848196

Retraction

Retracted: Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier

Applied Bionics and Biomechanics

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

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] Z. N. Ali, I. Askerzade, and S. Abdulwahab, "Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6670316, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 6670316, 9 pages
https://doi.org/10.1155/2021/6670316

Research Article

Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier

Zainab N. Ali,[1] **Iman Askerzade**,[1] **and Saddam Abdulwahab**[2]

[1]*Computer Engineering, Engineering Faculty, Ankara University, 06830, Turkey*
[2]*Department of Computer Engineering and Mathematics, Rovira i Virgili University, 43007, Spain*

Correspondence should be addressed to Zainab N. Ali; znali@ankara.edu.tr

Received 17 October 2020; Revised 7 January 2021; Accepted 6 February 2021; Published 26 February 2021

Academic Editor: Mohammed Yahya Alzahrani

Estimation of the quality of food products is vital in determining the properties and validity of the food concerning the baking and other manufacturing processes. This article considers the quality estimation of the wheat bread that is baked under standard conditions. The sensory data are collected in real-time, and the obtained data are analysed using the efficient data analytics to predict the quality of the product. The dataset obtained consists of 300 bread samples prepared in 15 days whose vital physical, chemical, and rheological measures are sensed. The measures of the read are obtained through sensory tools and are gathered as a dataset. The obtained data are generally raw, and hence, the required features are obtained through dimensionality reduction using the Linear Discriminant Analysis (LDA). The processed data and the attributes are given as input to the classifier to obtain final estimation results. The efficient Fuzzy Weighted Relevance Vector Machine (FWRVM) classifier model is developed for this achieving this objective. The proposed quality estimation model is implemented using the MATLAB programming environment with the required setting for the FWRVM classifier. The model is trained and tested with the input dataset with data analysis steps. Some state-of-the-art classifiers are also implemented to compare the evaluated performance of the proposed model. The estimation accuracy is obtained by comparing the number of correctly detected bread classes with the wrongly classified breads. The results indicate that the proposed FWRVM-based classifier estimates the quality of the breads with 96.67% accuracy, 96.687% precision, 96.6% recall, and 96.6% F-measure within 8.96726 seconds processing time which is better than the compared Support vector machine (SVM), RVM, and Deep Neural Networks (DNN) classifiers.

1. Introduction

Provision of good quality foods is the global regulation and the food manufacturers have been directed towards having specialized laboratories with established procedures to ensure quality and safety of the food products [1]. The World Health Organization (WHO) and the World Trade Organization (WTO) have provided regulations to monitor the production and storage of the food products [2]. There are many organizations in each country to ensure the quality of food products. The main common criteria that are analysed for food quality must have sufficiently adhered to the needs of the consumer [3]. Bread is one of the highly consumed food products in most countries. The baking sector for bread is considered as an important section in food industries. Quality of bread is dependent on many parameters, the most important being the raw materials and their preparation [4]. The time for baking and expected time to consume bread also depends on the preparation of bread from raw products. However, sometimes due to consumer demands, some criteria might not be followed effectively. This leads to changes in bread quality and causes dissatisfaction in consumers [5]. Therefore, the bread-making industries are trying to implement advanced technologies to balance the trade-off between the consumer satisfactions of bread quality to the profit of the organization [6]. Also, the boosting of bread quality can avoid wastage of bread which is one of the major visions of the United Nations.

The quantity and quality of wheat gluten are vital in determining the bread quality [7]. The other major factors are the water and moisture in the dough, blending property and the additives which are inseparable in determining the

bread quality [8]. The bread makers often consider better raw materials to obtain good quality breads. But in this modern world, the availability of equivalent raw materials is always not assured. In such cases, the makers are tilted towards using the available raw materials and producing the best possible breads. In these scenarios, the estimation of the finished bread products is vital in analysing and reboosting the bread quality [9]. The bread quality is also defined by the use of a genomic selection of wheat used. Although it is an effective process, it is a time-consuming process. Hence, it is suggested to use only the rheology of the wheat ingredients. In recent years, the sensory tools are used to analyse and monitor the quality metrics. The colour and crust of bread are considered to check the quality. It is considered as one of the prominent analysis methods. However, the colour and crust of breads can vary based on various components and external factors but still provide adequate quality. However, the quality analysis of all breads is a time-consuming process and requires high manpower. Hence, automatic and efficient quality estimation models are greatly required to improve the production of bread and also the quality of such products. Based on these suggestions, this study focuses on developing data analysis methods to estimate the bread quality.

The utilization of data mining and machine learning algorithms can be efficient in determining bread quality [10]. Many studies have applied fuzzy-based algorithms to improve prediction accuracy. Fuzzy-C-Means (FCM) clustering and Adaptive Neuro-Fuzzy Inference System (ANFIS) are two of the most common and efficient fuzzy-based classifiers. Integration of these algorithms with the deep learning algorithms [11] can enhance the prediction accuracy from time-series data through deep feature learning. Similar soft clustering techniques have been utilized by the authors [12, 13] in recent studies for predicting diseases from the clinical data with higher accuracy and less complexity. Motivated by those techniques, this study has initiated the design and development of fuzzy weighted classifier for predicting the quality of the wheat breads.

This paper envisions this objective and introduced FWRVM classifier model to improve the quality estimation of the bread samples. The bread dataset collected in real-time is pre-processed, and the LDA is applied to reduce the dimensionality. Then, the classification is performed using FWRVM based on the bread features to obtain good and bad breads. Experiments are performed to evaluate the proposed FWRVM model and compare it with the state-of-the-art methods. The remainder of this article is structured as Related works in Section 2. Explanation of the proposed bread quality estimation in Section 3 is followed by the evaluation results in Section 4. Section 5 concludes this article.

2. Related Works

Bread quality analysis is one of the most important research topics in food industries. Many studies have been developed to estimate the quality of breads. Most studies employed physical methods to record bread properties and analyse their quality. Recently, Adebayo-Oyetoro et al. [14] employed a physical method to assess the breads made from wheat and fermented banana flour. Selivanov and Voronina [15] developed a physical sensory system to monitor the bread quality from the wheat flour. Eshetu et al. [16] introduced proximate composition-based bread quality analysis Ethiopia breads. These methods are highly practical and most effective. However, it is very time consuming and requires extensive labour.

Automatic analysis models have been developed using the machine learning algorithms both for bread images and bread numerical datasets. Oury et al. [17] proposed bread wheat quality estimation using the phenotypic information and partial least square (PLS) regression. Mutlu et al. [18] developed artificial neural network- (ANN-) based bread quality prediction with the help of near-infrared spectroscopy. Although less expensive and efficient, the complexity of this model increases the time consumption. Bouachra et al. [19] introduced bread wheat baking quality prediction model using an optimized GlutoPeak-Test method. This test method increased the accuracy of bread quality estimation but it requires extensive tools to perform this operation.

Guha et al. [20] developed a prediction model to predict the rheological and chemical properties of wheat dough using deep neural network (DNN) where each layer is trained greedily using restricted Boltzmann machine (RBM) networks. The authors used a new algorithm in which each layer is tuned using RBM, and the final network is fine-tuned using deep neural network (DNN) to estimate the bread dough quality. However, this approach is complex and requires extensive training and testing to produce results. Giefer et al. [21] employed Optical Sensors and Deep Learning method for detecting the quality of bread dough. By using a combination of machine learning and super-ellipsoid model fitting, an instance segmentation and parameter estimation method were developed for dough objects to estimate the quality. However, the complexity of using deep learning models has significantly reduced its use for smaller models. Junior et al. [22] proposed a multitarget bread wheat quality prediction model using near-infrared spectroscopy and machine learning algorithms. The results showed that the proposed multitarget-based machine learning algorithms provided better results. Isleroglu and Beyhan [23] proposed the baking quality prediction model using nonlinear polynomial models (PLN) and nonlinear artificial neural network (ANN) models. This approach improved the baking quality of breads, but this model has limitations in handling large samples. From the literature studies, it can be understood that physical methods are time-consuming and require extensive labour. The suggestion to include machine learning algorithms for automatic and effective quality estimation seems promising. Hence, this study has proposed and developed the FWRVM based quality estimation model.

3. Methodology

The bread quality estimation framework has been developed using the fuzzy weighted relevance vector machine. The working model of the proposed framework is given in Figure 1. which is utilized. This data will be preprocessed to filter the noisy data and stabilize the dataset structure. Then,

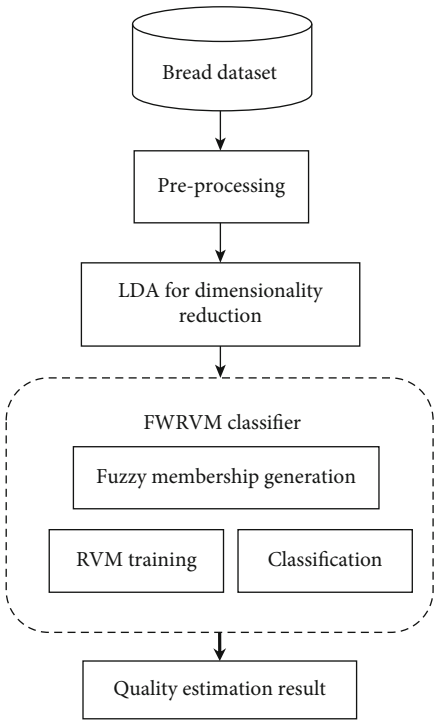


Figure 1: Proposed Bread Quality estimation framework.

the dimension is reduced by removing the irrelevant and replicated data samples using the LDA. The features of these data samples are then fed to the classifier where the weights are assigned to them and the fuzzy-based classification is performed to produce the estimation results.

3.1. Data Collection. The bread data for evaluation collected in real-time from bread-making industries for 15 days. For 15 days, twenty samples of bread per day were prepared which contains an equal number of good and bad breads. Thus, a total of 300 bread samples were prepared, and the data were collected for evaluation. The bread formation ingredients are white wheat flour (250 g), sugar (45 g), fat oil (17.5 g), yeast in wet form (8 g), salt (5 g), hydroxypropyl methylcellulose-HPMC (3.75 g), skim milk powder-SMP (5 g), and water (150-165 ml). First, the flour is mixed with the ingredients (sugar, salt, yeast, HPMC, and SMP). Then, the water is added and mixing is done for 10 minutes. Then, the fat oil is added and the fermentation is done at 24-28 degree Celsius for 120 minutes. Then, the dough is punched for 2 minutes and fermented again for 20 minutes. After fermentation, the dough is moulded into the required shape, and final proofing is done for 20 minutes at 30 degree Celsius. Finally, the baking is performed at 150 degree Celsius for 45 minutes and depanned to allow it to cool for 60 minutes at room temperature. This results in the required bread samples.

The properties of the bread samples are estimated to formulate the features through the sensor-based analysis used in the industries. The proximate analysis is performed to get the carbohydrates, protein, crude fibre, fat content, ash content, and moisture content of the breads. The sensory and statistical analyses were performed using the scoring tests and ANOVA analysis of variance. The sensory attributes such as crust colour, crumb colour, texture, taste, and aroma were estimated between 1 and 3 hours of baking to ensure the fresh breads are free of external factors. From the analysis results, a total of 9 features (attributes or variables) are used in this paper. The attributes are water content (%), mass temperature (Celsius), feed rate (g/min), malt, product moisture (%), horizontal expansion ratio, water absorption index, water solubility index (%), and bulk density (g/l). Other features, wheat flour, salt, yeast, and other factors are the raw materials. These raw materials are not considered as factors, since they are used in a fixed quantity. For instance, the wheat flour, fat oil, etc. are used in the specified quantity and are not varied during the entire process. The classes (good/bad) are also denoted in the dataset. A subset of the dataset is given in Table 1.

The full dataset contains 150 good and 150 bad bread samples. The total dataset of 300 bread samples is divided into the training and testing datasets. The training/testing ratio is set as 7 : 3, i.e., 70% training samples and 30% testing samples. Therefore, 210 samples are used for training, while 90 samples are used in testing. The number of good/bad bread samples in the training dataset is 105 each and that in the testing dataset is 45 each. This input dataset might contain noise or missing values. Hence, the preprocessing is performed as a precaution.

3.2. Preprocessing. In the preprocessing step, the noise removal and imputing missing values are vital tasks performed to improve the bread data analysis. The noisy data are formed due to the low-level errors in the data formation resulting in highly irrelevant or imperfect data. As the bread dataset is collected without such errors, the possibility of noise is very low. Hence, the noise removal is selectively utilized. The k-nearest neighbour based imputation is used to resolve the missing value problem which calculates the missing value as a related value to its nearest neighbour values. Thus, formed data is balanced and perfect for dimension reduction and classification.

3.3. Dimensionality Reduction Using LDA. The high number of features is a relatively challenging problem in pattern recognition tasks and machine learning techniques. The dimension reduction is the viable solution to attain higher classification accuracy. Dimension reduction is the representation of the high dimensional features in low dimensional space, i.e., minimizing the number of features in the dataset. The bread dataset contains nine features among which some might not contribute towards the accurate classification due to high computation and time complexity. Hence, the dimension reduction is performed to reduce the number of features and reduce the complexity in computations. For achieving this specialized objective, the LDA [24] is applied. LDA detects the linear features that have higher performance efficiency to maximize the between-class sample separation and minimize the within-class distribution.

Considering the training dataset X with N samples $\{x_1, x_2, \cdots, x_N\}$ and each of the samples x_i is represented as a

Table 1: A subset of the bread dataset.

Water content (%)	Mass temperature (Celsius)	Feed rate (g/min)	Malt	Product moisture (%)	Horizontal expansion ratio	Water absorption index	Water solubility index (%)	Bulk density	Class
7	14	21	30	40	48	54	61	72	2
7	14	24	32	37	47	55	61	70	2
7	13	21	30	39	44	52	63	68	2
7	13	20	31	39	45	55	62	70	2
8	16	21	30	38	44	54	61	70	2
2	26	27	26	26	49	65	69	76	1
3	29	30	22	29	49	61	68	76	1
6	25	29	17	33	52	63	68	75	1
4	26	26	27	33	49	59	68	74	1
1	28	29	14	34	50	64	68	77	1

column vector of length d. The training samples will possibly belong to one of the classes in $p \in \{1, 2\}$. Let C_p denotes the list of samples in class p with $n_p = |C_p|$ representing the number of samples in class p. The within-class distribution matrix in LDA is estimated as

$$D_w = \frac{\sum_p \sum_{i \in C_p} (x_i - m_p)(x_i - m_p)^T}{N}. \quad (1)$$

Similarly, the between-class distribution matrix is given by

$$D_b = \frac{\sum_p n_p (m_p - m)(m_p - m)^T}{N}. \quad (2)$$

Here, m is the mean of the dataset which is given by $m = 1/N \sum_i x_i$ and m_p is the mean of the pth class given by $m_p = 1/n_p \sum_{i \in C_p} (x_i)$.

The linear transformation $x \to W^T x$ maximizes the between-class variance in relevance to the within-class variance, where W is the $d \times ed$ matrix with ed is the desired number of dimensions. In LDA dimension reduction, it is shown that the columns of the optimal estimation of W are the comprehensive Eigen vectors on the condition that $D_b W = \lambda D_w W$. This corresponds to the largest Eigen values of ed with λ representing the Eigen value. The most common outcome of this condition is that the W also simultaneously transverses the distribution matrices $W^T D_b W$ and $W^T D_w W$. It means the LDA disconnects the relationship of the data of both the between-class and within-class.

The LDA reduced the dimension of the bread dataset by removing the low variance columns of the dataset. The dimension reduced dataset contains only 9 high informative features compared to that of the input dataset. Water content (%), mass temperature (Celsius), feed rate (g/min), and bulk density are the retained features after LDA dimensionality reduction. Table 2 shows the subset of the dataset after LDA dimensionality reduction.

Table 2: The subset of the bread dataset after dimension reduction.

Water content (%)	Mass temperature (Celsius)	Feed rate (g/min)	Bulk density	Class
7	14	21	72	2
7	14	24	70	2
7	13	21	68	2
7	13	20	70	2
8	16	21	70	2
2	26	27	76	1
3	29	30	76	1
6	25	29	75	1
4	26	26	74	1
1	28	29	77	1

3.4. Fuzzy Weighted Relevance Vector Machine Classifier. One of the most important statistical learning methods is the relevance vector machine which is based on the Bayesian learning model and effective kernel functions. Hence, it is much suitable for the quality estimation task. The major advantage of using RVM-based classification models is the adaptive solution utilization of sparser models that enhances the probabilistic prediction without requiring extensive parameter adjustment. The FWRVM is an improved model of RVM in which the fuzzy membership vectors are used for optimal selection of the weight vectors adapting the weighted features. As all the features are not equally impactful, the features from the LDA are obtained and weight values are assigned to improve the classification of FWRVM.

For the bread dataset, the FWRVM has been modelled to predict the posterior probability of each bread sample. Considering the training dataset, $X = \{x_1, x_2, \cdots, x_n\}$ with N samples belong to classes $p \in \{1, 2\}$. The statistical analysis is adopted by the FWRVM, and the comprehensive linear model is used by introducing the logistical sigmoid function $\sigma(y) = 1/(1 + e^{-l})$ to the estimated decision $y(x)$. Adopting

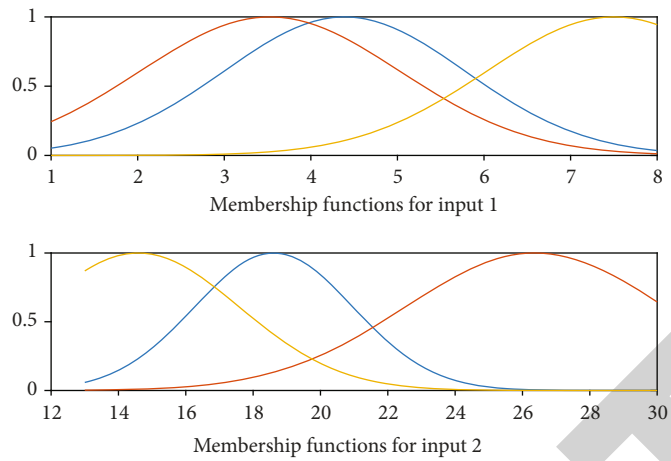


Figure 2: Fuzzy membership functions in FWRVM for two bread input.

Table 3: Confusion matrix of FWRVM.

Class	True positive	True negative	False positive	False negative
Good (1)	42	45	0	3
Bad (2)	45	42	3	0

this function and the Bernoulli distribution function for probability $P(p \mid X)$ towards the probability of the bread samples, the following equation is obtained.

$$P(p \mid X, w) = \prod_{n=1}^{N} \sigma\{y(x_n; w)^p\}[1 - \sigma\{y(x_n; w)\}]^{1-p}. \quad (3)$$

Here, $y(x; w) = \sum_{i=1}^{N} w_i K_{RBF}(x, x_i) + w_0$. $w = (w_0, w_2, \cdots, w_N)^T$ denote the adaptable parameters of FWRVM, and K_{RBF} denotes the radial basis function (RBF) in the FWRVM based classification. The RBF kernel function is adopted for the FWRVM mainly due to its efficiency over other kernel functions for classification.

Introducing the fuzzy membership vectors [25], $s = [s_1, s_2, \cdots, s_N]$ for the bread samples, the X from Eq. (3) becomes

$$P(p \mid S, w) = \prod_{n=1}^{N} \sigma\{y(s_n; w)^p\}[1 - \sigma\{y(s_n; w)\}]^{1-p}. \quad (4)$$

Here, $y(s; w) = \sum_{i=1}^{N} w_i K_{RBF}(s, s_i) + w_0$.

The fuzzy membership functions obtained for two inputs (training and testing) of the bread dataset in simulation is shown in Figure 2.

Once the fuzzy membership vector is adopted, the optimal weight vectors are needed to be found. Determining the value of w is equivalent to finding the optimal weight, and it maximizes the likelihood $P(w \mid p, S, \alpha) \propto P(p \mid S, w)P(w \mid \alpha)$. Here, $\alpha = [\alpha_0, \alpha_1, \cdots, \alpha_N]$ denotes the vector of $N + 1$ hyperparameters. As analytically determining the weights is not suitable, the closed-form functions for either marginal probability $P(w \mid \alpha)$ or weight posterior probability $P(w \mid p, S, \alpha)$ are avoided. Hence, the approximation of the weights is done by the Laplace's method.

For the present stable values of α, the best feasible weights w_{BF} are estimated using the position of the posterior approach. As $P(w \mid p, S, \alpha) \propto P(p \mid S, w)P(w \mid \alpha)$ is feasible, the maximization of weights is equivalent to this condition:

$$w_{BF} = \arg\max_{w} \log P(w \mid p, S, \alpha) = \arg\max_{w} \log \{P(p \mid S, w)P(w \mid \alpha)\}, \quad (5)$$

$$w_{BF} = (\arg_w \max)\left\{\sum_{(n=1)}^{N} [p \log d_n + (1-p)(1 - \log d_n]] - \frac{1}{2} w^T A w\right\}. \quad (6)$$

Here, $d_n = \sigma\{y(s_n; w)\}$ and $A = \text{diag}\,(\alpha_0, \alpha_1, \cdots, \alpha_N)$.

Laplace's approximation is based on the simple Gaussian approximation to the log-posterior mode of the weights w_{BF} from Eq. (6) which is differentiated to obtain the following:

$$\nabla_w \nabla_w \log P(w \mid p, S, \alpha)|_{w_{BF}} = -\left(\Phi^T B \Phi + A\right). \quad (7)$$

Here, B is the diagonal matrix given as $B = \text{diag}\,(\beta_0, \beta_1, \cdots, \beta_N)$ and $\beta_n = \sigma\{y(s_n; w)\}[1 - \sigma\{y(s_n; w)\}]$. Φ denotes the $N \times (N + 1)$ policy matrix where $\Phi_{nr} = K(s_n, s_{r-1})$ and $\Phi_{n0} = 1, n = (1, 2, \cdots, N)$ and $m = (1, 2, \cdots, N + 1)$. Using negation and inversion on the Eq. (7), the covariance matrix $\Sigma = (\Phi^T B \Phi + A)^{-1}$ can be formulated.

The hyperparameters α can be updated by an iterative re-approximation function. Initially, the randomly estimated α_i is obtained, and then, the estimation of $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ is done. Here, Σ_{ii} denotes ith diagonal component of the covariance matrix. Then, the α_i is reapproximated as

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{u_i^2}. \quad (8)$$

Here, $u = w_{BF} = \Sigma \Phi^T B p$. Assigning $\alpha_i \leftarrow \alpha_i^{\text{new}}$ and

Table 4: Summary of performance results.

Methods/metrics	Accuracy	Precision	Recall	F-measure	Chi-square statistics
SVM	0.8	0.85714	0.80	0.827	38.5714
RVM	0.87778	0.90179	0.87778	0.87592	54.6429
DNN	0.93333	0.94118	0.93333	0.93304	68.8235
Proposed FWRVM	**0.96667**	**0.96875**	**0.9677**	**0.96663**	**78.75**

Table 5: Time comparisons.

Methods/metrics	Processing time (seconds)	Training time (seconds)	Inference time (seconds)
SVM	11.299	8.2341	1.112
RVM	10.7941	7.8898	1.087
DNN	9.12983	7.1207	1.1255
Proposed FWRVM	8.96726	7.2353	1.0222

reapproximating the next α_i^{new} and γ_i continues until the convergence is obtained.

4. Results and Discussion

The performance of the FWRVM-based bread data quality estimation is evaluated in MATLAB R2016b on the collected bread dataset. The performance of FWRVM is compared with the existing algorithms, Support vector machine (SVM) [26], RVM and Deep Neural Networks (DNN) [20].

4.1. SVM. SVM is the most famous and proficient classification algorithm. It is used in many applications in different fields. However, the disadvantages of SVM limit its use in prediction models. SVM needs coefficients for error whose selection reduces the accuracy. SVM is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes bread data samples into either good or bad bread. In two-dimensional space, this hyperplane is a line dividing a plane in two parts wherein each class lay in either side.

4.2. RVM. RVM was also developed for the same function as that of SVM and follows the same structure. However, it is based on Bayesian algorithm and employs the Bayesian inference functions to perform probabilistic predictions. This makes the RVM much suitable for the prediction based classifications.

4.3. DNN. Deep learning algorithms are the latest and improved classifiers. DNN is based on the ANN and is advanced in the deep architecture. This deep architecture improves the learning rate of DNN and increases accuracy.

4.4. Performance Comparisons. The performance of the proposed FWRVM is evaluated and compared with the existing models in terms of accuracy, precision, recall, F-measure, time, and Chi-square statistics. Table 3 shows the confusion matrix of FWRVM for testing data with True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

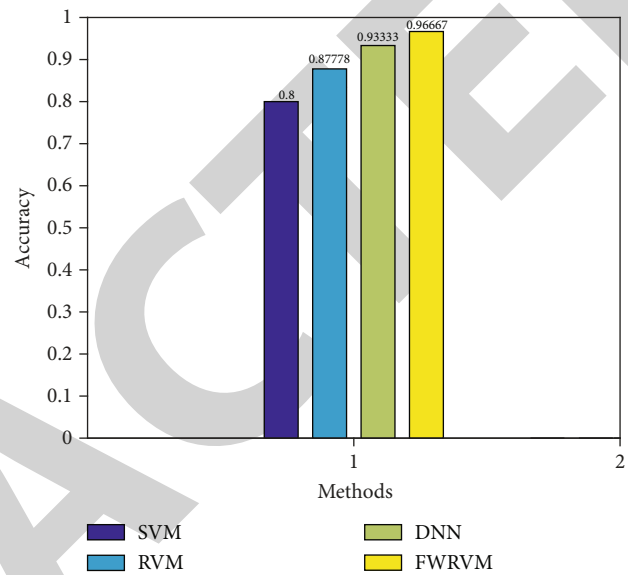


Figure 3: Accuracy comparison.

From the 90 testing samples (45 good and 45 bad), the FWRVM classifier estimated 87 samples correctly. The predicted results showed that the dataset has 48 good breads and 42 bad breads.

The performance of the proposed FWRVM is evaluated in terms of accuracy, precision, recall, F-measure, processing time, and Chi-square statistics. These parameters are calculated using the following equations.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}, \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{11}$$

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

$$\text{Chi} - \text{square} = \sum \frac{(\text{Observed value} - \text{Expected value})^2}{\text{Expected value}}. \tag{13}$$

The percentage difference between the proposed and

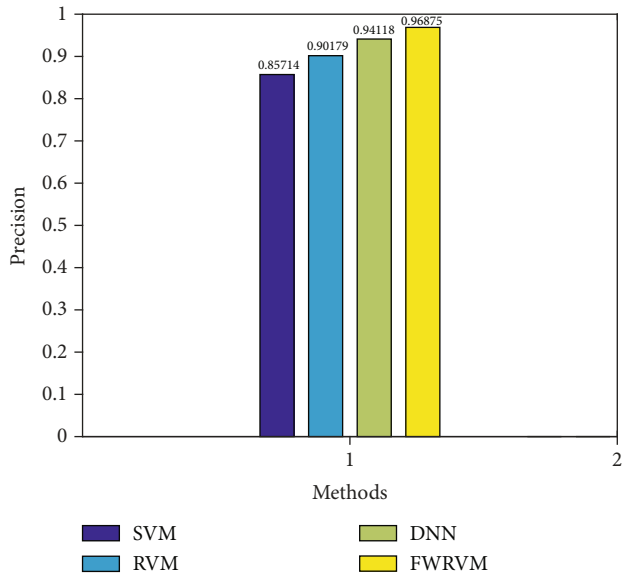


Figure 4: Precision comparison.

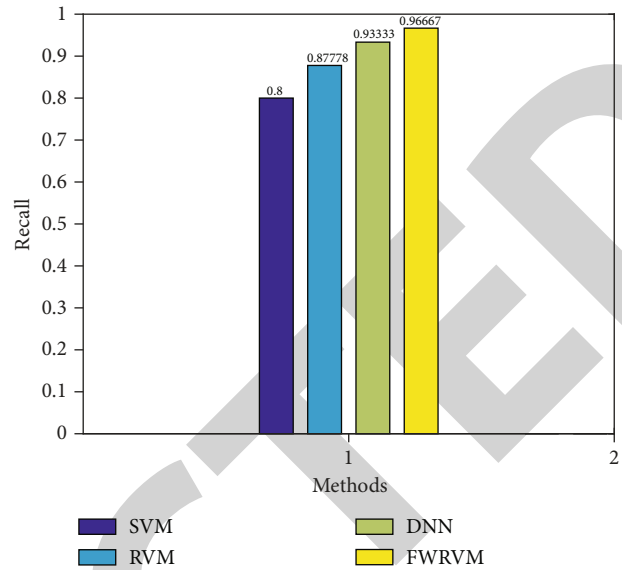


Figure 5: Recall comparison.

existing methods for all parameters including the time values can be computed using the following equations.

$$\%\text{Difference} = \begin{cases} (E-P) \text{ or } (P-E) \text{ if values are represented as\%} \\ (E-P)\times 100 \text{ or } (P-E)\times 100 \text{ If values are represented as ratio} \\ \dfrac{(E-P)}{E}\times 100 \quad \text{or} \quad \dfrac{(P-E)}{E}\times 100 \text{ otherwise} \end{cases}. \tag{14}$$

Here, E and P represent the existing and proposed methods. The formulas are chosen such that the smaller value is always subtracted from the larger value irrespective of the method. When the method having high value is considered better, the difference is calculated by subtracting existing method from the proposed method and vice versa.

Table 4 shows the performance results of FWRVM and also the compared existing methods.

From the above table, it can be seen that the proposed FWRVM has higher performance than the other compared methods in terms of all parameters. FWRVM has higher values of accuracy, precision, recall, F-measure, and Chi-square statistics. Table 5 shows the processing time and training/inference time comparisons.

From Table 5, it is clear that the proposed model has higher performance than the exiting model including the deep learning DNN. FWRVM has taken 2.33 seconds lesser processing time, 0.9988 seconds lesser training time, and 0.09 seconds lesser inference time than the SVM based classifier for the same bread data classification. The comparison graphs are also plotted to illustrate the performance results.

Figure 3 displays the accuracy evaluation of the suggested FWRVM model against the prevailing models. From the comparison, it is apparent that the suggested FWRVM has higher accuracy than the existing models. FWRVM has an accuracy of 0.96667, and it is 3.3%, 8.9%, and 16.6% greater than DNN, RVM, and SVM models, respectively.

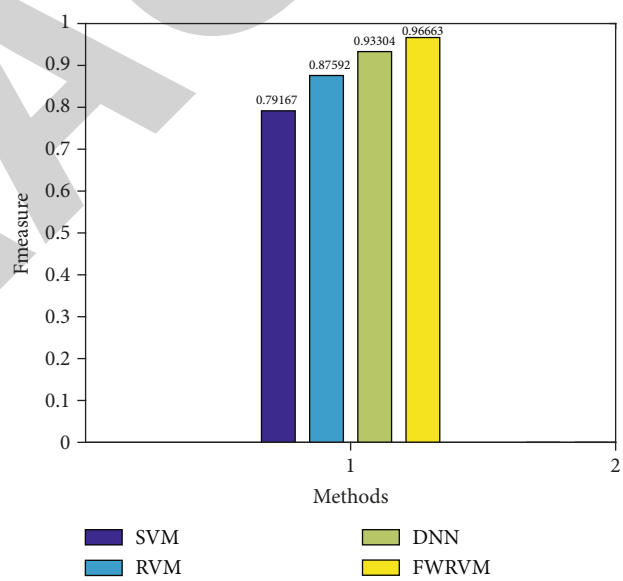


Figure 6: F-measure comparison.

Figure 4 shows the precision evaluation of the suggested FWRVM against the existing models. FWRVM has high precision of 0.96875 which is 2.7%, 6.7%, and 11.1% larger than DNN, RVM and SVM models, respectively.

Figure 5 shows the recall evaluation of the suggested FWRVM against the prevailing models. The results show that the FWRVM has high recall than other models. FWRVM has a recall of 0.96667, and it is 3.3%, 8.9%, and 16.6% greater than DNN, RVM, and SVM models, respectively.

Figure 6 illustrates the F-measure evaluation of the suggested FWRVM based bread quality estimation against the existing models. The plot results show that the FWRVM has high F-measure of 0.96663 which is 3.3%, 9.1%, and

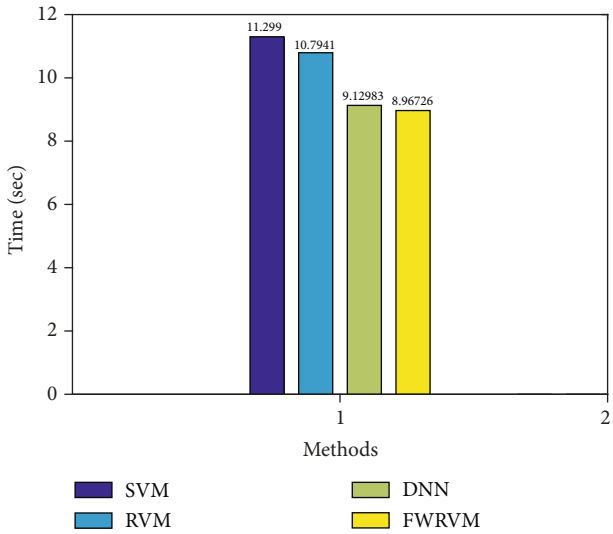


Figure 7: Time comparison.

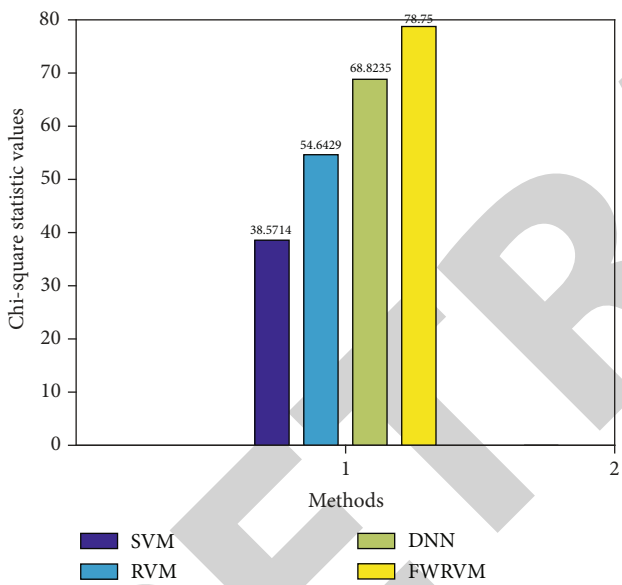


Figure 8: Chi-square statistics.

17.5% higher than DNN, RVM, and SVM models, respectively.

Figure 7 illustrates the processing time evaluation for the suggested FWRVM model against the prevailing models. The main evaluation of this study is to decrease the time complexity. From the figure, it is proved that the time complexity of the suggested FWRVM is reduced greatly. FWRVM consumes 8.96726 seconds which is 0.16 seconds, 1.83 seconds, and 2.33 seconds less than DNN, RVM, and SVM models, respectively.

Figure 8 illustrates the Chi-square statistics of the suggested FWRVM-based bread quality estimation against the existing models. Chi-square statistics illustrates the relationship between the observed results and the expected outcomes. The plot results show that the FWRVM has high Chi-square statistics value of 78.75 which is very higher than the existing models. The main reason for this enhancement is the use of fuzzy memberships to the RVM and weighted parameters. These improvements significantly enhance the performance of the FWRVM for bread quality estimation.

5. Conclusions

An efficient bread quality estimation model has been developed in this paper using the FWRVM classifier. This study focused on developing an automatic quality estimation model for the bread samples using efficient machine learning algorithms. The integration of fuzzy logic to the weighted RVM classifier has improved the overall accuracy. Also, the use of LDA for dimension reduction has increased quality estimation outcomes. The evaluation on a dataset of 300 bread samples illustrated that the proposed FWRVM based model achieved an accuracy of 0.96667 and less time of 8.96726 seconds. This concludes that the proposed bread quality estimation model has better efficiency and much suitable for bread-making industries. In future, the possibility of including more rheological properties to estimate the bread quality will be examined. Also, the possibility of using the proposed FWRVM for other food datasets will be investigated.

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] K. G. Grunert, "Food quality and safety: consumer perception and demand," *European Review of Agricultural Economics*, vol. 32, no. 3, pp. 369–391, 2005.

[2] R. P. Carpenter, D. H. Lyon, and T. A. Hasdell, *Guidelines for Sensory Analysis in Food Product Development and Quality Control*, Springer Science & Business Media, 2012.

[3] K. D. Dolan and D. K. Mishra, "Parameter estimation in food science," *Annual Review of Food Science and Technology*, vol. 4, no. 1, pp. 401–422, 2013.

[4] M. G. Gänzle, J. Loponen, and M. Gobbetti, "Proteolysis in sourdough fermentations: mechanisms and potential for improved bread quality," *Trends in Food Science & Technology*, vol. 19, no. 10, pp. 513–521, 2008.

[5] P. D. Ribotta, G. T. Pérez, M. C. Añón, and A. E. León, "Optimization of additive combination for improved soy–wheat bread quality," *Food and Bioprocess Technology*, vol. 3, no. 3, pp. 395–405, 2010.

[6] A. Loutfi, S. Coradeschi, G. K. Mani, P. Shankar, and J. B. B. Rayappan, "Electronic noses for food quality: a review," *Journal of Food Engineering*, vol. 144, pp. 103–111, 2015.

[7] L. Y. Lin, H. M. Liu, Y. W. Yu, S. D. Lin, and J. L. Mau, "Quality and antioxidant property of buckwheat enhanced wheat bread," *Food Chemistry*, vol. 112, no. 4, pp. 987–991, 2009.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9846843, 1 page
https://doi.org/10.1155/2023/9846843

Retraction

Retracted: Classification of Lactate Level Using Resting-State EEG Measurements

Applied Bionics and Biomechanics

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

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. A. Shaban, O. N. Ucan, and A. D. Duru, "Classification of Lactate Level Using Resting-State EEG Measurements," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6662074, 8 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 6662074, 8 pages
https://doi.org/10.1155/2021/6662074

Research Article

Classification of Lactate Level Using Resting-State EEG Measurements

Saad Abdulazeez Shaban,[1,2] **Osman Nuri Ucan**,[3] **and Adil Deniz Duru**[4]

[1]*Computer Science Department, College of Education for Pure Sciences, Diyala University, Diyala 32001, Iraq*
[2]*Electrical and Computer Engineering, School of Engineering and Natural Sciences, Altınbaş University, Istanbul 34217, Turkey*
[3]*Engineering Faculty, Electrical and Electronics Department, Istanbul University, 34850 Avcilar, Istanbul, Turkey*
[4]*Neuroscience and Psychology Research in Sports Lab, Faculty of Sport Science, Marmara University, 34668 Istanbul, Turkey*

Correspondence should be addressed to Saad Abdulazeez Shaban; saad.shaban@uodiyala.edu.iq

Received 9 November 2020; Revised 1 January 2021; Accepted 15 January 2021; Published 8 February 2021

Academic Editor: Mohammed Yahya Alzahrani

The electroencephalography (EEG) signals have been used widely for studying the brain neural information dynamics and behaviors along with the developing impact of using the machine and deep learning techniques. This work proposes a system based on the fast Fourier transform (FFT) as a feature extraction method for the classification of human brain resting-state electroencephalography (EEG) recorded signals. In the proposed system, the FFT method is applied on the resting-state EEG recordings and the corresponding band powers were calculated. The extracted relative power features are supplied to the classification methods (classifiers) as an input for the classification purpose as a measure of human tiredness through predicting lactate enzyme level, high or low. To validate the suggested method, we used an EEG dataset which has been recorded from a group of elite-level athletes consisting of two classes: not tired, the EEG signals were recorded during the resting-state task before performing acute exercise and tired, the EEG signals were recorded in the resting-state after performing an acute exercise. The performance of three different classifiers was evaluated with two performance measures, accuracy and precision values. The accuracy was achieved above 98% by the K-nearest neighbor (KNN) classifier. The findings of this study indicated that the feature extraction scheme has the ability to classify the analyzed EEG signals accurately and predict the level of lactate enzyme high or low. Many studying fields, like the Internet of Things (IoT) and the brain computer interface (BCI), can utilize the findings of the proposed system in many crucial decision-making applications.

1. Introduction

Electroencephalography is the brain neural signals which reflect the brain's electrical potentials and are mainly used for studying brain neural information dynamics processing and employed to diagnose brain disturbances [1]. Normally, those signals are time series signals [2] recorded by means of a specialized skull helmet which contains multiple electrodes distributed and attached to a specific position on the scalp, either in a wet or dry manner [3]. The data acquired from the recording of such signals are in a very large amount, and that is why they should be analyzed by specialized methods, rather than conventional visual ways [4]. Among those methods, data classification using machine learning techniques can play a vital role in analyzing and investigating EEG signals and exploitation results in different applications like diagnosing human mental diseases [5], predicting emotional states, decision discovery for patient rehabilitation devices, or assistive technology for interactive input devices like gaming controllers and wheelchairs drivers [6]. Many of the previous studies were investigating different brain activity by means of magnetic resonance imaging (MRI) that measures anatomical images noninvasively. Resting-state functional MRI activity is shown to differ between before and after performing aerobic exercise [7]. A recent study investigates the impact of a single acute exercise session on the brain's functional connectivity and showed an obvious increase in the functional connectivity of sensorimotor brain

networks that could be clearly assessed using functional MRI [8]. Even though the MR imaging shows high spatial resolution, but because of the blood oxygenated level dependency attributes (BOLD), it shows a limited temporal resolution of the measured signals. Using the electroencephalography signals as a measure of electrophysiological brain activity produces enhancement in temporal resolution to a range of milliseconds [9]. The use of EEG as a technique of analyzing in the field of psychology has been applied for a long time in many studies, but it was not that common in the field of exercises and sports until recent times. A study investigated the impact of severe physical short-term exercise and long-term workout training on the EEG resting-state alpha frequency (iAPF) of the individual shows that frequency has increased after performing intense exercise [10], while another study shows an increment in the power of the frontal area observed in the EEG signals after performing acute cycling exercises [11]. The high-intense running was found to have an effect on both EEG and the mood of the exerciser [12]. Brain activation during aerobic exercises was found to be increased, i.e., the EEG beta frequency band power is increased and the alpha frequency band is decreased during performing moderate-intensity short-time cycle ergometer exercise, then returning to the power baseline after finishing [13].

One of the emerging aspects in this field is predicting the lactate dehydrogenase enzyme levels, whether being high or low, in the human blood by classifying the collected resting-state EEG data [14] from a subject along with measuring lactate levels in the blood. The idea behind that is to study whether lactate levels in the human blood could be predicted of being increased or decreased affected by performing an acute exercise, as it was reported that the blood lactate level would reach its peak after maximal treadmill running exercise was made for a short period [15], meanwhile collecting EEG data before and after exercise and mark them as class 1 (before exercise or lactate-level-low) and class 2 (after exercise or lactate-level-high). This study is aimed at examining this idea by suggesting a classification system that should discriminate two states of lactate level using EEG signals having different frequency bands recorded from a group of healthy athlete subjects of the elite level, before and after performing a single bout of acute exercise. As EEG signals have different features that could be extracted using a variety of methods [16], we should dominate the best discriminant feature that gives us the best classification score in terms of accuracy. Among different features, the band power features which represent the energy (power) of EEG signals were chosen to represent a discriminant criterion and are computed by means of power spectral density of each EEG signal frequency band for a given channel. Frequency band power is regarded as a gold standard feature to be applied in applications like brain computer interface (BCI) by many studies [6, 17]. Band power features are calculated to evaluate the brain's activity changes over a given time window (typically of a few seconds) encountered by performing an acute exercise session. Then, the extracted feature data is arranged in a vector, manipulated and modified using preprocessing techniques to clean data from artifacts and enhance the model performance. These features are analyzed along with the relation to blood lactate levels before and after performing exercises. To the best of the author knowledge, until now, no one study in the literature is related to the assessment of classification performance using power spectral density- (PSD-) based feature extraction machine learning classifiers when applied to the fatigue problem after acute exercise. Compared with several studies, experimental results clarify that the suggested system could enhance the detection rate. Figure 1 shows the schematics of the proposed system.

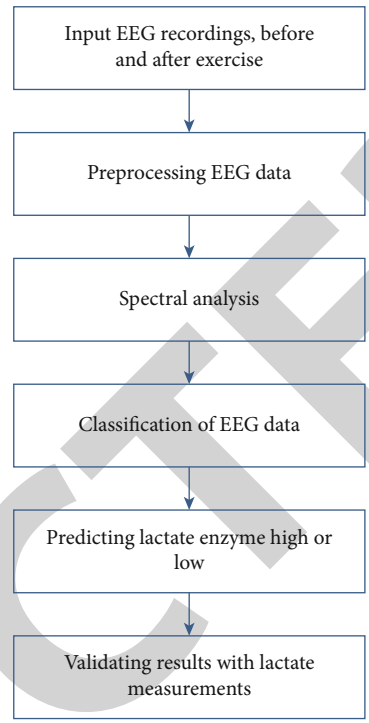


Figure 1: The proposed classification model.

2. Materials and Methods

2.1. Operational Tasks. In this study, the employed dataset that includes the resting-state EEG signals from [9], has been used. The proposed system in our study consists of two main parts: one involves feeding lactate enzyme level test measurements, and the other involves input the EEG signal recordings and manipulating them. EEG data was recorded from a volunteers' group of elite level athletes (no.of subjects = 10), and all are representing members of official karate team. These subjects had performed the blood lactate level test before doing the exercise, and the results were allocated to represent low-level lactate (not tired) class. Initial lactate measurements were found to be at the baseline value of around 2 millimoles/litre. In the first step, subjects were sitting in a calm fashion with eyes closing (EC) condition and asked to stay as-calm-as possible and thinking about nothing for 3 minutes. Meanwhile, the EEG signals are being collected from subjects with a sampling rate of 1000 Hz using BrainAmp ExG amplifier from 16 channels (Fz, Fp1, Fp2, F3, F4, Cz, C3, C4, T7, T8, Pz, P3, P4, Oz, O1, and O2) with dry

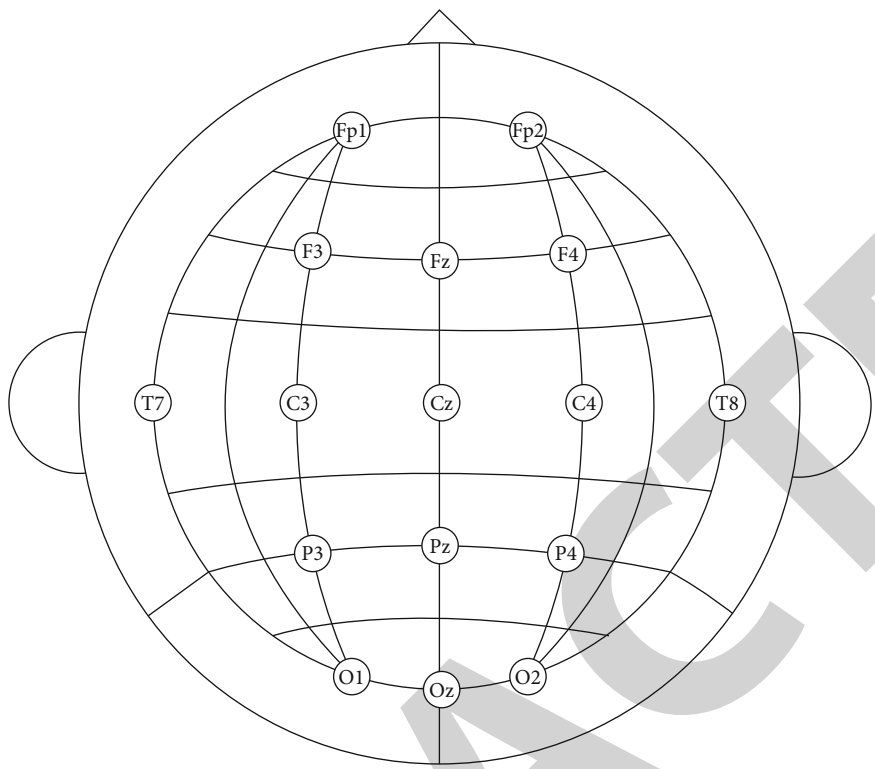


Figure 2: Cap 16-electrodes layout.

electrode caps. Figure 2 shows the distribution of 16 electrodes over the brain scalp.

The next step in this phase requires each subject to separately perform an acute exercise of a short-time shuttle run with 20 meters for each shuttle. This running protocol is an incrementally progressive test that is used to predict personal physical sensations experienced during exercises like maximum oxygen consumption, increased heart rate, muscle fatigue, and increased sweating. It consists of 20 m running that requires increasing running pace while time decreases as levels proceed with a beep stimulus between levels.

While performing the exercise, the performance is monitored using rated perceived exertion (RPE) scale. RPE is a scale for measuring physical activity intensity by asking the activist about how he feels his body is working without interrupting the exercise. The exercise ended when each subject reports a 16 RPE level according to the Borg rating of perceived exertion [18].

The next experimental phase starts, after a short resting period of one minute, by measuring blood lactate levels for each subject, and it was found at a high level of around 16 millimoles/litre. Then, the lactate test is repeated 4 times with 2 minutes between them, and each test result found to be at the same high levels with no drop to baseline within the EEG data measurement phase. Then, the EEG data measuring was repeated with EC condition for 3 minutes and assigned to be a high-level lactate (tired) class. Both datasets of measured EEG signal, pre, and postexercise contain artifacts generated by some muscular movement, eye blinking, and heartbeats that can contaminate the quality of EEG data [19], and that is why the data has been cleaned from noises by removing epochs that have an absolute amplitude greater than $100\,\mu$V by using band-pass filtering technique. Figure 3 shows epochs of 1 second EEG signal of one subject recorded from 16 channels before and after performing the exercise.

2.2. Feature Extraction. The EEG signals are nonstationary time-series signals, and once the raw version of EEG data was recorded then passed the preprocessing step, the next step is to get related attributes through the feature extraction process. To get the better distinguishing feature from EEG signals, we have applied the fast Fourier transform (FFT) method to provide frequency representation of the signals, which helps to measure the power spectrum of data for each frequency band, delta (0-4 Hz), theta (4-8), alpha (8-13), beta (13-30), and gamma (30-45), within a time window or epoch. For the frequency spectral analysis, the nonstationarity can be tolerated and the EEG signal assumed stationary for the epoch period. Fast Fourier transform (FFT) is a signal processing method which is used to transform the signal from its time domain to the equivalent frequency domain representation by dividing the signal function into a continuous frequency band known as frequency spectrum [20]. If $F(k)$ is the fast Fourier transform of a function $f(z)$, then it is defined by using equation (1) as follows:

$$F(k) = \sum_{z\,\text{even}} f(z) w_Z^{kz} + \sum_{z\,\text{odd}} f(z) w_Z^{kz}, \tag{1}$$

for $k = 0, 1, \cdots, Z-1$, where $F(k)$ is the Fourier coefficient of $f(z)$, which is assumed to have a complex value, z even and z odd correspond to the EEG samples of $f(z)$, which were

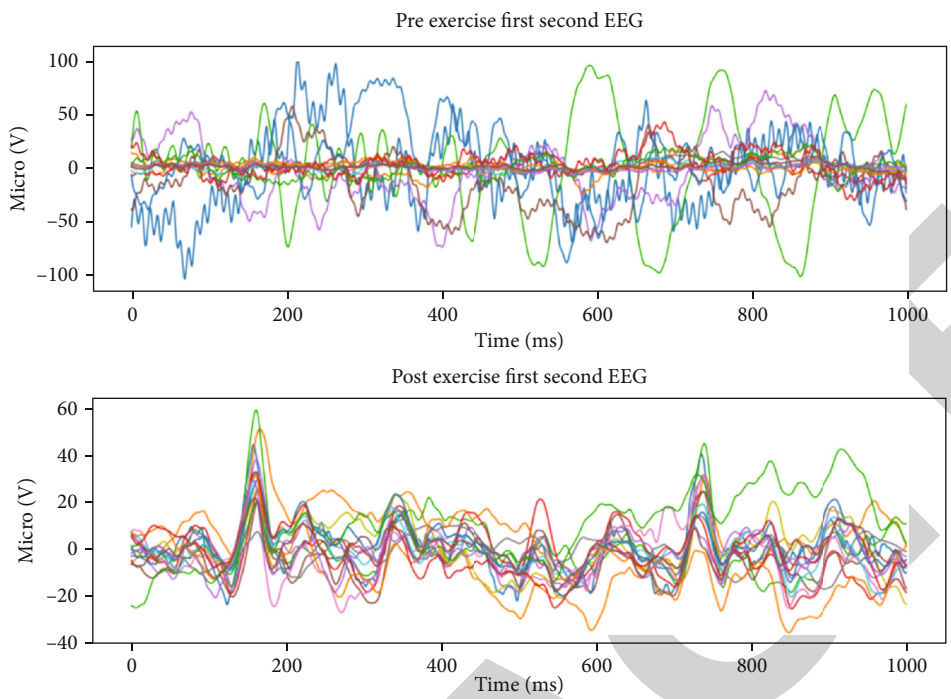


Figure 3: Pre and post exercise 16-channel EEG data.

even and odd numbered, respectively. $w = \exp\ (-2zj/Z)$, where z is equal to 3.14, and j is the imaginary part.

2.3. Classification. The process of EEG data formatting and performing frequency band power calculation is done by a self-developed MATLAB routine, and the resulting data, pre, and postexercise are then combined and formed a single bulk dataset of size 7759 rows and 80 columns and fed to a set of classification algorithms (classifiers) with 80 features and two class labels, "1" is representing lactate-level-low before exercise, and "2" is representing lactate-level-high after exercise. Among those applied classifiers, the KNN, decision tree (DT), and logistic regression (LR) have reported scoring better than others, like the linear discriminant analysis (LDA) classifier and support vector machine (SVM) classifier; thus, only the highest-scoring classifiers have been listed in Results.

2.3.1. Decision Tree. A decision tree is a machine learning model in which each nonleaf node denotes a test on a feature, each branch node represents an outcome of the test, and each terminal node holds a class label. The root node is the topmost node. Assume an X tuple with an unknown class label, the feature values of X are tested against the tree with path traced along from the root node to a leaf node, which represents the class prediction for the given tuple [21]. Each decision tree employs an attribute selection method that specifies a procedure for choosing the best attribute that discriminates the tuple depending on the class. This procedure uses attribute selection measure as a metric function to evaluate split for feature selection such as the information gain and the Gini index. Some attribute selection measures impose the tree to become binary like the Gini index; others like information gain are not. Gini index measures the impurity of data from a set of training tuples, as in equation (2).

$$\mathrm{Gini}(\mathrm{data}) = 1 - \sum_{i=1}^{k} Pi^2, \tag{2}$$

where Pi represents the probability that a tuple in data belongs to a specific class, say Ci. Information gain is an attribute selection measure that tries to find the attribute which has the highest information gain that minimizes the required information to classify tuple and is defined by equation (3).

$$\mathrm{Inf}(\mathrm{data}) = -\sum_{i=1}^{k} Pi \log_2(Pi), \tag{3}$$

where Pi is the probability that a tuple in data belongs to a specific class Ci.

2.3.2. K-Nearest Neighbor. The K-nearest neighbor (KNN) classifiers learn by comparing a given test vector with similar training vectors (SRs). The training vectors are described by number of n attributes. When given an unknown vector, a K-nearest neighbor classifier seeks the pattern field for the k training vectors (nearest neighbors) that are closest to the unknown vector. The "closeness" is defined in terms of a distance, such as Manhattan distance, which defines distance d between $x1$ and $x2$ vectors, as in equation (4):

$$d(j, l) = |xj1 - xl1| + |xj2 - xl2| + |xj3 - xl3|. \tag{4}$$

Table 1: Classification score of applied models.

Model	KNN	Decision tree	Logistic regression
Accuracy	98.4%	98.0%	70.0%

Then, the probability is used as a measure to assign the input x to the most probable class (nearest one), as in equation (5):

$$P(y = j \mid X = x) = \frac{1}{K} - \sum_{i \epsilon B} L\left(y^{(i)} = j\right), \tag{5}$$

where B represents the set of K neighbors of the training vector which are nearest to input x, and $L(x)$ represents indicator that acts as a function which sets to 1 if the input x is true and set to 0 if not.

2.3.3. Logistic Regression. Logistic regression is a popular model for solving classification problems, and the term "Logistic" comes from the underlying Logit function used in this model for classification, the natural logarithm of odds ratio [22]. Logistic regression estimates as probability the impacts of independent variables on the outcome variables. Simple logistic model is shown in equation (5).

$$\text{Logit}(y) = \text{naturallog}(\text{odds}) = \ln\left(\frac{y}{1-y}\right), \tag{6}$$

where the logit(y) is representing the probabilities from 0 to 1.

The technology used for classification was the Classification Learner applications available in the MATLAB R2018a software.

To validate entire input data, we used the technique of K-fold cross-validation, which splits data into K folds (parts). Among these K folds, K-1 folds are used to train proposed model and the remaining fold is used for testing purpose. The procedure is replicated for K times until all subsets are validated; then, all the results are averaged for final accuracy prediction [23].

3. Results

The extraction of EEG band power feature yields a significant enhancement in the classifier's accuracy scores, especially for KNN, decision tree and logistic regression classifiers. The main finding of our study is proving the ability to clearly predict human blood lactate levels using resting-state EEG signals when applying suitable techniques, power spectral density in our case. The classification score versus applied method results are listed in the Table 1. To the best of the authors' knowledge, there is no study in the literature related to the classification performance measure using FFT and machine learning classifiers, investigating the fatigue problem after acute exercise.

Another measure is to calculate the specificity and sensitivity of the classifier. Sensitivity is also referred to as the rate of true positive recognition (i.e., the proportion of the first-class belonging tuples that are correctly identified); on the other hand, specificity represents the rate of true negative recognition (i.e., the proportion of the second-class belonging tuples that are correctly identified) [24]. These two measures are defined as follows in equations (6) and (7), respectively:

Table 2: Sensitivity and specificity of the classifiers.

Model	KNN	Decision tree	Logistic regression
Sensitivity	98.8%	97.2%	76.5%
Specificity	97.0%	94.2%	64.0%

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FN}}, \tag{8}$$

where TP represents the positive tuples of data that were correctly classified by the model, whereas FP represents the positive tuples that were falsely classified. On the other hand, TN represents the negative tuples of data that were correctly classified by the model. In contrast, FN represents the negative tuples that were falsely classified by the model. Table 2 shows those measures for each of the applied classification models. Thus, we note that the KNN and decision tree classifiers have a high accuracy along with high sensitivity and specificity which indicates their ability to correctly classify both the positive and negative tuples, which are in contrast to the logistic regression classifier that showed a moderate sensitivity and specificity scores meaning that it can recognize positive and negative tuples at a lower rate.

Furthermore, classifiers show the following precision values, which represent percentage of instances labelled as positive and are actually such, for both classes, lactate-level-low and lactate-level-high denoting pre-exercise (not tired) and post-exercise (tired) tiredness recognition, respectively, for different classifiers as in Table 3a, b, and c as follows:

4. Discussion

In the present study, we had investigated the ability to predict whether the lactate level is low or high in the human body using EEG signals of subjects after performing an acute exercise. The subjects were athletes of the elite level from the national team of Turkey. The achieved results indicate that predicting blood lactate levels, high or low, using electroencephalogram brain data can be done accurately in terms of classification scores when implemented for healthy athlete who endures a single bout of acute exercises. The discrimination ability is driven by the changes encountered in the band power values of EEG signal bands after doing an exercise [25]. This hypothesis was proven by variations that occurred with alpha and beta frequency band power that investigated after implementing a maximal effort exercise and shows an increment in beta absolute power in a group of electrodes [26]. In our study, the best scoring classification model was KNN with 98.4% accuracy with a ratio of training data and testing data 80 : 20, which was found to be a high scoring

Table 3: Precision values for different classifiers.

(a) Precision values for decision tree

	Total	3867	3893	7760
Actual class	Lactate-level-low	3790	76	3866
	Lactate-level-high	77	3817	3894
		Lactate-level-low	Lactate-level-high	Total
		Predicted class		

(b) Precision values for KNN

	Total	3870	3890	7760
Actual class	Lactate-level-low	3804	62	3866
	Lactate-level-high	66	3828	3894
		Lactate-level-low	Lactate-level-high	Total
		Predicted class		

(c) Precision values for logistic regression

	Total	4359	3401	7760
Actual class	Lactate-level-low	2959	907	3866
	Lactate-level-high	1400	2494	3894
		Lactate-level-low	Lactate-level-high	Total
		Predicted class		

Table 4: Score results of various objective studies.

Study	Year	Electrode no.	Study objective	Applied algorithms	Accuracy %
[29]	2016	14	Classify facial movement and expressions by a noninvasive EEG signals	KNN	98.0
[30]	2018	1	Analyze EEG signals from different cognitive states to control BCI devices	LDA, SVM, KNN	95.0, 100.0, 100.0
[33]	2014	128	EEG signal classification recorded while doing a complex cognitive task	ANN, SVM, MLP, KNN, Naïve Bayes	88.89, 98.75 98.57, 98.21 83.57
[34]	2016	12	Monitoring EEG signal while driving to detect the driver fatigue state	SVM	91.28
[35]	2019	19	EEG signal features extraction and classification during fatigue exercise	SVM	90.0
Present study	2020	16	Classify resting-state EEG to predict lactate, high or low	KNN, DT, LR	98.4, 98.0, 70.0

model in other studies, especially when applied against applications with a low-dimensional [27] feature vector of data [28]. The KNN was found to perform effectively to extract and classify feature vector for different facial movements and expressions measured by noninvasive EEG devices. The accuracy was around 98% driven by implementing segmentation to the complete signal waveform [29]. Even though classification could be applied using other EEG features like the average spectral centroid, average standard deviation, or average energy entropy, but still the power spectral density offers the highest accuracy with all classifiers and was found to score 100% with KNN when analyzing EEG signals from different human cognitive states employed to control brain computer interface (BCI) devices [30]. In contrast to our work which investigated the effect of a single bout of acute exercise, the effect of increasing running exercise intensities on spontaneous EEG was investigated by a study, which found that the overall spectrum power in EEG significantly increased in all frequency bands with increasing intensities of exercise, lactate level has increased, and even after a period of 15- to 30-minute recovery, lactate enzyme level has decreased but still significantly higher than baseline and discernible [31]. The subsequent decrease in spectrum power was seen in a subset of frequency bands in some cortical regions suggesting a decrease in cortical activation after exercise intensities, as a hypothesis of brainstem inhibitory mechanism, may occur [32]. Table 4 shows the results of various objective studies comparable to our work.

5. Conclusions

The proposed work represents the use of band power spectral density along with machine learning techniques for classification and analysis of EEG signals recorded during resting-state tasks. The band's power feature of EEG signals was extracted using FFT for all of the 16 channels of each subject's EEG recording. Three different classification models (KNN, decision tree, and logistic regression) were applied, and their performance was reported. The classification accuracy of KNN and decision tree found to be above 98%. This makes the study the unique and pioneer one to discuss and prove the ability to use resting-state EEG signals as an accurate measure for the human tiredness level through predicting lactate enzyme level high or low. The band power was found to be a very useful EEG feature to classify these signals after performing acute exercise sessions. Hence, the proposed feature extraction and classification system have the significance to be applied on real-time EEG applications like BCI, IoT, military, or medical applications to predict the individual physical tiredness state that can assist in many crucial situations. As a suggested study expansion, the classifiers could be applied to EEG data collected for each subject individually with applying the same previous procedures, and the results could be compared in both cases. This may be implemented in future work possibly with applying more algorithms and preprocessing techniques for the purpose of achieving higher classification accuracy scores.

Data Availability

The EEG dataset which was used in the present study is available from the author, Adil Deniz Duru, through a reasonable request.

Conflicts of Interest

No conflict of interest is declared by the authors regarding this paper publication.

References

[1] S. Sanei and J. A. Chambers, *EEG Signal Processing, 312*, Wiley, England, 2007.

[2] T. H. H. Aldhyani and M. R. Joshi, "Integration of time series models with soft clustering to enhance network traffic forecasting," in *2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 212–214, Kolkata, India, September 2016.

[3] L.-W. Ko, C. Yang, P.-L. Wu et al., "Development of a smart helmet for strategical BCI applications," *Sensors*, vol. 19, no. 8, article 1867, 2019.

[4] R. Vigario, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja, "Independent component approach to the analysis of EEG and MEG recordings," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 5, pp. 589–593, 2000.

[5] T. H. Aldhyani, A. S. Alshebami, and M. Y. Alzahrani, "Soft computing model to predict chronic diseases," *Journal of Information Science and Engineering*, vol. 36, no. 2, pp. 365–376, 2020.

[6] F. Lotte, L. Bougrain, A. Cichocki et al., "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, article 031005, 2018.

[7] T. B. Weng, G. L. Pierce, W. G. Darling, D. Falk, V. A. Magnotta, and M. W. Voss, "The acute effects of aerobic exercise on the functional connectivity of human brain networks," *Brain Plasticity*, vol. 2, no. 2, pp. 171–190, 2017.

[8] A. S. Rajab, D. E. Crane, L. E. Middleton, A. D. Robertson, M. Hampson, and B. J. MacIntosh, "A single session of exercise increases connectivity in sensorimotor-related brain networks: a resting-state fMRI study in young healthy adults," *Frontiers in Human Neuroscience*, vol. 8, p. 625, 2014.

[9] A. D. Duru, T. H. Balcıoğlu, C. E. Ö. Çakır, and D. G. Duru, "Acute changes in electrophysiological brain dynamics in elite karate players," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, pp. 565–579, 2019.

[10] B. Gutmann, A. Mierau, T. Hülsdünker et al., "Effects of Physical Exercise on Individual Resting State EEG Alpha Peak Frequency," *Neural Plasticity*, vol. 2015, Article ID 717312, 6 pages, 2015.

[11] H. Enders, F. Cortese, C. Maurer, J. Baltich, A. B. Protzner, and B. M. Nigg, "Changes in cortical activity measured with EEG during a high-intensity cycling exercise," *Journal of Neurophysiology*, vol. 115, no. 1, pp. 379–388, 2016.

[12] S. Schneider, C. D. Askew, J. Diehl et al., "EEG activity and mood in health orientated runners after different exercise intensities," *Physiology and Behavior*, vol. 96, no. 4-5, pp. 709–716, 2009.

[13] K. A. Kubitz and A. A. Mott, "EEG power spectral densities during and after cycle ergometer exercise," *Research Quarterly for Exercise and Sport*, vol. 67, no. 1, pp. 91–96, 1996.

[14] A. M. A. Mohamed, O. N. Uçan, O. Bayat, and A. D. Duru, "Classification of resting-state status based on sample entropy and power spectrum of electroencephalography (EEG)," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8853238, 10 pages, 2020.

[15] N. Fujitsuka, T. Yamamoto, T. Ohkuwa, M. Saito, and M. Miyamura, "Peak blood lactate after short periods of maximal treadmill running," *European Journal of Applied Physiology*, vol. 48, no. 3, pp. 289–296, 1982.

[16] A. S. Al-Fahoum and A. A. Al-Fraihat, "Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains," *ISRN Neuroscience*, vol. 2014, Article ID 730218, 7 pages, 2014.

[17] D. Trad, T. Al-Ani, and M. Jemni, "Motor imagery signal classification for BCI system using empirical mode décomposition and bandpower feature extraction," *Brain*, vol. 7, no. 2, pp. 5–16, 2016.

[18] G. A. V. Borg, "Psychophysical bases of perceived exertion," *Medicine and Science in Sports and Exercise*, vol. 14, no. 5, pp. 377–381, 1982.

[19] S. Romeroa, M. A. Mañanasa, and M. J. Barbanoj, "A comparative study of automatic techniques for ocular artifact reduction in spontaneous EEG signals based on clinical target variables: a simulation case," *Computers in Biology and Medicine*, vol. 38, no. 3, pp. 348–360, 2008.

[20] M. R. Canal, "Comparison of wavelet and short time Fourier transform methods in the analysis of EMG signals," *Journal of Medical Systems*, vol. 34, no. 1, pp. 91–94, 2010.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9762945, 1 page
https://doi.org/10.1155/2023/9762945

Retraction

Retracted: Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network

Applied Bionics and Biomechanics

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

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] F. E. K. Al-Khuzaie, O. Bayat, and A. D. Duru, "Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network," *Applied Bionics and Biomechanics*, vol. 2021, Article ID 6690539, 9 pages, 2021.

Hindawi
Applied Bionics and Biomechanics
Volume 2021, Article ID 6690539, 9 pages
https://doi.org/10.1155/2021/6690539

Research Article

Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network

Fanar E. K. Al-Khuzaie,[1] **Oguz Bayat**,[1] **and Adil D. Duru**[2]

[1]*Graduate School of Science and Engineering, Altinbas University, Istanbul, Turkey*
[2]*Department of Physical Education and Sports Teaching, University of Marmara, Istanbul, Turkey*

Correspondence should be addressed to Fanar E. K. Al-Khuzaie; fanar@itnet.uobabylon.edu.iq

Received 26 November 2020; Revised 14 January 2021; Accepted 22 January 2021; Published 2 February 2021

Academic Editor: Mohammed Yahya Alzahrani

There are many kinds of brain abnormalities that cause changes in different parts of the brain. Alzheimer's disease is a chronic condition that degenerates the cells of the brain leading to memory asthenia. Cognitive mental troubles such as forgetfulness and confusion are one of the most important features of Alzheimer's patients. In the literature, several image processing techniques, as well as machine learning strategies, were introduced for the diagnosis of the disease. This study is aimed at recognizing the presence of Alzheimer's disease based on the magnetic resonance imaging of the brain. We adopted a deep learning methodology for the discrimination between Alzheimer's patients and healthy patients from 2D anatomical slices collected using magnetic resonance imaging. Most of the previous researches were based on the implementation of a 3D convolutional neural network, whereas we incorporated the usage of 2D slices as input to the convolutional neural network. The data set of this research was obtained from the OASIS website. We trained the convolutional neural network structure using the 2D slices to exhibit the deep network weightings that we named as the Alzheimer Network (AlzNet). The accuracy of our enhanced network was 99.30%. This work investigated the effects of many parameters on AlzNet, such as the number of layers, number of filters, and dropout rate. The results were interesting after using many performance metrics for evaluating the proposed AlzNet.

1. Introduction

The normal brain of humans consists of mainly three regions, namely, white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) [1]. The white matter is called as such because of its white appearance. It contributes about sixty percent to the total brain volume. The gray matter is responsible of the whole processing of the neural signals. It consists of dendrites and the neuron nuclei. It contributes almost about forty percent of the total brain volume. Cerebrospinal fluid is a colorless fluid that provides protection from mechanical shocks and also emits some important hormones to make the communication possible among the white matter, gray matter, and spinal cord of the central nervous system [2]. It is known that the family of artificial intelligence (AI) includes many algorithms and methods which could be used in different aspects of our life, for example, genetic algorithm [3–5] and neural networks [6]. Machine learning (ML) is a field of artificial intelligence that usually employs factual procedures to allow PCs to "learn" by utilizing information from saved data sets. At a very basic level, deep learning (DL) is a machine learning subset [7]. Deep learning can be defined as a neural network which uses a huge number of parameters and layers. There are many fundamental network architectures [8] like (i) convolutional neural networks (CNNs) which are basically a standard neural network that has been extended across space using shared weights [9]. A convolutional neural network (CNN) is designed to recognize images by having convolutions inside, which see the edges of a recognized object on the image [10]. (ii) Recurrent neural networks (RNNs) are a denomination of artificial neural networks where connections among nodes lay out a directed graph along the temporal

sequence. Unlike feedforward neural networks, RNNs have the ability to use their internal state for processing the sequences of inputs. RNN is designed to recognize sequences, for example, a speech signal or a text [9]. (iii) Recursive neural networks are more like a hierarchical network where there is really no time aspect to the input sequence, but the input has to be processed hierarchically in a tree fashion [8, 10]. Generally, different external stimuli match to different brain activities, and the different brain activities display different functional brain images [11]. For that, image classification plays a significant role in identifying different activities of the brain. Recently, many methods of deep learning were proposed to perform classification of image for different brain activities [12, 13]. To identify different activities of the brain including emotions, motor, social, relational and language activities, and working memory, Koyamada et al. [12] applied a feedforward deep neural network from images of functional magnetic resonance imaging (fMRI) to implement this mission. The feedforward deep neural network involved a softmax layer and multiple hidden layers. Similarly, these hidden layers were used to get high-level latent features, while the softmax layer was applied to calculate the probability of every subjects in a class. In addition, dropout, minibatch stochastic gradient descent, [14], and principal sensitivity analysis [15] were combined into the feedforward deep neural network to improve the performance of the final classification. Recently, to classify different sensorimotor tasks including auditory attention, visual stimulus, right-hand clenching, and left-hand clenching, Jang et al. [13] used fully connected feedforward deep neural networks and multiple hidden layers. In addition to the above classifications, the methods of deep learning classification of magnetic resonance imaging (MRI) images have been used also by other fields of classification, like stroke diagnosis [16], age prediction [17], classification of attention deficit hyperactivity disorder (ADHD) [18], discrimination of cerebellar ataxia types [19], and emotional response prediction [20]. Due to the science engineering field, it was doable to create systems of computer-aided diagnosis (CAD) that play a critical role in assisting the researchers and physicians when they interpret the medical imaging. Recently, the use of the machine learning approach, especially DL techniques in systems of CAD to diagnose and classify the healthy control normal (CN) people, Alzheimer's disease (AD), and mild cognitive impairment (MCI) patients, has exponentially increased [21, 22]. The Alzheimer's disease automatic diagnosis, especially in its early stage, plays a significant role in human health. Since Alzheimer's disease is a neurodegenerative illness, it has a long period of incubation. Thus, it is necessary to analyze the AD symptoms at different stages. Currently, a lot of researchers have proposed using the classification of images to perform diagnosis of AD. Moreover, many DL methods have been proposed to implement severity classification of different Alzheimer's disease patients by using MRI images [22, 23]. As known in image processing and analyzing, the better the image quality, then the better the results gained. However, the image quality depends on acquisition of the image, so, when the image acquisition is better, then the image quality is higher. Magnetic resonance imaging (MRI) not only keeps the features of noninvasive and good soft tissue contrast, but in addition does not expose humans to high ionizing radiation. Since MRI can provide a lot of invaluable information about structures of tissue, such as localization, size, and shape, it is attracting more of attention for computer-aided diagnosis and clinical routine [24, 25]. MRI can be divided into functional and structural imaging. Functional imaging contains tasking state functional MRI (ts-fMRI), resting state functional MRI (rs-fMRI), etc., structural imaging contains T1-weighted MRI (T1w), diffusion tensor imaging (DTI), and T2-weighted MRI (T2w) [26]. Medical data systems are diagnostic, and analytical systems are applied to help medical centers and physicians in disease treatment, and they are critical to improve treatment and diagnosis. Computer scientists have been interested in this domain given the vital role of medical data in the lives of humans. Physicians may refer to the classification of medical data, including medical analyses and symptoms of critical diseases, for making the decisions. A data set of disease contains symptoms of patients as attributes besides the number of instances of these symptoms. Health care may use the considerable medical data accessible. In the analyses of medical centers, data mining could be used to provide sufficient sources on illnesses for their prevention and timely detection and to avoid the expensive costs incurred by medical tests [27]. Representation of features plays a significant role in medical image analysis and processing. Deep learning has two obvious advantages in the representation of features:

(i) It can be applied to automatically discover features from a given data set for every specific application. Usually, methods of traditional feature extraction are based on some prior knowledge for extracting features in a certain application. So, these approaches are semiautomatic learning methods

(ii) It can discover new features that are appropriate to specific applications, which have never been discovered by researchers previously. Traditional methods of feature extraction are often restricted by some a priori knowledge, which can only extract some features which are associated with a certain application [28, 29]

Medical imaging is the mechanism and process of establishing visual representations of the interior of the body for medical intervention and clinical analysis [30]. Machine learning tools and medical image processing can help neurologists in estimating whether a subject is developing Alzheimer's disease [31]. Alzheimer's disease is a chronic neurodegenerative disease causing tissue loss throughout the brain, and the death of nerve cells usually starts slowly and worsens over time [32]. Alzheimer's disease is expected to affect more and more people by the year 2050. The cost of caring for patients of AD is also expected to rise [33]. Presently, AD is the sixth reason that leads to death in the United States [34]. For this reason, individual computer-aided systems are necessary for accurate and early diagnosis of this disease [33]. There are many approaches for accurate and automatic classification of brain MRI, and one of them is our work. The next part of this article is the related works, then we will talk about our methodology, the results, the discussion, and, at the end, our references.

2. Related Works

Researchers have been applying machine learning techniques to build classifiers by using clinical measures and imaging data for AD diagnosis. These studies have identified the important structural differences in the regions such as the entorhinal cortex and hippocampus entorhinal cortex between the brain with AD and healthy brain. Different imaging methods like the functional and structural magnetic resonance imaging (fMRI and sMRI, respectively), single photon emission computed tomography (SPECT), position emission tomography (PET), and diffusion tensor imaging (DTI) scans which can perceive the changes causing AD due to the degeneration of cells of the brain. In recent years, deep learning models, especially convolutional neural networks, have demonstrated outstanding performance for medical image analysis. Payan and Giovanni [35] produced and tested a pattern classification system which combines convolutional neural network and sparse autoencoders. Ehsan et al. [36] adapted a 3D-CNN model for diagnosis of AD. The 3D-CNN is built upon the 3D convolutional autoencoder, which is pretrained to catch anatomical shape variations in scans of structural brain MRI. Sergey et al. [37] proposed two different kinds of 3D convolutional network architectures to classify the brain MRI which are the amendments of residual and plain convolutional neural networks. Applied convolutional neural networks can tackle the two problems stated before. These networks can propagate local features into the metarepresentation of an object for classification or image recognition. In deep learning for image classification, modern advancements like residual network architectures and batch normalization mechanism alleviate the issues of having small data sets of training, while providing a frame for automatic feature generation. As a result, these models can be used to 3D MRI images in the absence of intermediate handcrafted feature extraction. Karim et al. [38] adapted three tasks of binary classification which are considered for separating the normal control (NC) subject from mild cognitive impairment (MCI) patients and Alzheimer's disease (AD). Two fusion methods on a fully connected (FC) layer and on the single-projection CNN output offer better achievement by about 91% accuracy. The outcomes are competitive with the SOA which utilizes a heavier algorithmic chain. Fan and Manhua [39] proposed a classification technique built on multiple clusters of dense convolutional neural networks (DenseNets) to pick up the various local features for images of the MR brain, which are collected for classification of AD. The total brain image is partitioned into different local parts and from each region, a number of 3D patches are extracted. By using the$K$-means clustering method for grouping the patches from each region into different clusters, the DenseNet had been constructed to pick up the patch features for each cluster, and the features learned from the characteristic clusters of each part are grouped for classification. The classification outputs from different local parts are combined to foster the final image classification. This method can progressively learn the features of MRI from the local patches to the global image level for the task of classification. For preprocessing images of MRI, there are no segmentation and rigid registration required. Shaik and Ram [40] provided an approach to extract the gray matter from the human brain and make the classification by using the CNN. To enhance the voxels, a Gaussian filter is applied, and to remove the irrelevant tissues, the skull stripping algorithm is used. After that, by applying a hybrid, enhanced, and independent component analysis, those voxels are segmented. The input to the CNN was segmented gray matter. Clinical valuation was performed using the provided approach and 90.47 accuracy was achieved. Hamed and Kaabouch [41] proposed a method that yielded good classification accuracy. The convolutional neural network with modified architecture was used to get the high quality features from the brain MRI to classify people into healthy, early mild cognitive impairment (EMCI), or late mild cognitive impairment (LMCI) groups. The results showed the classification between control normal (CN) and LMCI groups in the sagittal view with 94.54 accuracy.

3. Materials and Methods

Inside a CNN, a filter series, with an equivalent size to a small image patch, automatically searches the entire image to find images of similar spatial features. These filters can be learned and updated independently; thus, a collection of them can detect crucial information of a specific task and data set [42]. There are standard steps of CNN. The first step is named "convolution"; this step is responsible for finding the features and applying filters. It is a filter kernel that picks up its weights by convolving the input data tensor with such kernel. There are several variables that effect the convolutional operation output such as strides and number of filters. The distance in pixels between two pixels is the stride, while the number of filters states the output feature map number [43]. The operation of convolution is just a mathematical operation, which should be treated equally with other operations such as multiplication or addition and should not be discussed particularly in the literature of machine learning. But, it has still been discussed here for completeness. Convolution is a mathematical operation on two functions (e.g., f and g) and produces a third function h; this is an integral that expresses the amount of overlap of one function (f) as it is shifted over the other function (g) [44]. Formally, it is described as

$$h(t) = \int_{-\infty}^{\infty} f(T)g(t-T)dT, \tag{1}$$

And denoted as $h = f * g$.

A typically convolutional neural network works with two-dimensional convolution operation that could be summarized in Figure 1. As displayed in Figure 1, the input matrix is Figure 1(a), and Figure 1(b) is usually called a kernel matrix. So convolution is applied to these matrices, then the result is displayed as in Figure 1(c). The process of convolution can be considered as an element-wise product followed by a sum, like what is shown in the example of Figure 1. When the left upper matrix which is 3×3 convoluted with the kernel, then the result is 29. After that, the target 3×3 matrix slides one column to the right, then is convoluted with the kernel and gets the result 22. The sliding and recording of the results have been continued as a matrix. Every target matrix is 3×3, because the kernel is 3×3; thus, the whole 5

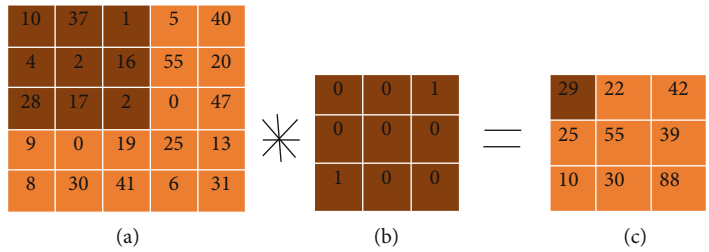


Figure 1: A simple illustration of a two-dimension convolution operation: (a) input matrix, (b) kernel matrix, and (c) output matrix after convolution.

$\times 5$ matrix is shrunk into a 3×3 matrix when every 3×3 matrix is convoluted to one digit. (Because of $5 - (3 - 1) = 3$, the first 3 means the kernel matrix size.) One should realize that the convolution process is a locally shifted invariant, which means that for many different combinations of how the nine numbers in the upper matrix 3×3 are placed, the convoluted result will be 29. This invariant property plays a crucial role in vision problem because the result of recognition should not be changed due to shift or rotation of features in an ideal case. This crucial property is applied to be solved elegantly by [45], but CNN brought the performance up to a new level.

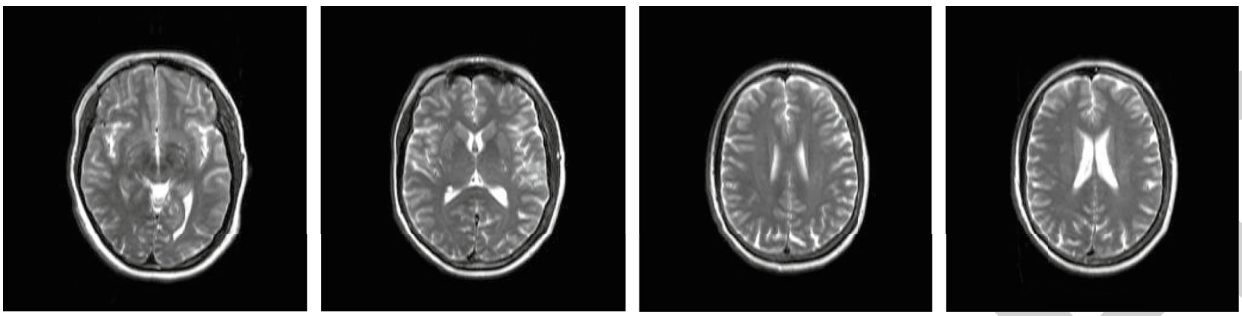
With each convolution layer, there is an activation function; the activation is an operation which converts the input from a linear data tensor to a nonlinear data tensor. In deep learning, many activation functions are popular such as rectified linear units (ReLU), sigmoid, and tanh [46]. Recently, the rectified linear unit (ReLU) has been used more than the other nonlinear functions, because it does not activate all the neurons at the same time [24]. The second step is named "Max Pooling"; this step is responsible for downsizing the image and keeping the important features. Pooling is the operation of downsampling which can be performed globally or locally. The function of global pooling returns for every 2D feature map a scalar value. The function of local pooling downsamples local image parts by a factor [43]. The third step named "flattening" converts to one dimension array (vector). The fourth step is named "full connection"; this step is responsible of building all needed connections. The fully connected layer (FC) is typically followed by an activation layer. FC is the layer where the receptive domain is a whole channel of the former layer [43, 46]. The last step is named "classifier"; it represents the classification stage to decide if the image is normal or abnormal [47]. The use of the dropout technique is so common in convolutional neural networks. Dropout was introduced in [14, 48]. This mechanism soon got influential, not only because it has good performance but also because of its simplicity of implementation. The idea is very easy: while training, randomly drop out some of the units. More formally: for each training case, every hidden unit is randomly omitted with a probability of p from the network. As suggested in [14], dropout can be seen as an efficient method to perform model averaging across a great number of different neural networks, where overfitting can be avoided with less cost of computation because of the actual performance which it introduces. Dropout became very popular upon when it was first introduced; many works have attempted to understand its technique in different perspectives, including [49]. It has also been used to train other models, such as SVM [50].

The CNN architecture which was used in this study is composed of five convolutional layers which take an input image (the brain's MRI slice) with a size of $200 * 200$. Figure 2 shows some slices of the brain's MRI; those were we used in our research. All five convolutional layers were followed by a max-pooling layer. The 64 filters with a kernel size of $9 * 9$ were considered for the first convolutional layer, and the max-pooling layer kernel size was set on $2 * 2$. The 64 filters with a kernel size of $7 * 7$ were considered for the second convolutional layer, and the max-pooling layer kernel size also was set on $2 * 2$. The 64 filters with a kernel size of $5 * 5$ were considered for the third convolutional layer, and the max-pooling layer kernel size was set on $2 * 2$. The 32 filters with a kernel size of $5 * 5$ were considered for the fourth convolutional layer, and the max-pooling layer kernel size was set on $2 * 2$.The 32 filters with a kernel size of $3 * 3$ were considered for the fifth convolutional layer, and the max-pooling layer kernel size was set on $2 * 2$. It is worthwhile to mention that the ReLU (rectified linear unit) function was used as the activation functions in all convolutional layers. The ReLU function is used commonly in models of DL; basically, if the function receives a negative value as input, it returns 0, and if the function receives a positive value, then the same positive value will return.

The function of ReLU is understood as $f(a) = \max(0, a)$.

Figure 3 demonstrates the block diagram of the proposed system (AlzNet). After the convolution layers and the flattening layer, there is a dense unit 121, and here, we used a ReLU as an activation function, then we used a dropout (0.2) to prevent overfitting, then there is a dense unit and sigmoid as an activation function; at the last stage, there is a binary classifier for displaying the results.

Table 1 demonstrates the number of MRI slices. There are samples for men and women such as a left-handed man (L.-handed male), left-handed woman (L.-handed female), right-handed man (R.-handed male), and a right-handed woman (R.-handed female); all brain MRIs were in the axial view manner. Keras provides a perfect tool to show a model's summary; Table 2 demonstrates that summary. This displays the number of trainable parameters and the output shape for each layer. Before starting to fit the model, this is a sanity check. So the total params = 414,419, the trainable params = 414,419, and the nontrainable params = 0.



Left-handed woman | Left-handed man | Right-handed woman | Right-handed man

Figure 2: Some brain MRI slices of Alzheimer's patients.

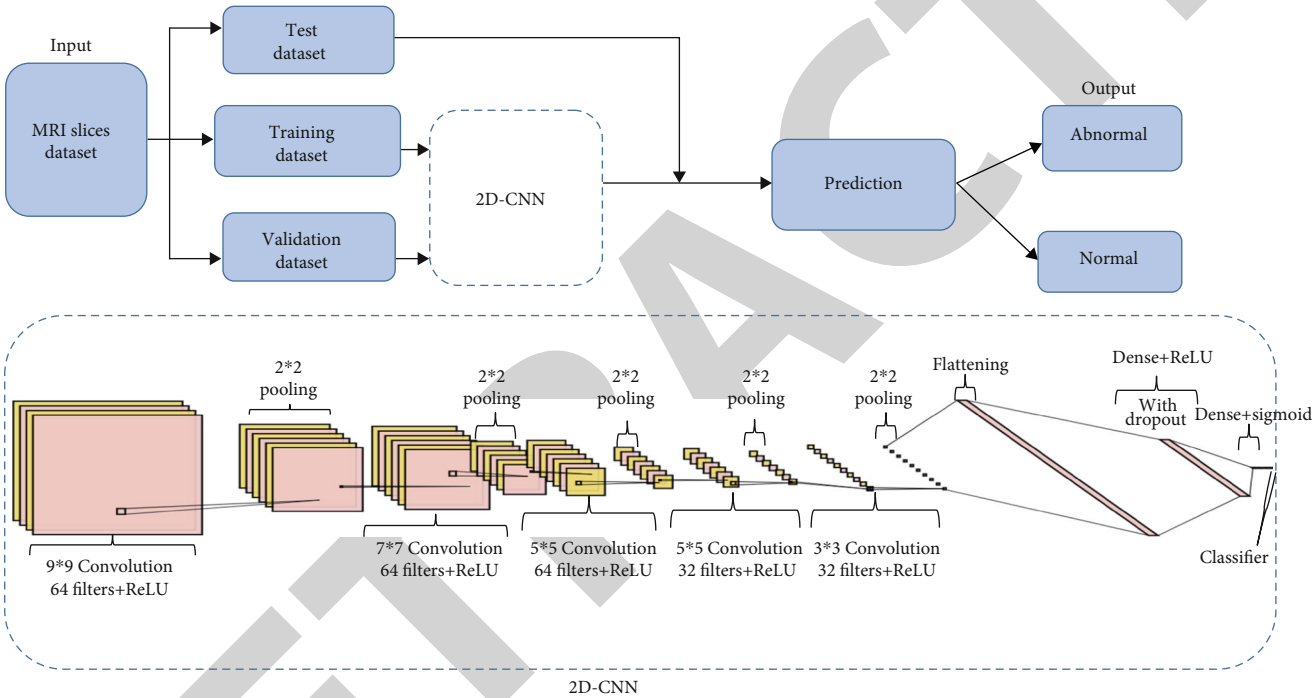


Figure 3: Block diagram of proposed system (AlzNet). There are five convolutional layers; after each convolutional layer there is a max-pooling layer; the activation function in every convolutional layer was ReLU. After the convolution layers and flattening layer, there is a dense unit 121 with a ReLU as an activation function, with a dropout to prevent overfitting, then there is a dense unit and sigmoid as an activation function; at the last stage, there is a binary classifier.

Table 1: Number of MRI slices.

Subjects	Number of subjects	Number of MRI slices
R.-handed male (AD patient)	60	3050
L.-handed male (AD patient)	40	2600
L.-handed female (AD patient)	30	2150
R.-handed female (AD patient)	40	2600
Female (NC)	25	2000
Male (NC)	45	2800

4. Results

Python language 3.6 and Keras have been used for programming this work. Keras is a high-level library; it is an open source machine learning library that is written in Python. Keras is used for numerical computation purposes; it is used to perform the computations more easily and efficiently in practice. The training data set was 75% and the validation data set was 25%. There were many practical experiments that had been done in this research for trying to find the best parameters of this convolutional neural network. So we try to find the best number of dense units for hidden layers depending on the result of accuracy, whereas the other researches used different numbers at each one. Another parameter we tested many times is the rate of dropout to find the fit rate

Table 2: Summary of the proposed model.

Layer (type)	Output shape	Param #
Conv2d_1 (Conv2D)	(None, 192, 192, 64)	15,616
Max_pooling2d_1 (MaxPooling2)	(None, 96, 96, 64)	0
Conv2d_2 (Conv2D)	(None, 90, 90, 64)	200,768
Max_pooling2d_2 (MaxPooling2)	(None, 45, 45, 64)	0
Conv2d_3 (Conv2D)	(None, 41, 41, 64)	102,464
Max_pooling2d_3 (MaxPooling2)	(None, 20, 20, 64)	0
Conv2d_4 (Conv2D)	(None, 16, 16, 32)	51,232
Max_pooling2d_4 (MaxPooling2)	(None, 8, 8, 32)	0
Conv2d_5 (Conv2D)	(None, 6, 6, 32)	9248
Max_pooling2d_5 (MaxPooling2)	(None, 3, 3, 32)	0
Flatten_1 (flatten)	(None, 288)	0
Dense_1 (dense)	(None, 121)	34,969
Dropout_1 (dropout)	(None, 121)	0
Dense_2 (dense)	(None, 1)	122

Table 3: Accuracy of AlzNet depending on dropout rate and dense unit.

		Dropout rate				
		0.5	0.4	0.3	0.2	0.1
Dense unit	130	94.90	94.21	94.78	94.02	94.12
	129	95.18	95.72	95.61	95.31	95.48
	128	94.85	94.56	94.71	94.65	94.58
	127	95.09	95.12	95.01	95.23	95.31
	126	96.61	96.72	96.40	95.97	96.01
	125	96.15	96.05	96.11	96.44	96.21
	124	96.09	96.12	96.02	96.30	96.05
	123	95.91	95.72	95.82	95.01	95.11
	122	96.85	96.56	96.90	95.77	95.83
	121	97.32	97.16	97.30	97.88	97.06
	120	95.43	95.66	94.94	94.88	94.90

for our convolutional neural network by observing the results of accuracy. That is figured out in Table 3.

So it is obvious from Table 3 that the highest accuracy value was when we utilized 0.2 for the dropout rate and 121 for the dense unit. In fact, the range of the dropout rate which we tested was (from 0.1 to 0.5) increasing by 0.1, when the number of dense units was 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, and 130. Figure 4 shows the accuracy value depending on the relationship between the dense unit number and the dropout rate.

There are several metrics for measuring the performance of binary classification [51], such as recall, precision, specificity, and Accuracy. Precision is very helpful because we want to be confident of our forecast, since it tells us how many of the values expected as positive are actually positive [50], as follows:

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}. \quad (2)$$

Recall (sensitivity) is another very valuable measure that helps one, for instance, to know the proportion of the number of values accurately labeled as positive on the overall values which are actually positive, as follows:

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}. \quad (3)$$

Using the F1 score is a safe way to get a complete impression of recall and precision. The F1 score provides us the harmonic mean of recall and precision [52, 53], as follows:

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

Accuracy is the proportion of accurate predictions (both true negative and true positive) among the entire number of cases examined [52], as follows:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True positive} + \text{True negative} + \text{False positive} + \text{False negative}}. \quad (5)$$

In this work, the measure metrics have been applied on the training data set and test data set (see Table 4).

5. Discussion

For our current work, we develop an efficient deep convolutional neural network based on a classifier and demonstrate very good performance by using the OASIS data set [54]. The OASIS-3 data set has been saved in the XNAT central repository [55]. In our work, a total of 15,200 MRI axial slices were used. The data was used to include the MRI scans of about 170 AD patients and 70 NC. They are all from different subjects that make the test of recognition performance more reliable. The age of each patient is between the range of 65-90 years old, both male and female in this work. At the first stage of data preprocessing, we obtained 2D slices from each MRI image. Then, the every last 20 dark slices with each time course was discarded because they included no functional information. Preprocessed images are augmented by rotating the slices to see whether or not the model can recognize the images; this increased the samples size and made a good training of the CNN model. The scans are T1-weighted whereas those of [40] were T2-weighted. Whereas [35, 41] used three convolution layers, our proposed system (CNN model) has 5 convolution layers; each convolution layer has ReLU as an activation function. Each convolution layer is followed by max pooling layers. Our proposed approach performs binary classification to fit the model in a batch size of 64 in 150 epochs. Table 2 summarizes the total architecture of the proposed system. When [40, 41] used Adam optimization, we trained the model by Adadelta optimization with a dropout rate of 0.2 for the dropout layer which had been utilized to prevent the overfitting like in [40], but [41] used a 0.5 dropout rate. The number of dense units was 121 when

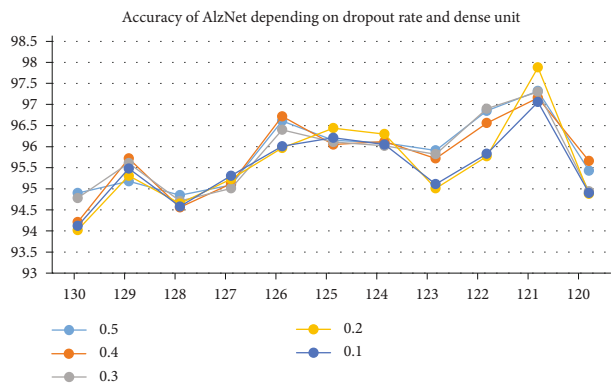


Figure 4: Accuracy of AlzNet depending on dropout rate and dense unit.

Table 4: Measure metrics of training data set and test data set.

Dataset	Precision	Recall	F1 score	Accuracy
Training	97.06	97.99	97.52	97.88
Test	98.92	99.53	99.22	99.30

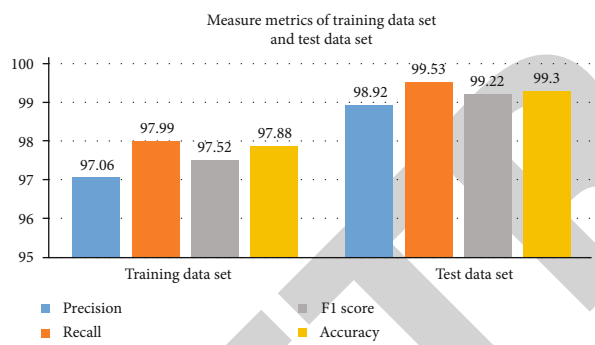


Figure 5: Measure metrics of training data set and test data set.

the number of dense units of [37] was 128; actually, we made many experiments to decide which is the best number of dense units we should take, and Table 3 shows that. In this work, we tried to put different values of the neural network parameters by trial and error, by relying on the accuracy value, and comparison with previous researches. During binary classification, we trained the classifier for AD and CN images, and the model resulted in 97.88% training accuracy and 99.30 test accuracy. It is required to mention that our proposed framework had been trained, and the prediction was made with utmost accuracy. Figure 5 demonstrates that. The accuracy of the proposed system has been compared with different models discussed in literature reviews as shown in Table 5.

It is observed that the proposed model achieves remarkable performance. The last thing we have to say is that neural networks have plenty of parameters, and any change in one of them will make the value of results different, and also, there is a big important reason for making a variation of results—it is the data set and its type.

Table 5: Literature reviews.

Methods	Year	Accuracy
Sergey et al. [37]	2017	80.00
Fan and Manhua [39]	2018	89.5
Shaik and Ram [40]	2019	90.47
Ehsan et al. [36]	2016	97.6
Karim et al. [38]	2017	91.41
Payan and Giovanni [35]	2015	95.39
Hamed and Kaabouch [41]	2019	94.54
Proposed method	2020	99.30

6. Conclusions

In order to diagnose Alzheimer's disease, deep neural networks, especially CNNs, can provide meaningful information. A CNN-based method for extracting discriminatory features from structural MRI was proposed in this paper, with the goal of classifying Alzheimer's disease and healthy subjects using 2D MRI slices. For potential AD individuals, the suggested approach can lead to many advantages and can also lead to an early diagnosis of AD. The experimental results of the OASIS database for 240 subjects demonstrated that our proposed method of extraction and classification of features provided high accuracy for AD and CN. The best results have been obtained for the classification between the CN and the AD axial view of the MRI. The proposed method yielded a classification accuracy of 99.30 percent. The above results indicate higher reliability, recall, precision, and F1 score of our proposed method for the diagnosis of AD and the classification between CN and AD.

Data Availability

Our data set was from the OASIS database; the website is https://www.oasis-brains.org. The OASIS-3 data set has been saved in the XNAT central repository; the website is https://central.xnat.org.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

Data were provided in MRI by OASIS-3. The principal investigators are T. Benzinger, D. Marcus, and J. Morris. The study was funded by NIH (P50AG00561, P30NS09857781, P01AG026276, P01AG003991, R01AG043434, UL1TR000448, and R01EB009352).

References

[1] M. Nazir, F. Wahid, and S. Ali Khan, "A simple and intelligent approach for brain MRI classification," *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 3, pp. 1127–1135, 2015.

[2] M. H. Fazel Zarandi, M. Zarinbal, and M. Izadi, "Systematic image processing for diagnosing brain tumors: a type-II fuzzy expert system approach," *Applied Soft Computing*, vol. 11, no. 1, pp. 285–294, 2011.

[3] H. Nabeel Kaghed, S. Eman al–Shamery, and F. E. K. al-Khuzaie, "Multiple sequence alignment based on developed genetic algorithm," *Indian Journal of Science and Technology*, vol. 9, no. 2, pp. 1–7, 2016.

[4] N. H. Kaghed, E. S. Al, and F. E. K. Al-Khuzaie, "Comparative study of genetic algorithm and dynamic programming of DNA multiple sequence alignment," *Journal of University of Babylon*, vol. 25, no. 2, pp. 403–414, 2017.

[5] F. Al-Khuzaie, "Multiple sequence alignment based on developed genetic algorithm," (Master thesis), University of Babylon, Iraq, 2016, https://www.researchgate.net/publication/347511906_Multiple_Sequence_Alignment_Based_on_Developed_Genetic_Algorithm?channel=doi&linkId=5fdf736a45851553a0db2ee7&showFulltext=true.

[6] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, pp. 110–117, Indianapolis, IN, USA, 2003.

[7] T. H. H. Aldhyani, M. Alrasheedi, A. A. Alqarni, M. Y. Alzahrani, and A. M. Bamhdi, "Intelligent hybrid model to enhance time series models for predicting network traffic," *IEEE Access*, vol. 8, pp. 130431–130451, 2020.

[8] 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.

[9] M. Z. Alom, T. M. Taha, C. Yakopcic et al., "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[10] D. Ravi, C. Wong, F. Deligianni et al., "Deep learning for health informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4–21, 2017.

[11] T. Horikawa, M. Tamaki, Y. Miyawaki, and Y. Kamitani, "Neural decoding of visual imagery during sleep," *Science*, vol. 340, no. 6132, pp. 639–642, 2013.

[12] S. Koyamada, Y. Shikauchi, K. Nakae, M. Koyama, and S. Ishii, "Deep learning of fMRI big data: a novel approach to subject-transfer decoding," 2015, https://arxiv.org/abs/1502.00093.

[13] H. Jang, S. M. Plis, V. D. Calhoun, and J. H. Lee, "Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: evaluation using sensorimotor tasks," *Neuro Image*, vol. 145, Part B, pp. 314–328, 2017.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

[15] S. Koyamada, M. Koyama, K. Nakae, and S. Ishii, "Principal sensitivity analysis," in *Proc. 19th Pacific-Asia Conf. Knowledge Discovery and Data Mining*, pp. 621–632, Ho Chi Minh City, Vietnam, 2015.

[16] T. Schmah, G. E. Hinton, R. S. Zemel, S. L. Small, and S. Strother, "Generative versus discriminative training of rbms for classification of fMRI images," in *Proc. 21st Int. Conf. Neural Information Processing Systems*, pp. 1409–1416, Vancouver, Canada, 2008.

[17] J. H. Cole, R. P. K. Poudel, D. Tsagkrasoulis et al., "Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker," *NeuroImage*, vol. 163, pp. 115–124, 2017.

[18] D. P. Kuang and L. H. He, "Classification on ADHD with deep learning," in *Proc., Int. Conf. Cloud Computing and Big Data (CCBD)*, pp. 27–32, Wuhan, China, 2014.

[19] Z. Yang, S. H. Zhong, A. Carass, S. H. Ying, and J. L. Prince, "Deep learning for cerebellar ataxia classification and functional score regression," in *Proc. 5th Int. Workshop on Machine Learning in Medical Imaging*, pp. 68–76, Boston, MA, USA, 2014.

[20] H. C. Kim and J. H. Lee, "Deep learning to predict the emotional response using functional MRI data," *NeuroImage*, vol. 186, pp. 607–627, 2019.

[21] H. T. Gorji and J. Haddadnia, "A novel method for early diagnosis of Alzheimer's disease based on pseudo Zernike moment from structural MRI," *Neuroscience*, vol. 305, pp. 361–371, 2015.

[22] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, "Early diagnosis of Alzheimer's disease with deep learning," in *Pro. IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pp. 1015–1018, Beijing, China, 2014.

[23] F. Li, L. Tran, K. H. Thung, S. W. Ji, D. G. Shen, and J. Li, "A robust deep model for improved classification of AD/MCI patients," *IEEE J Biomed Health Inform*, vol. 19, no. 5, pp. 1610–1616, 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, CA, USA, 2012.

[25] J. Liu, M. Li, Y. Pan et al., "Complex brain network analysis and its applications to brain disorders: a survey," *Complexity*, vol. 2017, Article ID 8362741, 27 pages, 2017.

[26] L. Jin, P. Yi, L. Min et al., "Applications of deep learning to MRI images: a survey," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 1–18, 2018.

[27] O. S. Soliman and E. A. El Hamd, "Classification of breast cancer using differential evolution and least squares support vector machine," *Int J Emerg Trends Technol Comput Sci*, vol. 3, no. 2, pp. 155–161, 2014.

[28] J. Liu, M. Li, J. Wang, F. Wu, T. Liu, and Y. Pan, "A survey of MRI-based brain tumor segmentation methods," *Tsinghua Science and Technology*, vol. 19, no. 6, pp. 578–595, 2014.

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

[30] M. Natarajan and S. Sathiamoorthy, "A novel pre-processing approach for the denoising of Alzheimer disease image dataset," *Asian Journal of Computer Science and Technology*, vol. 7, no. 2, pp. 107–112, 2018.

[31] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, J. Tohka, and Alzheimer's Disease Neuroimaging Initiative, "Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects," *NeuroImage*, vol. 104, pp. 398–412, 2015.

[32] C. Vaishnavi and P. Anna, "Alzheimer's disease: a comprehensive review," *EC Neurology*, vol. 8, no. 1, pp. 16–32, 2017.

[33] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. M. Arrighi, "Forecasting the global burden of Alzheimer's disease," *Alzheimer's & Dementia*, vol. 3, no. 3, pp. 186–191, 2007.

[34] S. L. Murphy, J. Xu, and K. D. Kochanek, "Deaths: final data for 2010," *Natl Vital Stat Rep*, vol. 61, no. 4, pp. 1–117, 2013.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9761657, 1 page
https://doi.org/10.1155/2023/9761657

Retraction

Retracted: Water Quality Prediction Using Artificial Intelligence Algorithms

Applied Bionics and Biomechanics

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

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] T. H. H. Aldhyani, M. Al-Yaari, H. Alkahtani, and M. Maashi, "Water Quality Prediction Using Artificial Intelligence Algorithms," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 6659314, 12 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2020, Article ID 6659314, 12 pages
https://doi.org/10.1155/2020/6659314

Research Article

Water Quality Prediction Using Artificial Intelligence Algorithms

Theyazn H. H Aldhyani,[1] **Mohammed Al-Yaari**,[2] **Hasan Alkahtani**,[3] **and Mashael Maashi**[4]

[1]*Community College of Abqaiq, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*
[2]*Chemical Engineering Department, King Faisal University, P.O. Box 380, Al-Ahsa 31982, Saudi Arabia*
[3]*College of Computer Science and Information Technology, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*
[4]*Software Engineering Department, King Saud University, Riyadh 11543, Saudi Arabia*

Correspondence should be addressed to Mohammed Al-Yaari; malyaari@kfu.edu.sa

Received 29 November 2020; Revised 12 December 2020; Accepted 16 December 2020; Published 30 December 2020

Academic Editor: Fahd Abd Algalil

During the last years, water quality has been threatened by various pollutants. Therefore, modeling and predicting water quality have become very important in controlling water pollution. In this work, advanced artificial intelligence (AI) algorithms are developed to predict water quality index (WQI) and water quality classification (WQC). For the WQI prediction, artificial neural network models, namely nonlinear autoregressive neural network (NARNET) and long short-term memory (LSTM) deep learning algorithm, have been developed. In addition, three machine learning algorithms, namely, support vector machine (SVM), *K*-nearest neighbor (K-NN), and Naive Bayes, have been used for the WQC forecasting. The used dataset has 7 significant parameters, and the developed models were evaluated based on some statistical parameters. The results revealed that the proposed models can accurately predict WQI and classify the water quality according to superior robustness. Prediction results demonstrated that the NARNET model performed slightly better than the LSTM for the prediction of the WQI values and the SVM algorithm has achieved the highest accuracy (97.01%) for the WQC prediction. Furthermore, the NARNET and LSTM models have achieved similar accuracy for the testing phase with a slight difference in the regression coefficient (RNARNET = 96.17% and RLSTM = 94.21%). This kind of promising research can contribute significantly to water management.

1. Introduction

Water is the most significant resource of life, crucial for supporting the life of most existing creatures and human beings. Living organisms need water with enough quality to continue their lives. There are certain limits of pollutions that water species can tolerate. Exceeding these limits affects the existence of these creatures and threatens their lives.

Most ambient water bodies such as rivers, lakes, and streams have specific quality standards that indicate their quality. Moreover, water specifications for other applications/usages possess their standards. For example, irrigation water must be neither too saline nor contain toxic materials that can be transferred to plants or soil and thus destroying the ecosystems. Water quality for industrial uses also requires different properties based on the specific industrial processes. Some of the low-priced resources of fresh water, such as ground and surface water, are natural water resources. However, such resources can be polluted by human/industrial activities and other natural processes.

Hence, rapid industrial development has prompted the decay of water quality at a disturbing rate. Furthermore, infrastructures, with the absence of public awareness, and less hygienic qualities, significantly affect the quality of drinking water [1]. In fact, the consequences of polluted drinking water are so dangerous and can badly affect health, the environment, and infrastructures. As per the United Nations (UN) report, about 1.5 million people die each year because of contaminated water-driven diseases. In developing countries, it is announced that 80% of health problems are caused by contaminated water. Five million deaths and 2.5 billion illnesses are reported annually [2]. Such a mortality rate is higher than deaths resulting from accidents, crimes, and terrorist attacks [3].

Therefore, it is very important to suggest new approaches to analyze and, if possible, to predict the water quality (WQ). It is recommended to consider the temporal dimension for forecasting the WQ patterns to ensure the monitoring of the seasonal change of the WQ [4]. However, using a special variation of models together to predict the WQ grants better results than using a single model [5–7]. There are several methodologies proposed for the prediction and modeling of the WQ. These methodologies include statistical approaches, visual modeling, analyzing algorithms, and predictive algorithms. For the sake of the determination of the correlation and relationship among different water quality parameters, multivariate statistical techniques have been employed [4]. The geostatistical approaches were used for transitional probability, multivariate interpolation, and regression analysis [5].

Massive increases in population, the industrial revolution, and the use of fertilizers and pesticides have led to serious effects on the WQ environments [8, 9]. Thus, having models for the prediction of the WQ is of great help for monitoring water contamination.

Currently, two main types for modeling and predicting water quality are available: mechanism- and non-mechanism-oriented models. The mechanism model is relatively sophisticated; it uses the advanced system structure data for simulating the WQ, and thus, it is considered as a multifunctional model that can be used for any water body. In addition, the Streeter–Phelos (S–P) model, one of the earliest WQ simulation model, has been used widely.

Later, some countries have developed a variety of WQ models including the QUAL model [10] and the WASP model [11], which have gained wide usage in mimicking the water quality of rivers. This was followed by Warren and Bach [12] who suggested to use MIKE21 for designing systems to model the estuaries, coastal waters, and seas.

Hayes et al. [13] have paired two models for improving the quality of downstream water, namely, quasi-static two-dimensional dissolved oxygen reservoir model (DORM-II) and a daily scale optimal dispatch model.

Using environmental fluid dynamics code (EFDC), a two-dimensional numerical model was developed to simulate the water environment of the Mudan River [14]. This is based on the distance between points and intervals [15].

Another study was conducted by Batur and Maktav [16] to predict the WQ of Lake Gala (Turkey) using satellite image fusion based on the principal component analysis (PCA) method. Jaloree et al. [17] have attempted to predict the WQ of the Narmada River with five WQ indicators using a decision tree model. Another study suggested the use of the deep Bidirectional Stacked Simple Recurrent Unit (Bi-S-SRU) [18] for the designing of a precise forecasting scheme of the WQ in smart mariculture.

Liao and Sun [19] developed a model to forecast the WQ of China's Chao Lake by pairing the ANN and decision tree algorithm. Yan and Qian [20] proposed an affinity propagation clustering model based on a least-squares support vector machine (AP-LSSVM). This model is highly sensitive to vacancies. Solanki et al. [21] analyzed and predicted the chemical eigenvalues of water, especially dissolved oxygen and pH using the deep learning network model which was reported to demonstrate more accurate results compared with supervised learning-based techniques. Li et al. [22] developed a novel hybrid model using a neural network and the Markov chain method. This model has helped in predicting dissolved oxygen, a primary measure of the WQ [23]. Khan and See [24] included dissolved oxygen, chlorophyll, conductivity, and turbidity in the developed WQ model using an artificial neural network (ANN). Yan et al. [25] suggested a genetic algorithm (GA) and particle swarm optimization (PSO) algorithm to enhance the backpropagation (BP) neural network to predict the oxygen demanded in a lake. An enhanced accuracy of the prediction results was reported.

Several studies have been performed to model and predict the water quality using different ANN models. These studies have approved the feasibility and effectiveness of employing ANN applications to predict the quality of drinking water.

Currently, researchers mostly emphasize enhancing the applicability and reliability of water quality prediction/modelling by using a variety of new technologies such as Fuzzy logic, stochastic, ANN, and deep learning [26, 27].

Shafi et al. [28] proposed four machine learning algorithms, namely, Support Vector Machines (SVM), Neural Networks (NN), Deep Neural Networks, and *k*-Nearest Neighbors (kNN), for the prediction of water quality. Using single feed-forward neural networks to classify water quality, 25 parameters have been included as input parameters [29].

Ranković et al. [30] estimated the dissolved oxygen (DO) by employing the ANN model. Gazzaz et al. [31] estimated the WQI by using an ANN model, and the Internet of Things (IOT) technology was applied to collect the dataset from water resources. Abyaneh [32] has applied the machine learning approaches like ANN and regression to predict the chemical oxygen demand (COD). Sakizadeh [33] used ANN with Bayesian regularization to estimate the water quality index (WQI). However, the radial-basis-function (RBF), a type of the ANN model, was used for the prediction and classification of water quality [34, 35].

In addition, it has been reported that deep learning methods showed high performance in predicting the WQ when compared to the traditional methods. Marir et al. [36] developed a model to find out the uncommon behavior from large-scale network traffic data. While a deep learning algorithm was employed for extracting features, a multilayer ensemble support vector machine model was used for classification. Fadlullah et al. [37] visualized a reward-based deep learning structure combining a deep convolutional neural network and a deep belief network.

For the analysis and prediction of the WQ of groundwater, different algorithms including ANN, Bayesian neural networks, adaptive neurofuzzy [38], decision support system (DSS), and autoregressive moving average (ARMA) have been applied [39]. However, these mimicking models have some limitations.

However, the contributions of the current study can be summarized as follows:

(i) Developing highly efficient advanced artificial intelligence models to predict the water quality index

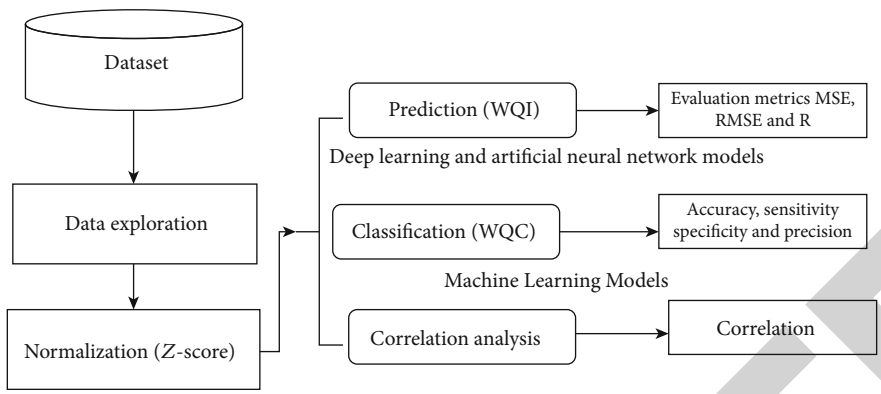


Figure 1: Framework of the proposed methodology.

(WQI) based on artificial neural networks and deep learning algorithms

(ii) Applying some machine learning models, namely, support vector machine (SVM), K-nearest neighbour (K-NN), and Naive Bayes algorithms, for the prediction of water quality classification (WQC).

The highly efficient developed models can be generalized and used to forecast the water pollution process which will help the decision-makers to make the right decisions at the right time.

2. Materials and Methods

Figure 1 displays the proposed methodology of the present study.

2.1. Dataset. The dataset used in this study is collected from certain historical locations in India. It contained 1679 samples from different Indian states during the period from 2005 to 2014. The dataset has 7 significant parameters, namely, dissolved oxygen (DO), pH, conductivity, biological oxygen demand (BOD), nitrate, fecal coliform, and total coliform. Data was collected by the Indian government to ensure the quality of the supplied drinking water. This dataset was obtained from Kaggle https://www.kaggle.com/anbarivan/indian-water-quality-data.

2.2. Data Preprocessing. The processing phase is very important in data analysis to improve the data quality. In this phase, the WQI has been calculated from the most significant parameters of the dataset. Then, water samples have been classified on the basis of the WQI values. For obtaining superior accuracy, the z-score method has been used as a data normalization technique.

2.2.1. Water Quality Index Calculation. To measure water quality, WQI is used to be calculated using various parameters that significantly affect WQ [40–42]. In this study, a published dataset is considered to test the proposed model, and seven significant water quality parameters are included. The WQI has been calculated using the following formula:

$$\mathrm{WQI} = \frac{\sum_{i=1}^{N} q_i \times w_i}{\sum_{i=1}^{N} w_i}, \tag{1}$$

where: N is the total number of parameters included in the WQI calculations q_i is the quality rating scale for each parameter i calculated by equation (2) below, and w_i is the unit weight for each parameter calculated by equation (3).

$$q_i = 100 \times \left(\frac{V_i - V_{\mathrm{Ideal}}}{S_i - V_{\mathrm{Ideal}}}\right), \tag{2}$$

where: V_i is the measured value of parameter i in the tested water samples V_{Ideal} is the ideal value of parameter i in pure water (0 for all parameters except DO = 14.6 mg/l and pH = 7.0), and S_i is the recommended standard value of parameter i (as shown in Table 1).

$$w_i = \frac{K}{S_i}, \tag{3}$$

where K is the proportionality constant that can be calculated as follows:

$$K = \frac{1}{\sum_{i=1}^{N} S_i}, \tag{4}$$

Tables 2 and 3 represent the unit weight of each parameter and the WQC, respectively.

2.2.2. Z-Score Normalization Method. Normalization is a way to simplify calculations. It is a dimensional expression transformed into a nondimensional expression and becomes a scalar. Z-score normalization (or normalization score) is a normalization method used to normalize parameters by using the mean (μ) and standard deviation (σ) values of the tested data. It can be calculated as follows:

$$Z\text{-score} = \frac{(x - \mu)}{\sigma}, \tag{5}$$

Table 1: Permissible limits of the parameters used in calculating WQI [43].

Parameters	Permissible limits
Dissolved oxygen, mg/l	10
pH	8.5
Conductivity, μS/cm	1000
Biological oxygen demand, mg/l	5
Nitrate, mg/l	45
Fecal coliform, Cfu/100 ml	100
Total coliform, Cfu/100 ml	1000

Table 2: Parameter unit weights.

Parameter	Unit weight (w_i)
Dissolved oxygen	0.2213
pH	0.2604
Conductivity	0.0022
Biological oxygen demand	0.4426
Nitrate	0.0492
Fecal coliform	0.0221
Total coliform	0.0022

Table 3: Water quality classification (WQC) [42].

Water quality index range	Classification
0-25	Excellent
26-50	Good
51-75	Poor
76-100	Very poor
Above 100	Unsuitable for drinking

where x is the measured value of the parameter i in the tested sample.

2.3. Prediction of Water Quality Index. For this purpose, ANN models, namely, nonlinear autoregressive neural network (NARNET) and long short-term memory (LSTM) deep learning algorithm, were used for the prediction of water quality index.

2.3.1. Artificial Neural Network (ANN) Model. In general, the neural network (NN) models are used as very powerful machine learning algorithms for time-series prediction of different engineering applications. The ANN model has consisted of an input layer, a hidden layer/s, and an output layer. Each hidden layer has weight and bias parameters to manage neurons. To transfer the data from the hidden layer into the output layer, the activation function is used. The learning algorithms are used to select the weights within the NN framework. The weight selection is based on the minimum performance measures such as mean square error (MSE).

The NARNET model is a very popular multilayer feedforward network. It starts with a guessed initial weight value, which is then updated using the actual data. Consequently, there is some sort of randomness in the prediction process

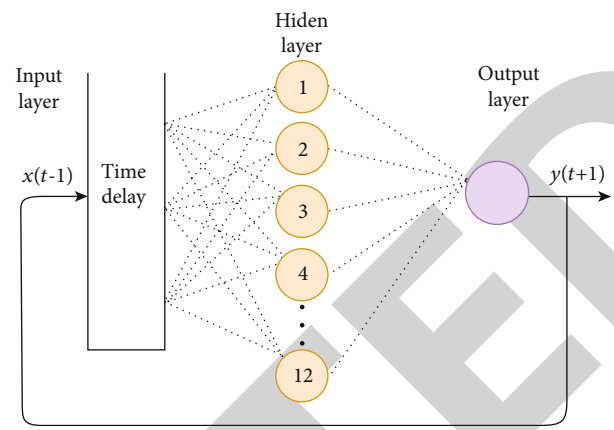


Figure 2: Computation of the NARNET model.

Table 4: Parameters of the developed ANN (NARNET) model.

Number of hidden layers	12
Number of delays	1 : 8
Maximum number of iterations	100
Maximum number of epochs	12
Number of gradients	1.734×10^3

performed by the NN model. The network is regularly trained many times using different random values for the initialization, and the results are averaged. In the NARNET model, the number of hidden layers and nodes must be identified in advance. Figure 2 displays the NARNET model scheme with multiple inputs and 4 hidden layers (as recommended for most of the research datasets). Equation (6) describes the NARNET time series model.

$$y(t) = h(y(t-1), y(t-2), \cdots, y(t-p)) + \epsilon(t), \qquad (6)$$

where y is the value of time-series data at time t and $y(t)$ for employing the p observation values of the series. The function (h) is used to optimize the network weights and neuron bias. Finally, the $\epsilon(t)$ is the error obtained from the model at time t.

In this work, the NARNET model has been developed to predict the WQI. The NARNET model is a time series model that is used to predict the stationary time series compared with other ANN models like the forward neural network model. The WQI parameters seem in the form of time series; therefore, the NARNET model is proposed to predict the WQI. Table 4 shows the significant parameters of the developed model. Figure 3 represents the topology of the developed NARNET model.

2.3.2. Deep Neural Network (DNN) Model. The DNN model is one type of feedforward NN algorithms, which is a fundamental technique for deep learning. DNN consists of 3 levels of nodes, and each node follows a nonlinear function, except for the input node. DNN presents a technique of backpropagation supervised learning. In this work, a WQI model was developed using the DNN algorithm and the simple DNN was compared with the proposed model. This model includes

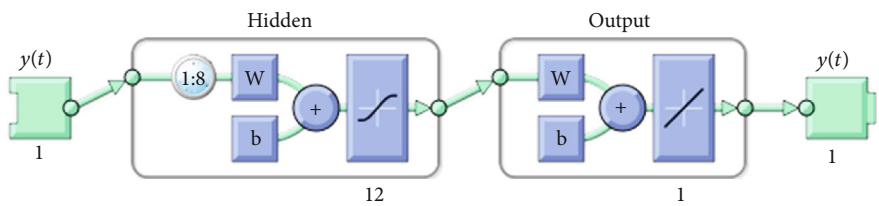


Figure 3: Architecture of the NARNET model.

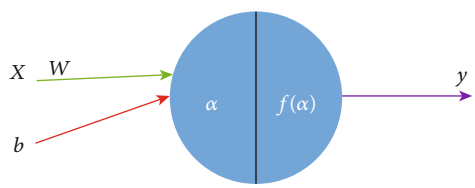


Figure 4: Architecture of the DNN model.

the following parameters and functions: bias (b), input (x), output (y), weight (w), calculation function (α), and activation function $f(\alpha)$. The neuron architecture of the DNN model is schematically shown in Figures 4and 5. Every single neuron in the DNN employs the following equations.

$$\alpha : sum = w \bullet x + b, \tag{7}$$

$$y : f(\alpha) = f(w \bullet x + b), \tag{8}$$

Recurrent neural network (RNN) is one type of deep learning techniques used in different domains such as computer vision, natural language processing, pattern recognition, and medical image diagnosis. As compared to different feed ANNs, RNN has a directional control loop that enables the previous states to be stored, recalled, and added to the current output. One of the most powerful RNN algorithms used to predict time series data is the LSTM model.

The long short-term memory (LSTM) model, a deep learning algorithm, is appropriate for estimating the time-series data whenever there is a randomized sized time step. The activating function used in the LSTM model is a logistic sigmoid. Providing that the forget gate is opened and the input gate is closed, the memory cell keeps reminding of the first entry and thus solving the typical RNN problems [44]. The formulas of the RNN model are as follows:

$$h_t = \tan h(W_i \bullet h_t + w_x x_t), \tag{9}$$

$$y_t = w_y \bullet w_t, \tag{10}$$

where h_t is the hidden layer of NN for the input training data (x_t). The output layer is represented by y_t. However, w_t and w_y are the weight of the neural cell and the matrix, respectively. The RNN model is used to create the LSTM model for the computing process. The LSTM consists of three significant parameters, namely, the input gate, forget gate, and output gate. The formulas used to compute the LSTM model are as follows:

$$\text{Input gate} : i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i), \tag{11}$$

$$\text{Forget gate} : f_t = \sigma(w_f \bullet [h_{t-1}, x_t] + b_f), \tag{12}$$

$$\text{Output gate} : o_t = \sigma(W_0 \bullet [h_{t-1}, x_t] + b_0), \tag{13}$$

$$\text{New memory cell} : \widetilde{c}_t = \tan h(W_c \bullet [h_{t-1}, x_t] + b_c), \tag{14}$$

$$\text{Final memory cell} : C_t = f_t \times C_{t-1} + i_t \times \widetilde{c}_t, \tag{15}$$

$$h_t = o_t \times \tan h(C_t), \tag{16}$$

where:

i_t, f_t, and o_t: input, forget, and output gates, respectively

h_t: number of hidden layers

σ: the logistic sigmoid function is used to transfer the training data from a hidden layer into the output gate

w_t: the weighted neural network

$\widetilde{c}_t$: an internal memory cell is used to compute in the hidden layer

C_t : the internal memory

h_t :the output of a hidden layer state is used to derive from the new memory

i, f, and o : are subscripts that stand for input, forget, and output gates, respectively

x_t: input training data

w_f, $w_o w_c$: weight vector of NN

b_f and b_o: bias vector in NN

The analysis of LSTM was performed utilizing MATLAB. Throughout the LSTM layer, 23 variables are open. We just set the units, activate the function, return the sequence, and dropout. Figure 5 illustrates the architecture of the LSTM, and the significant parameters of the LSTM model are presented in Table 5.

2.4. Prediction of Water Quality Classification. In this section, some machine learning algorithms, namely, support vector machine (SVM), K-nearest neighbor (KNN), and Naive Bayes, have been used to predict the water quality classification.

2.4.1. Support Vector Machine (SVM) Model. The SVM model was developed in 1995 by Corinna Cortes and Vapnik. It has several unique benefits in solving small samples, and nonlinear and high-dimensional pattern recognition. It can be extended to function in the simulation of other machine learning problems. It uses the hyperplane to separate the points of the input vectors and finds the needed coefficients.

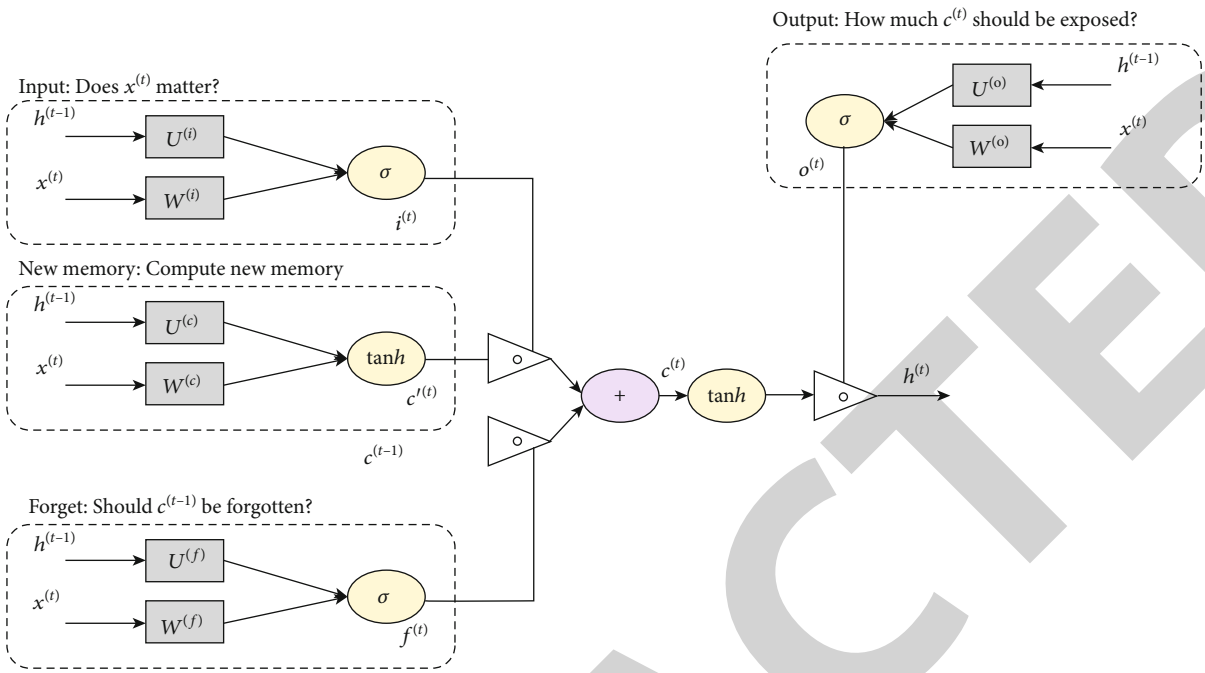


Figure 5: Architecture of the LSTM model.

Table 5: Parameters of the LSTM model.

Parameters	Numbers
Shallow hidden layer size	[30 80]
No. of hidden units 2	200
No. of hidden units 1	350
Delays	[1 3 4 7]
Maximum number of iterations	1500
Maximum number of epochs	150

Table 6: Performances of the NARNET LSTM and ANN models to predict WQI.

Models	Training data set		Testing data	
	MSE	R (%)	MSE	R (%)
NARNET	0.2815	95.97	0.1353	96.17
LSTM	0.1316	93.93	0.1028	94.21

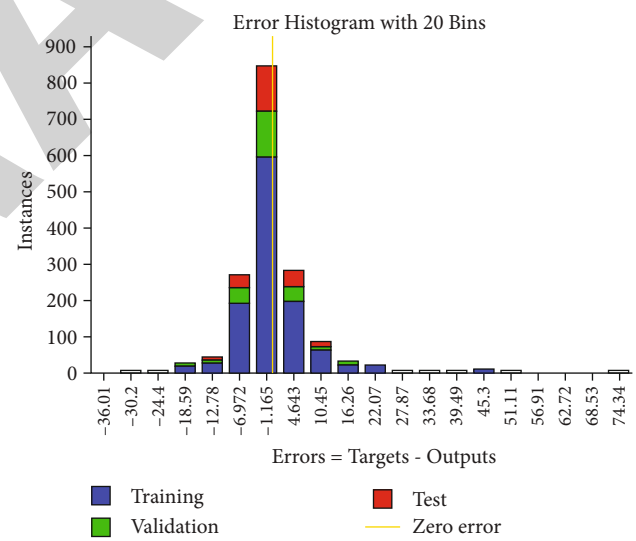


Figure 6: Histogram error of the NARNET model.

The best hyperplane is the line with the largest margin, which is meant the distance between the hyperplane and the nearest input objects. The input points defined in the hyperplane are called *support vectors*. In this work, the linear SVM model along with the Gaussian radial basis function (equation (17)) is used to classify the tested water samples based on their quality.

$$K\left(X, X'\right) = \exp\left(-\frac{\left\|X - X'\right\|^2}{2\sigma^2}\right), \tag{17}$$

where X and X' represent the feature vectors of the input dataset and the $\|X - X'\|^2$ is the squared Euclidean distance between the two feature inputs. The σ is a free parameter.

2.4.2. K-Nearest Neighbor (K-NN) Model. The K-NN algorithm is a basic classification and regression method. It is used to find the K values that are close to values in the training dataset. Most of these values belong to a certain class, and thus, tested data can be classified. The K value is used to find the closest points in the feature vectors, and the value should be unique. The following expression of the Euclidean distance function (Di) can be used.

$$D_i = \sqrt{(x_1 - x_2) + (y_1 - y_2)^2}, \tag{18}$$

where x_1, x_2, y_1, and y_2 are the variables for input data.

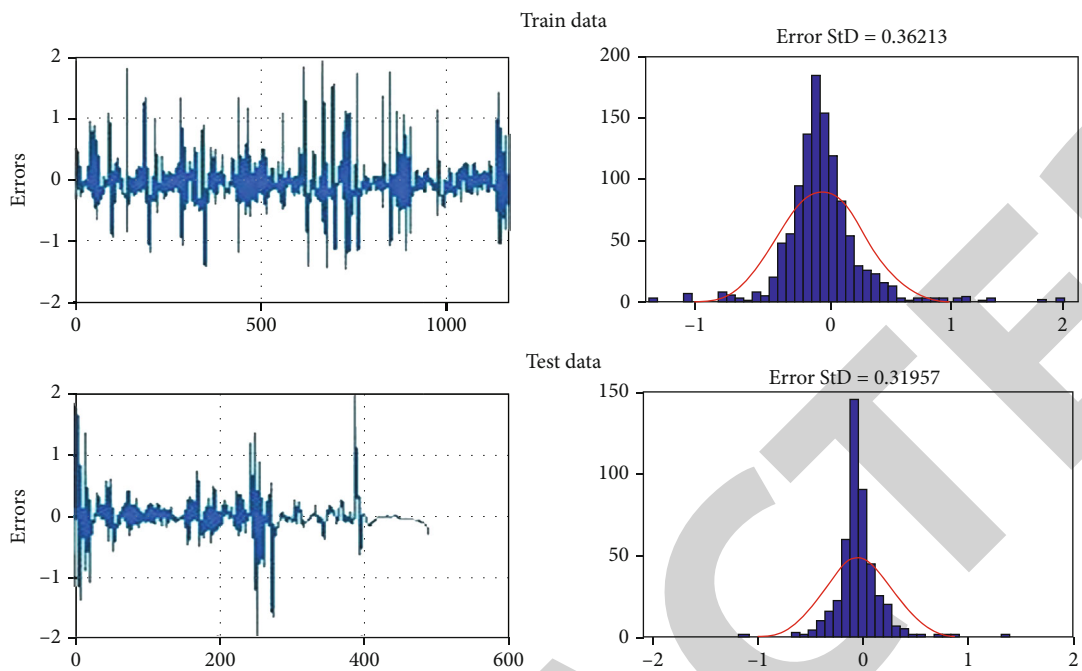


Figure 7: Histogram error and mean error of the LSTM model in the training and testing phases.

2.4.3. Naive Bayes Model. The Bayesian method uses the knowledge of probability statistics to predict and classify datasets. The Bayesian algorithm combines prior and posterior probabilities to avoid the supervisor's bias and the overfitting phenomenon of using sample information alone.

This Naive Bayes is a type of classification algorithms based on Bayes' theorem and the assumption of the independence of characteristic conditions. Attributes are assumed to be conditionally independent of each other when the target value is given. This method greatly simplifies the complexity of the Bayesian method.

In Bayesian analysis, the probability of an event A given an event B is not the same as the probability of B given A as in equation (18).

$$P(A \mid B) \neq P(B \mid A). \tag{19}$$

Assuming that $A_1, A_2 \cdots .A_n$ and C are the feature vectors and the class of the WQC dataset, respectively, the Bayes equation can be expressed as follows:

$$P(C \mid A) = \frac{P(C) \times P(A \mid C)}{P(A)}, \tag{20}$$

where the $P(A)$ is a prior probability representing the feature vectors of the WQC dataset and $P(A \mid C)$ is the prior probability of the class of the WQC dataset.

2.5. Performance Measurement. The statistical analysis, namely, mean square error (MSE), has been used to evaluate the robustness of the developed models to predict the WQI. However, the accuracy, specificity, sensitivity, precision, and F-score evaluation matrices were employed to evaluate the developed classification model to predict the WQC. The used statistical parameters were defined as follows:

(a) Mean Square Error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y\wedge_i)^2, \tag{21}$$

where y_i and $\hat{y}_i$ are the predicted and the observed responses, respectively, and N is the total number of variables.

(b) Accuracy

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \times 100\%, \tag{22}$$

(c) Specificity

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \times 100\%, \tag{23}$$

(d) Sensitivity

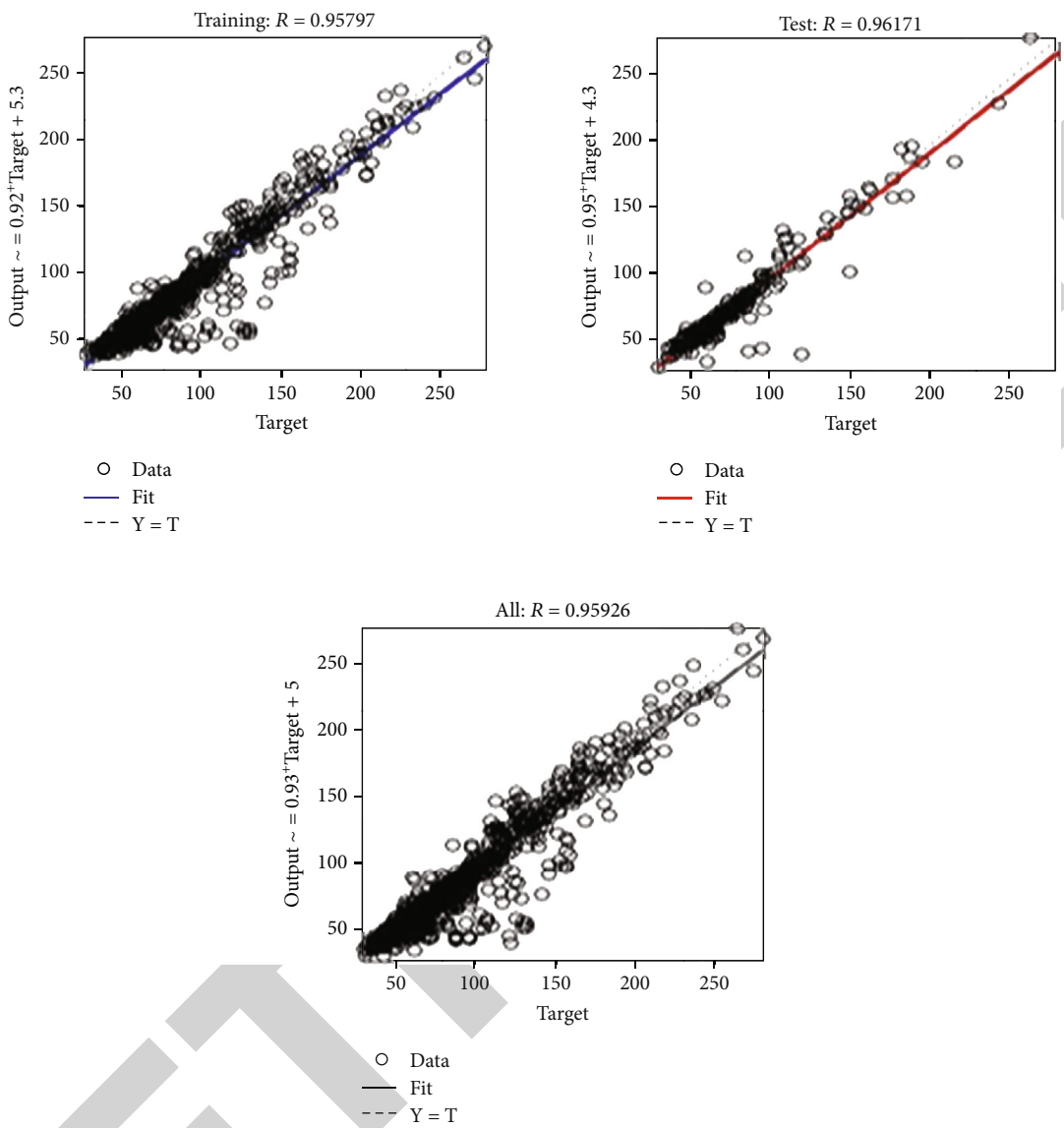


Figure 8: Regression plot of the NARNET model.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{24}$$

(e) Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \tag{25}$$

(f) F-score

$$F\text{-score} = \frac{2 \times \text{precision} \times \text{sensitivity}}{\textit{preision} + \text{sensitivity}} \times 100\%, \tag{26}$$

where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative, respectively.

2.6. Correlation Analysis. Pearson's correlation coefficient approach is applied to analyze the correlation between the significant parameters of the dataset used for the prediction of the QWI values.

$$R = \frac{n\sum(x \times y) - (\Sigma x)(\Sigma y)}{[n\sum(x^2) - \sum(x^2)] \times [n\sum(y^2) - \sum(y^2)]} \times 100\%, \tag{27}$$

where:
R: Pearson's correlation coefficient approach
x: input values in the first set of the training data
y: input values of the second set of the training data
n: total number of input variables

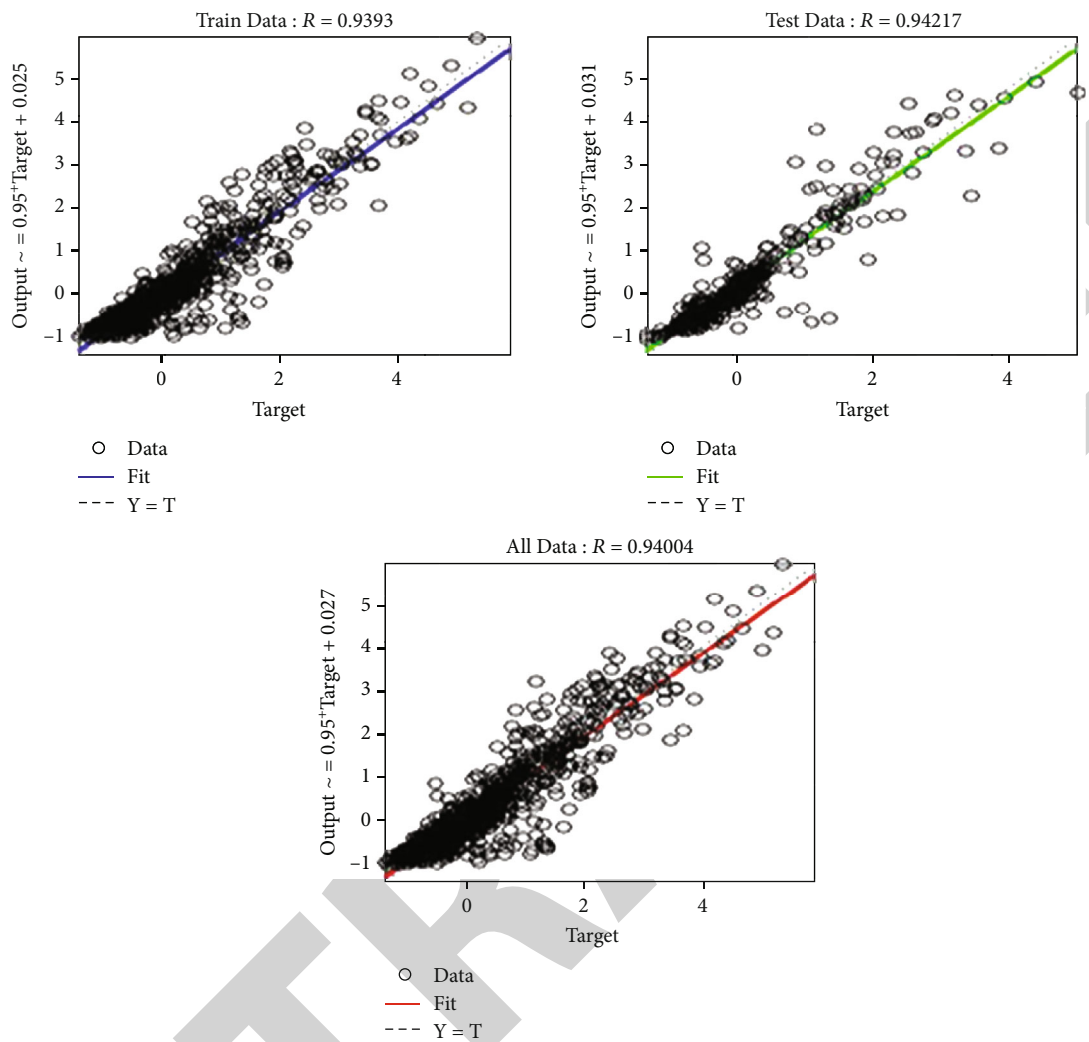


Figure 9: Regression plot of the LSTM model.

Table 7: Performance of Pearson's correlation coefficient approach.

Parameter	DO (mg/l)	pH	Conductivity (μS/cm)	BOD (mg/l)	Nitrate (mg/l)	Fecal coliform (MPN/100 ml)	Total coliform (MPN/100 ml)	WQI
DO (mg/l)	1.00	0.0466	-0.2914	-0.1819	-0.0347	0.1128	-0.1536	-0.3836
pH	0.0466	1.00	0.3268	0.2697	0.0562	-0.2082	-0.2170	0.5233
Conductivity (μS/cm)	-0.2914	0.3268	1.00	0.3288	0.1009	-0.1120	-0.0777	0.3935
BOD (mg/l)	-0.1819	0.2697	0.3288	1.00	0.2257	-0.1597	-0.1633	0.6130
Nitrate (mg/l)	-0.0347	0.0562	0.1009	0.2257	1.00	0.1408	0.0545	0.1768
Fecal coliform (MPN/100 ml)	-0.1128	-0.2082	-0.1120	-0.1597	0.1408	1.00	0.9119	0.2779
Total coliform (MPN/100 ml)	-0.1536	-0.2170	-0.0777	-0.1633	0.0545	0.9119	1.00	0.2679
WQI	-0.3836	0.5233	0.3935	0.6130	0.1768	0.2779	0.2679	1.00

2.7. Experimental Setup. The prediction experiments have been conducted in a specific environment (MATLAB 2018). The simulation has been performed using a system with i5 Processor and 4 GB RAM to process all required tasks.

3. Results and Discussion

For validating the developed model, the dataset has been divided into 70% training and 30% testing subsets. While the ANN and LSTM models were used to predict the WQI,

Table 8: Performance of the used machine learning models to predict WQC.

Models	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	*F*-score (%)
SVM	97.01	99.23	97.78	94.93	98.54
KNN	83.63	84.73	94.93	87.50	85.84
Naive Bayes	75.20	77.76	91.65	78.08	81.51

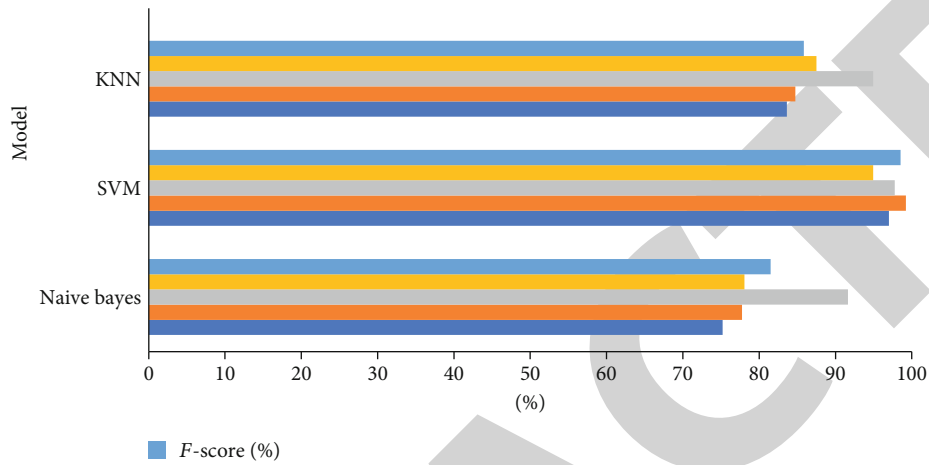


Figure 10: Performance of the machine learning algorithms used for the prediction of the WQC.

the SVM, KNN, and Naive Bayes were utilized for the water quality classification prediction.

3.1. Prediction of the WQI. A NARNET model, with 12 hidden layers, showed a good performance to predict the WQI values. As presented earlier, it has the following characteristics: 1 : 8 number of delays and 12 number of epochs. However, the developed LSTM model has a total number of 200 hidden layers,150 maximum number of epochs, and delays of [1, 3, 4, 7].

Table 6 summarizes the performance parameters of the developed models to predict WQI, although the prediction accuracy of LSTM for the testing data was slightly better than that for the training data. In addition, the LSTM model, in general, has shown a slightly better performance compared with the NARNET model according to the MSE values. However, based on the R value, the NARNET model has shown a better performance. In general, both models demonstrated an excellent prediction of the WQI values with $R\% > 93.93$.

Figure 6 illustrate the histogram error of the NARNET model. The histogram metric is used to find errors between the target values and the predicted values of training and testing datasets. The total error range is divided into 20 smaller bins, where the y-axis refers to the number of samples located in a particular bin. Figure 7 displays the histogram metric and mean errors of the LSTM model in the training and testing phases. The mean error and histogram metric are used to find the deviation between the observation values and the predicted values of training and testing.

Figures 8 and 9 display the regression plots for the predicted values of training, testing, and whole datasets for the NARNET and LSTM models, respectively. This plot is used to find the relationship between the predicted values and actual values. The "target" values in the plot are the actual dataset, whereas the "output" is the predicted values obtained from the NARNET and LSTM models. As shown in both figures, there is a clear good agreement ($R > 95.7\%$ (NARNET) and $R > 93.3\%$ (LSMT)) between the predicted WQI values and the ones calculated from the measured parameters. This implies the highly efficient performance of both developed models.

Table 7 summarizes the Pearson's correlation coefficient approach is used to predict the WQI values. The correlation between the WQI parameters for selecting the optimal parameters has been obtained. Results revealed that all parameters have a strong relationship with WQI parameters. This indicates that these parameters are very important for predicting the quality of water.

3.2. Prediction of the Water Quality Classification. This section presents the results of the classification algorithms are used to predict the WQC. Table 8 shows the results of the used machine learning algorithms. It is noted that the performance of the SVM algorithm is very superior as compared to the KNN and Naive Bayes models. However, the Naive Bayes algorithm has shown the poorest performance. Figure 10 shows the performance of the used algorithms to predict the WQC.

4. Conclusions

Modeling and prediction of water quality are very important for the protection of the environment. Developing a model by using advanced artificial intelligence algorithms can be used to measure the future water quality. In this proposed methodology, the advanced artificial intelligence algorithms, namely, NARNET and LSTM models were used to predict the WQI. Moreover, machine learning algorithms such as

SVM, KNN, and Naive Bayes were used to classify the WQI data. The proposed models were evaluated and examined by some statistical parameters. For the WQI prediction, the result has revealed that the performance of the NARNET model is slightly better than the LSTM model based on the obtained R value. However, the SVM algorithm has achieved the highest accuracy of the prediction of the WQC as compared with KNN and Naive Bayes algorithms. After examining the robustness and efficiency of the proposed model for predicting the WQI, in future work, the developed models will be implemented to predict the water quality in Saudi Arabia for different types of water.

Data Availability

The dataset used in this study is collected from certain historical locations in India. It contained 1679 samples from different Indian states during the period from 2005 to 2014. The dataset has 7 significant parameters named dissolved oxygen (DO), pH, conductivity, biological oxygen demand (BOD), nitrate, fecal coliform, and total coliform. The data was collected by the Indian government to ensure the quality of the supplied drinking water. This dataset was obtained from Kaggle https://www.kaggle.com/anbarivan/indian-water-quality-data.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

All authors contributed significantly to the completion of this article.

Acknowledgments

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number IFT20111.

References

[1] P. Zeilhofer, L. V. A. C. Zeilhofer, E. L. Hardoim, Z. M. . Lima, and C. S. Oliveira, "GIS applications for mapping and spatial modeling of urban-use water quality: a case study in District of Cuiabá, Mato Grosso, Brazil," *Cadernos de Saúde Pública*, vol. 23, no. 4, pp. 875–884, 2007.

[2] M. A. Kahlown, M. A. Tahir, and H. Rasheed, *National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006)*, Pakistan Council of Research in Water Resources Islamabad, Islamabad, Pakistan, 2007, http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/Water%20Quality%20Monitoring%20Report%202005-06.pdf.

[3] UN water, "Clean water for a healthy world," Development, 2010, https://www.undp.org/content/undp/en/home/presscenter/articles/2010/03/22/clean-water-for-a-healthy-world.html.

[4] K. Farrell-Poe, W. Payne, and R. Emanuel, *Water Quality & Monitoring*, University of Arizona Repository, 2000, http://hdl.handle.net/10150/146901.

[5] T. Taskaya-Temizel and M. C. Casey, "A comparative study of autoregressive neural network hybrids," *Neural Networks*, vol. 18, no. 5–6, pp. 781–789, 2005.

[6] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data," *Applied Soft Computing*, vol. 23, pp. 27–38, 2014.

[7] X. Zhang, N. Hu, Z. Cheng, and H. Zhong, "Vibration data recovery based on compressed sensing," *Acta Physica Sinica*, vol. 63, no. 20, pp. 119–128, 2014.

[8] M. M. S. Cabral Pinto, C. M. Ordens, M. T. Condesso de Melo et al., "An inter-disciplinary approach to evaluate human health risks due to long-term exposure to contaminated groundwater near a chemical complex," *Exposure and Health*, vol. 12, no. 2, pp. 199–214, 2020.

[9] M. M. S. Cabral Pinto, A. P. Marinho-Reis, A. Almeida et al., "Human predisposition to cognitive impairment and its relation with environmental exposure to potentially toxic elements," *Environmental Geochemistry and Health*, vol. 40, no. 5, pp. 1767–1784, 2018.

[10] Y. C. Lai, C. P. Yang, C. Y. Hsieh, C. Y. Wu, and C. M. Kao, "Evaluation of non-point source pollution and river water quality using a multimedia two-model system," *Journal of Hydrology*, vol. 409, no. 3-4, pp. 583–595, 2011.

[11] J. Huang, N. Liu, M. Wang, and K. Yan, "Application WASP model on validation of reservoir-drinking water source protection areas delineation," in *2010 3rd International Conference on Biomedical Engineering and Informatics*, pp. 3031–3035, Yantai, China, October 2010.

[12] I. R. Warren and H. K. Bach, "MIKE 21: a modelling system for estuaries, coastal waters and seas," *Environmental Software*, vol. 7, no. 4, pp. 229–240, 1992.

[13] D. F. Hayes, J. W. Labadie, T. G. Sanders, and J. K. Brown, "Enhancing water quality in hydropower system operations," *Water Resources Research*, vol. 34, no. 3, pp. 471–483, 1998.

[14] G. Tang, J. Li, Z. Zhu, Z. Li, and F. Nerry, "Two-dimensional water environment numerical simulation research based on EFDC in Mudan River, Northeast China," in *2015 IEEE European Modelling Symposium (EMS)*, pp. 238–243, Madrid, Spain, October 2015.

[15] L. Hu, C. Zhang, C. Hu, and G. Jiang, "Use of grey system for assessment of drinking water quality: a case S study of Jiaozuo city, China," in *2009 IEEE International Conference on Grey Systems and Intelligent Services (GSIS 2009)*, pp. 803–808, Nanjing, China, November 2009.

[16] E. Batur and D. Maktav, "Assessment of surface water quality by using satellite images fusion based on PCA method in the Lake Gala, Turkey," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2983–2989, 2019.

[17] S. Jaloree, A. Rajput, and G. Sanjeev, "Decision tree approach to build a model for water quality," *Binary Journal of Data Mining & Networking*, vol. 4, pp. 25–28, 2014.

[18] J. Liu, C. Yu, Z. Hu et al., "Accurate prediction scheme of water quality in smart mariculture with deep Bi-S-SRU learning network," *IEEE Access*, vol. 8, pp. 24784–24798, 2020.

[19] H. Liao and W. Sun, "Forecasting and evaluating water quality of Chao Lake based on an improved decision tree method," *Procedia Environmental Sciences*, vol. 2, pp. 970–979, 2010.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9808790, 1 page
https://doi.org/10.1155/2023/9808790

Retraction

Retracted: An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning

Applied Bionics and Biomechanics

Received 15 August 2023; Accepted 15 August 2023; Published 16 August 2023

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. M. A. Alkadhmi, O. N. Uçan, and M. Ilyas, "An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8888904, 13 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2020, Article ID 8888904, 13 pages
https://doi.org/10.1155/2020/8888904

Research Article

An Efficient and Reliable Routing Method for Hybrid Mobile Ad Hoc Networks Using Deep Reinforcement Learning

Murtadha M. A. Alkadhmi, Osman N. Uçan, and Muhammad Ilyas

ALTINBAŞ University, Engineering and Naturel Science, Mahmutbey, Istanbul, Turkey

Correspondence should be addressed to Murtadha M. A. Alkadhmi; murtadha.alkadhmi1@ogr.altinbas.edu.tr

Received 21 September 2020; Revised 23 October 2020; Accepted 12 November 2020; Published 11 December 2020

Academic Editor: Mohammed Yahya Alzahrani

With the reliance of humans on mobile smart devices that have wireless communication, modules have significantly increased in recent years. Using these devices to communicate with the survivors during a disaster or its aftermath can significantly increase the chances of locating and saving them. Accordingly, a method is proposed in this study to extend the lifetime of the nodes in a Mobile Ad Hoc Network (MANET) while maintaining communications with the nearest base station (BS). Such a methodology allows the rapid establishment of temporary communications with these survivors, as restoring the complex infrastructure is a time-consuming process. The proposed method achieves the longer lifetime of the network by balancing the load throughout the nodes and avoids exhausting those with limited remaining energy. The proposed method has shown significant improvement in the lifetime of the MANET while maintaining similar Packet Delivery Rate (PDR) and route generation time, compared to existing methods.

1. Introduction

Reliance on smart mobile devices, such as smartphones and tablets, has rapidly increased in recent years, according to the enormous number of services that are being provided through these devices, besides traditional communication services [1–3]. These devices are mainly connected to two types of wireless networks, a cellular network that provides basic communication services, such as phone calls, messaging, and internet, and Wi-Fi networks, which provide much less-expensive internet, compared to cellular data [4–6]. Providing these networks requires intense infrastructure to connect these devices to the other parts of the networks and the internet. This infrastructure provides an access point, i.e., base station (BS), for each device to connect it to the remaining parts of the network and other networks [7].

Such infrastructures can suffer from severe damages when a disaster, such as wars, natural disaster, and terrorist attacks, occurs. Reestablishing these networks is a time-consuming process, according to the complexity and intensity of the required infrastructure to reestablish communications. Moreover, the importance of these communications during such a disaster, or during the aftermath, is significant to establish communications with any survivors or those in need of assistance. Thus, a wireless ad hoc network (WANET) can be established by using the same devices as nodes to deliver the data packets to the nearest online BS. Moreover, as these devices are handheld and according to the high possibility that the owners of these devices are on the move, to escape the disastrous area, Mobile Ad Hoc Network (MANET) is more suitable to such a scenario [8, 9].

With the absence of the infrastructure and the use of mobile nodes to route the packets, finding the optimal route has raised as an emerging problem. Several studies have used overlay network technology [10–12] to route the packets in a hybrid network that uses MANET to establish communications with a base station. These methods are presented as improvements to the standard protocols with the aim of preventing crashing in these networks. However, these methods mainly are aimed at optimizing energy consumption [13, 14] or focusing on three-layered

routing to optimize the packet flow by reducing the routing overhead [15–18]. Recently, a Software Defined Network (SDN) has attracted significant attention in optimizing the routing in such networks [19–23].

Naser and Kadhim [19] present three SDN-based routing methods for MANETs. The aim of the proposed method is to reduce the power consumption in the network by minimizing the routes the data travel through, in a clustered MANET. However, despite the relatively improved network lifetime, emphasizing on reducing power consumption can exhaust the resources available on certain nodes. If such nodes are located in key-positions, in which a lot of traffic is routed through these nodes to shorten their paths, these nodes can die significantly faster than when the loading is distributed on multiple nodes. Using slightly longer paths to reduce the loading on certain nodes can maintain a larger number of nodes alive for a longer time, despite the possibility of increasing the end to end (E2E) delay of packets. However, extending the lifetime of each and every node in the MANET is a very important aspect in disastrous scenarios, as these nodes are used to establish communications with their owners.

In addition to the extension of the lifetime of the nodes in the MANET when the loading is distributed among the nodes, the results of the experiments conducted by Poularakis et al. [24] show that the use of backup paths has been able to improve the performance of the networks. Poularakis et al. [24] propose a hybrid routing method that allows the SDN controller to define the routes but also allows the nodes to make their own decision whether to follow that route or adapt to the current state of the network. This method is intended to be used by tactical networks, in which communications among the devices at a certain geographical area are more intense than those with squads in other areas. Hence, this method is aimed at maintaining communications within reachable devices even if the communications with the SDN are no longer available. However, in a disastrous scenario, communications among isolated devices are useless, unless one of these devices can reach a BS to deliver the messages.

The method proposed by Lee [8] uses deep learning to classify the node degree in order to assign virtual routes. This method is proposed to establish communications among MANET nodes in ad hoc topology to establish communications with the nearest BS during a disaster. Accordingly, this method considers the connectivity of the nodes as the utmost important factor. However, this method does not take the energy remaining at each node into consideration when producing the routes. Hence, it is possible that this method assigns a route through a node with very low remaining energy despite the ability to use another route that can avoid exhausting this node. Moreover, this method supposes a limited movement speed of the node (0-3 km/h), which may not be the case when people are escaping certain disasters. The speed limitation is according to the use of a feed-forward neural network in the classification task, which cannot take into consideration historical data that can accurately represent the movement of the nodes.

Kadam and Srivastava [25] proposed a routing method for wireless sensor networks (WSNs) based on Q-learning, i.e., reinforcement learning. This method takes into consideration the energy remaining at each node and has been able to significantly improve the lifetime of the network, by avoiding the use of nodes with low remaining energy in routing packets, as long as alternative routes exist. However, the nodes in the WSN are considered static in this method, so that it cannot be applied in MANETs. Moreover, this routing method requires each node to make a decision for each incoming packet, based on its destination, which can in return be a power-consuming task. However, as the data in WSNs may flow from one node to another; i.e., there is no sink node that collects all data; maintaining a routing table is important for each node.

Based on this review, the methods that exist in the literature have addressed the problem of establishing temporary communications using MANETs, even by using deep learning [8] and reinforcement learning [25]. However, these methods can handle networks with limited speed of nodes, which may be a dramatic limitation to these methods as survivors tend to flee the location of a disaster as fast as possible. Moreover, despite the ability of certain methods [19, 24] to minimize the power consumption of MANETs by using the shortest possible paths, such aim can exhaust certain nodes in the MANET, which can affect the lifetime of these nodes, despite the extension of the network's lifetime. Thus, in order to maximize the chances to locate and save survivors, it is important to extend the lifetime of the MANET, taking into consideration maximizing the lifetime of each individual node in the network.

In this paper, we proposed a new SDN-based routing method for MANETs. The proposed method is aimed at providing temporary communications to devices in a disastrous area, during the disaster of the aftermath. The established temporary network uses ad hoc topology, in which each node may act as a hop in the route to deliver a certain packet initiated from another node. Mainly, the proposed method is aimed at extending the lifetime of the nodes in the network for the maximum possible lifetime by avoiding exhausting the resources of certain devices as long as alternative routes are available, in addition to emphasizing the connectivity of each node. The use of alternative routes also allows multiple sources to communicate with the BS simultaneously, as no bottleneck nodes exist in the network. Moreover, the proposed method takes into consideration the movement of the nodes in the area, regardless of their speed. Such a consideration can be achieved by collecting and using historical data, which allows predicting the future position of the node based on its speed and trajectory.

2. Materials and Methods

2.1. Overview of the Proposed Method. The proposed method requires the existence of a BS that can handle all the required applications to the devices in the network. Then, the devices in the disastrous area act as nodes to create a MANET, in which all devices aimed at connecting to the BS, which makes it the sink node. In case that all the nodes are out of the range of the BS, an additional node can be positioned in the region to close such a gap. However, the proposed method considers

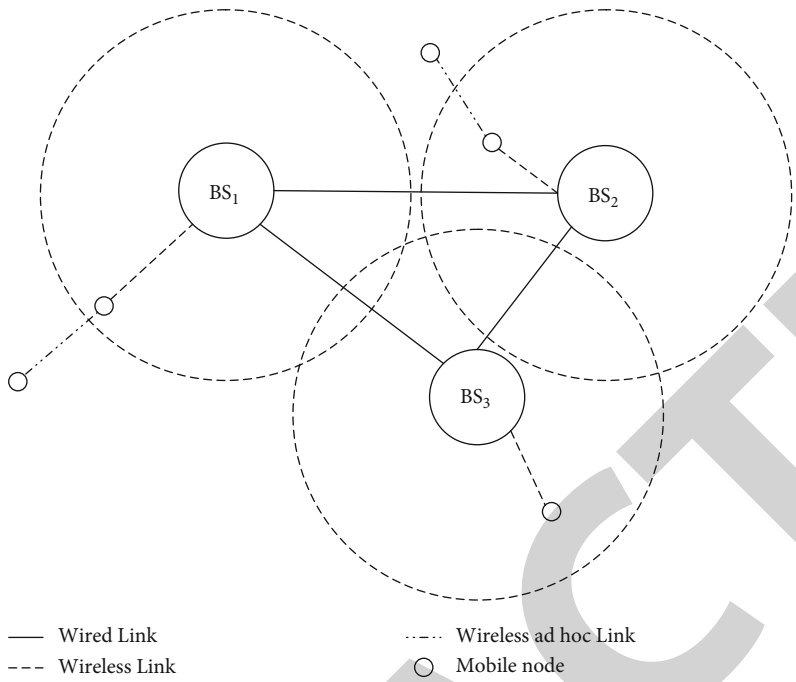


Figure 1: A sample hybrid wireless ad hoc network.

that a virtual connection exists between the BS from one side and each device from another, directly or through one or more nodes. This requirement is based on the fact that all devices must connect to the service provider in order to deliver their messages. Local networks that cannot reach BS have no importance as the devices in these networks cannot deliver their messages to the service provider. Hence, a dedicated server can be located at the BS side, as all BSs are already connected to each other using the infrastructure built prior to the disaster, in order to communicate with the nodes, collect their information, and control the packets flow in the network, as shown in Figure 1.

2.2. Network Initialization. Initially, the proposed method collects information about the nodes in the MANET in order to produce the optimal routes for each node to establish communications with the BS. This discovery uses a reverse-routing approach, which is an inefficient routing method that relies on reversing the path that a packet travels from the BS to the node. First, a "Hello" packet is sent from the BS to all the nodes that are within its range. Each node then forwards the packet to all adjacent nodes, i.e., within its range. If the packet is received by a node that already has received it from another node, the packet is dropped. Otherwise, the node sends an acknowledgment to the node that delivered the packet and appends its ID to the packet and forwards it to the nodes within its range. If the node receives an acknowledgment from the nodes is has sent the packet to, it waits for a response from that node, as the node has received the Hello packet but still discovering further nodes. Otherwise, the node sends back its position, range, and remaining energy (percentage) to the node that delivered the Hello packet. This node appends the data received from all the nodes that have acknowledged the arrival of the hello packet and adds its own information then sends it to the node the delivered Hello packet. This process is repeated until all the nodes' information is delivered to the BS, as shown in Figure 2.

2.3. Deep Reinforcement Learning. The main aim of reinforcement learning (RL) is to recognize the outcome of each action an agent can execute in a certain environment, based on the state of the agent in that environment [26]. Theoretically, the agent attempts all possible actions per each possible state in the environment, so that the action that produces the best outcome is executed during the runtime of the agent. However, especially in complex environments, it is impossible to replicate all the possible states during the training of the agent. Thus, an approximation function is required to predict the outcome of each action based on the state of the agent, so that even the outcome of the possible actions for states that the agent has not been through during the training can be predicted based on similar states that the agent has been through.

According to the outstanding performance of artificial neural networks in approximating complex functions, these networks are being used by RL agents to solve the problem of predicting outcomes of the possible actions for a certain state that is not included in the training. The state of the agent is fed to the neural network, and the outcome, denoted as Q, of each action is collected from the output. During the training, the reward that represents how good the response of the environment is for a certain action is used to train the neural network, by assigning this value to the neuron correspondent to the selected action. As the neural network has no initial knowledge about the environment, random actions are executed and used to train the neural network. As more knowledge is gained, the selected actions start to rely on the predictions of the neural

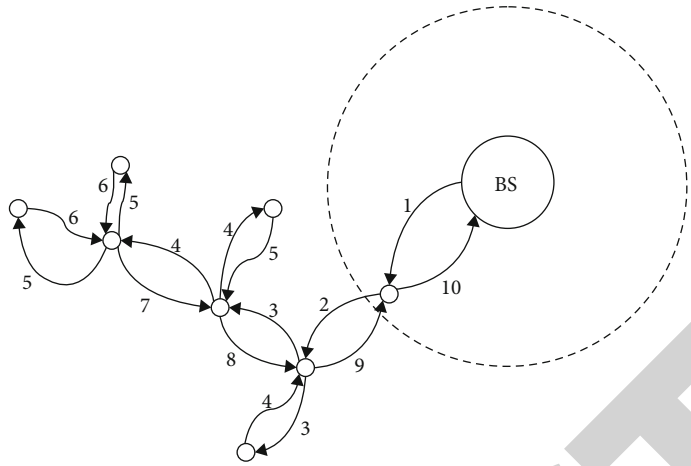


Figure 2: Propagation of the "Hello" packet and replies from the nodes.

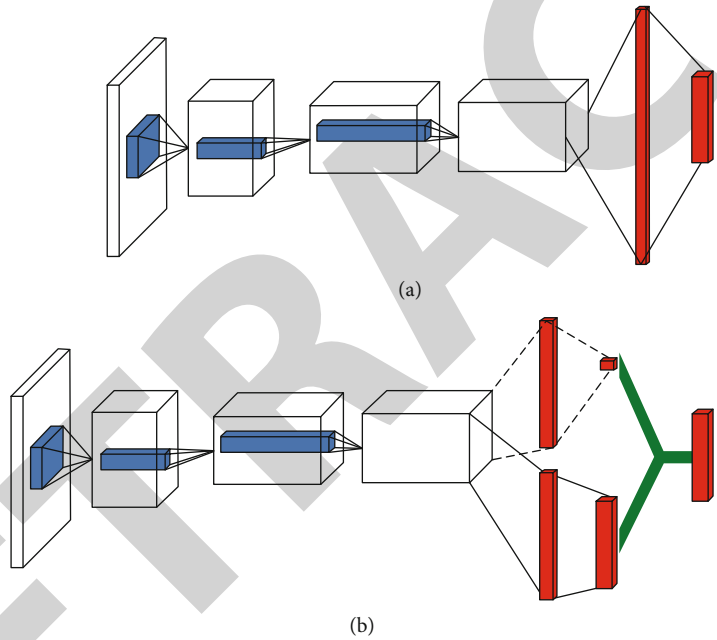


Figure 3: DQN and dueling Q-Network: (a) traditional DQN and (b) improved dueling Q-Network [27].

network, by selecting the action that the neural network predicts to have the highest reward, i.e., Q, value. The selection of the action, random or based on the neural network, is governed by a value, denoted as epsilon, that starts with a high value and is reduced as the neural network gains more knowledge. A random number generated and compared to the epsilon and a random action is selected if the random number is less than epsilon; otherwise, the action that has the highest Q value from the neural network is selected.

In some applications, such as routing, the outcome of an action cannot be recognized immediately. Alternatively, a series of actions must be conducted before evaluating the response of the environment, i.e., whether the packet is delivered or not. Thus, in such a scenario, the training of the neural network is postponed until all the required actions are executed and a reward is provided by the environment. Then, depending on the importance of each action, based on its position in the series, the reward value is discounted using a discount factor and used to train the executed action at the state it is executed in. A higher discount factor, close to one, indicates that the position of the action has less importance; i.e., earlier actions can have the same influence on the outcome as the recent ones. Such a high value is suitable for the required application, i.e., routing packets in MANET, as the selection of each hop can have equal influence on the selected route.

Traditionally, a Deep Q-Network (DQN) is used to predict the Q value of each action at a certain state, which represents the reward value expected from the environment at that state if the corresponding action is selected. Further improvement to the performance of RL is proposed by branching the last hidden layer into a group of neurons, equal to the number of possible actions, and a single neuron [27], as shown in

0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.83	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.81	0.0	0.0
0.0	0.0	0.0	0.0	0.0

0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.8	0.0

Figure 4: Sample slices of the $512 \times 512 \times 3$ energy array, which shows the rate of change in the remaining energy and the movement of the node.

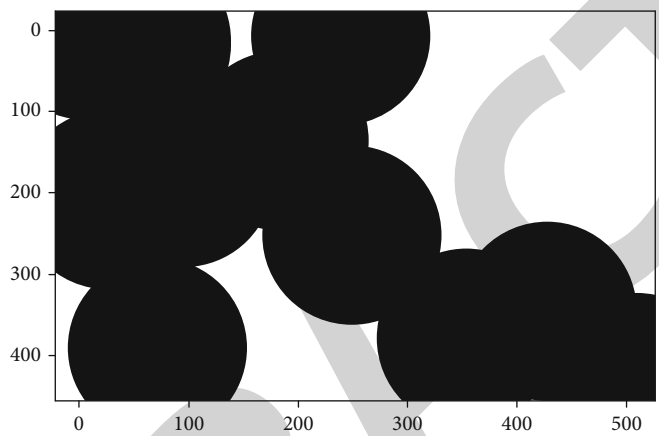


Figure 5: Illustration of the array that represents the ranges of the nodes.

Figure 3. Then, the output layer adds the value outputted by the single neuron, which represents the quality of the state the agent is currently at, to each value in the action group of neurons. Hence, the number of neurons in the output layer is equal to the number of actions and the outputted values also represent the Q value per each action. However, the use of the average of Q values to describe the state of the agent can be beneficial to evaluate being in that state, as overall, as well as the ability to evaluate each action whether to improve the state of the agent or worsen it. Thus, better training can be provided to the neural network, which also improves the predictions collected from it during runtime.

Several types of neural networks exist, which have shown different capabilities of processing different types of inputs. For instance, if each input is characterized using a one-dimensional feature vector, fully connected; i.e., dense, neural networks have shown good performance processing such inputs, in terms of complexity and quality of predictions. Recurrent Neural Networks (RNN) have shown better performance handling time-series data, in which the positions of the values influence the characteristics of the input. Moreover, Convolutional Neural Networks (CNN) have shown significantly better performance in processing three- and four-dimensional inputs, such as images. This type of neural networks convolutes two- or three-dimensional filters over the input to detect multidimensional features, based on the number of dimensions of the filters.

2.4. MANET Representation and Neural Network Implementation. An accurate and efficient representation of the environment to the agent can significantly improve its performance, by providing accurate predictions rapidly. Accordingly, only information relative to the routing task is collected from each node, which is sent by the node in a reply to the "Hello" message that is sent by the BS during the network initialization. The data required from each node are as follows:

(i) The position of the device: all modern devices are equipped with a positioning system, mostly based on the Global Positioning System (GPS), which can provide the coordinates of the device with high accuracy.

(ii) The remaining energy: all smart devices measure and display the percentage of the energy remaining in their energy sources, mostly batteries. This value is also sent to the BS in percentage.

(iii) Range: this is the distance that the wireless module on the device can achieve, as these devices are equipped with different communication modules that can achieve different ranges.

As the nodes in the MANET are moving, the position can change over time and the faster the node is moving the more change in position is detected in a fixed time window.

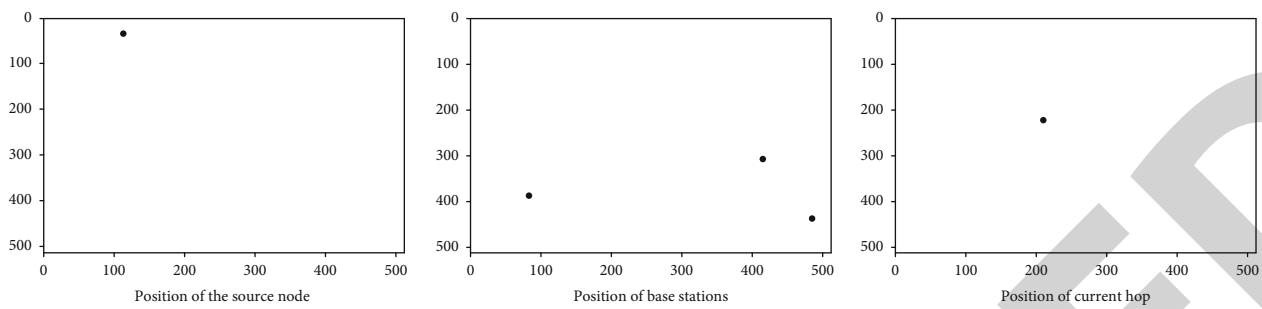


Figure 6: Positioning of the source node, base stations, and the node that the packet is currently at.

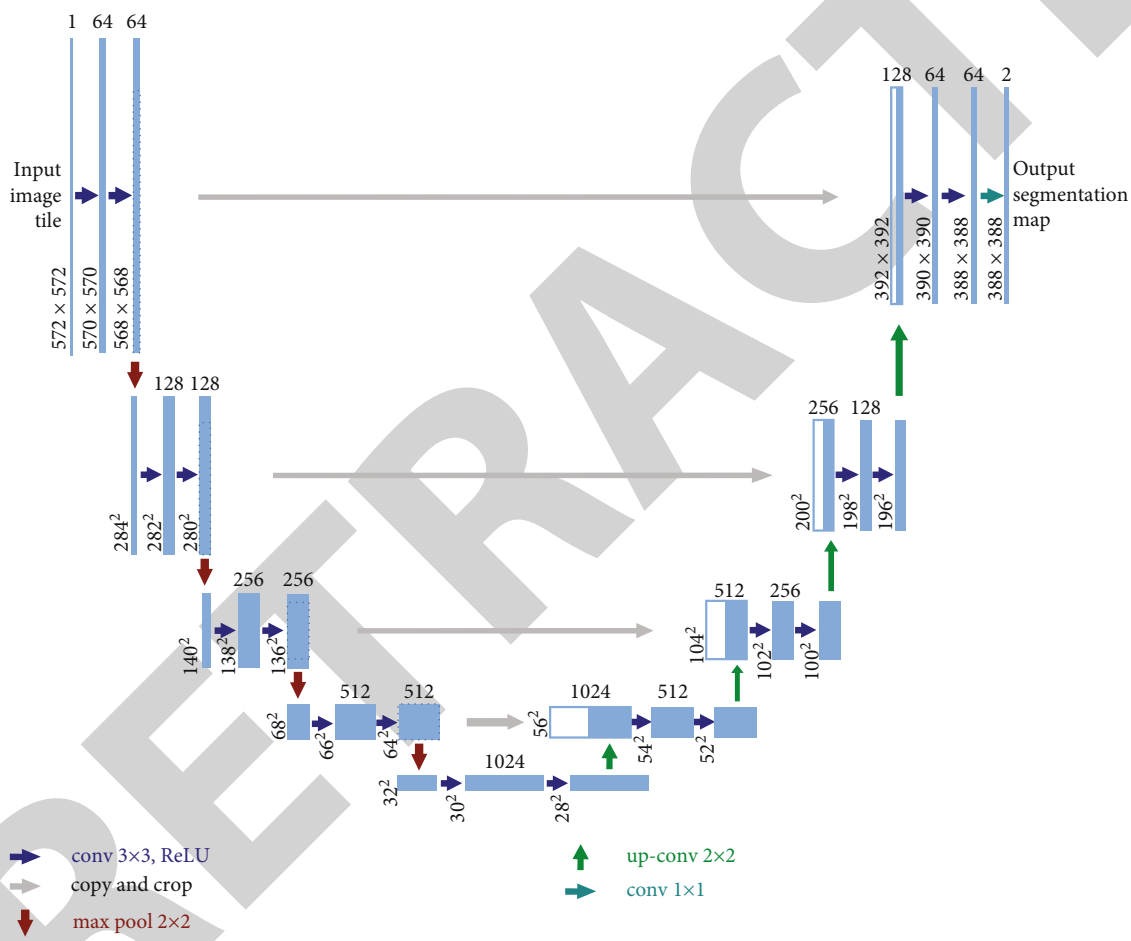


Figure 7: Structure of the U-net [28].

Moreover, the remaining energy can also be changing over time, mainly decreasing, as the device is being used. Thus, it is important to represent the changes in these two values to the implemented neural network. However, to maintain efficient representation, i.e., to avoid providing huge data that require intensive processing, a single array is used to represent both changes. This array has dimensions of $512 \times 512 \times 3$. The last three energy measurements are mapped according to the position of the node at the time the energy is measured, as shown in Figure 4, which shows that the neural network can detect the movement of the node N_1, as well as the change in its remaining energy.

Another array is also used as the input of the neural network, which represents the current position of the nodes in the MANET. To emphasize these positions and allow the neural network to exploit this input at several levels, and all the values in this array are set to zero, except the mapped positions of the nodes in the network, which are set to ones. Hence, this layer has $512 \times 512 \times 1$ dimensions. Another array is created to represent the

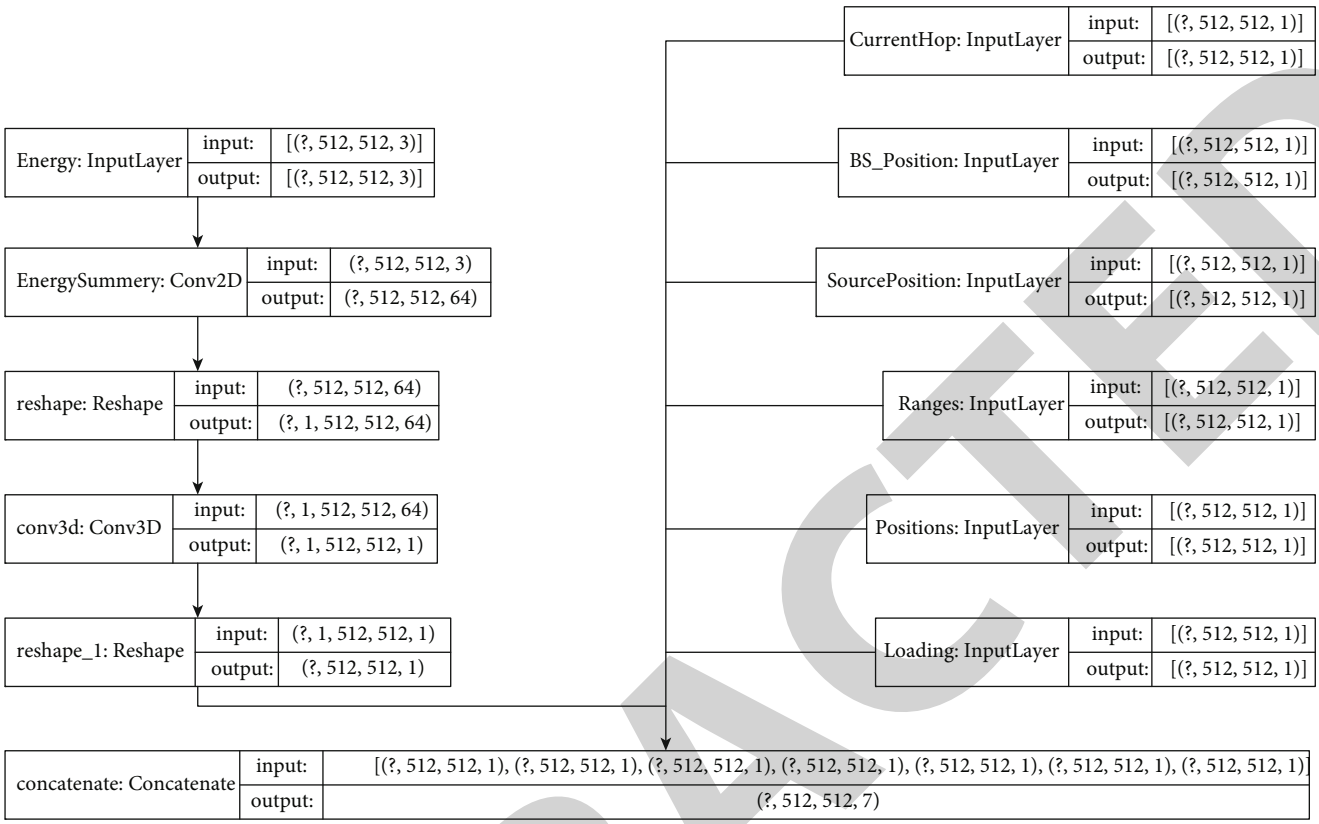


Figure 8: Processing of the energy array and concatenation with the remaining inputs.

range that each node can reach, mapped to the 512×512 dimension. This array is also filled with zeros before setting a circle filled with ones around each node, based on its mapped range. Thus, the ranges of nodes that can communicate with each other interfere as shown in Figure 5.

Additionally, three arrays are created, each has a dimension of $512 \times 512 \times 1$ and all are filled with zeros except the position of the source node in the first array, the position of the BSs, i.e. sink nodes, in the second layer and the position of the node that the packet is currently at, as the proposed method predicts the next-hop once per each prediction. Finally, an array is created and the ratio between the numbers of routes that pass through each node to the total number of nodes is calculated, up to the current routing step. Figure 6 summarizes these three arrays for the current routing task of the sample MANET shown in the same figure.

Inspired by the U-net neural network [28], the neural network implemented for the proposed method uses a similar approach, in which the input of the neural network and the outputs of certain layers are appended to the inputs of the layers closer to the output layer, as shown in Figure 7. This approach allows the proposed neural network to consider the characteristics of the MANET, especially the exact positions of the nodes, in the outputted values.

The neural network that is implemented for the proposed method consists of three main parts. The first part processes the $512 \times 512 \times 3$ array that represents the energy change in energy and position, extracts the required features, and produces a $512 \times 512 \times 1$ array, as shown in Figure 8. Two convolutional layers, one two-dimensional and one three-dimensional layers are used, where the three-dimensional layer uses a filter with a size of $1 \times 1 \times 64$ to summarize the 64 features detected by the 64 two-dimensional filters to produce the required shape. The outputted array is then concatenated with the remaining inputs to produce a single $512 \times 512 \times 6$ array.

The output of the concatenation layer is then processed by a CNN similar to the structure of the standard U-net but uses the "same" padding techniques, in which the dimension of the output array is identical to the input one. Such output is achieved by padding the image with additional zeros, i.e., extending the dimension of the input array, according to the size of the filters in the convolutional layer. Figure 9 summarizes the structure of the implemented neural network.

The output of this part of the neural network is then forwarded to the network shown in Figure 10. The output of this part, which is the output of the entire neural network, consists of two components, a two-dimensional array "Hop" and a single value "State." The value outputted by the "State" neuron represents the overall quality of the agent when being at that state. For instance, packets currently in nodes closer to the BS are expected to have higher "State" value. Moreover, the values in the output of the "Hop" layer represent the advantage of forwarding the packet to any position in the

region. However, according to the use of the positions array in the input and by considering avoiding outputting values at positions that do not contain a node, the output that has the maximum value is expected to be located on a position of a node. Nevertheless, the proposed method selects the node closest to the position that has the highest output to forward the packet to.

2.5. Training the CNN and Routing the MANET. As illustrated earlier, RL has the ability to postpone the training process until all the required actions are executed and a reward value can be calculated. Thus, the proposed neural network is used to route the packets and then a reward is calculated based on the performance of the MANET using the selected routes. To define the route that the packets must follow from a certain node to the BS, the information received with the "Hello" message is fed to the neural network. Initially, the position of the value one in the array that represents the current hop the packet is at is identical to the array that represents the source packet. The position of the node nearest to the maximum value in the output of the Hop layer is then selected as the next hop. The position of the value one is then mapped according to the newly selected node and fed to the neural network until a node that is directly connected to the BS is reached. When the route is defined, the information is sent to each node involved in the selected path, indicating that any packet incoming from the source node must be forwarded to the designated node.

When a node loses connection to one of the other nodes that exist in its routing table, this node instantly sends the same set of information to the nearest node that also exists in its routing table, which is used to route packets incoming from other nodes. If such a node does not exist, it sends the information to the nearest available node, in order to update the BS with the new position and to request a new route. Additionally, the proposed method also sends periodic "Hello" messages in order to discover any newly connected devices and update the positions of the existing nodes. This allows updating and producing the required $512 \times 512 \times 3$ array that reflects the changes in the energy and position.

When a node is selected based on the predictions of the neural network, the selection is validated against the path defined so far. If the node is found to exist in the path, the neural network is trained immediately with a reward value of -1, i.e., a punishment, and a new prediction is collected. This ensured avoiding producing loops in the network, which can dramatically reduce the performance of the network. As the aim of the proposed method is to extend the lifetime of each device in the network and maintain their connectivity, the reward value is calculated based on these factors, as shown in Equation (1). The exact same network is routed using the standard Ad hoc On-Demand Distance Vector (AODV) protocol, and the performance of the MANET is measured based on the lifetime of the device that dies first and the Packet Delivery Rate (PDR). According to the formula, a reward of one is used to train the neural network if it achieved identical performance to the use of AODV. Any additional improvement in the performance increases the reward value, while any lower performance reduces the reward value. Accordingly, the proposed method learns to improve the overall performance of the MANET, without emphasizing one factor over the other, which is the

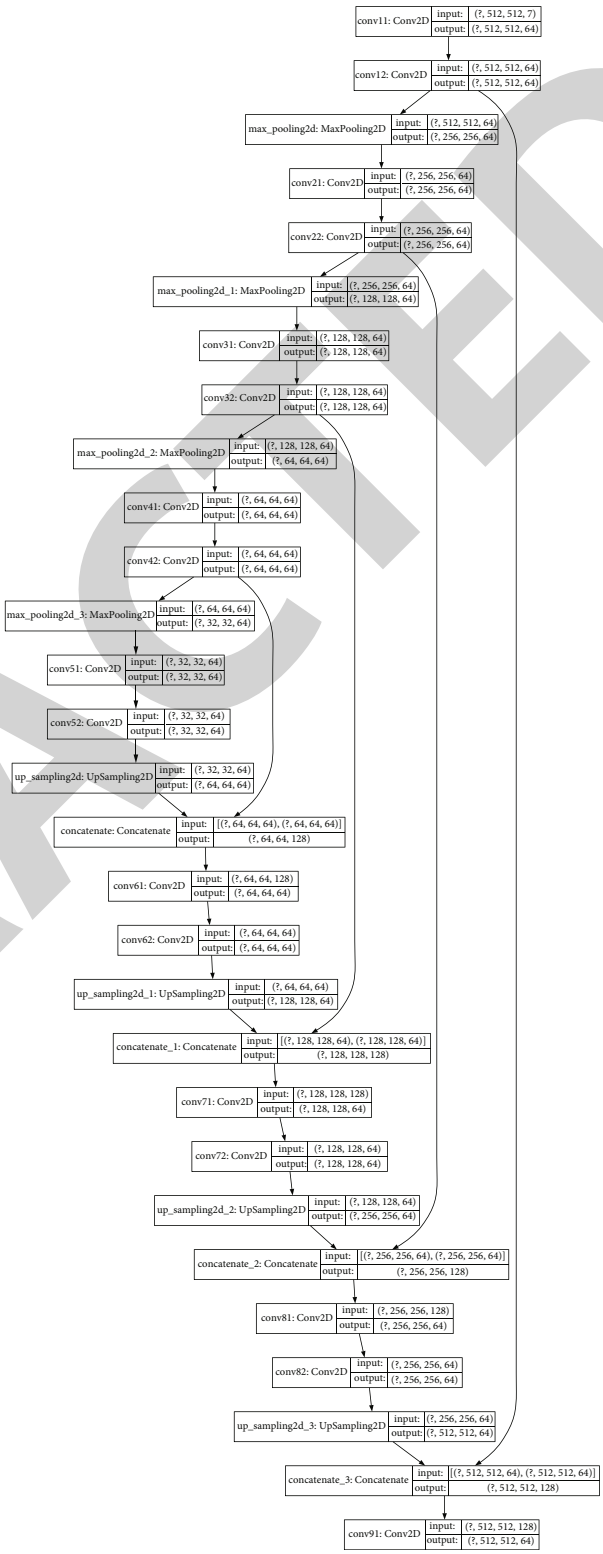


Figure 9: Structure of the part of the implemented CNN similar to the U-net.

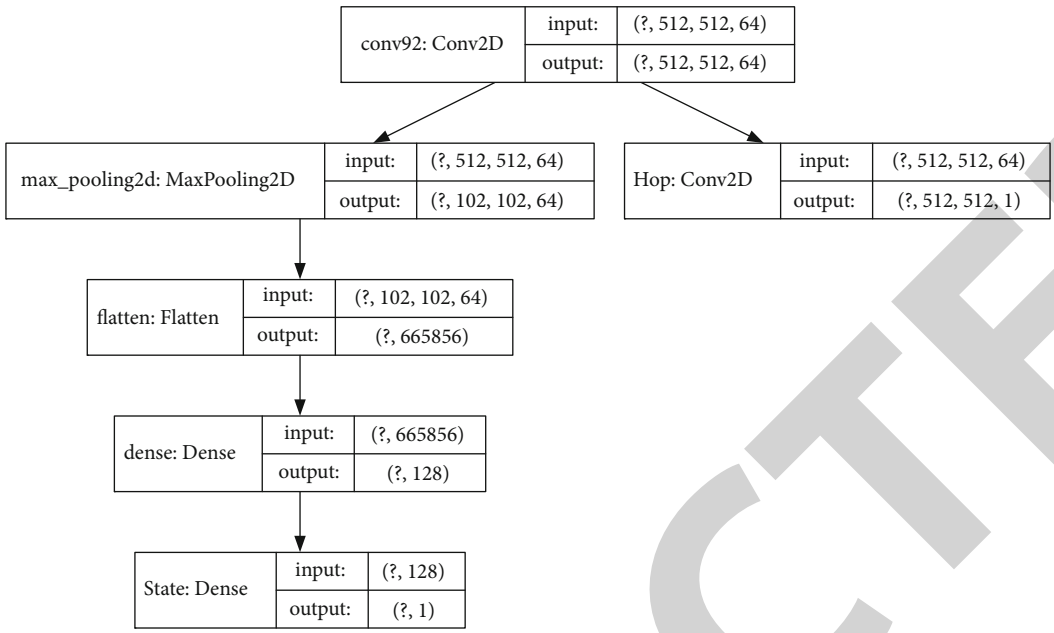


Figure 10: Processing the output of the U-net-similar part to achieve dueling Q-learning.

reason behind multiplying each performance measure by the other from the AODV-based performance.

$$R = \frac{L_{\mathrm{AODV}} \times \mathrm{PDR}_{\mathrm{RL}} + L_{\mathrm{RL}} \times \mathrm{PDR}_{\mathrm{AODV}}}{L_{\mathrm{AODV}} \times \mathrm{PDR}_{\mathrm{AODV}}}, \quad (1)$$

where the L_{AODV} and $\mathrm{PDR}_{\mathrm{AODV}}$ are the lifetime and PDR of the AODV-based MANET and L_{RL} and $\mathrm{PDR}_{\mathrm{RL}}$ are the lifetime and PDR for the MANET using the proposed RL-based routing method.

3. Results and Discussion

A model is implemented for the proposed method using Python programming language, in which the Tensorflow library is used to implement and operate the neural network. The Sim2Net (https://pypi.org/project/sim2net/) library is used to simulate the wireless sensor networks and interact with the proposed model that is responsible for routing the packets in the network. This implementation allows direct interaction between the simulated wireless network and the implemented rout selection model; i.e., the node vectors are directly collected from the network and processed by the model, in order to recognize the optimal route. Mainly, the performance of the proposed method is validated by comparing the performance of the network when the proposed method and three of the standard and widely used routing protocols in MANETs are used. These protocols are the Ad hoc On-Demand Distance Vector (AODV) [29], Optimized Link State Routing (OSLR) [30], and Zone Routing Protocol (ZRP) [31]. The main parameters of the simulated network are shown in Table 1.

Table 1: Parameters of the experimental setup.

Parameter (unit)	Value
Simulation time (sec)	300
Pause time (sec)	60
Data type	Constant Bit Rate (CBR)
Size of each packet (byte)	512
Network area (m^2)	3000×3000
Range of each node (m)	MN: 375, BS: 375
Number of nodes	MN: [50, 100, 150], BS: [1–5]
Energy (joule)	MN: 1, BS: unlimited
Packet transmission energy (joule)	5×10^{-9}
Nodes movement	Random walk
Maximum node speed (km/h)	10

3.1. Packet Delivery Rate (PDR). As illustrated in the experimental setup, the performance of the proposed method is evaluated using fifteen different scenarios, according to the existence of three numbers of mobile nodes and five numbers of base stations; i.e., per each number of base stations, the three numbers of mobile nodes are evaluated. Per each scenario, the ratio between the numbers of packets that are delivered to the base stations to the total number of packets initiated by the mobile nodes is measured for the proposed method, as shown in Figure 11. These results show that the proposed method has been able to maintain similar performance when the network has a high density of nodes. Such behavior is according to the existence of several alternative routes that the proposed method can use to deliver the packet. Additionally, the probability of delivering a packet is also increased when the number of base stations in the region is increased. The existence of these stations also allows

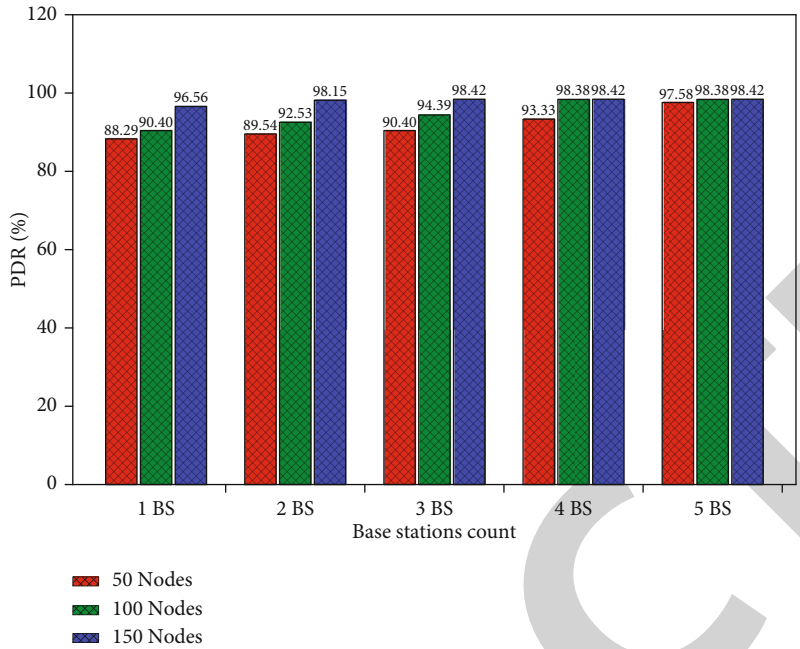


Figure 11: PDR versus the number of base stations in the region.

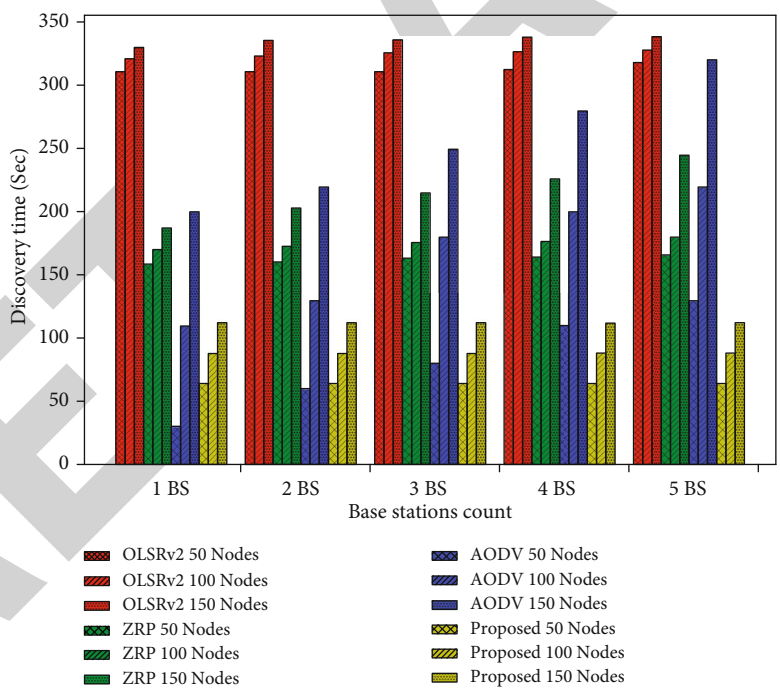


Figure 12: Influence of the number of base stations and mobile nodes on the route discovery time.

more flexibility to deliver the packet, by providing more possible routes.

3.2. Route Discovery Time. As each node waits for the remaining nodes that are in its range and have not received the "Hello" message from the base station to discover their network and reply back to that node, the arrival of the nodes vectors from a certain node indicates that all the network behind that node is discovered. Accordingly, when the base station receives the node vectors from all the nodes that are directly connected to it, network discovery can be considered complete and the routing table generation process can be initiated. As shown in Figure 12, in addition to the lower time required by the proposed method to discover and assign a

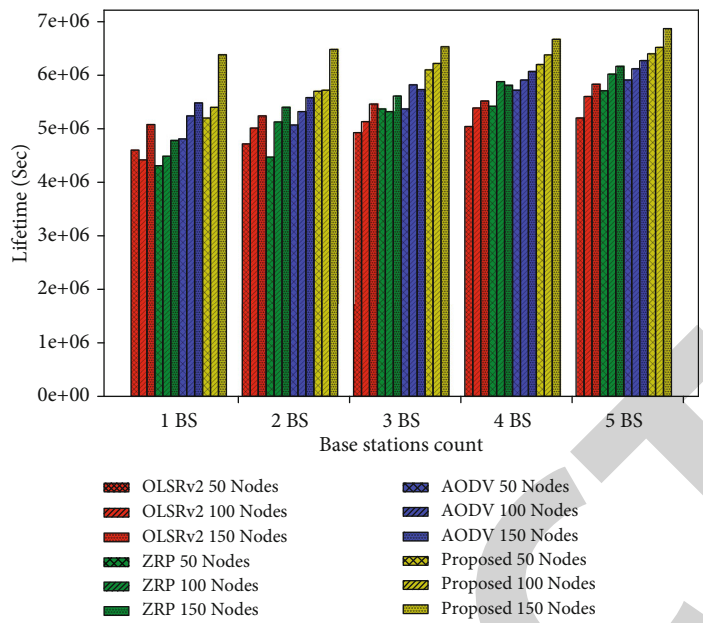


Figure 13: Lifetime of the wireless network versus the number of base stations.

route to each node in the network, the relation between the time and number of nodes is almost linear and irrelative to the number of base stations. Such a relationship is according to the fact that the input to the neural network contains all the information of the network, so that the neural network requires no additional processing if an additional base station exists. However, as increasing the number of nodes in the network increases the number of routes to be discovered, the proposed method requires additional time to discover these routes. Similar behavior is noticed in the OLSRv2 protocol, despite the significantly higher discovery time, which is the highest among all protocols.

In contrast, the time required by the AODV protocol to discover the routes is exponentially dependent on the number of nodes in the network and the number of base stations. Such an increase in time is according to the need for route discovery per every query, so that more computations are required when more nodes exist in the network, according to the additional traffic generated by the nodes. Similarly, the number of zones is also increased when the number of nodes is increased while using the ZRP. Hence, the time required to discover the routes increases significantly when additional nodes are added to the network. Thus, in terms of route discovery time, the proposed method has been able to achieve significantly better performance, especially by maintaining similar times even when the number of base stations is increased; i.e., more possible routes exist in the network.

3.3. Network Lifetime. Extending the lifetime of each node in the network is the main aim of the proposed method, according to the significant importance of this extension to maintain communications during or after the disaster for as long as possible. Accordingly, this experiment evaluates the lifetime of these devices by monitoring the energy of each node. When the energy of one of the nodes becomes less than the energy required to transmit a data packet, the node is considered dead. The lifetime of the network is measured between the initialization of the network and the loss of the first node. Accordingly, the simulation is not governed by any interval and continued until one of the nodes exhausts all its energy. As shown in Figure 13, the proposed method has been able to extend the lifetime of the nodes, which indicates that the loading has been balanced among the nodes, to avoid exhausting the resources of certain nodes. However, the gap between the lifetimes of the networks with different numbers of nodes is larger at a lower number of base stations. The lack of alternative routes in such cases is the main reason behind forcing the proposed method to exhaust certain nodes that are in the route of the base stations to deliver the packets. Providing alternative routes, by increasing the number of nodes or base stations, has been able to significantly improve the performance of the network when using the proposed routing method.

4. Conclusions

Establishing communications during a disaster or the aftermath is a vital feature to locate and save any survivors. However, during such a disaster, the infrastructure that is required to establish wireless communications, such as cellular, is most probably affected by a disaster. The establishment of such an infrastructure is a time-consuming process, which cannot normally be accomplished during the search and rescue operations. Thus, MANETs are being used to establish temporary communications with the survivors, through their everyday digital devices. With the limited energy available in these devices and the need for those devices to route the packets to the BS, it is important to improve the efficiency of energy usage, so that the overall lifetime of the MANET as well as the lifetime of each node is extended to maximum, to improve the chances of locating and saving the survivors.

A new routing method is proposed in this paper to efficiently connect the nodes in a MANET to an operating BS to establish communications with survivors during the occurrence or the aftermath of a disaster. The proposed method is aimed at increasing the lifetime of the network by balancing the loading on the nodes and avoiding exhausting the ones with limited remaining energy. This aim is achieved by using RL, in which a neural network considers the status of the network to predict the route for a node to the BS. The use of the proposed method can extend the lifetime of the devices, which can significantly increase the chances of saving lives. In addition to the better lifetime, the proposed method has been able to achieve similar performance, compared to existing routing protocols, in terms of the PDR and route discovery time. Moreover, as the nodes in the wireless network are mobile with relatively high speeds, the proposed method can provide efficient communications to the search and rescue teams during the search for and extraction of survivors.

In a future work, the ability to use Generative Adversarial Networks (GANs) to generate the route directly based on the inputted state of the MANET is according to the ability of these neural networks in producing complex multidimensional output. The routes that are going to be used to train the neural network are going to be generated using the method proposed in this study. Hence, the same routes can be generated while significantly reducing the route generation time.

Data Availability

There have been no data collected during this study. All evluation was conducted by simulating the networks. Hence, the same simulation environment can be replicated based on the parameters described in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Javed, M. A. Shahid, M. Sharif, and M. Yasmin, "Energy consumption in mobile phones," *International Journal of Computer Network & Information Security*, vol. 9, no. 12, pp. 18–28, 2017.

[2] R. Fojtik, "The use of mobile devices in education," *New Trends and Issues Proceedings on Humanities and Social Sciences*, vol. 3, no. 3, pp. 41–47, 2017.

[3] T. H. Aldhyani and M. R. Joshi, "An integrated model for prediction of loading packets in network traffic," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '16*, pp. 1–6, Udaipur, India, 2016.

[4] T. Wang, P. Li, X. Wang, Y. Wang, T. Guo, and Y. Cao, "A comprehensive survey on mobile data offloading in heterogeneous network," *Wireless Networks*, vol. 25, no. 2, pp. 573–584, 2019.

[5] Y. Huo, X. Dong, W. Xu, and M. Yuen, "Cellular and WiFi co-design for 5G user equipment," in *2018 IEEE 5G World Forum (5GWF)*, pp. 256–261, Silicon Valley, CA, USA, 2018.

[6] M. Joshi and T. H. Hadi, "A review of network traffic analysis and prediction techniques," 2015, https://arxiv.org/abs/1507.05722.

[7] F. Anthony, M. Gabriel, and B. Shao, "Open source cellular technologies for cost effective cellular connectivity in rural areas," *International Journal of Computer Applications*, vol. 146, 2016.

[8] Y. Lee, "Classification of node degree based on deep learning and routing method applied for virtual route assignment," *Ad Hoc Networks*, vol. 58, pp. 70–85, 2017.

[9] R. Verma, A. Prakash, N. Tyagi, and R. Tripathi, "A hybrid wireless ad-hoc network model for critical services," in *2010 Sixth International conference on Wireless Communication and Sensor Networks*, pp. 1–6, Allahabad, India, 2010.

[10] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, 2004.

[11] Z. Li and Z. Yang, "A wireless ad hoc overlay networks indexing mechanism based on petri nets theory," in *2010 International Conference on Computational Aspects of Social Networks*, pp. 474–478, Taiyuan, China, 2010.

[12] Z. Li and A. Ji-Yu, "A hybrid routing algorithm based on overlay network for wireless ad hoc network," in *2012 International Conference on Industrial Control and Electronics Engineering*, pp. 1138–1141, Xi'an, China, 2012.

[13] Y. Zhong, "Cross layer power efficient cooperative diversity scheme for wireless hybrid ad hoc networks," in *2010 Second International Conference on Communication Software and Networks*, pp. 502–506, Singapore, Singapore, 2010.

[14] A. Antoo and R. Koshy, "Enhanced hybrid drive scheme for power saving in wireless ad-hoc networks," *2013 International Conference on Control Communication and Computing (ICCC)*, 2013, pp. 221–226, Thiruvananthapuram, India, 2013.

[15] N. A. M. Maung and M. Kawai, "Hybrid RSS-SOM localization scheme for wireless ad hoc and sensor networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, Sydney, NSW, Australia, 2012.

[16] N. Zaman, T. J. Low, and T. Alghamdi, "Enhancing routing energy efficiency of wireless sensor networks," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pp. 587–595, Seoul, South Korea, 2015.

[17] D. Choudhury, D. Kar, K. R. Biswas, and H. N. Saha, "Energy efficient routing in mobile ad-hoc networks," in *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, pp. 1–7, Vancouver, BC, Canada, 2015.

[18] T. H. Aldhyani, M. Alrasheedi, A. A. Alqarni, M. Y. Alzahrani, and A. M. Bamhdi, "Intelligent hybrid model to enhance time series models for predicting network traffic," *IEEE Access*, vol. 8, pp. 130431–130451, 2020.

[19] J. I. Naser and A. J. Kadhim, "Multicast routing strategy for SDN-cluster based MANET," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 10, 2020.

[20] K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN-enabled tactical ad hoc networks: extending programmable control to the edge," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 132–138, 2018.

[21] K. Liu, J. K. Ng, V. C. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9819236, 1 page
https://doi.org/10.1155/2023/9819236

Retraction

Retracted: Adaptive Anomaly Detection Framework Model Objects in Cyberspace

Applied Bionics and Biomechanics

Received 15 August 2023; Accepted 15 August 2023; Published 16 August 2023

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] H. Alkahtani, T. H. H. Aldhyani, and M. Al-Yaari, "Adaptive Anomaly Detection Framework Model Objects in Cyberspace," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 6660489, 14 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2020, Article ID 6660489, 14 pages
https://doi.org/10.1155/2020/6660489

Research Article

Adaptive Anomaly Detection Framework Model Objects in Cyberspace

Hasan Alkahtani,[1] Theyazn H. H. Aldhyani,[2] and Mohammed Al-Yaari[3]

[1]*College of Computer Science and Information Technology, King Faisal University, P.O. Box 4000, Al-Ahsa 31982, Saudi Arabia*
[2]*Community College of Abqaiq, King Faisal University, P.O. Box 4000, Al-Ahsa 31982, Saudi Arabia*
[3]*Chemical Engineering Department, King Faisal University, P.O. Box 380, Al-Ahsa 31982, Saudi Arabia*

Correspondence should be addressed to Theyazn H. H. Aldhyani; taldhyani@kfu.edu.sa

Received 27 October 2020; Revised 23 November 2020; Accepted 28 November 2020; Published 10 December 2020

Academic Editor: Mohammed Yahya Alzahrani

Telecommunication has registered strong and rapid growth in the past decade. Accordingly, the monitoring of computers and networks is too complicated for network administrators. Hence, network security represents one of the biggest serious challenges that can be faced by network security communities. Taking into consideration the fact that e-banking, e-commerce, and business data will be shared on the computer network, these data may face a threat from intrusion. The purpose of this research is to propose a methodology that will lead to a high level and sustainable protection against cyberattacks. In particular, an adaptive anomaly detection framework model was developed using deep and machine learning algorithms to manage automatically-configured application-level firewalls. The standard network datasets were used to evaluate the proposed model which is designed for improving the cybersecurity system. The deep learning based on Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) and machine learning algorithms namely Support Vector Machine (SVM), K-Nearest Neighbor (K-NN) algorithms were implemented to classify the Denial-of-Service attack (DoS) and Distributed Denial-of-Service (DDoS) attacks. The information gain method was applied to select the relevant features from the network dataset. These network features were significant to improve the classification algorithm. The system was used to classify DoS and DDoS attacks in four stand datasets namely KDD cup 199, NSL-KDD, ISCX, and ICI-ID2017. The empirical results indicate that the deep learning based on the LSTM-RNN algorithm has obtained the highest accuracy. The proposed system based on the LSTM-RNN algorithm produced the highest testing accuracy rate of 99.51% and 99.91% with respect to KDD Cup'99, NSL-KDD, ISCX, and ICI-Id2017 datasets, respectively. A comparative result analysis between the machine learning algorithms, namely SVM and KNN, and the deep learning algorithms based on the LSTM-RNN model is presented. Finally, it is concluded that the LSTM-RNN model is efficient and effective to improve the cybersecurity system for detecting anomaly-based cybersecurity.

1. Introduction

The end of the Cold War has led to many challenges and threats that the international community has never seen before, known as asymmetric or asymmetric cross-border threats that recognize neither borders and national sovereignty nor the idea of a nation-state. These threats led to shifts in the field of security and strategic studies as well as at the level of political practice. The explosion of the information revolution and the entry of the digital age, especially in the 21st century resulted in many repercussions manifested in the emergence of cyber threats and crimes. Such threats are regarded to be a major challenge to the national as well as international security making cyberspace as the fifth area of war after land, sea, air, and space. These repercussions entailed the need for security guarantees within this digital environment which led to the emergence of cybersecurity as a new dimension within the field of security studies that has acquired the interests of many researchers in this area. Having said that, we need to understand what cybersecurity is as a new variable in international relations. The task of adjusting concepts and terminology is a challenge facing

various researchers and scholars in different disciplines because of the problems it poses making it difficult to agree on clear, comprehensive, and unified definitions among members of the scientific community. Cybersecurity is one of the complex concepts that have been presented by many different definitions. In this sense, researchers in the field of international relations and other subfields in security and strategic studies are increasingly focusing on the impact of technology on national and international security, including related concepts such as power and sovereignty, global governance, and securitization. As a matter of fact, the expansion of the internet has reshaped traditional forms and norms of the international force that are working extensively to enter a new era of geopolitics.

Cybersecurity is the technical, regulatory, and administrative means that are used to prevent unauthorized use, abuse, and recovery of electronic information over communication systems and the information they contain. In addition, the aim of cybersecurity is to ensure the availability and continuity of the work of information systems and enhance the protection, confidentiality, and privacy of personal data by all measures. Cybersecurity is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks, also known as information security or cyberwarfare [1]. One of the major challenges of network traffic analysis is intrusion detection. The Intrusion Detection Systems (IDS) are designed to find out malicious activities that attempt to compromise the confidentiality, integrity, and assurance of computer systems. The intrusion detection system has become the most widely used security technology [2]. Certainly, intrusion detection systems have become critical components in network security. Consequently, two factors need to be considered to guarantee an effective performance of IDS. First, the intrusion detection should deliver consistent detection results. The detection method should be effective in discovering intrusions since poor detection performance ruins the trustworthiness of the IDS. Second, the IDS should be able to survive in hostile environments (i.e., under attacks). The main challenge for IDS is to maintain high detection accuracy. As new intrusions increase, IDS tools are becoming incapable of protecting computers and applications. Consequently, a robust approach that is able to discover new attacks is necessary for building reliable IDS. Machine learning provides insights for identifying novel attacks. Machine learning enhances the capability of a machine that automatically improves its performance through learning from experience [3–7]. Machine learning techniques are employed to study normal computer activities and identify anomalous behaviors that deviate from the normal as intrusions. Even though these anomalies-based IDSs are able to detect novel attacks, most of them suffer from misclassification.

The algorithms of machine learning have played a crucial role in the area of cybersecurity. Deep learning networks performed incredibly in solving problems from a wide variety of fields. Furthermore, as can be observed, it gained a significant increase in its usage for artificial intelligence (AI) and unsupervised challenges [8]. The artificial neural network is part of machine-learning simulating the processes of the human brain. Deep learning refers to simple building blocks that are organized in a complex hierarchical order. These building blocks have the ability to solve high-level problems. Recently, the applications of deep learning methods are oriented towards various uses, especially cybersecurity [9–11]. In recent years, deep learning has evolved as an important research area in machine learning. Viewing it as a special architecture of deep learning, DNN has proved to be of effective applications, particularly in different tasks of pattern recognition such as visual classification and speech recognition. Having said that, recent studies from 2013 onwards have shown that DNN is prone to serious attacks [12, 13]. An example that shows the vulnerability of DNN is image classification where it extracts only few features which hinder its performance, especially with images that have nuanced differences. Thus, it is easy for attackers to evade anomaly detection. Szegedy et al. [12] suggested using a slightly blurred image to trick the pretrained DNN. This was followed by many works that suggested impersonation models for the sake of attacking DNNs and proposing corresponding intelligent systems (e.g., face recognition, speech recognition, and autonomous driving) [14–17]. Fiore et al. [18] explored the use of a semisupervised model for network intrusion detection. They used a discriminative restricted Boltzmann machine to combine the expressive power of generative models with good classification abilities. They employed the KDD Cup'99 dataset with a set of 41 features and 97,278 instances. Salama et al. [19] paired the Restricted Boltzmann Machine (RBM) with Support Vector Machine (SVM) to build a traffic intrusion detection system.

The dataset used in the study was NSL-KDD, and its training set had a 22 training attack types along with 17 types in the testing set. The study demonstrated that such combination showed better performance in classification compared to the classification of support vector machine alone. Alrawashdeh and Purdy [20] implemented the RBM with a deep belief network for anomaly detection. They employed the KDD Cup'99 dataset which consisted of 494,021 training records and 311,029 testing records. They carried out the deep learning architecture in C++ in Microsoft Visual Studio 2013. Their study demonstrated that the use of a Restricted Boltzmann Machine improved the accuracy of classifying attacks to 92%. The article showed better results than those implemented by Salama et al. [19] in both accuracy and speed detection. Aldwairi et al. tested the effect of appropriate features on the performance of Restricted Boltzmann Machines and compared its performance with conventional machine learning algorithms [21]. Their study demonstrated that Restricted Boltzmann Machines can be trained to accurately classify and distinguish between normal and anomalous Net Flow traffic. The study employed the ISCX dataset [22] and applied it in the intrusion detection area. They employed a restricted Boltzmann machine in which its deep neural network training is made of two steps: (i) training restricted Boltzmann machine, and (ii) tuning the parameters of the whole RBM. The results demonstrated that using the restricted Boltzmann machine on the KDD Cup'99 dataset of deep belief network outperformed the performance of a support vector machine and an artificial neural network. Fu

et al. [23] improved a framework for detecting fundamental patterns of deceptive behaviors, such as the detection of fake credit cards. The framework is based on the convolutional neural network. The convolutional neural network was also implemented by Zhang et al. [24]. They used the data of the commercial bank B2C online transactions. The transaction data of one month were classified into training and testing datasets. The results showed an accuracy rate of 91% and a recall rate of 94%. Compared with the results of Fu et al. [23], the results of Zhang et al. showed an increase in the accuracy and recall rate with 26% and 2%, respectively. Nasr et al. designed a particular system that is called DeepCorr which is based on deep learning architecture to learn a flow correlation function tailored to Tor's complex network. In their experiment, DeepCorr achieved the best performance with a learning rate of 0.0001, and for a false-positive rate of 10-3, achieved a true positive rate close to 0.8. Zhang et al. designed an anomaly traffic detection model leveraging two layers of the neural network [25]. The first layer is made of the improved LetNet-5 convolutional neural network with the function to extract the spatial features. The second layer makes use of long short-term memory with the function to extract temporal features. On the CIC-IDS2017 dataset, the performance exceeded 94%. Their suggested system achieves better accuracy, F1-measure, and higher recall rate compared to other machine learning algorithms. Thus, the framework proposed by Zheng et al. is regarded as light-weight with the ability to detect new attacks and classify encrypted traffic. Yu et al. applied a convolutional autoencoder to test the efficiency of the detection system on network intrusion [26]. Two datasets were employed: they are the CTU-UNB and Contagio-CTU-UNB. To develop the neural network model, the Theano tool was used. The learning rates were 0.001, and the pertaining and fine-tuning process was 0.1. Using the Contagio-CTU-UNB dataset, the classification tasks included 6 class and 8 class with the ROC curve value 0.99. Moreover, the study achieved a high rate of accuracy (i.e., 99.59%) in the binary classification. With the use of the deep belief network and probabilistic neural network, Zhao et al. [27] proposed an IDS framework. In their study, they used the KDD Cup'99 dataset for monitoring the efficiency of the intrusion detection model. The dataset was divided into 10% training and 10% testing dataset. The results demonstrated that the adopted method outperformed the other three models: (i) the traditional probabilistic neural network, (ii) principal component analysis with traditional probabilistic neural network, and (iii) optimized deep belief network with probabilistic neural network. Zhang et al. [28] attempted to design a self-adaptive model to modify the structure of the network enabling it to face different types of attacks. Thus, they presented an intrusion detection model based on both improved Genetic Algorithm (GA) and Deep Belief Network (DBN). DBN module is mainly divided into two steps in the training phase: (i) each RBM is trained separately, and (ii) the last layer of the DBN is set as the BP neural network. Using the NSL-KDD dataset, the performance of the proposed model showed a high detection rate of 99%. The main advantage of the intrusion detection system is recognizing malicious cyberattacks on a network. Besides that, the intrusion detection system can help in monitoring and evaluating the activities in a network or computer system [29, 30]. The area of cybersecurity has gained much attention from many researchers where they focused on developing systems that are able to detect security risks and prevent attacks. One of the well-known cybersecurity systems is the signature-based network intrusion detection system [31] which works by looking for specific patterns, for example, byte sequences in network traffic. This system has gained commercial success with widespread of applications. Another system which is regarded as superior to signature based is the anomaly-based system. This system has the ability to detect unknown attacks [32, 33]; it is based on machine learning which creates a model of trustworthy activity and compares new behavior against this model [34–36]. A shortcoming of this approach is that it may raise a false-positive alarm for previously unknown legitimate activity and classifying it as a malicious [32]. Therefore, developing intrusion systems with the ability to minimize the false-positive rates must be of primary concern. Hence, such issues can be solved by considering detection approaches based on machine learning. Machine learning is regarded as a discipline within artificial intelligence; other disciplines within artificial intelligence are computational statistics, data mining, and data science. Machine learning is based on the idea that computers can learn from data [36, 37]. It is closely related to mathematical theories, methods, statistical analysis, optimization, and many application areas in the field.

Therefore, machine learning plays a primary role in the area of cybersecurity where building an intelligent security model for predictions is based on understanding the raw security data. It is known that the association analysis is considered in machine learning techniques for building rule-based intelligent systems [38–40]. However, in the current study, the main focus is on the learning techniques of classification [35, 41], which leverages a given training dataset for the sake of building a predictive system. For example, building a data-driven predictive model requires many techniques like naive Bayes classifier, support vector machines, k-nearest neighbors, logistic sigmoid function, and rule-based classification [35, 36]. Plenty of studies focused on detecting intrusions or cyberattacks and have used the abovementioned machine learning classification techniques. Li et al. [42] employed the hyperplane-based support vector machine classifier to classify identified attack categories, for example, DoS, Probe or Scan, U2R, R2L, and normal traffic leveraging the highly popular KDD'99 Cup dataset. Amiri et al. created faster systems through using a least-squared support vector machine classifier. This classifier helped in training the designed model with the use of large datasets [43].

Over the past five to ten years, nearly every company and organization has undergone a digital transformation through the adoption of cloud, mobile technologies, and the internet. These technologies have opened up new organizational capabilities. However, they created new complexities and vulnerabilities that, once cybercriminals learn about them, can quickly be exploited. A new wave of creative, sophisticated, and multichannel attacks floods companies with thousands of alerts, and hundreds of thousands of potential malicious

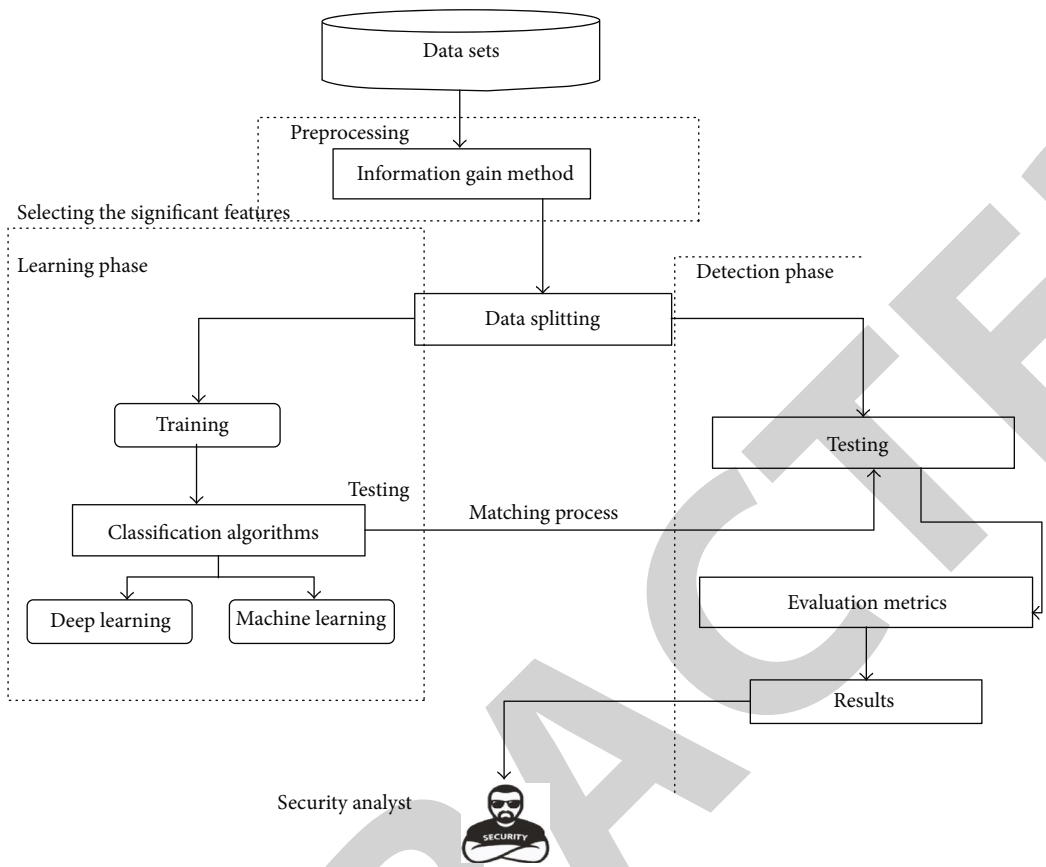


Figure 1: Framework for proposed methodology.

files are analyzed every day. Currently, artificial intelligence based on the machine learning and deep learning algorithms for data-processing capabilities provides the most effective value to the areas of cyber defenses through uncovering patterns, shapes, and outliers that indicate potential incidents, even if these solutions do not align with known attack patterns. The current research contributes to the area of cybersecurity by developing a system based on the deep learning algorithm (LSTM-RNN) to detect an anomaly, thus making the system able to detect unknown attacks. The proposed system was tested and evaluated by using four standard network datasets and two types of attacks have been considered in developing the system, namely Denial-of-Service attack (DoS) and Distribute Denial-of-Dervice (DDoS).

2. Materials and Methods

Figure 1 displays the framework of the proposed system for detecting anomaly based on cybersecurity.

2.1. Datasets. In this experiment, the four standard datasets were conducted to test the proposed system for cybersecurity. The detailed description of these data is presented in the next subsubsections.

2.1.1. KDD Cup'99 Dataset. The KDD (Data Mining and Knowledge Discovery) cup dataset was developed for the intrusion detection system; it was represented in the 3rd international knowledge discovery and data mining and machine learning tools. These datasets were collected from Local-Area Network (LAN) by Lincoln Lab, which contains a record of around five million connection networks. It contains four major types of attacks: Denial of Service (DOS), Probe, User to Root (U2R) and Remote to Local (U2R) attacks, and 41 features. In this study, a deep learning algorithm was developed to detect the DoS attack. The dataset is available in the following link http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

2.1.2. NSL-KDD Dataset. The NSL-KDD is an updated dataset of KDD Cup'99, developed by McHugh. It contains four major types of attacks: Denial of Service (DoS), Probe, User to Root (U2R) and Remote to Local (U2R), and 41 features. The dataset is available on this website: https://www.unb.ca/cic/datasets/index.html.

2.1.3. ISCX Dataset. The ISCX2012 was gathered from the University of New Brunswick in 2012. This dataset consists of two profiles: the Alpha-profile, which carries out DDoS attacks, and the Beta-profile, which is the benign network traffic generator. The dataset has been collected from network traffic which contains different protocols like HTTP, SMTP, SSH, IMAP, POP3, and FTP. The dataset is available on this website: https://www.impactcybertrust.org/dataset_view?idDataset=916.

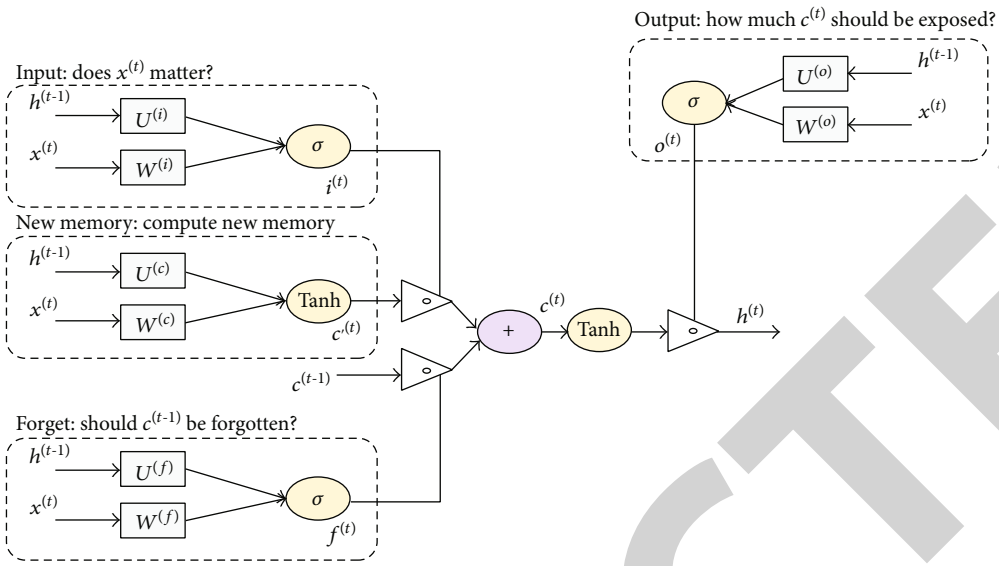


Figure 2: Structure of LSTM model.

2.1.4. CIC-IDS2017 Dataset. This dataset was collected from the Canadian Institute for cybersecurity. It contains benign networks generator and attacks, which looks like the true real-world data (PCAPs). The dataset was gathered in period starting at 9 a.m., Monday, July 3, 2017, and ended at 5 p.m. on Friday, July 7, 2017, for a total of 5 days. The normal network traffic collected on Monday. The network traffic included different types of protocols such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS. In this study, the Friday network traffic is considered for developing the deep learning system. It contains only DDoS attack and normal traffic. The dataset is available on https://www.unb.ca/cic/datasets/ids-2017.html.

2.2. Preprocessing. In this section, a detailed description of preprocessing techniques is presented. This is very important and significant in network traffic analysis, because the network traffic patterns have various types of format and dimensionality. Preprocessing is the main stage in data analysis; it is employed to manage real-world datasets into an intelligible format. Undoubtedly, most of the real-world datasets have been imperfect, noisy, and very difficult for determining the behavior of this data [44]. Preprocessing plays a vital role in analyzing patterns from network data for achieving accurate results. The information gain method was suggested to handle the important features from network datasets for detecting the malicious attacks.

2.2.1. Information Gain (IG). Information gain, which is calculated based on information entropy, represents the degree of uncertainty of information elimination, and feature selection can be performed by sorting variables by the magnitude of information gain. The amount of information has a monotonically decreasing relationship with probability. The smaller the probability, the greater the amount of information. Information gain, that is, the reduced part of the prior entropy to the posterior entropy, reflects the degree of information elimination uncertainty [45]. The information gain method is one of the ranking feature selection methods which is used to score the variable by using a threshold method for removing variable below the value of the threshold.

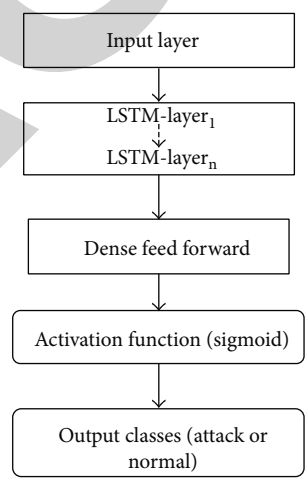


Figure 3: LSTM model for cyberattack detection.

Table 1: Parameter values of LSTM model used in the proposed system.

Parameter name	Values
LSTM units	32
Drop out	0.2
Dense feed forward layer (DFFL)	265
Dense output layer	2
Epochs	10
Batch size	205

$$H(Y) = -\sum_{y \in Y} P(y)\ \log_2 (p(y)), \tag{1}$$

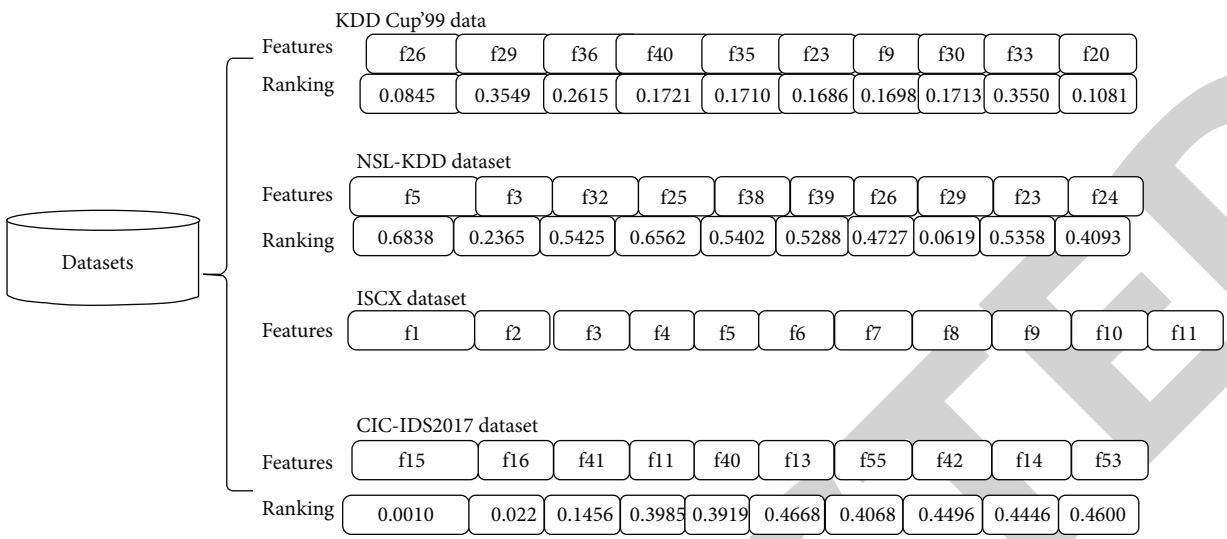


Figure 4: The significant features with ranking values from the datasets.

where $H(Y)$ is an entropy for cybersecurity datasets (y) which quantifies the uncertainty involved in the predictive value of a random variable.

$$H(Y/X) = -\sum_{x\in X} P(x) \sum_{y\in Y} P\left(\frac{y}{x}\right) \log_2 \left(p\left(\frac{y}{x}\right)\right), \quad (2)$$

where $H(Y/X)$ is a condition entropy of the x, p and GI is information gain:

$$\mathrm{GI} = H(Y) - H(\mathrm{Y/X}) \quad (3)$$

2.3. Machine Learning Algorithms. The traditional machine learning, namely Support Vector Machine (SVM) and K-Nearest Neighbor (K-NN), was presented to detect anomalies used in cybersecurity. The detailed description of classification algorithms is as follows:

2.3.1. Support Vector Machine (SVM) Algorithm. Support Vector Machines (SVM) is a binary classification model. Its basic model is a linear classifier with the largest interval defined in the feature space. The largest interval makes it different from the perceptron. SVM also includes kernel techniques, which makes it an essential nonlinear classifier that is also equivalent to the problem of minimizing the regularized hinge loss function. The learning algorithm of SVM is the optimal algorithm for solving convex quadratic programming. The basic idea of SVM learning is to solve the separation hyperplane that can correctly divide the training dataset and have the largest geometric interval. For a linearly separable dataset, there are infinitely many such hyperplanes but the separating hyperplane with the largest geometric interval is the only one.

$$K\left(X, X'\right) = \exp\left(-\frac{\left\|X - X'\right\|^2}{2\sigma^2}\right), \quad (4)$$

where the X, X' are training data of the dataset and represent the features vectors of the input dataset and the $\|X - X'\|^2$ is the squared Euclidean distance between the two features input. The σ is a free parameter. Its decision boundary is the maximum margin for solving learning samples. SVM is one of the most robust and accurate methods among all well-known data mining algorithms. It belongs to a two-class classification algorithm and can support linear and nonlinear classification. In this research work, the Radial Basis Function (RBF) was applied to detect the malicious attacks.

2.3.2. K-Nearest Neighbor (K-NN) Algorithm. The KNN algorithm is classified by measuring the distance between different feature values. K is usually an integer not greater than 20. In the KNN algorithm, the selected neighbors are all objects that have been correctly classified. This method only determines the category of the sample to be classified based on the category of the nearest one or several samples in the decision-making of classification.

The K-nearest neighbor algorithm is used to find the K values that are close to values in the training dataset, and most of these K values belong to a certain class; then, the input instance is classified into this category.

$$D_i = \sqrt{(x_1 - x_2) + (y_1 - y_2)^2}. \quad (5)$$

The K value is used to find the closet points in the feature vectors; the value should be a unique value.

2.3.3. Long-Short Term Memory Recurrent Neural Network (LSTM-RNN). Recurrent Neural Network (RNN) is one type of deep learning technique. The RNN model has a directional control loop which enables the previous states to be stored, recalled, and added to the current output [1, 2]. RNN has the gradient vanishing problem, so in order to sort out this problem, Long Short Memory (LSTM) is presented [46–

Table 2: Top ranked features of KDD Cup'99 using information gain method.

Feature's number	Feature's name
f26	srv_serror_rate
f29	same_srv_rate
f36	dst_host_same_src_port_rate
F40	dst_host_rerror_rate
F35	dst_host_diff_srv_rate
F23	Count
F9	Urgent
F32	diff_srv_rate
F29	dst_host_srv_count
F21	host_login

Table 3: Top ranked features of NSL-KDD using information gain method.

Feature's number	Feature's name
F5	src_bytes
F3	Service
F32	dst_host_srv_count
F25	serror_rate
F37	dst_host_serror_rate
F39	dst_host_rerror_rate
F26	srv_serror_rate
F29	same_srv_rate
F23	Count
F24	srv_coun

Table 4: Top ranked features of CIC-ID2017 using information gain method.

Feature's number	Feature's name
F15	Flow bytes/s
F16	Flow packets/s
F41	Packet length mean
F11	Bwd packet length max
F40	Max packet length
F13	Bwd packet length mean
F55	Avg Bwd segment size
F42	PSH flag count
F14	Bwd packet length Std
F53	Average packet size

48]. Figure 2 shows the structure of LSTM model for classifying the cyberattacks.

The hidden layer is referred to as h_t, input as x_t, and output as y_t. In addition, the RNN has internal loops which perform a series of instructions for expressing the output as being a function of a past hidden layer besides being a function of a new input. In this way, the network continues growing. The RNN enables tackling the issue of exploding and vanishing, thus preserving information. The process of the cell state is supported by the RNN, which helps in the transmission of the input data into a certain network element, and then, they are integrated with subsequent element. RNN is different from the normal neural network where it can be visualized as multiple copies of a neural network; each passes information to the next one. The state of the cell is like a conveyer belt carrying the whole architecture of the network through the entire chain. The cells have gates, which have the function of regulating the information carried throughout the conveyer belt. These gates are composed of sigmoid type activation where the output gate value and y_t are subject of multiplication. The sigmoid function has the values of 0 and 1, where the 0 value represents the transition information and 1 value represents the whole information [49].

$$h_t = \text{sigm}\left(W_{xt} + Uh_{t-1} + b^{(h)}\right), \tag{6}$$

$$O_t = \text{sigm}\left(Vh_t + b^{(o)}\right), \tag{7}$$

where h_t refers to the hidden layer that corresponds to the output x_t, h_t ¯1 refers to the hidden state of recurrent neural network, x_t refers to the input data, and O_t refers to the output value. The weight vector of neural network is represented by W, U, and V. The b refers to the bias vector in a neural network. The structure of the long-short term memory cell is shown in Figure 3. The forget gate is represented as (f_t), input gate (i_t), input modulation gate (m_t), output gate (O_t), memory cell (C_t), and hidden state (h_t). The gates are computed:

$$f_t = \text{sigm}\left(W^{(f)} + X_t + U^{(f)}\ h_{t-1} + b^{(f)}\right), \tag{8}$$

$$i_t = \text{sigm}\left(W^{(i)} + X_t + U^{(i)}\ h_{t-1} + b^{(i)}\right), \tag{9}$$

$$m_t = \tanh\ \left(W^{(m)} + X_t + U^{(m)}\ h_{t-1} + b^{(m)}\right), \tag{10}$$

$$o_t = \text{sigm}\left(W^{(o)} + X_t + U^{(o)}\ h_{t-1} + b^{(o)}\right), \tag{11}$$

where x_t is a training input data, W and U are parameters used to adjust the weight matrices, and h_{t-1} is the previous hidden layer in the long-short term memory network. In order to transfer the data from input into output, the logistic sigmoid function is used. The hyperbolic tangent function is based on the tanh function, and the b is the bias vector of training data. We computed memory cell (c_t) and hidden state (h_t) by these equation:

$$c_t = i_t.m_t + f_t.c_{t-1}, \tag{12}$$

$$h_t = o_t.\tanh\left(c_t\right). \tag{13}$$

In this research, the following specific structure of LSTM model was utilized to detect the cybersecurity attacks. Figure 3 shows LSTM model for cyberattack detection.

$$f\left(x\right) = \max\left(0, x\right). \tag{14}$$

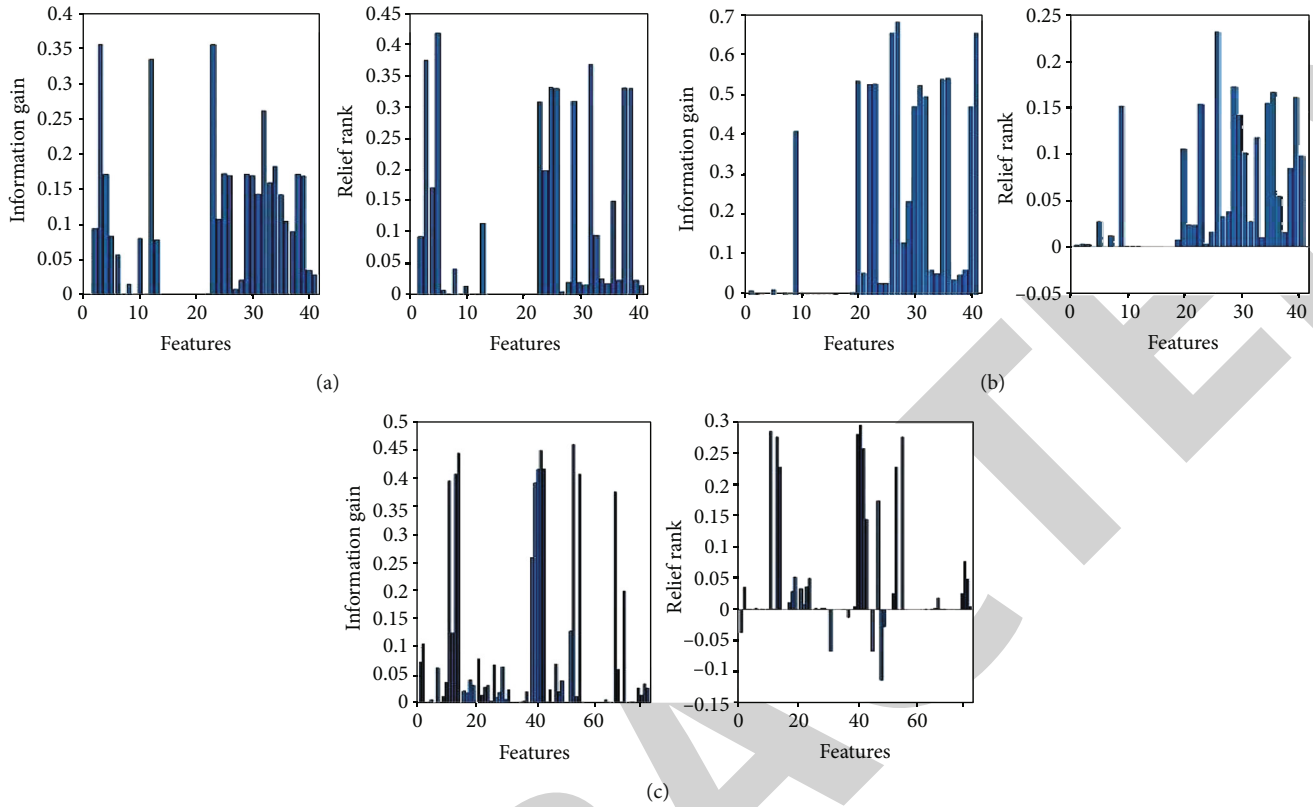


Figure 5: Performance of information gain method (a) KDD Cup'99, (b) NSL-KDD, and (c) CIC-IDS 2017 datasets.

Table 5: Distribution and splitting of the used datasets.

Dataset name	Total of samples	Training set (70%)	Testing set (30%)	Total of normal class	Total and type of attack class
KDD Cup'99	133193	93235	39958	1131107	2086 DDOS attack
NSL-KDD	29175	20422	8753	15601	13574 DOS attack
ISCX	24431	17101	7330	18426	6005 DOS attack
CIC-ID2017	19933	13953	5980	11833	8100

Sigmoid activation function is used to perform classification of the intrusion classes. The significant parameter values of LSTM model is presented in Table 1. The formula of sigmoid function is expressed as follows:

$$\sigma = \frac{1}{1 - e^{2x}}. \tag{15}$$

3. Experiment Environment Setup

In order to develop a robust cybersecurity system for detecting the cyberattacks, we should provide answers for the following set of questions; this will grant developing a successful system.

(1) Do the selected features score the highest ranking by using information methods?

(2) Can these features help in reducing the negligible features that obstructed getting accurate results by the proposed system?

(3) Do the advanced learning algorithms like deep learning have the ability to make the system more secure?

(4) Why should we compare the results of the basic machine learning and the deep learning used in detecting cyberattacks?

The answers of the above questions begin by using four standard network datasets to test the proposed system. The proposed system focused on detecting the DoS and DDOS attacks from these datasets. For the selection purity of network features, the information gain method was applied. These important features can help to obtain the highest classification accuracy. The implementation of this research has been done by using Python 3.7 with tensor flow 1.14 library and Matlab 2018 programming. The experiments were conducted on the system with I5 Processor and 4 GB RAM to process all tasks of the system. The evaluation metrics were used to evaluate the proposed system.

3.1. Significant and Ranking Features Using Information Gain Method. To answer questions one and two, the feature

selection method was used to handle the dimensionality reduction and select the subset features from the network dataset. The information gain method was applied for enhancing the accuracy of the classifying algorithms with less cost and time saving. 10 features were selected which scored the highest rank from KDD Cup'99, NSL-KDD, and CID-ID2017 datasets. Figure 4 displays the significant selection features and their ranking obtained from information gain method for three datasets (KDD Cup'99, NSL-KDD, and CID-ID2017), whereas the ISCX dataset has 11 features. These features were considered to examine the proposed system for detecting cyberattacks.

The information gain method was applied to select the significant features for improving the classification process. The information gain method depends on the ranking of the features that have lower entropy. In this research, four network datasets were considered to evaluate the proposed system, and two types of attacks were employed to test the efficiency of this system; these attacks are DoS and DDOS. Table 2 shows the important features of KDD Cup'99 dataset. The KDD cup'99 dataset has 41 features in general, the highest ranking features obtained by information gain method were selected. The significant features of NSL-KDD dataset obtained by using information gain method are presented in Table 3. 10 important features were selected which have the highest ranking among the 41 features compared with another dataset features. The CIC-ID2017 dataset contains 78 features, we have selected the 10 important features using information gain method. The 10 significant features are shown in Table 4. Figure 5 displays the ranking of KDD Cup'99, NSL-KDD, and CIC-ID2017 features of ICI-ID2017 dataset that were obtained by using information gain method.

3.2. Evaluation Metrics. In order to evaluate and measure the effectiveness of the proposed system to detect cyberattacks, the evaluation metrics like accuracy, sensitivity, specificity, precision, recall, and F1 score were employed. The equations are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}}, \tag{16}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%\text{Specifity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%, \tag{17}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%\text{Sensivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%\text{Sensivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{20}$$

Table 6: Confusion matrix of SVM algorithm.

Datasets	True positive	False positive	True negative	False negative
KDD Cup'99	39331	9	615	3
NSL-KDD	4508	151	3941	153
ISCX	1773	5	5413	139
ICI-ID2017	2313	1265	2396	4

Table 7: Confusion matrix of KNN algorithm.

Datasets	True positive	False positive	True negative	False negative
KDD Cup'99	39222	124	500	112
NSL-KDD	3918	865	3754	216
ISCX	5553	1	1774	2
ICI-ID2017	2382	43	3515	40

$$\text{F1 score} = 2 * \frac{\text{precision} * \text{Recall}}{\text{precision} * \text{Recall}} \times 100\%, \quad \text{Sensivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{21}$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

3.3. Splitting of Datasets. The following table provides a description of the types of datasets used in these experiments. Table 5 shows the splitting of the datasets.

4. Experimental Results

In this section, classification results of machine learning and deep learning based on the LSTM-RNN algorithm are presented. The empirical results of the system were examined by using the evaluative metrics: accuracy, sensitivity, specificity, precision, recall, and F1 score. The system was developed to detect the DoS and DDoS attacks. The classification algorithms were processed the significant features that have obtained from information gain method. The detailed description of the empirical results of the proposed system for detecting cyberattacks is presented in the following subsection.

4.1. Results of Machine Learning Algorithms. In this section, the results of machine learning, namely SVM and KNN algorithms, for detecting DoS and DDoS attacks are presented. The datasets were divided into 70% training and 30% testing. Tables 6 and 7 show the confusion matrix of SVM and KNN algorithms of four standard datasets. It is noted that the SVM algorithm results are better compared with the KNN algorithm.

The empirical results obtained from the machine learning approaches are calculated by making a confusion matrix.

Table 8: Testing results of SVM algorithm.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	Recall (%)	F1 score (%)	Time (second)
KDD Cup'99	99.97	99.98	99.51	99.99	99.98	99.98	85.22
NSL-KDD	96.53	96.76	96.26	96.72	96.72	96.76	286.67
ISCX	98.03	99.71	97.49	92.73	99.71	96.09	34.53
ICI-ID2017	78.77	64.64	99.83	99.98	64.64	78.47	78.18

Table 9: Testing results of KNN algorithm.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	Recall (%)	F1 score (%)	Time (second)
KDD Cup'99	99.40	99.71	81.69	99.68	81.69	99.70	255.42
NSL-KDD	87.65	81.27	94.77	94.55	81.27	87.41	168.09
ISCX	99.95	99.94	99.64	99.88	99.94	99.91	94.83
ICI-ID2017	98.61	98.79	98.35	98.79	98.83	98.57	53.67

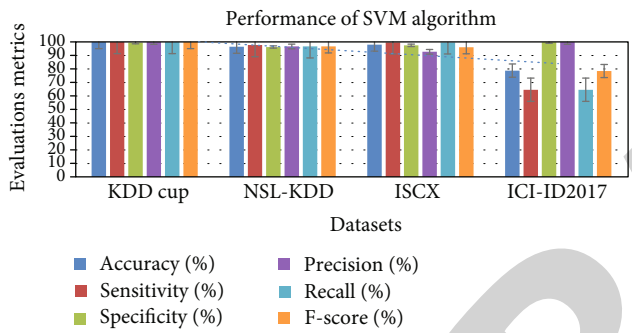


Figure 6: Performance of SVM algorithm of testing results to classify cyberattacks.

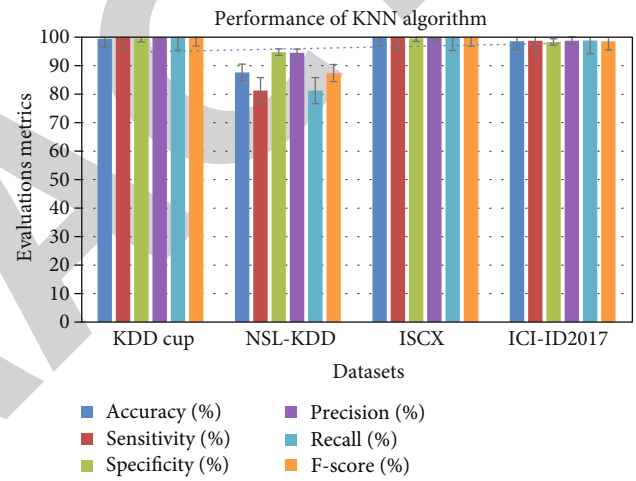


Figure 7: Performance of KNN algorithm of testing results to classify anomaly based on cyberattacks.

The confusion matrix reported the results of false positives, false negatives, true positives, and true negatives. Based on these numbers, the evaluation metrics namely accuracy, sensitivity, specificity, precision, recall, and F1 score are computed to test the proposed system. Table 8 shows the empirical results of SVM algorithm to detect the DOS and DDOS attacks from network traffic. The prediction results of KNN algorithm is presented in Table 9. It is noted that both SVM and KNN algorithms have shown satisfactory results; nevertheless, the performance of the KNN algorithm is better with CIC-ID2017 dataset, whereas the SVM algorithm is better with KDD Cup'99 and NSL-KDD datasets. Finally, the SVM algorithm demonstrates slightly better performance over most datasets.

Figures 6 and 7 show the performance of the machine learning algorithms, namely SVM and KNN for detecting cyberattacks. It is observed that the machine learning algorithms are able to detect the normal and DoS and DDoS attacks from patterns in the network dataset according to the obtained results from unseen testing data.

4.2. Results of LSTM-RNN Algorithm. To answer the third question, the prediction results of deep learning based on the LSTM-RNN algorithm to detect the DoS, DDoS attacks and normal from standard network datasets are demonstrated. Experimental results were carried out on four different standard datasets. Table 10 summarizes the confusion matrix of the LSTM-RNN algorithm.

Table 10: Confusion matrix of LSTM-RNN algorithm.

	True positive	False positive	True negative	False negative
KDD Cup'99	39284	74	600	0
NSL-KDD	4288	366	3901	198
ISCX	5552	6	1769	3
ICI-ID2017	3576	6	2340	58

The confusion matrix reported the number of false positives, false negatives, true positives, and true negatives. For analyzing the classifications of the LSTM-RNN algorithm, we used dissimilar evaluation parameters along with their formulas as cited above. These are accuracy, sensitivity, specificity, precision, recall, and F1 score. While calculating these parameters, it is noticed that the proposed model provides

Table 11: Testing results of LSTM-RNN algorithm.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	Recall (%)	F1 score (%)	Time (second)
KDD Cup'99	99.81	100	89.02	99.78	100	99.90	120.60
NSL-KDD	93.55	95.58	91.42	92.13	95.58	93.82	48.20
ISCX	99.87	99.94	99.66	99.89	99.94	99.91	40.25
CIC-ID2017	98.92	98.40	99.74	99.83	98.40	99.11	68.12

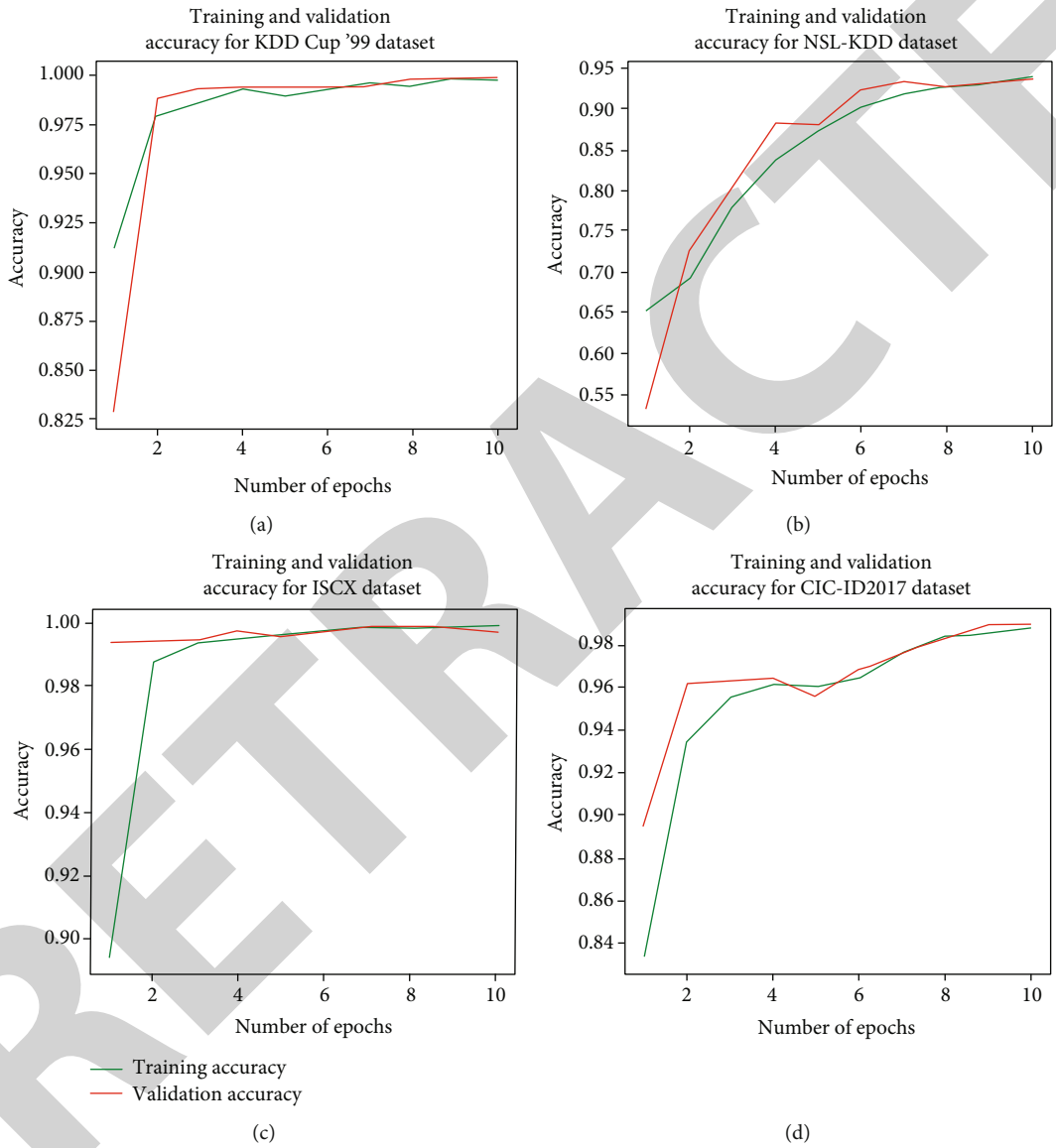


Figure 8: Performance of LSTM-RNN model for testing data.

better performance in all network datasets. Table 11 shows the empirical results obtained from the LSTM-RNN algorithm.

Figure 8 shows the performance of LSTM-RNN model to classify the cyber-attack by using four standard network datasets. The graphical representation shows the validation result of the LSTM-RNN model and the number of epochs considered to run the system. Overall, the LSTM-RNN model achieved optimal results compared with traditional machine learning algorithms.

4.3. Results Discussion. To answer the fourth question, a comparative presentation of the prediction results of the traditional machine learning and deep learning based on LSTM-RNN algorithms is given in order to approve the effectiveness of the proposed system for detecting the

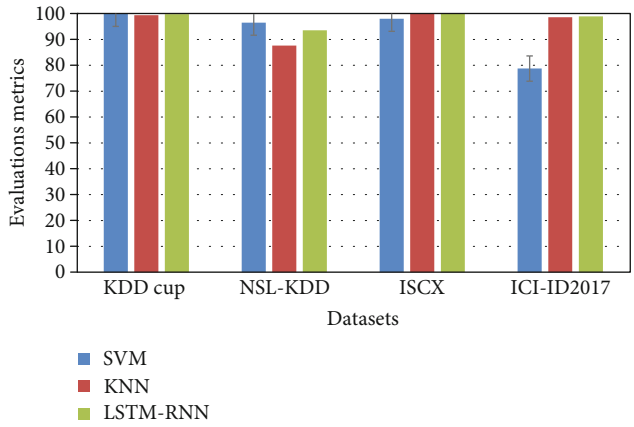


Figure 9: Comparison of the LSTM-RNN against machine learning algorithms in terms of accuracy metric.

cyberattacks. We use the same training and testing set of data for all the algorithms.

The result outcome from the machine learning, namely SVM and KNN and deep learning, based on the LSTM-RNN algorithms for detection cyber-attack is approved by using evaluation metrics. The empirical results were calculated using the confusion matrix obtained from the proposed model. We calculate the validation results only for finding the capability of the proposed system to identify the DOS and DDoS attacks. In order to save the time of building the model and the accuracy, the preprocessing method is important for handling the datasets features. The information methods were applied to select the highest ranking features and these features are significant for detecting cyberattacks. These features were processed by using the machine learning and LSTM algorithms; it is noted that the LSTM-RNN model has achieved the highest accuracy over all the network datasets. The LSTM-RNN model gave significant results in terms of accuracy, sensitivity, specificity, precision, recall, and F1 score which ensures the model effectiveness while predicting anomalies or intrusions. In addition, Figure 9 shows the outcome results of the LSTM-RNN against the machine learning SVM and KNN algorithms in terms of accuracy values.

5. Conclusion

In this paper, we presented the machine learning and deep learning algorithms to detect anomalies in cybersecurity attacks. Taking into account the multidimensional nature of the network features due to the different formats of the network dataset, we find the preprocessing stage is very important to handle this multidimensionality. Furthermore, the information gain method was applied to select the highest ranking network features for building the system. For making the system more secure, we selected the important network features. These features were processed by classifying algorithms to detect the anomaly in the cybersecurity attacks. The machine learning algorithms like SVM and KNN algorithms and deep learning based on the LSTM-RNN model were implemented. The effectiveness of the proposed system was examined by conducting a number of experiments on cybersecurity datasets. The proposed system was tested by using evaluation metrics for unseen dataset. The experimental results showed the effectiveness of the proposed system to detect the intrusion attacks on cybersecurity. Overall, deep learning based on the LSTM-RNN algorithm achieved the highest accuracy. Comparison of outcome results of LSTM-RNN model with traditional machine learning approaches for analyzing the effectiveness of these approaches is also presented. In a future work, we will apply the propped system in Internet of Things (IoT) security services on cybersecurity attacks.

Data Availability

The KDD (Data Mining and Knowledge Discovery) cup dataset was developed for the intrusion detection system; it was represented in the 3rd international knowledge discovery and data mining and machine learning tools. These datasets were collected from Local-Area Network (LAN) by Lincoln Lab, which contains a record of around five million connection networks. It contains four major types of attacks: Denial of Service (DOS), Probe, User to Root (U2R) and Remote to Local (U2R) attacks, and 41 features. In this study, a deep learning algorithm was developed to detect the DoS attack. The dataset is available in the following link http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. In "NSL-KDD Dataset" subsubsection, the NSL-KDD is an updated dataset of KDD Cup'99, developed by McHugh. It contains four major types of attacks: Denial of Service (DoS), Probe, User to Root (U2R) and Remote to Local (U2R), and 41 features. The dataset is available on this website: https://www.unb.ca/cic/datasets/index.html. In "ISCX Dataset" subsubsection, the ISCX2012 was gathered from University of New Brunswick in 2012. This dataset consists of two profiles: the Alpha-profile, which carries out DDoS attacks, and the Beta-profile, which is the benign network traffic generator. The dataset has been collected from network traffic which contains different protocols like HTTP, SMTP, SSH, IMAP, POP3, and FTP. The dataset is available on this website: https://www.impactcybertrust.org/dataset_view?idDataset=916. In "CIC-IDS2017 Dataset" subsubsection, this dataset was collected from the Canadian Institute for cybersecurity. It contains benign networks generator and attacks, which looks like the true real-world data (PCAPs). The dataset was gathered in period starting at 9 a.m., Monday, July 3, 2017, and ended at 5 p.m. on Friday, July 7, 2017, for a total of 5 days. The normal network traffic collected on Monday. The network traffic included different types of protocols such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS. In this study, the Friday network traffic is considered for developing the deep learning system. It contains only DDoS attack and normal traffic. The dataset is available on https://www.unb.ca/cic/datasets/ids-2017.html.

Conflicts of Interest

The authors declare that they have no conflicts interest.

Acknowledgments

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia, for funding this research work through the project number IFT20177.

References

[1] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 343–356, 2010.

[2] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[3] A. Avaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BIONETICS)*, pp. 21–26, New York, NY, USA, 2016.

[4] H. Hindy, D. Brosset, E. Bayne et al., "A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets," *arXiv*, vol. 1806, p. 03517, 2018.

[5] R. Williams and D. Zipser, "Gradient based learning algorithms for recurrent networks and their computational complexity. In Back propagation: Theory, Architectures, and Applications," in *Lawrence Erlbaum Associates*, pp. 433–486, Hillsdale, NJ, USA, 1995.

[6] L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surv*, vol. 18, no. 2, pp. 1153–1176, 2016.

[7] T. Garcia, J. Diaz, G. Maciá, and E. Vázquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[8] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection. In Security and Privacy (SP)," *IEEE Symposium*, vol. 2010, pp. 305–316, 2010.

[9] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URLs," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1333–1343, 2018.

[10] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Detecting malicious domain names using deep learning approaches at scale," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1355–1367, 2018.

[11] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Evaluating deep learning approaches to characterize and classify the DGAs at scale," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1265–1276, 2018.

[12] C. Szegedy, "Intriguing properties of neural networks," *arxiv*, vol. 1312, p. 6199, 2014.

[13] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 1310–1318, Honolulu, HI, USA, 2017.

[14] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur*, pp. 1528–1540, Vienna, Austria, 2016.

[15] N. Carlini and P. Mishra, "Hidden voice commands," in *Proc. 25th USENIX Secur. Symp*, pp. 513–530, Austin, TX, USA, 2016.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," *Arxiv*, p. 160702533, 2017.

[18] U. Fiore, F. Palmieri, A. Castiglione, and A. de Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neuro computing*, vol. 122, no. 122, pp. 13–23, 2013.

[19] M. Salama, H. Eid, R. Ramadan, A. Darwish, and A. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft Computing in Industrial Applica- tions*, pp. 293–303, Springer, 2011.

[20] N. Gao, L. Gao, Q. Gao, H. Wang, K. Alrawashdeh, and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 195–200, 2016, IEEE.

[21] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of restricted Boltzmann machines as a model for anomaly network intrusion detection," *Comput. Netw*, vol. 144, no. 144, pp. 111–119, 2018.

[22] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Second International Conference on Advanced Cloud and Big Data. IEEE*, pp. 247–252, Huangshan, China, 2014.

[23] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*, pp. 483–490, Springer, 2019.

[24] Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang, "A model based on convolutional neural network for online transaction fraud detection," *Security and Communication Networks*, no. 2, 9 pages, 2018.

[25] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37004–37016, 2019.

[26] Y. Yu, J. Long, and Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders," *Security and Communication Networks*, vol. 2017, Article ID 4184196, 10 pages, 2017.

[27] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, no. 1, pp. 639–642, 2017.

[28] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[29] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–41, 2015.

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9759506, 1 page
https://doi.org/10.1155/2023/9759506

Retraction

Retracted: Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)

Applied Bionics and Biomechanics

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

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] A. M. A. Mohamed, O. N. Uçan, O. Bayat, and A. D. Duru, "Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8853238, 10 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2020, Article ID 8853238, 10 pages
https://doi.org/10.1155/2020/8853238

Research Article

Classification of Resting-State Status Based on Sample Entropy and Power Spectrum of Electroencephalography (EEG)

Ahmed M. A. Mohamed,[1,2] **Osman N. Uçan**,[1] **Oğuz Bayat**,[1] **and Adil Deniz Duru**[3]

[1]*School of Engineering and Natural Sciences, Altinbas University, 34217, Turkey*
[2]*Department of Computer Science, The Libyan Academy, 16063 Benghazi, Libya*
[3]*Faculty of Sport Science, Marmara University, 34668, Turkey*

Correspondence should be addressed to Ahmed M. A. Mohamed; ahmed.mohamed@ogr.altinbas.edu.tr

Received 6 September 2020; Revised 2 October 2020; Accepted 22 October 2020; Published 11 November 2020

Academic Editor: Mohammed Yahya Alzahrani

An electroencephalogram (EEG) is a significant source of diagnosing brain issues. It is also a mediator between the external world and the brain, especially in the case of any mental illness; however, it has been widely used to monitor the dynamics of the brain in healthy subjects. This paper discusses the resting state of the brain with eyes open (EO) and eyes closed (EC) by using sixteen channels by the use of conventional frequency bands and entropy of the EEG signal. The Fast Fourier Transform (FFT) and sample entropy (SE) of each sensor are computed as methods of feature extraction. Six classifiers, including logistic regression (LR), K-Nearest Neighbors (KNN), linear discriminant (LD), decision tree (DT), support vector machine (SVM), and Gaussian Naive Bayes (GNB) are used to discriminate the resting states of the brain based on the extracted features. EEG data were epoched with one-second-length windows, and they were used to compute the features to classify EO and EC conditions. Results showed that the LR and SVM classifiers had the highest average classification accuracy (97%). Accuracies of LD, KNN, and DT were 95%, 93%, and 92%, respectively. GNB gained the least accuracy (86%) when conventional frequency bands were used. On the other hand, when SE was used, the average accuracies of SVM, LD, LR, GNB, KNN, and DT algorithms were 92% 90%, 89%, 89%, 86%, and 86%, respectively.

1. Introduction

The electrical activities of the brain can be used to identify the mental state of a person. Additionally, brain health can be monitored by electrical signals, which can be noninvasively measured from the scalp surface. Several sensors can be attached to the scalp surface, and the electrical activity of the brain can be investigated using electroencephalography (EEG). Since the number of sensors (electrodes) is limited (1-256), the spatial resolution of the measured EEG is low when compared with the other brain activity measurement techniques. On the other hand, the temporal resolution of the EEG is in milliseconds. Among the brain imaging techniques, none of the methods can work at such a high temporal sensitivity except magnetoencephalography (MEG) which is not as practical as EEG. The electrical activities of the brain are projected through the scalp surface because their impulses pass through the skull, which filters the data acting as a low-pass filter. Thus, the scalp EEG should be carefully analyzed and processed to find out the mental state of the subject. For instance, the resting state of the brain can be assessed through the ongoing EEG measurements. To identify the resting state, EEG measurement is taken for a few seconds and power spectral analysis is performed. Signal processing is widely used to extract relevant information, which is hidden in the measured EEG signals. EEG data classification is used to identify the mental diseases in emotion recognition and mental state determination using a brain-computer interface (BCI) as well as through the monitoring of mental workload. The basic state that can be identified from the scalp measurements is the resting state of the brain when eyes are open or closed [1]. The identifiers of these two states are hidden in the

frequency content of the measured signal; however, this identifier may also vary in groups of subjects [2].

1.1. Aim of the study. We aimed to identify the brain's resting status using short-length EEG epochs using both linear and nonlinear features derived from EEG. Conventional EEG band power values are generally linear, while sample entropy has been measured as complexity metrics of the multivariate signal. The concept behind this study is to adopt machine learning techniques, which are used for feature classification. Logistic regression (LR), K-Nearest Neighborhood (KNN), linear discriminant (LD), decision tree (DT) classifier, support vector machine (SVM), and Gaussian Naive Bayes (GNB) algorithms are implemented and evaluated when these techniques are performed with precision.

2. Literature Review

When the alpha band power of a depressed group of participants was compared to a normal group, low alpha band power was observed in both conditions (EC and EO) [3]. In a recent study, connectivity metrics of frontal and centro-parietal lobes were formed to classify the cases into EO and EC and the obtained accuracy values were high [4]. The same condition was studied based on wavelet fuzzy approximate entropy (WFAPEN) feature values based on support vector machine (SVM), and they reported an accuracy value of 88.2% [5]. Several studies have been conducted to analyze machine learning methodologies as a part of EEG classification. For instance, a new proposed model was applied to detect the case of vigilance or drowsiness for fast train drivers. In that study, the Fast Fourier Transform (FFT) was used to extract the power spectrum density (PSD) of EEG. They achieved 90.70% accuracy using SVM [6]. According to Saghafi et al. in [7], changing eye states EC and EO without any notice can affect the brain signals. They applied logistic regression (LR), support vector machine (SVM), and artificial neural network (ANN) classifiers. Their highest obtained accuracy was 88.2% for ANN, which detects the eye change in less than two seconds. Stepwise linear discriminant analysis (SWLDA) and Fisher's linear discriminant (FLD) showed the best performance because they were applied as parts of two kinds of methods, which were linear and nonlinear, to compare the classification techniques for the P300 Speller [8]. In a prediction study of the eye states, while using EEG signals, stacked autoencoders (SAE) and deep belief network (DBN) classifiers were used with 98.9% accuracy for the designed SAE models [9]. An effective technique was introduced that can be implemented to identify sleep stages using new statistical features, which are applied to individual EEG signals for 10-second epoch durations [10]. The distance computation like Manhattan, Minkowski, Euclidean, Hamming, and Chebyshev can affect the accuracy of a classifier. Isa et al. in [11], showed 70.08% KNN accuracy as the highest classification with Minkowski distance computation. Two classified conditions, which are represented in positive and negative emotions collected by EEGÖzerdem and Polat in [12], indicated 77.14% accuracy for multilayer perceptron neural network (MLPNN) and 72.92% for K-Nearest Neighborhood (KNN). Based on linear and nonlinear features derived from EEG, cognitive activity and resting-state conditions were classified by applying SVM and 92.1% was achieved applying nonlinear features whereas 87.5% of SVM was observed applying linear features [13].

3. Materials and Methods

3.1. Data Collection. Nine subjects participated during the resting-state EEG measurements. 16 electrodes (F3, Fp1, P3, O1, C3, FZ, T7, CZ, Fp2, F4, C4, T8, PZ, P4, O2, and OZ) were placed on the scalp surfaces of the participants using an active electrode cap with a V-amp device. The sampling rate was set at 1 kHz. Subjects were asked to close their eyes without focusing on an idea for 3 minutes while their brain signals were being collected using EEG. Then, the other 3 minutes of measurement was taken while the eyes were in an open condition. According to studies of functional resting-state magnetic resonance imaging (rs-fMRI), the keeping choice of EC and EO tasks in the resting-state studies is considered a critical factor that has a series of effects on the brain activity patterns [14, 15]. The detection of neural mechanisms of various diseases has been widely used by rs-fMRI [16] because it is suitable for patients who are unable to cooperate and respond to the task paradigms [17]. Figures 1–3, respectively, show the signals for different durations, the first second, the half-second, and the end time second of recording signals from one subject in the case of close eyes.

Figures 4–6 show the signals in the time of the first second, the half-second, and the end time second of recording signals from the same subject in the case of open eyes.

3.2. Preprocessing and Feature Extraction. For further analysis, the absolute amplitude of epochs which is greater than 100 μV was removed additionally, this study requires feature extraction, for which, the Fast Fourier Transform and sample entropy were used along with logistic regression (LR) classifier, K-Nearest Neighbor (KNN), linear discriminant (LD) analysis, decision tree (DT) classifier, support vector machine (SVM), and Gaussian Naïve Bayes (GNB) algorithms, which were used for classification purposes. Figures 7 and 8 show the power spectrum as an example of extracted features for closed and open eyes, respectively.

Fourier Transform (FT) is used to transform time-domain measurements into the frequency domain. FT divides the function into a continuous band called the spectrum of frequencies [18]. Fast Fourier Transform (FFT) is an algorithm that computes the FT at a fast pace [19].

The general formula of the Fast Fourier Transform is shown in Eq. (1).

$$X(k) = \sum_{n=0}^{N-1} x[n] w_N^{kn} = \sum_{n\,\text{even}} x(n) w_N^{kn} + \sum_{n\,\text{odd}} x(n) w_N^{kn}, \quad k = 0, 1, \cdots \cdots, N-1, \tag{1}$$

where $X(k)$ represents the Fourier coefficients of $x(n)$, which is assumed to have a complex value (sample of the time series, which consists of N samples), even n and odd n

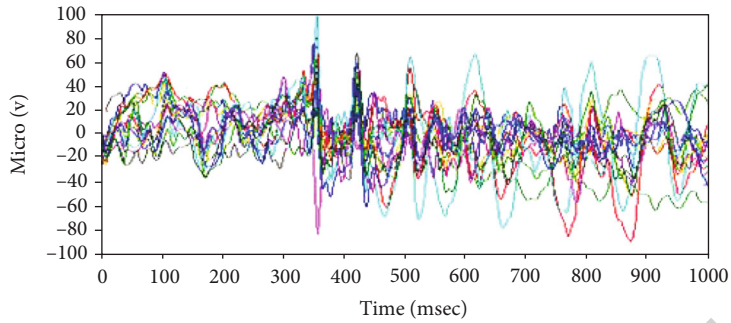


Figure 1: Time series for a subject in the first second when the subject closed the eyes.

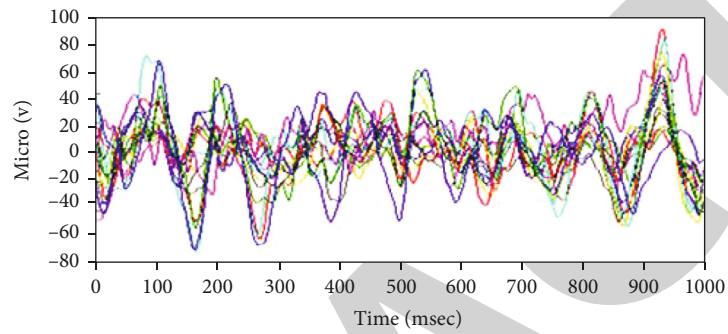


Figure 2: Time series for half second when eyes of the subject were close.

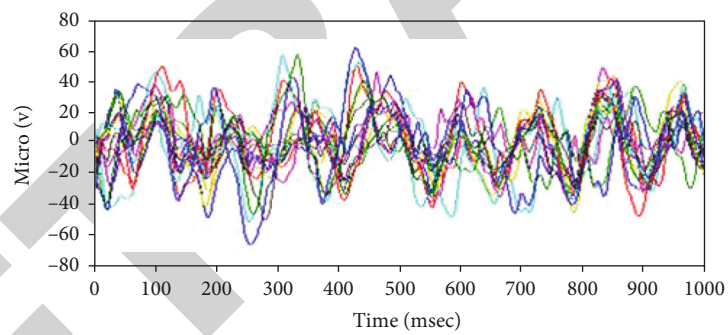


Figure 3: Time series for subject at the end time seconds in the case of close eyes.

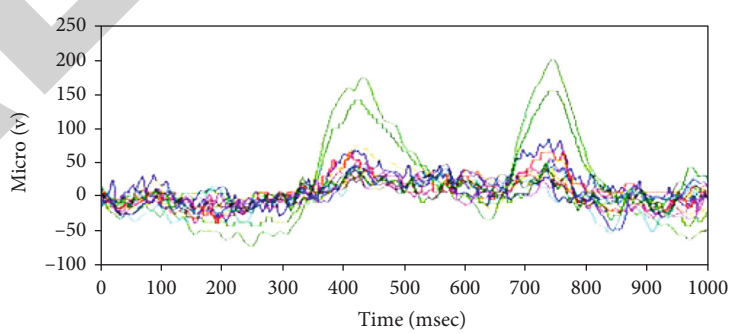


Figure 4: Time series for subject during the first second with open eyes.

correspond to the even-numbered and odd-numbered samples of $x(n)$, for frequency k, respectively. Here, $w = \exp(-2\pi j/N)$, and $j = \sqrt{-1}$; hence, j can be considered as an imaginary unit.

The second method used for feature extraction in this study was the sample entropy (SE). It is used to measure the complexity and regularity of time series [20]. The general formula of sample entropy is shown in Eq. (2).

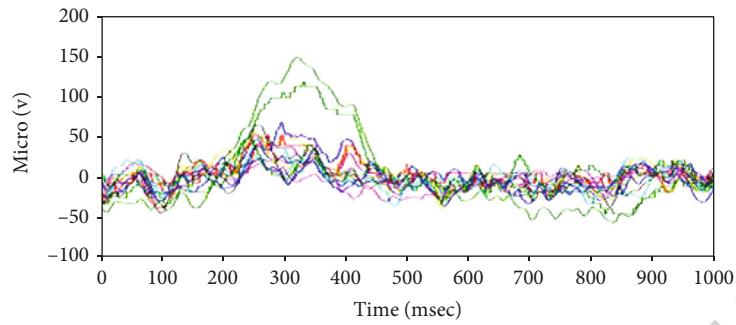


Figure 5: Time series for subject at the half seconds in the case of open eyes.

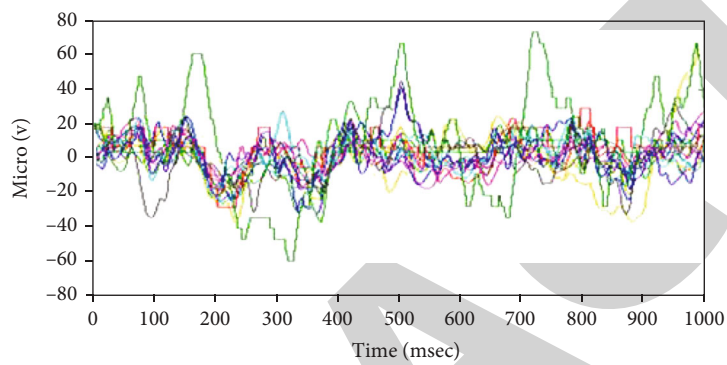


Figure 6: Time series for subject at the end time seconds in the case of open eyes.

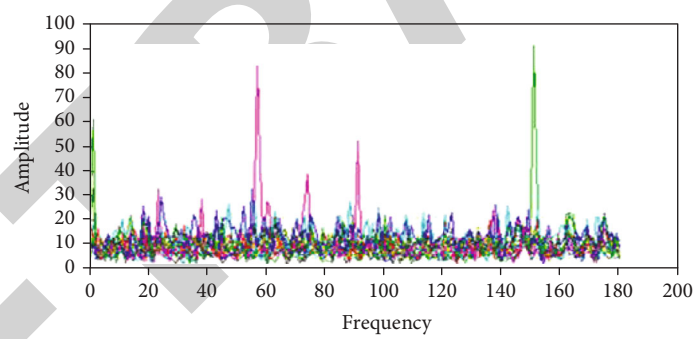


Figure 7: Power spectrum in the case of close eyes. Different colors denote different EEG channels.

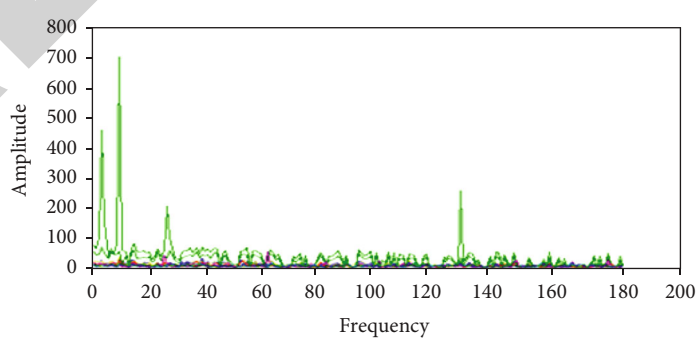


Figure 8: Power spectrum in the case of open eyes.

$$H(R) = -\sum_{i=1}^{n} \text{pi} \ \log 2(\text{pi}), \tag{2}$$

where R is a random variable that takes on values from the set {R1, R2, . . ., Rn} with respective probabilities p1, p2, . . ., pi, where $\sum i \, \text{pi} = 1$. Then, the entropy of R, $H(R)$, represents the

average amount of information contained in R, and pi is a proportion of samples that belongs to class n for a particular node.

3.3. Classification. For classification, K-Nearest Neighbor (KNN) was implemented as a first step, which was presented in the early 1950s, and it works with the enormous datasets for pattern recognition. Classifiers of the nearest neighbor depend on the comparison of similarity between training tuples with a given test tuple that represents an analogy learning step between them. This method works by using an n-dimensional pattern space, and the training tuples are put into it. KNN works as the density evaluator to distribute the training data. Based on the extracted features, which are also considered as training patterns, the data can be classified by applying KNN [21]. By Euclidean distance formula, this distance can be determined; the formula of Euclidean distance is shown below in Eq. (3).

$$\mathrm{dis}(\mathrm{x1}, \mathrm{x2}) = \sum_{i=1}^{n} \sqrt{(\mathrm{x1i} - \mathrm{x2i})^2}, \tag{3}$$

where x1 refers to tuple x1, x2 refers to tuple 2, and x11, x12,……, x1i express the number of features in tuple 1, whereas x21, x22,……,x2i represent the number of features in tuple 2.

The second method that has been used for analysis in this paper is logistic regression. It is a statistical method that deals with two kinds of classes. Also, logistic regression helps to make predictions, and it is used to develop a regression model based on a categorically dependent variable. It deals with the variable vector that evaluates the input variable coefficients [22]. Respectively, the regression model is defined by Eqs. (4) and (5).

$$z = a_0 + \sum_{i=1}^{n} a_i x_i, \tag{4}$$

$$P(z) = \frac{e^z}{1 + e^z}, \tag{5}$$

where z is the contribution measure of the explanatory variables; the regression coefficients can be represented by $x_i\ (i = 1 \cdots n)$, a_i, which were obtained by maximum likelihood in conjunction with their standard errors represented in Δa_i and $P(z)$.

Logistic regression has three types: (I) binary, which deals with a variable response as a binary response; (II) multinomial that has more than two unordered sets; (III) ordinal that has ordered sets.

The next classifier was the decision tree (DT) that works on huge stored data and transforms the data into helpful knowledge. DT is considered as a tree, and this tree has internal nodes (non leaf node) while each of them expresses a test based on an attribute, and the outcome of the test acts as a branch and a class label holed by each end node (leaf node) [23]. The process of DT learning from a class-labelled training data tuples is known as induction of DT. Attribute selection measures like information Gain and Gini index are used during tree construction to select the attributes that best partition data tuples into distinct classes. Gini index measures the impurity of data D from a set of training tuples as written in Eq. (6).

Table 1: Confusion matrix.

	Actual positive	Actual negative
Predicted positive	TP	FP
Predicted negative	FN	TN

$$\mathrm{Gini}\,(D) = 1 - \sum_{i=1}^{m} \mathrm{pi}^2, \tag{6}$$

where pi represents the probability of a tuple in D belonging to a class Ci and can be estimated using $|\mathrm{Ci}, D| / |D|$.

Information gain is a measure that finds the attribute with the highest information gain which helps to minimize the information required to classify a data tuple. Information gain is defined as in Eq. (7).

$$\mathrm{Info}\,(D) = -\sum_{i=1}^{m} \mathrm{pi}^2 \log_2 \mathrm{pi}, \tag{7}$$

where pi is the probability that a tuple in dataset D could belong to a specific class, say Ci.

Linear discriminant analysis is a method of alleviation of linear dimensions. It is used to identify the linear features, which increase the separation between classes of data and reduce the scattering within a class [24]. The classifier of LDA is used to estimate both mean and variance of the entered data using a function, which is given below in Eq. (8).

$$\mu = \frac{1}{nk} \sum x. \tag{8}$$

For the class (k), (n) is the total number of observations and μ represents the mean of the input (x).

By using Eq. (9), variance is computed for all the model inputs.

$$\sigma^2 = \frac{1}{n-k} \sum (x - \mu)^2, \tag{9}$$

where σ^2 represents the variance of all the inputs of the model.

The Naive Bayes method is known as a supervised classification algorithm, which does not need a huge amount of data for training. Naive Bayes classifiers perform very fast. With a normal distribution and big data, the Gaussian process can generate them [22, 25, 26], as shown in Eq. (10).

$$p(x_i/y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\pi\sigma_y^2}\right), \tag{10}$$

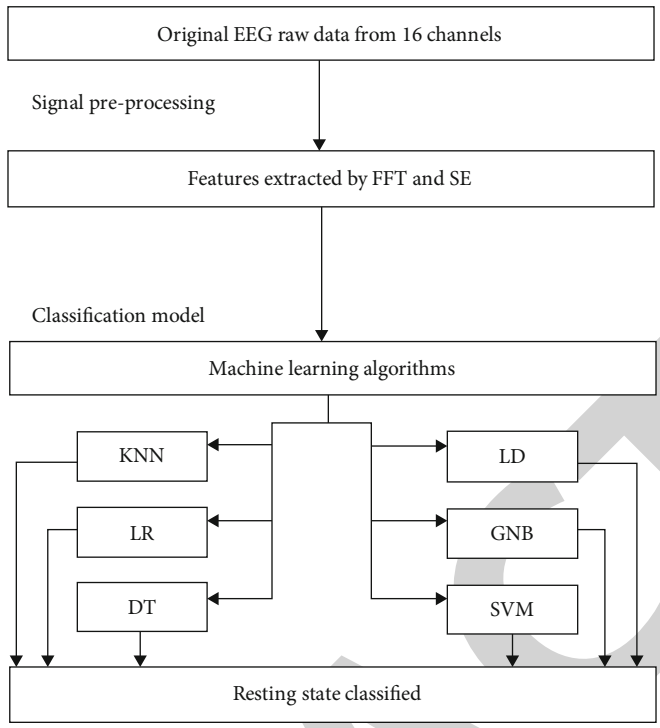


Figure 9: Sequential steps of our study.

where parameters σ_y and μ_y are estimated by maximum likelihood.

The last classifier used in this paper is the support vector machine (SVM), which is a classification method for linear and nonlinear datasets. SVM provides a learning model by separating between separable classes through constructing a hyperplane. The goal is to find the hyperplane that best separates and provides the highest distance margin between points of data tuple [26, 27]. Let W be the data vector normal to hyperplane and b the displacement of that vector; then, the decision function D for input z can be defined by Eq. (11).

$$D(z) = W \bullet z - b, \tag{11}$$

where $z \in \begin{cases} A & \text{if } D(z) > 0 \\ B & \text{if } D(z) < 0 \end{cases}$.

The distance from z to the hyperplane is defined by Eq. (12).

$$\frac{D(z)}{||W||}. \tag{12}$$

3.4. Confusion Matrix. The confusion matrix evaluation has been applied in the current study. The classification is quantifiable using the confusion matrix. According to Han et al. in [28], classifier analysis can be applied to recognize different classes with the help of confusion matrix tools. The performance matrix can be expressed in terms of confusion matrix using options such as true negative (TN), true positive (TP), false negative (FN), and false positive (FP).

Table 2: Classification accuracies of classifiers applied to extracted features by FFT.

Subjects	KNN	LR	DT	LD	GNB	SVM
1	0.97	0.99	0.99	0.98	0.97	0.99
2	0.92	0.97	0.85	0.97	0.71	0.98
3	0.70	0.77	0.73	0.75	0.71	0.82
4	0.88	0.97	0.84	0.92	0.89	0.94
5	0.99	1.00	0.97	0.99	0.63	1.00
6	1.00	1.00	0.97	0.99	0.99	1.00
7	1.00	1.00	0.99	1.00	0.98	0.98
8	0.96	1.00	1.00	1.00	0.99	0.99
9	0.99	0.99	0.97	0.99	0.90	0.99
Average accuracy	0.93	0.97	0.92	0.95	0.86	0.97

True negatives (TNs): they are correctly labelled negative records. TNs represent the number of true negatives.

True positives (TPs): it classifies positive tuples that have been accurately labelled. TPs represent the number of true positives.

False negatives (FNs): they are mainly positive records mislabelled as negatives, for instance, in our case study; the class with open eyes was wrongly predicted as the one with closed eyes by the classifier. FNs represent the number of false negatives.

False positives (FPs): they are negative records that are incorrectly labelled "positive," such as a record of class with closed eyes while the classifier prediction was open eyes. FPs represent the number of false positives. "Davis and Goadrich

Table 3: Confusion matrix and parameter evaluation of classifiers applied to extracted features by FFT.

Classifiers	Confusion matrix			Parameter evaluation		
		Close	Open	Precision	Recall	F1-score
KNN	Close	35	1	0.92	0.97	0.94
	Open	3	34	0.96	0.91	0.93
		Close	Open	Precision	Recall	F1-score
LR	Close	36	1	0.97	0.97	0.97
	Open	1	36	0.97	0.97	0.97
		Close	Open	Precision	Recall	F1-score
DT	Close	34	3	0.94	0.92	0.94
	Open	2	35	0.92	0.95	0.93
		Close	Open	Precision	Recall	F1-score
LD	Close	36	1	0.97	0.97	0.97
	Open	1	36	0.97	0.97	0.97
		Close	Open	Precision	Recall	F1-score
GNB	Close	35	1	0.84	0.97	0.89
	Open	9	28	0.95	0.76	0.82
		Close	Open	Precision	Recall	F1-score
SVM	Close	35	1	0.97	0.97	0.96
	Open	3	34	0.97	0.97	0.97

in [29]" described the confusion matrix, which is illustrated in Table 1.

Besides the mentioned ones, there are some other confusion matrix concepts such as "precision and recall", which are commonly applied as classification methods. Recall measures completeness while precision measures exactness. Precision implies the real records' percentage, which is labelled "positive". Recall implies the records' percentage, which is also labelled "positive". Other ways of using precision and recall are to convert them into a single measure, which is termed as F1 score/F measure. Precision, recall, and F1 score considered as evaluation parameters. They can be computed by, respectively, using Eqs. (13), (14), and (15).

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

where TP is a true positive and FP is a false positive.

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

where TP is a true positive and FN is a false negative.

$$\text{F1} - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{Precision} + \text{recall}}. \tag{15}$$

The sequence of this work is shown in Figure 9

4. Results

The results are presented in the following forms: average accuracy of classification/confusion matrix/parameter evaluation represented in precision, recall, and F1-score. The average accuracy value of the individual subjects, which was obtained from the features deduced by FFT, was the highest one (97%), which were obtained using LR and SVM algorithms. The accuracies achieved from the LD, KNN, and DT were 95%, 93%, and 92%, respectively. The minimum accuracy was computed for GNB (86%). Additionally, the average accuracy value of the individual subjects, which were obtained from the features extracted by SE, was 92% for SVM, 90% for LD, 89% for LR and GNB, and 86% for KNN and DT algorithms.

Tables 2 and 3 respectively show the accuracies of classifiers, confusion matrix, and parameter evaluation applied to the extracted features by FFT.

Table 4: Classification accuracies of classifiers applied to extracted features by SE.

Subjects	KNN	LR	DT	LD	GNB	SVM
1	0.71	0.77	0.72	0.78	0.72	0.82
2	0.88	0.93	0.89	0.93	0.93	0.96
3	0.83	0.88	0.87	0.87	0.89	0.92
4	0.79	0.81	0.70	0.81	0.71	0.77
5	0.85	0.90	0.90	0.92	0.96	0.96
6	0.99	0.99	0.96	0.99	0.98	0.99
7	0.94	0.97	0.95	0.97	0.98	0.98
8	0.87	0.87	0.90	0.89	0.91	0.93
9	0.92	0.93	0.89	0.92	0.93	0.97
Average accuracy	0.86	0.89	0.86	0.90	0.89	0.92

Table 5: Confusion matrix and parameter evaluation of classifiers applied to extracted features by SE.

Classifiers	Confusion matrix			Parameter evaluation		
		Close	Open	Precision	Recall	F1-score
KNN	Close	90	2	0.82	0.97	0.89
	Open	22	69	0.96	0.77	0.85
		Close	Open	Precision	Recall	F1-score
LR	Close	83	9	0.89	0.90	0.89
	Open	10	81	0.90	0.89	0.89
		Close	Open	Precision	Recall	F1-score
DT	Close	79	13	0.89	0.86	0.86
	Open	11	80	0.86	0.89	0.87
		Close	Open	Precision	Recall	F1-score
LD	Close	85	8	0.89	0.92	0.90
	Open	11	80	0.92	0.88	0.90
		Close	Open	Precision	Recall	F1-score
GNB	Close	84	8	0.90	0.91	0.90
	Open	10	81	0.90	0.89	0.89
		Close	Open	Precision	Recall	F1-score
SVM	Close	87	5	0.92	0.94	0.93
	Open	9	82	0.94	0.91	0.92

Table 6: Different results of different objective studies compared with our study.

Researchers	Objectives of study	Algorithms used	Accuracy (%)
[6]	They proposed a fatigue detection system based on monitoring high-speed train drivers by wireless EEG.	SVM	90.70
[12]	Classification for positive and negative emotions used in the EEG signals.	MLPNN KNN	77.14 72.92
[13]	Classifying the resting states of the human brain using linear and nonlinear EEG features	SVM with non-linear features SVM with linear features.	92.1 87.5
[33]	Improving the three-class motor imagery (MI) with BCI classification accuracy	LDA SFFS	86.06 93
[34]	Imposing to increase the mental workload (mw)	Average accuracies of KNN, SVM, and DT	94, 88, 89
[35]	Effectiveness of the discrete wavelet transform (DWT) in load recognitions signatures	KNN, SVM DT, RF Adaboost, GBoosting GaussianNB, LDA QDA	98.93, 64.93 100, 95.33 61.20, 100 66.53, 69.06 19.06
Present work	Identifying the resting-state status of brain using short-length EEG epochs	Extracted features by FFT: KNN, LR, DT, LD, GNB, SVM Extracted features by SE: KNN, LR, DT LD, GNB, SVM	 93, 97, 92, 95, 86, 97 86, 89, 86 90, 89, 92

Tables 4 and 5 respectively show the accuracies of classifiers, confusion matrix, and parameter evaluation applied to the extracted features by SE.

5. Discussion

EEG played a significant role in several studies because its functions are based on pure brain signals. Using these signals, a lot of information can be obtained, specifically when this information is processed through cleaning, filtering, and sorting. It was noticed that different studies used different kinds of EEG. Besides, the numbers of electrodes, which can be 1, 4, 16, or even up to 256, record the signals directly from the brain tissue as an invasive technique or just from the scalp. Besides, EEG can be measured under different planned conditions, such as eyes closed, eyes open, or during the implementation of a cognitive task from healthy participants or patients [30]. In the concept of this study, we adopted three-minute measurements of EC and EO tasks as the resting-state paradigm.

Some studies were conducted to compare the accuracy when they twice used the SVM classifier with linear and nonlinear features. Moreover, the eyes' open condition was sustained for five minutes, during which, measurement signals were obtained from 128 electrodes in [13]. Other studies focused on the comparison between the results used in different algorithms. For instance, a comparative analysis was conducted to compare SVM and ANN to detect the events of eyes, such as closed, open, and blinking eyes. The highest accuracy was assured in the case of SVM [31]. The results of previous studies showed that the accuracy of the algorithms varied, depending on the different conditions they were operating with. In some of the EEG classification studies, varying accuracy values were reported as a result of different classifiers. However, in our study, similar accuracy values were achieved as in [32]. In EC and EO tests, the number of electrodes in our study was different compared to the ones used in previous studies and volunteers of our study showed a very high degree of compliance with the instructions given to them. The execution time of classification algorithms was less than a minute in the present study. Some different factors that can affect the results of a lot of studies represented in the numbers of electrodes of EEG, the length of windows, number of subjects who may be healthy or patients' volunteers, the ways of feature extraction, the stimulus which can be used, numbers of classifiers, and so on. For instance, according to Zhang et al. in [6], there were ten participants and they used eight electrodes, whereas in our study, there were nine participants and we used sixteen channels. In [12], the number of channels of the EEG was thirty-two with twenty participants whereas our data came from sixteen channels. According to Ahmad et al. in [13], the number of channels was 128 with eight healthy participants. Djemal et al. [33] used two public datasets of a BCI competition, which was provided by Graz University represented in dataset IIa with nine participants and dataset IV with three participants. According to Duru in [34], the states of EO, EC, and increased mental workload MW were tried to be identified using the scalp of EEG measurements epoched with a duration of a second. Hence, the different factors can affect the result from such a study to another. Table 6 shows the different results of different objective studies, which are comparable to our study.

From a BCI approach, the speed of the information extraction rate depends on the length of the EEG time series. Thus, when compared with the other studies, it can be observed that we can classify the two states using shorter time series, with similar accuracy values.

We achieved consistent accuracy values based on the usage of either FFT-based features or the SE features. The former feature set represents the oscillations in the time series while the latter mimics the regularity of the fluctuations. Finally, the response of the brain to the EC or EO stimuli can be discriminated by the computation of the features (by FFT or by SE) even from a one-second time period.

Data Availability

The data used for this study are available from the corresponding authors, [AMAM and ADD], upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. D. Duru and M. Assem, "Investigating neural efficiency of elite karate athletes during a mental arithmetic task using EEG," *Cognitive Neurodynamics*, vol. 12, no. 1, pp. 95–102, 2018.

[2] A. D. Duru, T. H. Balcıoğlu, C. E. Özcan Çakır, and D. Göksel Duru, "Acute Changes in Electrophysiological Brain Dynamics in Elite Karate Players," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 1, pp. 565–579, 2020.

[3] D. P. X. Kan and P. F. Lee, "Decrease alpha waves in depression: An electroencephalogram(EEG) study," in *2015 International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)*, pp. 156–161, Kuala Lumpur, Malaysia, May 2015.

[4] D. La Rocca, P. Campisi, B. Vegso et al., "Human brain distinctiveness based on EEG spectral coherence connectivity," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 9, pp. 2406–2412, 2014.

[5] W. Y. Hsu, "Assembling a multi-feature EEG classifier for left–right motor imagery data using wavelet-based fuzzy approximate entropy for improved accuracy," *International Journal of Neural Systems*, vol. 25, no. 8, article 1550037, 2015.

[6] X. Zhang, J. Li, Y. Liu et al., "Design of a fatigue detection system for high-speed trains based on driver vigilance using a wireless wearable EEG," *Sensors*, vol. 17, no. 3, p. 486, 2017.

[7] A. Saghafi, C. P. Tsokos, M. Goudarzi, and H. Farhidzadeh, "Random eye state change detection in real-time using EEG signals," *Expert Systems with Applications*, vol. 72, pp. 42–48, 2017.

[8] D. J. Krusienski, E. W. Sellers, F. Cabestaing et al., "A comparison of classification techniques for the P300 Speller," *Journal of Neural Engineering*, vol. 3, no. 4, pp. 299–305, 2006.

[9] S. Narejo, E. Pasero, and F. Kulsoom, "EEG based eye state classification using deep belief network and stacked autoencoder," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, pp. 3131–3141, 2016.

[10] K. A. I. Aboalayon, M. Faezipour, W. S. Almuhammadi, and S. Moslehpour, "Sleep stage classification using EEG signal analysis: a comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, p. 272, 2016.

[11] N. E'zzati Md Isa, A. Amir, M. Z. Ilyas, and M. S. Razalli, "The performance analysis of K-nearest neighbors (K-NN) algorithm for motor imagery classification based on EEG signal," *MATEC Web of Conferences*, vol. 140, article 01024, 2017.

[12] M. S. Özerdem and H. Polat, "Emotion recognition based on EEG features in movie clips with channel selection," *Brain Informatics*, vol. 4, no. 4, pp. 241–252, 2017.

[13] R. F. Ahmad, A. S. Malik, H. U. Amin, N. Kamel, and F. Reza, "Classification of cognitive and resting states of the brain using EEG features," in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–5, Benevento, Italy, May 2016.

[14] T. H. Brandt, "How to see what you are looking for in fMRI and PET—or the crucial baseline condition," *Journal of Neurology*, vol. 253, no. 5, pp. 551–555, 2006.

[15] E. Marx, T. Stephan, A. Nolte et al., "Eye closure in darkness animates sensory systems," *NeuroImage*, vol. 19, no. 3, pp. 924–934, 2003.

[16] T. Jao, P. E. Vértes, A. F. Alexander-Bloch et al., "Volitional eyes opening perturbs brain dynamics and functional connectivity regardless of light input," *NeuroImage*, vol. 69, pp. 21–34, 2013.

[17] J. L. Roland, C. D. Hacker, A. Z. Snyder et al., "A comparison of resting state functional magnetic resonance imaging to invasive electrocortical stimulation for sensorimotor mapping in pediatric patients," *NeuroImage: Clinical*, vol. 23, article 101850, 2019.

[18] M. R. Canal, "Comparison of wavelet and short time Fourier transform methods in the analysis of EMG signals," *Journal of Medical Systems*, vol. 34, no. 1, pp. 91–94, 2010.

[19] W. T. Cochran, J. W. Cooley, D. L. Favin et al., "What is the fast Fourier transform?," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.

[20] J. M. Yentes, N. Hunt, K. K. Schmid, J. P. Kaipust, D. McGrath, and N. Stergiou, "The appropriate use of approximate entropy and sample entropy with short data sets," *Annals of Biomedical Engineering*, vol. 41, no. 2, pp. 349–365, 2013.

[21] H. Latifpour, M. Mosleh, and M. Kheyrandish, "An intelligent audio watermarking based on KNN learning algorithm," *International Journal of Speech Technology*, vol. 18, no. 4, pp. 697–706, 2015.

[22] A. Prabhat and V. Khullar, "Sentiment classification on big data using Naïve Bayes and logistic regression," in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Coimbatore, India, January 2017.

[23] H. Sharma and S. Kumar, "A survey on decision tree algorithms of classification in data mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016.

[24] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*, pp. 531–542, Springer, Berlin, Heidelberg, 2006.

[25] R. D. S. Raizada and Y. S. Lee, "Smoothness without smoothing: why Gaussian naive Bayes is not naive for multi-subject searchlight studies," *PLoS One*, vol. 8, no. 7, article e69566, 2013.

[26] S. V. Wawre and S. N. Deshmukh, "Sentiment classification using machine learning techniques," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 819–821, 2016.

[27] T. Santhanam and M. S. Padmavathi, "Application of K-means and genetic algorithms for dimension reduction by integrating

Hindawi
Applied Bionics and Biomechanics
Volume 2023, Article ID 9796728, 1 page
https://doi.org/10.1155/2023/9796728

Retraction

Retracted: The Use of Hellinger Distance Undersampling Model to Improve the Classification of Disease Class in Imbalanced Medical Datasets

Applied Bionics and Biomechanics

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

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] Z. Z. R. Al-Shamaa, S. Kurnaz, A. D. Duru, N. Peppa, A. H. Mirnezami, and Z. Z. R. Hamady, "The Use of Hellinger Distance Undersampling Model to Improve the Classification of Disease Class in Imbalanced Medical Datasets," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 8824625, 10 pages, 2020.

Hindawi
Applied Bionics and Biomechanics
Volume 2020, Article ID 8824625, 10 pages
https://doi.org/10.1155/2020/8824625

Research Article

The Use of Hellinger Distance Undersampling Model to Improve the Classification of Disease Class in Imbalanced Medical Datasets

Zina Z. R. Al-Shamaa,[1] **Sefer Kurnaz**,[1] **Adil Deniz Duru**,[2] **Nadia Peppa**,[3] **Alex H. Mirnezami**,[3] **and Zaed Z. R. Hamady**[3]

[1]*Graduate School of Science and Engineering, Altınbaş University, Istanbul, Turkey*
[2]*Sports and Health Sciences Department, Marmara University, Istanbul, Turkey*
[3]*Southampton University Hospital NHSFT, Southampton, UK*

Correspondence should be addressed to Zina Z. R. Al-Shamaa; zina.shamaa@gmail.com

Received 18 August 2020; Revised 19 September 2020; Accepted 5 October 2020; Published 4 November 2020

Academic Editor: Mohammed Yahya Alzahrani

Imbalanced class distribution in the medical dataset is a challenging task that hinders classifying disease correctly. It emerges when the number of healthy class instances being much larger than the disease class instances. To solve this problem, we proposed undersampling the healthy class instances to improve disease class classification. This model is named Hellinger Distance Undersampling (HDUS). It employs the Hellinger Distance to measure the resemblance between majority class instance and its neighbouring minority class instances to separate classes effectively and boost the discrimination power for each class. An extensive experiment has been conducted on four imbalanced medical datasets using three classifiers to compare HDUS with a baseline model and three state-of-the-art undersampling models. The outcomes display that HDUS can perform better than other models in terms of sensitivity, F1 measure, and balanced accuracy.

1. Introduction

Classification is a standard data mining process. It consists of two steps, building a model and testing a model. A classification model is built to learn from training data which is then tested to predict the category of unknown samples. Most classification algorithms were mainly built to classify the balanced dataset, whereas a problem occurs when a dataset is imbalanced, which degrades the recognition power of the classifier [1]. The imbalanced problem appears when the data is composed of very different sample numbers for the various classes; i.e., the number of samples of one class is greater than those in the second class, the former is called the majority class, and the latter is called the minority class [2]. Imbalanced datasets usually influence the classification process. If the problems of imbalanced class distribution are not addressed before implementing the classification procedures, the classifier appears to be biased towards the majority class cases while ignoring to classify the minority class cases correctly [3]. However, the problems of classifying imbalanced data often occur in real-life applications such as analyzing medical datasets, where the cases of patients with the disease are significantly lower than those without the disease. For instance, in cancer detection, the cases of patients diagnosed with cancer are much smaller than those of patients who do not have cancer [4]. The classification model to predict cancer results in lower classification performance of abnormal class and incorrect prediction disease which leads to serious health risk.

In general, the problem of classifying imbalanced data is due to the lack of training with a few minority class samples which are inadequate to predict accurately [5]. Previous

studies have proposed resampling techniques to solve the problem of class imbalance. These techniques are mainly categorized into oversampling and undersampling [6]. The oversampling method is aimed at generating samples for the minority class [7], and the undersampling method is aimed at reducing samples for the majority class [8].

In this work, we propose a novel undersampling technique, named the Hellinger Distance Undersampling (HDUS) model, aimed at solving the imbalanced classification problem in medical datasets. The proposed model reduces healthy class samples to improve the classifying performance of the rare disease class. It adopts the Hellinger distance to measure the similarity between majority class instance and its neighbouring minority class instances, then chooses a number of the highest Hellinger distance values, and sums them up to be a similarity value for each majority instance. Finally, the model selects a subset from the majority instances, having top similarity values, and combined with the original minority class instances. This model could effectively separate major class instances and minor class instances and boost the discrimination power for each class, thereby improving the classification accuracy for rare class. We compared the HDUS with four models, including a baseline model without using any sampling technique and three state-of-the-art undersampling models. The experiment was conducted on four imbalanced medical datasets using three classifiers.

The remainder of this paper is organized as follows. Section 2 mentions a related work. Section 3 presents the proposed model. Section 4 reviews the experiment setup. Section 5 presents the results of the experiment. Section 6 demonstrates the discussion of the results. Section 7 is the conclusion of the study.

2. Related Work

Recently, the problem of imbalanced classification has drawn much attention in the literature because the traditional classification algorithms were not initially built to train the imbalanced dataset [9]. This problem usually emerges from the different distribution of classes in the feature space. Furthermore, there are some other problematic features of imbalanced datasets such as overlapping samples, small disjoints, and small sample sizes. The overlapping denotes to the data samples in various classes that overlap in the feature space. The small disjoint denotes to the few samples in the minority class that are spread separately in the feature space. Finally, the small sample sizes refer to an insufficient number of data samples in the minority class. The aforementioned imbalanced features would raise the complexity of the classifier, which in turn makes it difficult to classify the minority class samples correctly [5, 10].

To overcome the imbalanced data problem, current approaches may be categorized to the algorithm level and data level. The first group works to change the classification algorithm, to support the minority class cases, by specifying weights to cases from various classes or by ensemble methods [11, 12]. The second group applies before the classification procedure to modify the distribution of imbalanced dataset through data sampling techniques [13].

Previous studies indicated that solving the imbalanced problem at the data level is simple and efficient for unbalanced classification [1]. Therefore, data sampling techniques have been widely used to alleviate the unbalanced classification problem by modifying the distribution of classes in the training dataset. Generally, sampling techniques are categorized into over- and undersampling [14]. The oversampling technique is aimed at generating instances artificially for a minority class by adding copies of already existing data from minor class instances [7]. Many methods of oversampling have been applied earlier. Random oversampling (ROS) is a common oversample approach that randomly adds samples to the minor class. Although ROS adjusts the class distribution, it may increase the overfitting problem by making similar copies of the minor class that influence the classification process [14]. Another standard oversample approach is the synthetic minority oversampling technique (SMOTE) [15]. It is used to generate artificial samples. Unlike ROS, SMOTE avoids the overfitting problem, but it may cause the overlapping with the surrounding samples that increase the overall training data size and hinder the training process [16, 17]. Generally, with the oversampling technique, the problem of an imbalanced class is diluted, but the training data is going to get more crowded. Therefore, the classification performance is affected [18].

Undersampling is another reasonable data sampling technique which attempts to reduce the number of samples in the majority class. The undersampling concept is how to eliminate majority class instances in a manner that retains the practical distinction among classes [8]. Numerous undersampling methods have been implemented and used earlier. The most naive approach is random undersampling (RUS), which eliminates instances from major class randomly. It tends to balance the distribution of classes but causes waste of valuable information that could be essential for the classification process [14]. Tomek link (Tml) is another undersampling method used to address the overlapping problem. It looks for pairs of samples belonging to different classes but are each other's nearest neighbour and eliminates the majority sample of the pair [19]. Another method is the edited nearest neighbour, which is applied to eliminate major class samples based on the nearest K neighbour that belongs to the minor samples. When the number of neighbours in each major class is higher in the minor class, the major class sample shall be omitted as noise or borderline [20].

Previous research studies revealed that there is no optimal rule to attain the best fit with over- or undersampling. They have shown that usually undersampling process of the major class is used to outperform the results obtained through the oversampling of the minor class [15]. More than that, as the data size has been increasing, the undersampling method would be a better option than the oversampling method [21]. Instance selection was used in previous studies to remove the outlier from the training dataset, which can make the classifier perform better than the original dataset [22–24]. However, the existing instance selection techniques have programmed to choose a portion of the initial dataset which cannot be used directly to choose instances from just one class of the dataset, such as selecting from the major class

instances. Kubat and Matwin in [25] proposed one-sided instance selection to remove noise samples, redundant samples, and borderline samples from the majority class while keeping the original samples belonging to the minority class.

Recently, a lot of undersampling methods have been reported in the literature to improve the imbalanced data classification. Tsai et al. in [10] introduced an undersampling method by clustering the majority class into groups of similar data samples; then, the instance selection extracts the non-representative data samples from each group. Nwe and Lynn in [20] suggested an undersampling approach began by determining the closest major class neighbours to each minor class sample, then evaluating the number of correlation of each neighbour from the major class with the minor class samples. Finally, the required number of major class instances is taken from the number of correlations. Besides, the authors in [26] adopted the one-sided undersampling technique. They proposed a method for reducing the major class size that modifies the distribution of initial imbalanced classes by measuring the similarities of each major class case with the corresponding minor class cases. The method effectively separates the major and minor class cases to optimize the identity value for each class.

3. Proposed Model

The work is aimed at providing a method that handles the problem of imbalanced data distribution which affects classification performance of minority class samples. In the imbalanced dataset, the class with a larger number of instances takes up most of the space. Unequal class distribution makes the classifier to be inadequately qualified to classify the smaller class instances, and the class with a larger number of instances overlaps the identification ability of the class with a smaller number of instances. In this case, the classifier would favour the majority class instances and scoring false high accuracy.

In this work, we proposed an undersampling model by following the principle of one-sided selection to extract instances from the major class, while the data in the minor class will remain without change. This is based on the premise that it is better to keep the instances of a minor class as real as they are, in such a manner that no greater or no less quantity is exercised on them. So, the classifier will be provided by an accurate recognition power for the original minor class samples.

Instance selection in the undersampling technique depends on how to select majority class instances in a manner that retains the compatible distinction among classes. In our proposed model, we used Hellinger distance (HD) [27, 28] to choose instances from the major class based on their Hellinger similarity degree with the minor class instances. Hellinger distance is a measure of the variance in distribution [29]. In [30], Cieslak et al. demonstrated analytically that HD is very robust in the presence of a skew distribution of class and it is not affected by the class imbalanced rate due to its isometric contours. This is the motivation of using HD in our proposed method. To express the equation of Hellinger distance, let $a(x)$ and $b(x)$ be two probability functions; then the HD between $a(x)$ and $b(x)$ can be expressed as follows:

$$\mathrm{HD}(a, b) = \frac{1}{\sqrt{2}} \sqrt{\int \left(\sqrt{a(x)} - \sqrt{b(x)} \right)^2 dx}. \quad (1)$$

Considering the problem of classification of the imbalanced class dataset and being motivated by the properties of the HD, we proposed an undersampling model using Hellinger similarity measure. The proposed model works to reduce the number of major class instances, aimed at upgrading the prediction performance of minority class which is the class of teh highest interest in medical datasets. Algorithm 1 the pseudocode of the proposed HDUS model:

4. Experiment Setup

In this section, we display the details of the experiment to test the proposed HDUS model. We present the nature of the datasets, the used classification algorithms, the evaluation metrics, and the undersampling methods used for comparison.

The code for the whole experiment was conducted in Python Programming language and spyder tools using the available utilities to provide all the necessary preprocessing and classification techniques besides the evaluation functions.

4.1. Datasets. In this work, we have exercised four imbalanced medical datasets to evaluate the performance of the suggested (HDUS) model. For each dataset, the number of features (attributes), the number of instances, the number of majority cases, and the number of minority cases are presented. These datasets are described in the following.

4.1.1. A Novel Colorectal Cancer Dataset (CRC). This dataset is from the Southampton University Hospital and has been used with approval from the responsible surgeon (co-author), and the data are all anonymous. The data are for patients having primary cancer at 12 colorectal sites, who then have cancer resection surgery. There are 1005 instances (patients), each of which acts as a record of a single patient with 14 features (attributes), including the target label. Out of 1005 instances, 760 are for patients having primary CRC who do not have metastasis, representing the majority samples, and another 245 cases are for patients having primary CRC growing to metastasis in other organs of the body, representing the minority samples. The data type is categorical (groups into multiple categories) and mapped to numeric values. Table 1 shows the features of colorectal cancer dataset.

4.1.2. PIMA Indians Dataset. The dataset of PIMA Indians was taken from the UCI machine learning repository [31]. It has nine features, including the class feature. The class feature indicates if there are patients with diabetes or not. The dataset has 768 samples, including 268 having diabetes (the minority samples) and 500 without diabetes (the majority samples). The information of features is shown in Table 2.

```
Input: Imbalanced Training dataset (ITrD)
Output: Balanced Training dataset (BTrD)
Group the ITrD according to the classes
  C1= ITrD (class1) //C1 indicates the minor class which contains less number of instances
  C2= ITrD (class2) //C2 indicates the major class which contains more number of instances
  For i in rows of (C2)
    For j in rows of (C1)
      Sim_{i,j} = calculate the similarity between C2(i) and C1(j) using Hellinger Distance
        append Sim_{i,j} To HD(i)
    Next j
      select m top values from HD (i) // where m is a given number of neighbouring minority class
       HD_sum(i)= sum the selected m top values
  Next i
C2HD=select w majority class instances according to the highest similarity value in HD_sum(i),
// where w is a given number
return (BTrD= C2HD +C1)
```

Algorithm 1: Hellinger Distance Undersampling (HDUS) pseudocode

Table 1: The features of colorectal cancer dataset.

No.	Attribute name	Data type
1	Tumour site (in colorectal)	Categorical
2	Surgery type	Categorical
3	Operation type (on which part of colorectal the operation was done)	Categorical
4	Differentiation	Categorical
5	Dukes stage of tumour	Categorical
6	T stage 5th edition	Categorical
7	N stage 5th edition	Categorical
8	EMVI	Categorical
9	Tumour perforation	Categorical
10	Resection margin	Categorical
11	Neoadjuvant therapy n-CRT	Categorical
12	Chemotherapy	Categorical
13	Radiotherapy	Categorical
14	CRC metastasis (class)	Categorical

Table 2: The features of PIMA Indian dataset.

No.	Attribute name	Data type
1	Number of times pregnant	Numeric
2	Plasma glucose concentration	Numeric
3	Diastolic blood pressure	Numeric
4	Triceps skinfold thickness	Numeric
5	Amount of insulin	Numeric
6	Body mass index	Numeric
7	Diabetes pedigree function	Numeric
8	Age	Numeric
9	Class	Categorical

4.1.3. Thoracic Surgery Dataset (THS). The thoracic data was taken from the UCI machine learning repository [31]. This data was collected from patients who experienced tumour resections for primary lung cancer. The dataset has 17 features, including the class feature. It has 470 samples, including 70 patients who died during the one year after surgery (the minority samples) and 400 who are alive (the majority samples). The information of features is shown in Table 3.

4.1.4. Breast Cancer (BC) Dataset. The BC dataset was taken from the UCI machine learning repository [31] which is provided by the Oncology Institute. It has ten features, including the class feature. The dataset indicates if a breast cancer recurred or not. The dataset has 286 samples, including 85 cases of the minority class and 201 cases of the majority class. The information of features is shown in Table 4.

4.2. Classification Data Mining Algorithms. In this study, three classification algorithms with different characteristics were explored: decision tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbour (KNN). The primary purpose of using these classifiers was to evaluate the performance of the proposed model on four imbalanced medical datasets. The experiment was initially done on a CRC dataset and then tested on three datasets selected from the UCI repository.

4.2.1. K-Nearest Neighbour (KNN). KNN is a classification technique that relies on feature similarity measures to find the closest neighbours. For the classification of a new point, the KNN reviews each training sample as a tuple (X) with the particular label denoting its class. KNN counts the spaces between X and all training tuples, then specifies to X the maximum repeat class in the nearest k tuple [32].

4.2.2. Support Vector Machine (SVM). SVM is a supervised kernel-based classification algorithm that can be used for binary classification problems. It uses a mathematical function to define an optimal hyperplane that splits two classes in a training dataset with a maximum margin. Then, SVM increases the space between the closest training data points (support vectors) and the class boundaries trying to find the optimal hyperplane that removes some insignificant data from the training data set. However, when the data is intrinsically nonlinear, SVM will use kernel function to construct a

Table 3: The features of THS dataset.

No.	Attribute name	Data type
1	Diagnosis	Categorical
2	Forced vital capacity	Numeric
3	A volume that has been exhaled at the end of the first of forced expiration	Numeric
4	Performance status	Categorical
5	Pain before surgery	Categorical
6	Hemoptysis before surgery	Categorical
7	Dyspnoea before surgery	Categorical
8	Cough before surgery	Categorical
9	Weakness before surgery	Categorical
10	Size of the original tumour	Categorical
11	Type 2 diabetes mellitus	Categorical
12	Myocardial infarction up to six months	Categorical
13	Peripheral arterial diseases	Categorical
14	Smoking	Categorical
15	Asthma	Categorical
16	Age at surgery	Numeric
17	One-year survival period (class)	Categorical

Table 4: The features of BC dataset.

No.	Attribute name	Data type
1	Tumor size	Categorical
2	Inv nodes	Categorical
3	Node caps	Categorical
4	Menopause	Categorical
5	deg malig	Categorical
6	Breast side	Categorical
7	Breast quad	Categorical
8	Irradiat	Categorical
9	Age	Categorical
10	Class (recurrence/no-recurrence)	Categorical

separating hyperplane that transforms the data from the original dimension into a high-dimensional space. Popularly used kernel functions are the linear, polynomial, sigmoid, and Gaussian kernel [33, 34].

4.2.3. Decision Tree (DT). A decision tree classifier involves several simpler decisions to build a tree model. The tree model builds three types of nodes: root, internal, and leaf. The root represents the starting point which has no incoming edges but outgoing edges. The internal nodes are represented by the data attribute, which has only one incoming branch and at least two leaving branches for each possible attribute. The leaf nodes are represented by the classes. These patterns of the decision tree express sets of if-then rules that can be employed to classify novel samples [35].

4.3. Evaluation Metrics. A classifier is, typically, evaluated by a confusion matrix which contains four values from classification outputs that report the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The TP refers to the rate of correctly recognizing the rare positive. The TN refers to the rate of correctly recognized negative. The FP refers to the rate of frequent negative incorrectly recognized as rare positive, and the FN refers to the rate of rare positive incorrectly recognized as frequent negative. In the experiment, the minority class refers to positive and the majority class refers to negative. The most used performance measure of classification tasks is accuracy. However, it is not an appropriate metric when evaluating the imbalanced class distributions because the classifier has a strong bias towards the majority class and fails to classify the few samples of minority class [36].

More proper metrics could be used to assess the performance measurement of classifying imbalanced class distribution, such as sensitivity or recall (True positive rate (TPR)), specificity (True negative rate (TNR)) [37], precision (positive predictive value (PPV)) [32], F1-measure [35], and balanced accuracy (BACC) [38].

These metrics are given by equations in (2) as follows:

$$\begin{aligned} \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{TNR} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}, \\ \mathrm{PPV} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ \mathrm{F1m} &= \frac{2\times\mathrm{TPR}\times\mathrm{PPV}}{\mathrm{TPR}+\mathrm{PPV}}, \\ \mathrm{BACC} &= \frac{\mathrm{TPR}+\mathrm{TNR}}{2}. \end{aligned} \tag{2}$$

To ensure an unbiased evaluation of the models, the n-fold cross-validation is used as an evaluation criterion. In n-fold cross-validation, the data were divided into n equal folds, then the model was trained on all folds except one fold as a validation set on which the prepared model was tested. The process repeats so that each fold gets an opportunity to act as the test set. Then, the n-test outcome was averaged [35]. In our work, the n value is set to 5.

4.4. Comparative Method. To allow a fear valuation of the validity of our proposed method, HDUS is compared against three other undersampling methods:

(i) Tomek link (Tml): it is aimed at removing the noise and border points from majority class instances by examining pairs of samples belonging to different classes but are each other's nearest neighbour and eliminates the majority sample of the pair [19].

(ii) Random undersampling (RUS): it eliminates instances from major class randomly until the desired balance of class distribution is achieved [14].

(iii) Edited nearest neighbour (ENN): the basic idea of ENN is to eliminate samples of the major class based

on the K-Nearest Neighbour that belong to the minor samples. If the number of neighbours is predominant in each majority instance from minority instances, certain instances of the majority class are eliminated as overlap instances [20].

5. Results Analysis

To investigate the performance measures of the proposed HDUS method, we used four imbalanced medical datasets using three classification algorithms including DT, SVM, and KNN and they were compared with the baseline model (without any resampling method) and with three state-of-the-art undersampling methods (Tomek link, RUS, and ENN). The results of the four datasets (CRC, PIMA, THS, and BC) are shown in Tables 5–8, respectively, in terms of sensitivity, specificity, precision, F1 measure, and balanced accuracy.

As shown in Tables 5–8, the first column of the baseline model confirms that the imbalanced classification problem exists in all used datasets. This presents a low average rate of sensitivity to predict minority class; it ranges from 7.94% in the THS dataset to 39.7% in the PIMA dataset, while it assigns a high average rate of specificity to predict the instances of the majority class.

The 2nd, 3rd, and 4th columns of the tables represent the result of used undersampling methods: Tomek link, RUS and ENN, respectively. We can observe the improvement achieved by using these methods, as expressed by the values of sensitivity that reflects the ability of the models to detect the class of interest, i.e., the minor class. Although the Tomek link obtained worse performance in all datasets, it is better than the baseline model except in the THS dataset, which scored lower than the baseline model.

More improvement is achieved in the 5th column of all tables by the proposed HDUS method in terms of sensitivity, F1_m, and Bacc. It can be observed that the HDUS performance shows significant improvement over the baseline and the three undersampling methods. The HDUS results in the top rate of sensitivity overall datasets (that refers to the highest ability to detect the class of interest, i.e., the minority class). It scores over 80% in both CRC and PIMA, near 70% in BC and near 60% in THS which is the lowest sensitivity. It also results in the highest rate for both F1_m and balanced accuracy.

Table 5: Evaluation results for CRC dataset using classifiers (KNN, SVM, and DT) and models (baseline, Tml, RUS, ENN, and HDUS).

CRC		Baseline	Tml	RUS	ENN	HDUS
KNN	Sensitivity (%)	29.41	37.25	62.75	62.75	80.35
	Specificity (%)	85.38	75.38	60	60.31	50.23
	Precision (%)	44.12	37.25	38.1	38.51	33.91
	F1_m (%)	35.29	37.25	47.41	47.73	47.69
	Bacc (%)	57.4	56.32	61.38	61.53	65.29
SVM	Sensitivity (%)	5.88	29.41	62.75	47.06	76.40
	Specificity (%)	92.31	82.31	54.62	70.77	55.85
	Precision (%)	23.08	39.47	35.16	38.71	35.40
	F1_m (%)	9.37	33.71	45.07	42.48	48.38
	Bacc (%)	49.1	55.86	58.68	58.91	66.13
DT	Sensitivity (%)	35.29	45.1	62.75	66.67	81.00
	Specificity (%)	68.46	59.23	46.92	43.85	56.91
	Precision (%)	30.51	30.26	31.68	31.78	39.90
	F1_m (%)	32.73	36.22	42.1	43.04	53.46
	Bacc (%)	51.88	52.16	54.83	55.26	68.96
AVG	Sensitivity (%)	23.53	37.25	62.75	58.83	79.25
	Specificity (%)	82.05	72.31	53.85	58.31	54.33
	Precision (%)	32.57	35.66	34.98	36.33	36.40
	F1_m (%)	27.32	36.44	44.92	44.92	49.85
	Bacc (%)	52.79	54.78	58.3	58.57	66.79

6. Discussion

This study discussed about a preprocessing undersampling method named HDUS. The model handles the class inequality problem in medical datasets to improve the prediction performance of the minority class samples by using the instance selection based Hellinger distance similarity measure.

It is crucial to refer to the need for handling the issue of class inequality by choosing appropriate approaches that address the skewed distributions of data. As noted in the previous section, the baseline classification of original datasets shows a very high value of specificity to predict the majority class samples but a very poor sensitivity to predict the minority class samples, which is the class of interest in the imbalanced medical datasets. The use of traditional undersampling techniques shows good progress, mainly by RUS. However, using RUS seems to be not convenient since it eliminates meaningful samples randomly and can also cause overfitting due to the expanding of scar samples without limitations [26]. The performance of ENN is lower than that of RUS except in PIMA dataset, and Tml is the worst one in the experiment. However, the proposed HDUS method has proved to overcome all the other methods in the experiment for all datasets due to the robust measure of Hellinger distance which has the property of skew intensive that is not affected by the class imbalance [30].

To simplify the comparison among the different undersampling methods used in the experiment and to evaluate their efficiency, Figures 1–3 provide a graphical representation for the average values of (sensitivity, F1_m, and Bacc) resulting from the five models applied on four imbalanced medical datasets. As can be seen from the figures, the performances vary when different undersampling techniques are utilized. From Figure 1, it is evident that our HDUS method has made good progress in predicting minority class samples on all datasets in terms of sensitivity. The similar situation can be found through Figure 2 for F1_m, which is the trade-off between precision and recall, and Figure 3 for Bacc, which is the trade-off between sensitivity and specificity.

Table 6: Evaluation results for PIMA dataset using classifiers (KNN, SVM, and DT) and models (baseline, Tml, RUS, ENN, and HDUS).

PIMA		Baseline	Tml	RUS	ENN	HDUS
KNN	Sensitivity (%)	60.06	64.52	70.77	70.57	83.87
	Specificity (%)	83.85	80	73.85	74.92	62.50
	Precision (%)	63.16	56.61	56.41	56.46	50.35
	F1_m (%)	61.57	60.31	62.78	62.73	62.92
	Bacc (%)	70.96	72.26	72.41	72.75	73.19
SVM	Sensitivity (%)	0	66.13	74.42	74	79.03
	Specificity (%)	100	83.85	73.85	76.92	71.50
	Precision (%)	0	58.13	58.54	58.04	57.00
	F1_m (%)	0	61.87	65.53	65.06	66.23
	Bacc (%)	50	74.99	74.63	75.86	75.27
DT	Sensitivity (%)	61.29	69.35	70.97	70.97	91.90
	Specificity (%)	79.23	66.92	58.46	68.46	66.77
	Precision (%)	58.46	50	44.9	50.76	56.98
	F1_m (%)	59.84	58.11	55	59.19	70.34
	Bacc (%)	70.26	68.14	64.71	69.71	79.34
AVG	Sensitivity (%)	39.78	66.67	72.05	71.85	84.93
	Specificity (%)	87.69	76.92	68.72	73.43	66.92
	Precision (%)	40.54	54.91	53.28	54.09	54.78
	F1_m (%)	40.16	60.22	61.26	61.72	66.50
	Bacc (%)	63.74	71.8	70.58	72.77	75.93

Table 7: Evaluation results for THS dataset using classifiers (KNN, SVM, and DT) and models (baseline, Tml, RUS, ENN, and HDUS).

THS		Baseline	Tml	RUS	ENN	HDUS
KNN	Sensitivity (%)	0.00	0.00	42.86	4.76	23.81
	Specificity (%)	100.00	98.97	60.82	91.75	75.26
	Precision (%)	0.00	0.00	19.15	11.11	19.24
	F1_m (%)	0.00	0.00	26.47	6.67	21.28
	Bacc (%)	50.00	49.48	51.84	48.26	49.53
SVM	Sensitivity (%)	0.00	0.00	66.67	4.76	71.43
	Specificity (%)	100.00	100.00	47.42	91.75	44.27
	Precision (%)	0.00	0.00	21.54	11.11	21.13
	F1_m (%)	0.00	0.00	32.56	6.67	32.61
	Bacc (%)	50.00	50.00	57.04	48.26	57.84
DT	Sensitivity (%)	23.81	14.29	42.86	38.10	80.95
	Specificity (%)	87.63	91.75	48.45	81.44	40.02
	Precision (%)	29.41	27.27	15.25	30.77	25.99
	F1_m (%)	26.32	18.75	22.50	34.04	39.33
	Bacc (%)	55.72	53.02	45.66	59.77	60.48
AVG	Sensitivity (%)	7.94	4.76	50.79	15.87	58.73
	Specificity (%)	95.88	96.91	52.23	88.32	53.18
	Precision (%)	9.80	9.09	18.65	17.66	22.12
	F1_m (%)	8.77	6.25	27.18	15.79	31.07
	Bacc (%)	51.91	50.83	51.51	52.09	55.95

Table 8: Evaluation results for BC dataset using classifiers (KNN, SVM, and DT) and models (baseline, Tml, RUS, ENN, and HDUS).

BC		Baseline	Tml	RUS	ENN	HDUS
KNN	Sensitivity (%)	33.33	44.44	57.11	50	61.11
	Specificity (%)	84.91	70.36	65.81	70.36	73.58
	Precision (%)	42.86	40	40.74	42.86	44.00
	F1_m (%)	37.5	42.1	47.56	46.16	51.16
	Bacc (%)	59.12	57.4	61.46	60.18	67.35
SVM	Sensitivity (%)	22.22	38.89	66.67	44.44	66.67
	Specificity (%)	94.34	88.68	64.15	83.02	69.81
	Precision (%)	57.14	53.85	38.71	47.06	42.86
	F1_m (%)	32	45.16	48.98	45.71	52.17
	Bacc (%)	58.28	63.78	65.41	63.73	68.24
DT	Sensitivity (%)	38.89	38.89	44.44	44.44	77.78
	Specificity (%)	66.04	64.15	62.26	69.81	66.04
	Precision (%)	28	26.92	28.57	33.33	43.75
	F1_m (%)	32.56	31.82	34.78	38.09	56.00
	Bacc (%)	52.46	51.52	53.35	57.13	71.91
AVG	Sensitivity (%)	31.48	40.74	56.07	46.29	68.52
	Specificity (%)	81.76	74.4	64.07	74.4	69.81
	Precision (%)	42.67	40.26	36.01	41.08	43.54
	F1_m (%)	36.23	40.5	43.85	43.53	53.11
	Bacc (%)	56.62	57.57	60.07	60.35	69.17

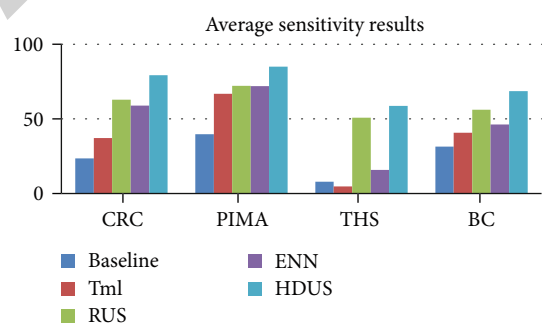


Figure 1: The average sensitivity results using models (baseline, Tml, RUS, ENN, and HDUS) for four datasets.

Regarding the classification methods, it is worth to remark that the benefit of carrying out classification increases when the class imbalance issue is appropriately addressed. In our experiment, different classification algorithms may benefit from the adoption of the HDUS model. In particular, DT achieves the best performance with HDUS. It also obtained the best results with Tml and ENN, whereas SVM is more appropriate for the RUS method. As shown in Table 9, the average results of used classifiers with experimented undersampling methods in the four datasets achieved improvement in predicting the minority class samples through sensitivity, F1_m, and Bacc.

Finally, the results of the proposed HDUS model should be considered as a preliminary experiment, but a promising

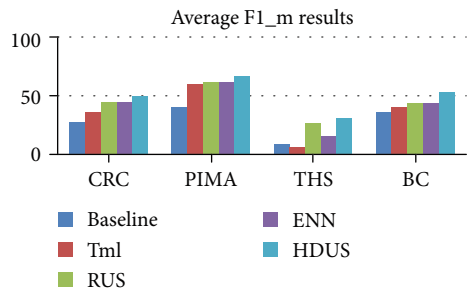


Figure 2: The average F1_m results using models (baseline, Tml, RUS, ENN, and HDUS) for four datasets.

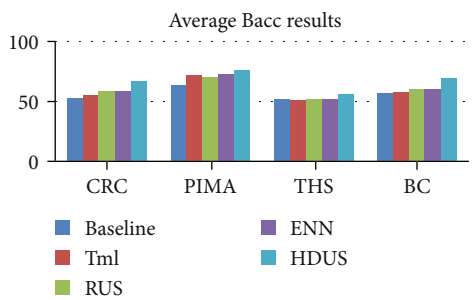


Figure 3: The average Bacc results using models (baseline, Tml, RUS, ENN, and HDUS) for four datasets.

Table 9: The average results using classifiers (KNN, SVM, and DT) with models (baseline, Tml, RUS, ENN, and HDUS) on four datasets.

		Baseline	Tml	RUS	ENN	HDUS
KNN	Sensitivity (%)	30.70	36.55	58.37	47.02	62.28
	Specificity (%)	88.54	81.18	65.12	74.34	65.39
	Precision (%)	37.54	33.47	38.60	37.24	36.88
	F1_m (%)	33.59	34.92	46.06	40.82	45.76
	Bacc (%)	59.37	58.87	61.77	60.68	63.84
SVM	Sensitivity (%)	7.03	33.61	67.63	42.57	73.38
	Specificity (%)	96.66	88.71	60.01	80.62	60.36
	Precision (%)	20.06	37.86	38.49	38.73	39.10
	F1_m (%)	10.34	35.19	48.03	39.98	49.85
	Bacc (%)	51.85	61.16	63.94	61.69	66.87
DT	Sensitivity (%)	39.82	41.91	55.25	55.04	82.91
	Specificity (%)	75.34	70.51	54.02	65.89	57.44
	Precision (%)	36.60	33.61	30.10	36.66	41.65
	F1_m (%)	37.86	36.23	38.60	43.59	54.79
	Bacc (%)	57.58	56.21	54.64	60.47	70.17

method in the application of undersampling the imbalanced medical dataset to improve the classification performance of minor class samples.

7. Conclusion

This paper proposed a novel model, HDUS, that handles the imbalanced classification problem in the medical datasets to improve the classification of the minority disease class. HDUS works to reduce the majority class instances by using the Hellinger distance to calculate the similarity between majority class instance and minority class instances. Then, HDUS selects a subset from the majority class instances having the highest similarity values that are shown to perform well in combination with the original minority class instances. The experiment was conducted on four imbalanced medical datasets using three classifiers to compare HDUS with a baseline model and three selective undersampling models. The performance results show that HDUS could achieve significant improvement over the selective models in terms of sensitivity, which is highly desirable in the medical domain, F1_measure, and balanced accuracy. HDUS has proved to be a promising model for rebalancing the imbalanced medical datasets which contain a few but important cases of disease class.

In a future work, we encourage comparing HDUS with other sampling techniques for the same classifiers or using other classifiers or even utilizing a larger number of medical datasets with different characteristics. We also suggest integrating the proposed model with other sampling techniques to handle the imbalanced classification problem in medical datasets.

Data Availability

The Colorectal Cancer Dataset is not publicly available; it is from the Southampton University Hospital and has been used with approval from the responsible surgeon (co-author), and the data are all anonymous. The datasets "PIMA", "Thoracic surgery", and "Breast Cancer" are openly available at the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] A. Mahani and A. R. B. Ali, "Classification problem in imbalanced datasets," in *Recent Trends in Computational Intelligence*, pp. 1–23, IntechOpen, 2020.

[2] Y. Liu, Y. Wang, X. Ren, H. Zhou, and X. Diao, "A classification method based on feature selection for imbalanced data," *IEEE Access*, vol. 7, pp. 81794–81807, 2019.

[3] C. Seiffert, T. M. Khoshgoftaar, J. van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[4] F. Feng, K. C. Li, J. Shen, Q. Zhou, and X. Yang, "Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification," *IEEE Access*, vol. 8, pp. 69979–69996, 2020.

[5] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern*

Recognition and Artificial Intelligence, vol. 23, no. 4, pp. 687–719, 2009.

[6] S. Abdellatif, M. A. Ben Hassine, S. Ben Yahia, and A. Bouzeghoub, "ARCID: a new approach to deal with imbalanced datasets classification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10706, pp. 569–580, LNCS, 2018.

[7] K. K. Bejjanki, J. Gyani, and N. Gugulothu, "Class imbalance reduction (CIR): a novel approach to software defect prediction in the presence of class imbalance," *Symmetry*, vol. 12, no. 3, p. 407, 2020.

[8] V. S. Akondi, V. Menon, J. Baudry, and J. Whittle, "Novel K-means clustering-based undersampling and feature selection for drug discovery applications," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*no. Mcl, pp. 2771–2778, San Diego, CA, USA, USA, November 2019.

[9] K. Yoon and S. Kwek, "A data reduction approach for resolving the imbalanced data issue in functional genomics," *Information Sciences*, vol. 16, no. 3, pp. 295–306, 2007.

[10] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, and G.-T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," *Information Sciences*, vol. 477, pp. 47–54, 2019.

[11] Q.-Y. Yin, J.-S. Zhang, C.-X. Zhang, and N.-N. Ji, "A novel selective ensemble algorithm for imbalanced data classification based on exploratory undersampling," *Mathematical Problems in Engineering*, vol. 2014, Article ID 358942, 14 pages, 2014.

[12] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, and M. Buckland, "A hybrid feature selection with ensemble classification for imbalanced healthcare data: a case study for brain tumor diagnosis," *IEEE Access*, vol. 4, pp. 9145–9154, 2016.

[13] K. Polat, "Similarity-based attribute weighting methods via clustering algorithms in the classification of imbalanced medical datasets," *Neural Computing and Applications*, vol. 30, no. 3, pp. 987–1013, 2018.

[14] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," *Neurocomputing*, vol. 135, pp. 32–41, 2014.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[16] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, no. 1, p. 106, 2013.

[17] R. Blagus and L. Lusa, "Evaluation of SMOTE for high-dimensional class-imbalanced microarray data," in *2012 11th International Conference on Machine Learning and Applications*, vol. 2no. 1, pp. 89–94, Boca Raton, FL, USA, December 2012.

[18] A. Wosiak and S. Karbowiak, "Preprocessing compensation techniques for improved classification of imbalanced medical datasets," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, vol. 11, pp. 203–211, Prague, September 2017.

[19] C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis, "Uncertainty based under-sampling for learning naive Bayes classifiers under imbalanced data sets," *IEEE Access*, vol. 8, pp. 2122–2133, 2020.

[20] M. M. Nwe and K. T. Lynn, "KNN-based overlapping samples filter approach for classification of imbalanced data," in *Studies in Computational Intelligence*, vol. 845, pp. 55–73, Springer International Publishing, 2020.

[21] N. Garcia-Pedrajas, J. Pérez-Rodríguez, and A. de Haro-García, "OligoIS: scalable instance selection for class-imbalanced data sets," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 332–346, 2013.

[22] A. de Haro-García, G. Cerruela-García, and N. García-Pedrajas, "Instance selection based on boosting for instance-based learners," *Pattern Recognition*, vol. 96, article 106959, 2019.

[23] M. Blachnik, "Instance selection for classifier performance estimation in meta learning," *Entropy*, vol. 19, no. 11, p. 583, 2017.

[24] G. Yu, J. Tian, and M. Li, "Nearest neighbor-based instance selection for classification," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 75–80, Changsha, China, August 2016.

[25] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets : one-sided selection," *In Icml*, vol. 97, pp. 179–186, 1997.

[26] J. Li, S. Fong, S. Hu, R. K. Wong, and S. Mohammed, "Similarity majority under-sampling technique for easing imbalanced classification problem," in *Communications in Computer and Information Science*, vol. 845, pp. 3–23, Springer Singapore, 2018.

[27] A. Kumari and U. Thakar, "Hellinger distance based oversampling method to solve multi-class imbalance problem," in *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 137–141, Nagpur, India, November 2017.

[28] G.-H. Fu, Y.-J. Wu, M.-J. Zong, and J. Pan, "Hellinger distance-based stable sparse feature selection for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 21, no. 1, p. 121, 2020.

[29] P. Harsha, "Hellinger distance," in *Wiley StatsRef: Statistics Reference Online*, vol. 2011, pp. 1–8, John Wiley & Sons, Ltd, Chichester, UK, 2014.

[30] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012.

[31] "UCI machine learning repository," https://archive.ics.uci.edu/ml/datasets.php.

[32] M. B. Rodrigues, R. V. M. da Nobrega, S. S. A. Alves et al., "Health of things algorithms for malignancy level classification of lung nodules," *IEEE Access*, vol. 6, pp. 18592–18601, 2018.

[33] M. Çınar, M. Engin, E. Z. Engin, and Y. Ziya Ateşçi, "Early prostate cancer diagnosis by using artificial neural networks and support vector machines," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6357–6361, 2009.

[34] M.-W. Huang, C.-W. Chen, W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "SVM and SVM ensembles in breast cancer prediction," *PLoS One*, vol. 12, no. 1, article e0161501, 2017.

[35] E. Aličković and A. Subasi, "Breast cancer diagnosis using GA feature selection and rotation forest," *Neural Computing and Applications*, vol. 28, no. 4, pp. 753–763, 2017.