Sakarya University

Journal of Computer and Information Sciences

e-ISSN 2636-8129

VOLUME 8 ISSUE 1

MARCH 2025

VOLUME: 8 ISSUE: 1
E-ISSN 2636-8129

MARCH 2025
http://saucis.sakarya.edu.tr/tr/

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

SAKARYA
ÜNİVERSİTESİ

The Owner on Behalf of Sakarya University

Editor in Chief

Managing Editors

Editorial Board

Language Editors

Layout Editor

Indexing & Abstracting & Archiving

Scopus®

DOAJ

ULAKBİM

TR DİZİN

EBSCO Central & Eastern European Academic Source

Applied Science & Technology Source

SHERPA/RoMEO

Contents

Research Article

Review

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 8, No. 1, 1-11, 2025
DOI: 10.35377/saucis...1564937

RESEARCH ARTICLE

Comparative Analysis of Machine Learning Models for CO Emission Prediction in Engine Performance

Beytullah Eren[1], **İdris Cesur**[2]

[1]Sakarya University, Faculty of Engineering, Department of Environmental Engineering, Sakarya, Türkiye, ror.org/04ttnw109
[2] Sakarya University of Applied Sciences, Faculty of Technology, Department of Mechanical Engineering, Sakarya, Türkiye, ror.org/01shwhq58

Corresponding author:
Beytullah Eren, Sakarya University,
Faculty of Engineering,
Department of Environmental Engineering
beren@sakarya.edu.tr

Article History:
Received: 10.10.2024
Revised: 17.12.2024
Accepted: 15.01.2025
Published Online: 27.03.2025

ABSTRACT
This study presents a comparative analysis of machine learning models for predicting carbon monoxide (CO) emissions in automotive engines. Four models—Linear Regression, Decision Tree, Random Forest, and Support Vector Regression—were evaluated using a dataset of engine performance parameters and emission measurements. Among these, the Random Forest model demonstrated the highest predictive accuracy, achieving an R^2 score of 0.8965. Feature importance analysis identified nitrogen oxides (NO_X), engine speed (RPM), and hydrocarbons (HC) as the most significant predictors of carbon monoxide emissions. Learning curve analysis provided insights into model generalization and highlighted potential limitations. The study underscores the value of data-driven approaches in optimizing engine design and controlling emissions. The findings contribute to the development of cleaner, more efficient vehicles, supporting sustainability efforts in the automotive industry. This research bridges data science and automotive engineering, offering a framework for advanced emission prediction and control that can be applied to other pollutants and engine types.

Keywords: Carbon monoxide emissions, Machine learning, Random Forest, Engine performance optimization, Emission control, Sustainability, Automotive engineering

1. Introduction

The automotive industry is at a critical crossroads, tasked with enhancing engine performance while significantly reducing harmful emissions. Among these emissions, carbon monoxide (CO) poses a considerable threat to both human health and the environment [1]. As a byproduct of incomplete combustion, CO can lead to severe respiratory issues and, at high concentrations, may even be life-threatening [2]. Moreover, CO contributes to ground-level ozone formation, a major component of smog that exacerbates air quality concerns [3]. To combat these challenges, stringent global regulations, such as the European Union's Euro 6 standards and the United States' Tier 3 regulations, have been implemented, driving the need for innovative emission reduction strategies [4, 5].

Traditional approaches to emission control in internal combustion engines, such as optimizing engine design and using after-treatment systems, often involve trade-offs with engine performance and fuel efficiency [6, 7]. These methods also struggle to address the complex, non-linear interactions between engine parameters and emission outputs, highlighting the limitations of conventional techniques.

Recent advancements in data analytics and machine learning (ML) have introduced new opportunities to tackle these challenges. ML techniques are particularly adept at modeling complex, non-linear relationships between variables, enabling more accurate emission predictions and optimized control strategies [8, 9]. Studies have applied machine learning algorithms to predict pollutants like NO_X and CO_2, demonstrating promising results. Artificial Neural Networks (ANNs) have been effective in predicting NO_X emissions, capturing intricate relationships between operating conditions and outputs [10]. Similarly, Support Vector Machines (SVMs) and Random Forests have shown success in estimating particulate matter and CO_2 emissions, respectively, due to their ability to manage complex feature interactions and robust performance across diverse datasets [11, 12]. Deep learning techniques, particularly hybrid models like Convolutional Neural Networks (CNN)

Cite as: B. Eren and İ. Cesur, "Comparative analysis of machine learning models for CO emission prediction in engine performance," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 1-11, 2025. doi: 10.35377/saucis...1564937

combined with Long Short-Term Memory (LSTM) networks, have further enhanced the modeling of temporal emission patterns, particularly under transient operating conditions [13].

Despite these advancements, limited research has focused on comparing multiple ML models specifically for CO emission prediction. CO emissions are uniquely sensitive to various engine parameters and operating conditions, making their prediction particularly challenging [14]. Moreover, studies rarely address feature importance, which is critical for understanding the key engine parameters driving CO emissions. The lack of comprehensive comparative analyses leaves gaps in identifying the most effective ML techniques and their practical implications for emission reduction.

This study addresses these gaps by evaluating the performance of four widely used machine learning models—Linear Regression, Decision Tree, Random Forest, and Support Vector Regression (SVR)—in predicting CO emissions based on engine performance parameters. The study aims to: (1) assess the predictive accuracy of these models across varying engine operating conditions, (2) identify the most influential engine parameters through feature importance analysis, (3) evaluate the generalization capabilities and limitations of the models using learning curves, and (4) provide actionable insights for integrating ML techniques into emission control strategies and engine design. By offering a comparative analysis of machine learning models and their practical applications, this research contributes to the growing body of knowledge on data-driven emission reduction approaches and provides a foundation for future studies and industrial applications in optimizing cleaner and more efficient automotive technologies.

2. Materials and Method

2.1. Dataset Description

The dataset used in this study was derived from a series of controlled engine performance tests conducted on a two-cylinder, electronic injection, naturally aspirated, water-cooled, four-stroke, spark ignition engine. This engine type was selected due to its widespread use in small- to medium-sized passenger vehicles, making it highly representative of real-world applications. Additionally, its relatively simple design and operation allow for a more focused analysis of the relationship between engine parameters and CO emissions without interference from complex subsystems, such as turbocharging or advanced after-treatment devices.

The experiments were performed under full-load conditions in a controlled laboratory environment to ensure consistency and reproducibility. Key engine parameters were varied systematically, including fuel composition (gasoline-ethanol blends of E10, E20, and E30) and engine speed (RPM) across a broad operational range. These conditions were chosen to represent typical and extreme scenarios encountered during real-world engine operation, providing a comprehensive dataset for modeling.

Emission values, including nitrogen oxides (NOx), hydrocarbons (HC), and carbon monoxide (CO), were measured using a calibrated MRU Delta 1600 L exhaust gas analyzer. To minimize variability, five repeated measurements were taken for each test point, and the average values were used for analysis. In total, 90 data points were collected, representing a diverse range of operating conditions and configurations relevant to CO emission prediction.

While the dataset provides a representative sample of typical spark ignition engine conditions, it is important to note its limitations. The controlled laboratory environment excludes external variables such as temperature, humidity, and altitude, which can influence real-world emission outputs. Furthermore, the focus on a single engine type does not account for variations in design and operation seen in turbocharged or diesel engines. Future research should expand the dataset to include these factors for broader applicability.

Feature scaling was applied using the StandardScaler function from the scikit-learn library to normalize input variables, ensuring consistency across features and facilitating model training [15].

2.2. Machine Learning Models

In this study, four machine learning models were implemented to predict carbon monoxide (CO) emissions based on the input features, each with unique characteristics and strengths for handling different data relationships. Below is a detailed explanation of each model:

Linear Regression: Linear Regression is a fundamental statistical model that serves as a baseline for comparison. It assumes a linear relationship between the independent variables (engine parameters) and the dependent variable (CO emissions). The model calculates coefficients for each feature by minimizing the sum of squared residuals, resulting in a straightforward predictive framework. While efficient and interpretable, this method struggles with non-linear relationships or feature interactions [16].

Decision Tree: The Decision Tree model is a non-linear algorithm that partitions the dataset into subsets based on feature values. The model splits the data at decision nodes to minimize an impurity measure, such as the Gini index or entropy, in classification or Mean Squared Error (MSE) in regression. This model is particularly effective at capturing non-linear patterns and interactions between features. However, a single tree can be prone to overfitting, especially in datasets with noise [17].

Random Forest: Random Forest is an ensemble learning method that combines the predictions of multiple decision trees. Each tree is built on a random subset of data and features, introducing diversity and reducing overfitting. The final prediction is derived by averaging the predictions (for regression tasks) from all the individual trees. This robustness to overfitting and ability to handle complex, non-linear relationships makes Random Forest particularly suited for high-dimensional and noisy datasets [18].

Support Vector Regression (SVR): SVR is a versatile algorithm that maps input features into a high-dimensional space using a kernel function (e.g., linear, polynomial, or radial basis function) to capture linear and non-linear relationships. The model aims to fit a regression hyperplane within a tolerance margin, minimizing prediction error while controlling model complexity. This makes SVR particularly effective for datasets with intricate, non-linear patterns. However, its performance can be sensitive to parameter tuning, such as the choice of the kernel, regularization parameter (C), and epsilon (ε), which defines the margin of tolerance [19].

All models were implemented using the Python programming language and the scikit-learn library, a well-documented toolkit for machine learning applications [15]. The best-performing model also facilitated feature importance analysis, providing valuable insights into the relative contributions of input features to CO emission predictions.

2.3. Experimental Procedure

The experiments were conducted using a two-cylinder spark plug test engine on an engine test stand to determine engine performance parameters and exhaust emissions. An electric dynamometer was employed to provide full load conditions, while a mass-scale fuel measurement system was used to measure fuel consumption at varying engine speeds. Exhaust emissions were measured using an MRU Delta 1600L emission analyzer. The experiments were performed under full load conditions at engine speeds ranging from 1400 to 3400 rpm, in 400 rpm intervals. Initially, gasoline was used as the fuel, followed by ethanol-blended fuels. The experimental results obtained were subsequently used as input data for model development.

The experimental procedure included key steps to ensure a rigorous evaluation of machine learning models for predicting carbon monoxide (CO) emissions. The dataset consisted of 90 observations, incorporating engine performance parameters such as NO_X, RPM, HC, and fuel composition, along with the corresponding CO emission values. The data was split into training (80%) and testing (20%) subsets, allowing the models to learn patterns from the training data and evaluate their predictive performance on unseen testing data. The overall experimental workflow is illustrated in Figure 1.

The models' predictive performance was assessed using two primary metrics: the coefficient of determination (R^2) and the Mean Squared Error (MSE). R^2 quantified the proportion of variance in CO emissions explained by the model, reflecting its predictive accuracy, while MSE measured the average squared difference between predicted and actual CO values, highlighting the error magnitude. Together, these metrics provided a comprehensive evaluation of the models' performance.

To further enhance the interpretability of the best-performing model, a feature importance analysis was conducted to identify the relative contributions of input features to CO emission predictions [20]. Additionally, learning curves were generated to examine how model performance varied with the size of the training dataset, offering insights into the bias-variance trade-off for each model [21].

Finally, scatter plots and residual analyses were prepared for each model to visually compare actual and predicted values, and to identify any systematic patterns or biases in the predictions. These methodological steps ensured a robust evaluation process, facilitating reliable insights into the models' applicability for engine performance optimization and emission control strategies.

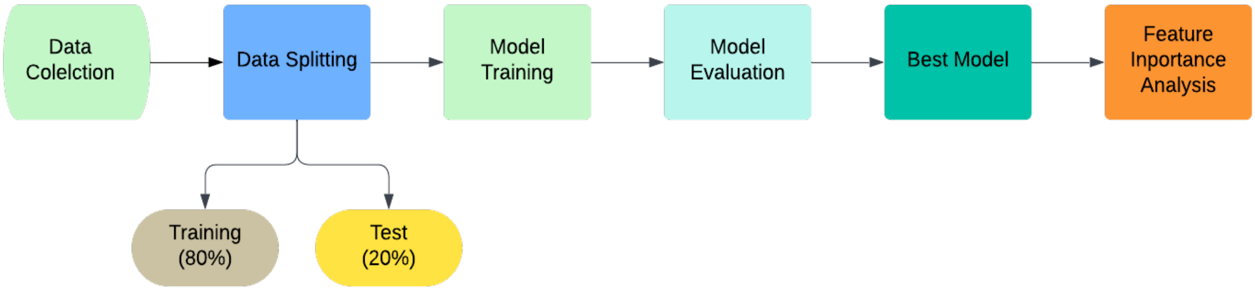


Figure 1. Workflow of the Study for CO Emission Prediction

3. Results and Discussion

In this study, we compared the performance of four different models - Linear Regression, Decision Tree, Random Forest, and Support Vector Regression (SVR) - in predicting CO emissions based on engine performance parameters. The results of our analysis are presented and discussed below.

3.1. Model Comparison and Evaluation

Figure 2 presents a comparative analysis of the performance of all four models. The graph shows each model's Mean Squared Error (MSE) and R2 scores. This analysis shows that the Random Forest model consistently outperformed the other models, achieving the lowest MSE and highest R^2 score. The Linear Regression model, on the other hand, showed the poorest performance, indicating that the relationship between the input features and CO emissions is likely non-linear. Linear Regression's inability to capture complex patterns stems from its strict assumption of linearity, which does not reflect real-world combustion dynamics. Similarly, SVR demonstrated moderate performance due to its sensitivity to parameter tuning and reliance on kernel functions. The limited dataset size may have also constrained SVR's ability to generalize effectively, highlighting the need for robust models like Random Forest in capturing non-linear relationships

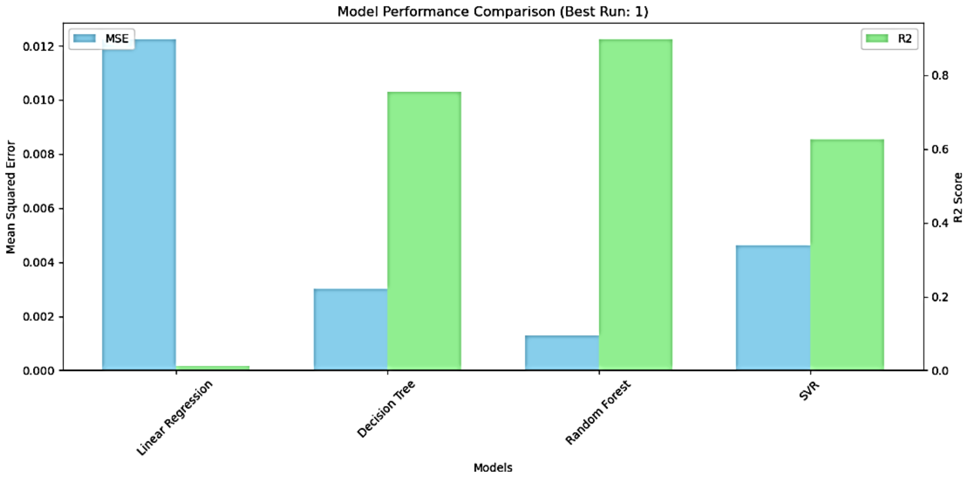


Figure 2. Comparative Performance of Machine Learning Models

To assess the consistency of our models across multiple runs, we analyzed the distribution of R^2 scores, as shown in Figure 3. This box plot reveals that the Random Forest model achieved the highest median R^2 score and demonstrated the least variability across runs. This suggests that Random Forest is not only the most accurate but also the most reliable model for this prediction task.

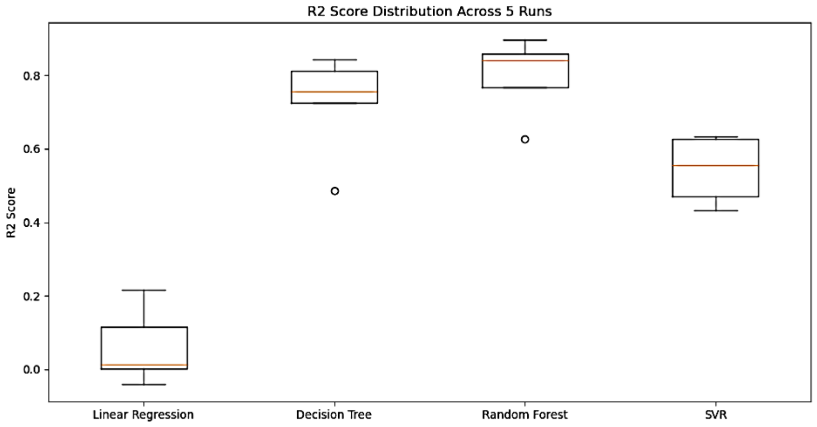


Figure 3. Distribution of R^2 Scores Across Multiple Runs

Figure 4 illustrates the importance of different features in predicting CO emissions, as determined by the Random Forest model. The analysis reveals that NO_X (nitrogen oxides), RPM (engine speed), and HC (hydrocarbons) are the most significant predictors of CO emissions. NO_X and RPM show almost equal importance, with the highest scores, followed closely by HC.

Interestingly, while the fuel composition (Gasoline and Ethanol) does impact CO emissions, their importance is considerably lower compared to the engine operation parameters and other emission components. This suggests that the engine's operating conditions and the formation of other pollutants play a more crucial role in determining CO emissions than the fuel mixture alone. These insights are particularly valuable for engine designers and environmental engineers focusing on emission reduction strategies. The high importance of NO_X and HC in predicting CO emissions indicates a complex interplay between different pollutants in the combustion process. As a top predictor, engine speed (RPM) suggests that optimizing engine operation across different speed ranges could be a key factor in controlling CO emissions.

The relatively lower importance of fuel composition (Gasoline and Ethanol) is noteworthy. While altering fuel mixtures is often considered a strategy for emission control, this analysis suggests that more significant gains might be achieved by focusing on engine operating parameters and technologies that simultaneously reduce NO_X, HC, and CO emissions. This feature importance analysis provides a clear direction for prioritizing efforts in emission control: focusing on technologies and strategies that address NO_X and HC emissions while optimizing engine speed could potentially yield the most significant reductions in CO emissions. Additionally, it highlights the importance of a holistic approach to emission control, considering the interdependencies between different pollutants and engine operating conditions.

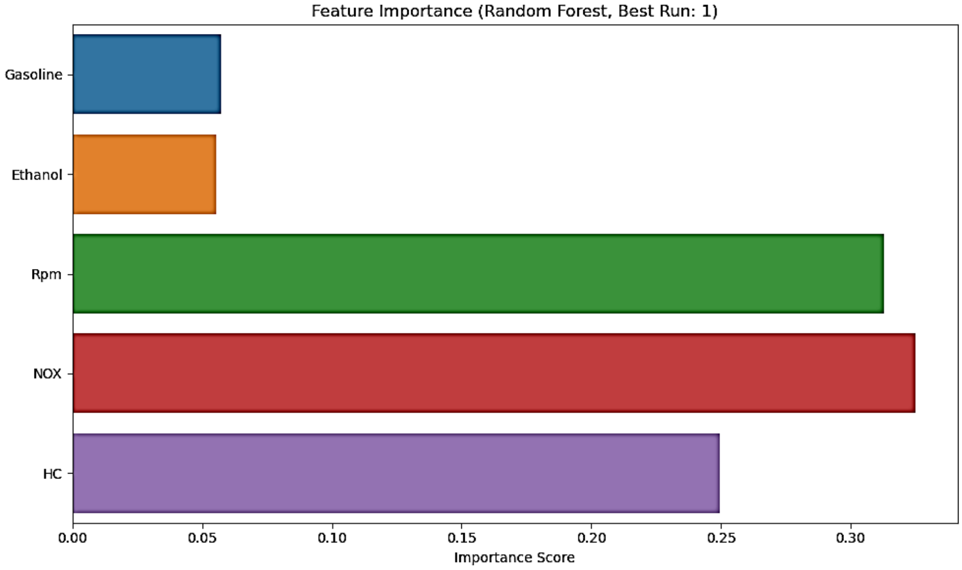


Figure 4. Feature Importance in CO Prediction

The correlation heatmap presented in Figure 5 reveals several significant relationships between input features, providing crucial insights into engine behavior and emission patterns. As expected, Gasoline and Ethanol show a perfect negative correlation (-1.0), reflecting their complementary nature in the fuel mixture. Engine speed (RPM) demonstrates strong correlations with both NO_X (0.7) and HC (-0.9), indicating that higher speeds tend to increase NO_X production while reducing HC emissions, likely due to changes in combustion conditions. The moderate negative correlation (-0.39) between NO_X and HC emissions highlights the typical trade-off in emission control strategies. Interestingly, Gasoline content correlates positively with both NO_X (0.59) and HC (0.32) emissions, while Ethanol shows inverse correlations of equal magnitude, suggesting that increasing ethanol content might help reduce these emissions. The absence of a significant correlation between RPM and fuel composition implies that engine speed-based optimization strategies could be effective across various fuel mixtures. These relationships underscore the complex relationship between engine operations, fuel composition, and emissions, emphasizing the need for a holistic approach in developing effective CO emission prediction models and reduction strategies.

3.2. Best Model Analysis

Figure 6 presents a comparative analysis of actual vs predicted CO values for four models, revealing significant variations in their predictive capabilities. The Random Forest model emerges as the best performer with an impressive R^2 score of 0.8965, demonstrating a strong alignment between predicted and actual values across the CO emission range. In contrast, the Linear Regression model (R^2: 0.0111) shows poor performance, indicating the highly non-linear nature of the CO emission prediction problem. The Decision Tree model (R^2: 0.7542) and SVR (R^2: 0.6236) fall between these extremes, with the

Decision Tree showing better consistency than SVR, particularly at lower CO values. Notably, even the top-performing Random Forest model exhibits slight tendencies to underpredict at higher CO values and overpredict at lower ones, suggesting potential areas for further refinement. These results underscore the complexity of CO emission processes and the superiority of ensemble methods like Random Forest in capturing the complex, non-linear relationships between engine parameters and emissions. The significant performance gap observed across models highlights the importance of selecting appropriate machine learning techniques for accurate CO emission predictions in engine performance analysis.

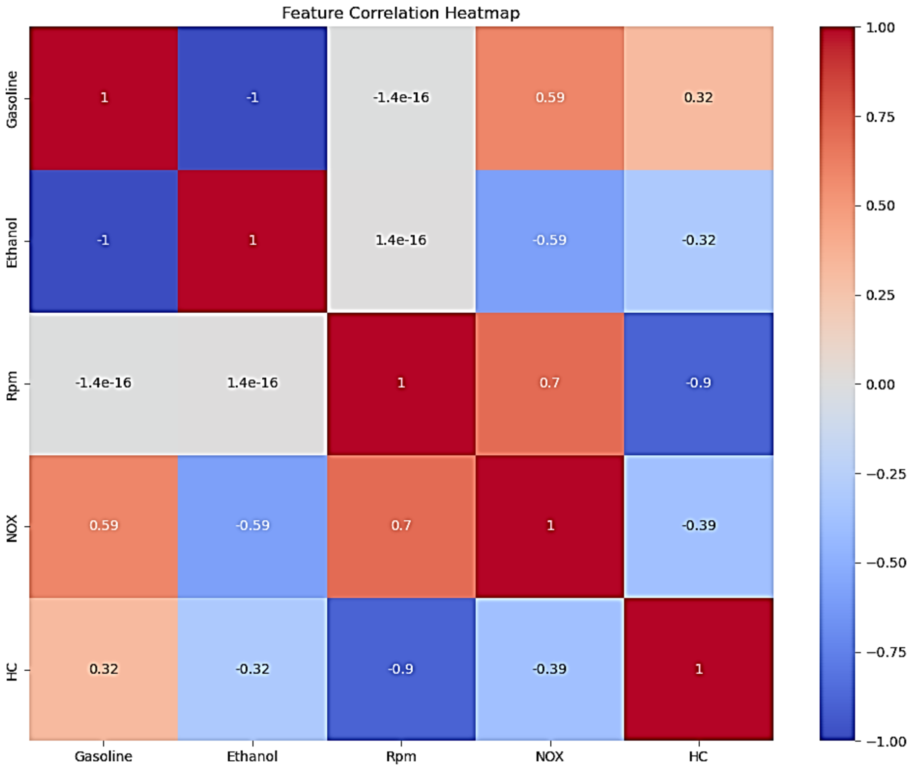


Figure 5. Correlation Analysis of Input Features

The residual plot for the best performing Random Forest model (Figure 7) displays a random scatter of points around the zero line, with residuals primarily falling within a -0.06 to 0.06 range. This pattern indicates that the model performs consistently across different CO emission levels without significant systematic bias. The even distribution of residuals above and below the zero line suggests that the model has captured most of the underlying patterns in the data. However, a slightly wider spread of residuals in the mid-range predictions (1.30 to 1.40) and the presence of a few outliers' hint at some complexity in CO emission behavior that the model doesn't fully capture. Notably, the residuals appear more concentrated around zero for lower and higher predicted values, potentially indicating better model performance at these extremes. These observations confirm the Random Forest model's strong overall predictive capability while highlighting areas for potential improvement, particularly in handling mid-range predictions and addressing outlier cases. The residual analysis thus supports the model's reliability while pointing to opportunities for further refinement in CO emission prediction accuracy.

Figure 8 presents the learning curve for the best performing Random Forest model. The graph shows a converging trend between the training and cross-validation scores as the number of training examples increases. Initially, there's a significant gap between the two scores, with the training score starting high and the cross-validation score starting low. As more data is introduced, this gap narrows considerably, stabilizing both scores at higher levels. The training score remains consistently high, slightly decreasing as more data is added, while the cross-validation score shows a steep increase before leveling off. This pattern indicates that the model has good generalization ability, effectively learning from the data without overfitting. The convergence of scores suggests that the model has reached a point of reducing returns in terms of performance gain from additional data. However, the slight upward trend in the cross-validation score at the highest number of training examples hints that there might still be small potential for improvement with even more data. This learning curve demonstrates that the Random Forest model is well-tuned for the current dataset, balancing complexity with generalization, and is likely to perform consistently on unseen data within the same domain.

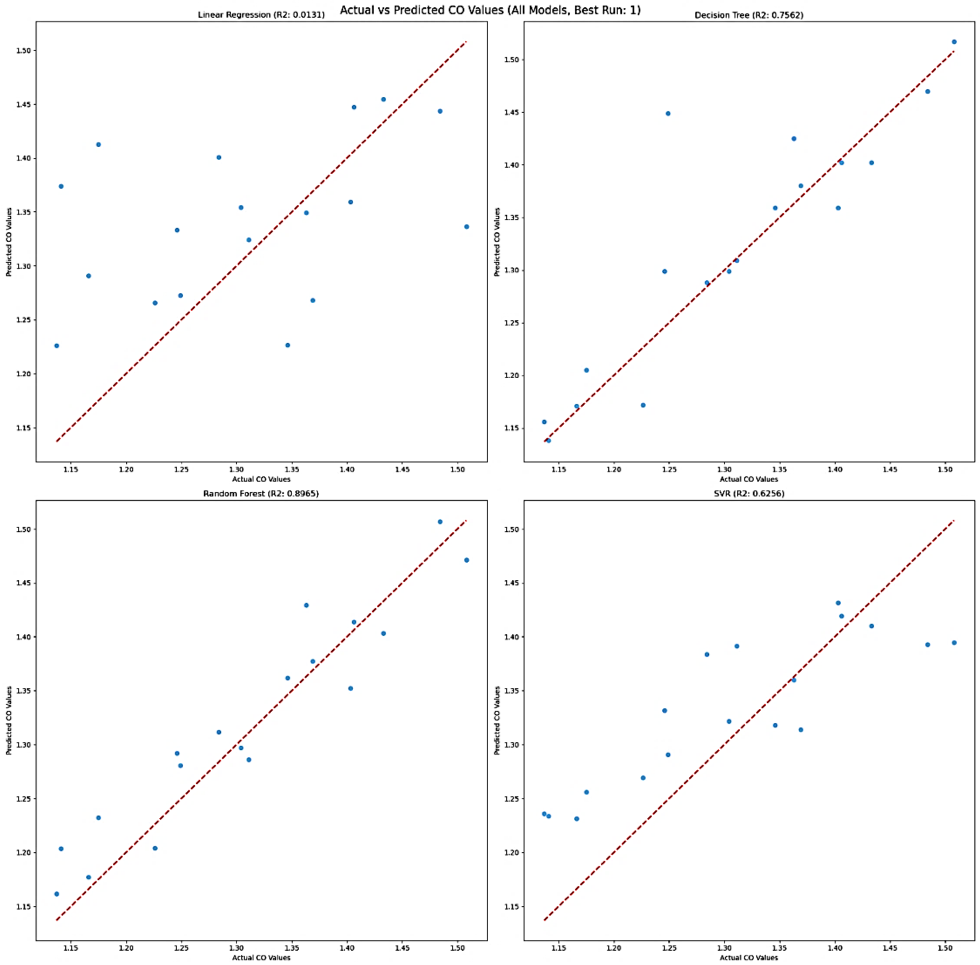


Figure 6. Actual vs Predicted CO Values for the Best Performing Model

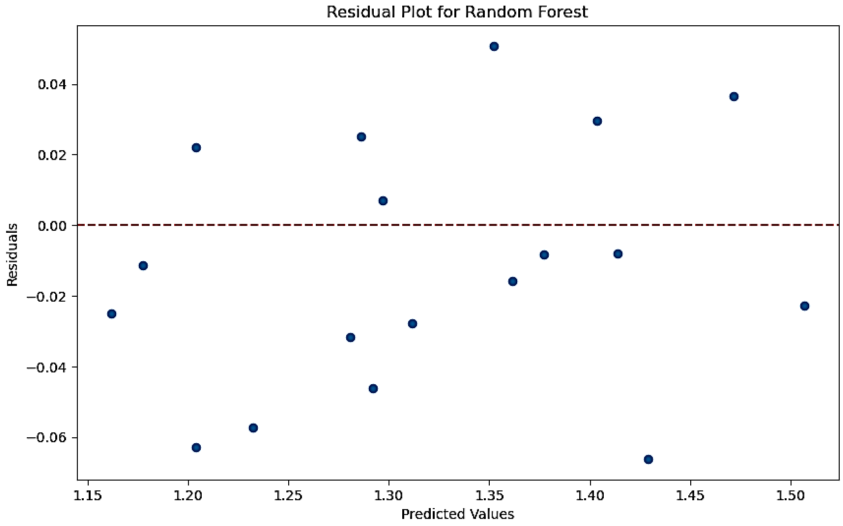


Figure 7. Residual Analysis of the Best Performing Model

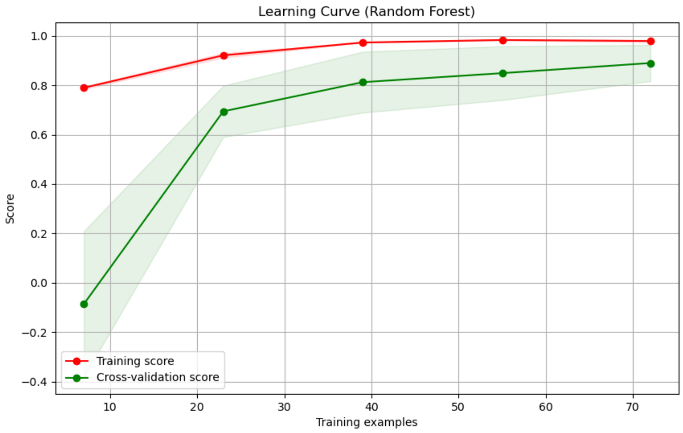


Figure 8. Learning Curve Analysis of the Best Performing Model

Finally, Figure 9 compares actual and predicted CO values for the best performing Random Forest model (R^2: 0.8965). This analysis reveals that our model captures the overall trends and significant fluctuations in CO emissions remarkably well across the sample range. The predicted values closely follow the actual values' pattern, accurately reflecting gradual changes and sharp peaks in CO emissions. The model demonstrates a strong ability to capture the trends and fluctuations in the actual CO values across the sample range. The Random Forest model demonstrates strong predictive capability, effectively tracking CO emissions' complex, non-linear behavior over the sample range. This performance underscores the model's potential for reliable CO emission forecasting in engine performance analysis.

The Random Forest model demonstrated superior performance, achieving an R^2 score of 0.8965, reflecting a strong alignment between actual and predicted CO values. This result underscores the model's ability to capture the complex, non-linear relationships inherent in the dataset. However, its applicability beyond the specific dataset warrants further investigation. While the dataset includes a range of engine speeds and fuel compositions, it is limited to a single engine type and controls laboratory conditions. Real-world scenarios introduce additional variables that may significantly influence combustion processes and emissions, such as variations in temperature, humidity, and altitude; long-term engine wear and maintenance inconsistencies; and diverse driving patterns, including stop-and-go traffic or highway cruising. These factors, absent from the current dataset, highlight the need for broader validation.

Future research should address these limitations by validating the Random Forest model on datasets encompassing diverse engine configurations, such as turbocharged and diesel engines, and incorporating external variables like environmental conditions. Such efforts would enhance the model's robustness across a wider range of operating scenarios. Additionally, while the model showed excellent performance, its tendency to underpredict higher CO emission values suggest opportunities for refinement. Incorporating advanced ensemble techniques, such as Gradient Boosting or hybrid models, could improve generalization across diverse datasets. Furthermore, integrating time-series data and transient operating conditions (e.g., rapid acceleration or deceleration) could enhance the model's adaptability to real-world applications. By addressing these gaps, the Random Forest model could become a more versatile tool for real-time CO emission monitoring and control in the automotive industry.

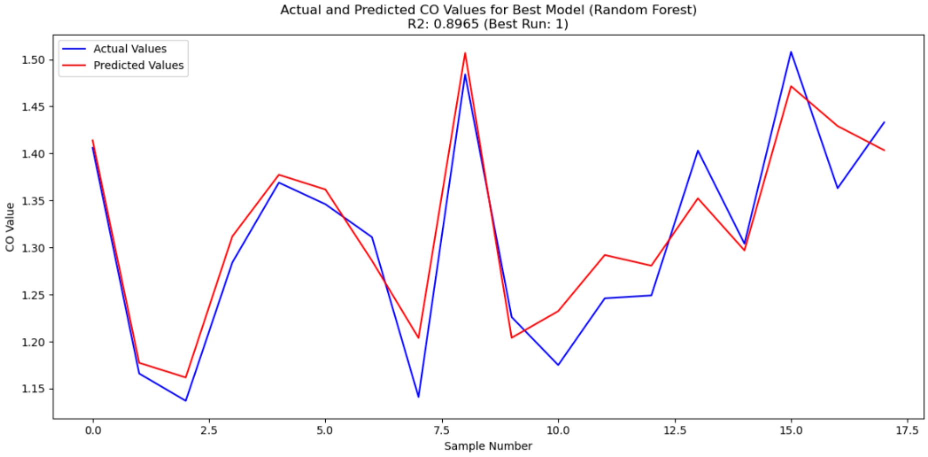


Figure 9. Comparison of Actual and Predicted CO Values

Our analysis demonstrates that machine learning models, particularly Random Forest, can effectively predict CO emissions based on engine performance parameters. The Random Forest model achieved the highest R^2 score of 0.8965, significantly outperforming other models such as Linear Regression, Decision Tree, and SVR. This superior performance can be attributed to Random Forest's ability to capture complex, non-linear relationships and its robustness to outliers. The model shows strong predictive capability across a range of CO emission values, as evidenced by the residual analysis and comparison. These findings have significant implications for engine design and emission control strategies. The feature importance analysis revealed that NO_X, RPM, and HC are the most important predictors of CO emissions, providing valuable insights for targeted emission reduction efforts. Manufacturers can optimize engine designs by accurately predicting CO emissions based on engine parameters to minimize emissions without compromising performance. For instance, they can focus on optimizing engine speed ranges and developing technologies that simultaneously address NO_X, HC, and CO emissions. Furthermore, the model's ability to capture trends in CO emissions can aid in real-time monitoring and control systems. This could lead to more adaptive and efficient emission control strategies, potentially improving overall engine performance while meeting stringent environmental regulations.

3.3. Analysis of Model Performance

The performance of the four machine learning models (Linear Regression, Decision Tree, Random Forest, and Support Vector Regression (SVR)) revealed distinct patterns in their ability to predict carbon monoxide (CO) emissions. Linear Regression and SVR exhibited relatively lower performance than Random Forest and Decision Tree, as shown in Figure 2. Below, we discuss the potential reasons for this underperformance.

Linear Regression, a baseline model, assumes a strict linear relationship between the independent variables (engine parameters) and the dependent variable (CO emissions). However, the combustion process in engines is inherently complex and involves non-linear interactions among various parameters such as NO_X, RPM, and HC. The model's inability to capture these non-linear relationships results in poor predictive performance, with an R^2 score of 0.0111. Additionally, Linear Regression lacks the flexibility to handle feature interactions, making it unsuitable for datasets where input variables exhibit strong dependencies, as observed in the correlation analysis in Figure 5.

Support Vector Regression demonstrated moderate performance (R^2: 0.6236) but underperformed compared to the ensemble-based Random Forest model. While SVR can handle non-linear relationships using kernel functions, its effectiveness depends on appropriate parameter tuning. In this study, an RBF (Radial Basis Function) kernel was used, which, while generally effective, may not have fully captured the complexity of CO emission patterns due to the dataset's limited size and diversity. Furthermore, SVR struggles with outliers, as the epsilon-insensitive loss function can inadvertently exclude essential data points, reducing predictive accuracy in highly variable scenarios.

In contrast, Random Forest consistently outperformed Linear Regression and SVR, achieving an R^2 score of 0.8965. This model's superior performance can be attributed to its ability to capture non-linear relationships, handle complex feature interactions, and remain robust to noise and outliers. By leveraging multiple decision trees and averaging their predictions, Random Forest effectively generalizes across a wide range of input conditions.

The underperformance of Linear Regression and SVR underscores the importance of selecting models that align with the underlying data characteristics. Models like Random Forest, which are inherently flexible and robust, are better suited for CO emissions' complex, non-linear nature. These findings highlight the necessity of comparative analyses to identify the most suitable algorithms for specific applications.

4. Conclusion

This study has demonstrated the efficacy of machine learning techniques, notably the Random Forest algorithm, in predicting carbon monoxide (CO) emissions based on engine performance parameters. Among the models evaluated (Linear Regression, Decision Tree, Support Vector Regression (SVR), and Random Forest), the ensemble-based Random Forest model achieved the highest predictive accuracy, with an R^2 score of 0.8965. Feature importance analysis revealed that NO_X, RPM, and HC levels were the most significant predictors of CO emissions, providing valuable insights into targeted emission reduction strategies. The learning curve analysis highlighted the generalization capabilities of the models and identified areas for further refinement.

The findings have significant practical applications in the automotive industry. The machine learning models developed in this study can be integrated into the engine design process to optimize configurations and reduce emissions without compromising performance. Real-time emission monitoring systems based on these models could dynamically adjust engine parameters to minimize CO emissions under varying conditions, promoting more environmentally friendly driving practices. Additionally, these systems could guide drivers with feedback on optimal acceleration patterns and gear shifts, supporting sustainability goals in the transportation sector.

This study has certain limitations that should be acknowledged. The dataset used was specific to certain engine types, which may restrict the generalizability of the findings across other engine designs or operating conditions. Future work should address these limitations by including diverse engine configurations, external factors such as environmental conditions, and real-world driving scenarios to enhance the robustness and applicability of the models. The models also exhibited limitations in predicting extreme values, suggesting opportunities for improvement in handling outliers. Furthermore, real-world factors such as environmental conditions, fuel quality variations, and long-term engine wear were not considered in this study, potentially limiting the applicability of the models in broader contexts.

Future studies should incorporate data from diverse engine types (e.g., turbocharged and diesel) and environmental conditions (e.g., temperature, humidity, altitude) to improve model generalizability. Developing advanced ensemble methods or hybrid models could enhance prediction accuracy, particularly for extreme values. Integrating real-time data and time-series analysis techniques could enable the creation of adaptive predictive models capable of responding dynamically to changing conditions. Expanding the study to include more diverse scenarios and pollutants could further improve the utility and relevance of the models in supporting cleaner and more efficient engine technologies.

This research represents a significant step forward in applying machine learning to emission prediction in automotive engines. By providing accurate, data-driven insights, the models developed here contribute to more informed decision-making in engine design and emission control. As the automotive industry faces the dual challenges of enhancing performance while reducing emissions, the approach outlined in this study offers a promising pathway toward achieving both objectives simultaneously.

References

[1] World Health Organization, “Air pollution,” WHO, 2021. [Online]. Available: https://www.who.int/health-topics/air-pollution

[2] J. A. Raub, M. Mathieu-Nolf, N. B. Hampson, and S. R. Thom, “Carbon monoxide poisoning—a public health perspective,” Toxicology, vol. 145, no. 1, pp. 1–14, 2000.

[3] Environmental Protection Agency, “Ground-level Ozone Basics,” EPA, 2021. [Online]. Available: https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics

[4] W. J. Requia, M. Mohamed, C. D. Higgins, A. Arain, and M. Ferguson, “How clean are electric vehicles? Evidence-based review of the effects of electric mobility on air pollutants, greenhouse gas emissions and human health,” Atmospheric Environment, vol. 185, pp. 64–77, 2018.

[5] T. Johnson and A. Joshi, “Review of vehicle engine efficiency and emissions,” SAE Int. J. Engines, vol. 11, no. 6, 2018.

[6] J. Gao, H. Chen, G. Tian, C. Ma, and F. Zhu, “An analysis of energy flow in a turbocharged diesel engine of a heavy truck and potential for recovery of exhaust heat,” Energy Convers. Manage., vol. 185, pp. 1040–1051, 2019.

[7] R. D. Reitz et al., “IJER editorial: The future of the internal combustion engine,” Int. J. Engine Res., vol. 21, no. 1, pp. 3–10, 2020.

[8] V. M. Janakiraman, X. Nguyen, and D. Assanis, “Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines,” Neurocomputing, vol. 177, pp. 304–316, 2016.

[9] J. D. Wu and J. C. Liu, “Development of a predictive system for car fuel consumption using an artificial neural network,” Expert Syst. Appl., vol. 38, no. 5, pp. 4967–4971, 2014.

[10] S. Roy, R. Banerjee, and P. K. Bose, “Performance and exhaust emissions prediction of a CRDI assisted single cylinder diesel engine coupled with EGR using artificial neural network,” Appl. Energy, vol. 119, pp. 330–340, 2014.

[11] T. Wang, M. Jerrett, P. Sinsheimer, and Y. Zhu, “Estimating PM2.5 in Southern California using remote sensing data and light use efficiency modeling: Implications for policy,” Environ. Sci. Technol., vol. 50, no. 9, pp. 4724–4733, 2016.

[12] S. C. De Lima Nogueira et al., “Prediction of the NOx and CO2 emissions from an experimental dual fuel engine using optimized random forest combined with feature engineering,” Energy, vol. 280, 128066, 2023.

[13] Q. Shen et al., “Prediction Model for Transient NOx Emission of Diesel Engine Based on CNN-LSTM Network,” Energies, vol. 16, no. 14, 5347, 2023.

[14] J. B. Heywood, Internal combustion engine fundamentals, New York, NY, USA: McGraw-Hill Education, 2018.

[15] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[16] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to linear regression analysis, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012.

[17] Y. Liu, Y. Wang, and J. Zhang, “A novel hybrid model based on data preprocessing and optimized decision tree for diesel engine NOx emission prediction under transient conditions,” Energy, vol. 239, 122207, 2022.

[18] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[19] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” Stat. Comput., vol. 14, no. 3, pp. 199–222, 2004.

[20] C. Strobl, A. L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” BMC Bioinformatics, vol. 9, no. 307, 2008.

[21] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, “Predicting sample size required for classification performance,” BMC Med. Inform. Decis. Mak., vol. 12, no. 8, 2012.

Authors Contributions
Authors are solely responsible for the design, execution, analysis, and writing of this study.

Funding Statement
Authors have not received any financial support for the research, authorship, or publication of this study.

Conflict of Interest Declaration
Authors declares that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent
It is hereby declared that all scientific and ethical standards were adhered to during the preparation of this study. All sources utilized in the study are appropriately cited in the bibliography.

Availability of Data and Materials
All data and materials related to this study are available from the corresponding authors upon reasonable request.

Plagiarism Statement
This article has been screened for plagiarism using iThenticate™ software and has been confirmed to be original.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 8, No. 1, 12-26, 2025
DOI: 10.35377/saucis...1607145

RESEARCH ARTICLE

Blockchain-Based IoT Security and Performance Analysis

Selami Terazi[1], **Arafat Şentürk**[1*]

[1]Duzce University, Faculty of Engineering, Department of Computer Engineering, Duzce, Türkiye, ror.org/04ttnw109

Corresponding author:
Arafat Şentürk, Duzce University, Faculty of Engineering, Department of Computer Engineering, Duzce, Türkiye
arafatsenturk@duzce.edu.tr

Article History:
Received: 25.12.2024
Revised: 05.02.2025
Accepted: 27.02.2025
Published Online: 27.03.2025

ABSTRACT
The Internet of Things (IoT) enables devices to connect and exchange data, revolutionizing industries and daily life. However, the rapid growth of IoT devices has introduced significant security challenges, including cyber-attacks, data breaches, and unauthorized access. This study explores the integration of blockchain technology, particularly Hyperledger Fabric, to enhance IoT security. With its permissioned structure and decentralized approach, blockchain ensures secure data storage, integrity, and confidentiality. Hyperledger Fabric's modular architecture offers organizations the flexibility to address these security needs effectively. Using the OPNET simulation tool, the study analyses the performance of IoT networks transmitting blockchain-encrypted packets. Results show that blockchain integration enhances security, strengthens user authentication, and prevents unauthorized access. These findings highlight blockchain's transformative potential for IoT security, offering practical solutions for industrial applications and emphasizing the need for continued research in this critical field.

Keywords: Blockchain, IoT, Cyber security, Hyperledger Fabric, OPNET

1. Introduction

The number of IoT devices is growing rapidly, reaching 17 billion devices worldwide by 2024 [1]. This number is projected to increase to 30 billion by 2030 [1]. These Internet-connected devices range from smart thermostats in homes to complex industrial control systems in factories [2]. In healthcare, for example, IoT devices enable real-time patient monitoring, enabling faster response times and more personalized care [3]. Similarly, IoT systems in manufacturing provide real-time data on machine performance, helping to minimize downtime and increase productivity [4]. The potential of IoT extends to how smart infrastructure can optimize traffic flow, reduce energy consumption and improve public safety [5].

However, this vast interconnected ecosystem also poses significant security risks [6]. Each device connected to the internet represents a potential entry point for cyber-attacks [6]. A study by HP highlighted the widespread security concerns in IoT systems, revealing that 70% of IoT devices are vulnerable to attacks [7]. A notable example of IoT vulnerabilities was the 2016 Mirai botnet attack, in which thousands of compromised IoT devices were used to launch a distributed denial of service (DDoS) attack, bringing down major websites [8]. With IoT devices often deployed without proper security measures in place, the attack surface for potential breaches has expanded significantly [6].

Security challenges in IoT are exacerbated by the limitations of many internet-connected devices [6]. Unlike traditional computing systems, many IoT devices have limited processing power and memory, making it difficult to implement traditional security mechanisms such as firewalls and encryption [9]. This has led to an increasing focus on lightweight cryptography and secure communication protocols adapted for resource-constrained environments [9]. Furthermore, the heterogeneous nature of IoT devices, ranging from simple sensors to complex machines, poses interoperability challenges and complicates security measures across different platforms [9].

In recent years, blockchain technology has emerged as a promising tool for improving IoT security [10]. Originally developed as the technology that created cryptocurrencies such as Bitcoin, blockchain is a decentralized ledger that records transactions across multiple nodes in a network [11]. Its characteristics of transparency, immutability and decentralized control make it an ideal solution for securing IoT systems [12]. Using Blockchain, data generated by IoT devices can be recorded in a hacker-proof manner and data integrity can be ensured [12]. Furthermore, the decentralized nature of Blockchain reduces the risk of single points of failure in the system by eliminating the need for a central authority [10].

Cite as: S. Terazi and A. Şentürk, "Blockchain-based IoT security and performance analysis," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 12-26, 2025, doi: 10.35377/saucis...1607145

Table 1. Blockchain Mechanisms for IoT Security

Reference	Security Threats	IoT Applications	Observation
[20]	Sybil attack, self-promoting attack, bad-mouthing attack.	IoT devices	Hyperledger's TABI is an access control mechanism designed for Edge-IoT networks that builds trust using blockchain technology. This Trust-Based Access Control Mechanism ensures secure and reliable access management, specifically tailored for the unique demands of IoT environments at the network edge.
[21]	Malicious software or physical attacks	IoT devices	IoTCop is an IoT monitoring framework that utilizes blockchain technology for enhanced security. Leveraging Hyperledger Fabric and modular hardware plugins, it swiftly identifies and isolates compromised devices to maintain network integrity.
[22]	"Impersonation," "man-in-the-middle," "ephemeral secret leakage (ESL)," and "replay" attacks.	IoT-enabled smart grid system	DBACP-IoTSG is a newly developed IoT-enabled smart grid system that operates independently of a Trusted Third Party (TTP). It employs leader election and PBFT (Practical Byzantine Fault Tolerance) consensus for secure block verification, while ECC (Elliptic Curve Cryptography) encryption ensures transaction privacy.
[23]	Jamming and impersonation attacks	IoT blockchain network	Through the study of obfuscation and impersonation attacks on a RAFT-based IoT blockchain network, a path-loss-based identification method was proposed, demonstrating strong detection rates against these types of threats.
[24]	Man-in-the-middle attack, eavesdropping attack, impersonation attack, replay attack.	IoT network	This solution provides a lightweight, blockchain-based authentication method for IoT, utilizing MSR encryption to enable decentralized and privacy-preserving authentication.
[25]	Malicious attacks	Industrial IoT network	A secure framework has been proposed that combines trust management with blockchain technology to address issues arising from varying levels of malicious devices in industrial IoT networks. This approach enhances network reliability by effectively managing and mitigating threats posed by compromised devices.

Table 2. Some Studies Leveraging Blockchain for IoT Security

Security areas in IoT	Proposed solutions	Blockchain features
Access control	[20]	TABI Mechanism for Edge-IoT Networks
	[26]	Access Control through Smart Contracts
	[22]	Respective smart meters (SMs)
	[27]	Manage and organize groups using a group key (GK)
	[28]	ABAC grants access based on the qualifications specified by the target
Data integrity	[23]	Applying a binary hypothesis test for identifying transmission nodes.
	[25]	
	[29]	
Data confidentiality	[30]	Asymmetric Scalar-product Preserving Encryption (ASPE)
	[31]	Attribute-based security authentication using the Hyperledger Fabric blockchain framework
	[24]	The framework integrates blockchain technology with the modular square root algorithm
	[19]	IoT powered by blockchain with dynamic device management and conditional traceability
	[32]	Blockchain-based model for IoT authentication and security protection
Data availability	[23]	Stochastic geometry tool

Blockchain can also help address some of the key challenges related to IoT security [12]. For example, it can be used to secure device-to-device communication by establishing trust between devices without relying on a centralized server [12]. This is particularly important for the traditional constrained client-server model [13]. Furthermore, smart contracts (self-executing contracts where the terms of the agreement are written directly into the code) can be used to automate processes in IoT systems, further improving security and efficiency [10]. For example, in supply chain management, smart contracts can reduce the risk of fraud by automatically triggering payments when goods are delivered [14].

Despite its advantages, there are also disadvantages in the integration of blockchain with IoT. One of the biggest issues is scalability [15]. Traditional blockchain networks such as Bitcoin and Ethereum struggle to handle large transaction volumes, making them inefficient for the high data throughput of IoT systems [16]. To address this issue, new blockchain platforms such as Hyperledger Fabric are being developed and offer more efficient consensus mechanisms that can support enterprise-level applications [17]. Additionally, the energy consumption of blockchain networks, especially those based on proof-of-work consensus algorithms, raises sustainability concerns, especially in IoT environments where energy efficiency is critical [18].

2. Related Works

In [19], S. Basudan introduces a scalable framework that integrates IoT with blockchain technology to enable secure transactions in dynamic environments. The framework leverages dynamic device management and conditional traceability through the DABG protocol, offering rapid transaction confirmations, enhanced data security, and privacy protection. Future developments in this framework aim to incorporate federative learning and advanced privacy protection techniques. Table 1 and Table 2 provide a detailed analysis of how blockchain is effectively utilized with IoT devices and the key features involved.

A. Pathak et al. [20] explore the application of blockchain to enhance security in IoT networks, addressing issues such as computational overhead and high energy consumption. By employing edge computing, their proposed Trust-Based Access Control Mechanism (TABI) (see Figure 1) provides a solution for ensuring end-to-end security in IoT networks, particularly those with limited resources. TABI integrates trust evaluation and access control to mitigate risks from malicious devices and users. Its performance indicates suitability for IoT applications requiring low latency and resource optimization. Future research will focus on improving service quality and identifying malicious devices within IoT ecosystems.

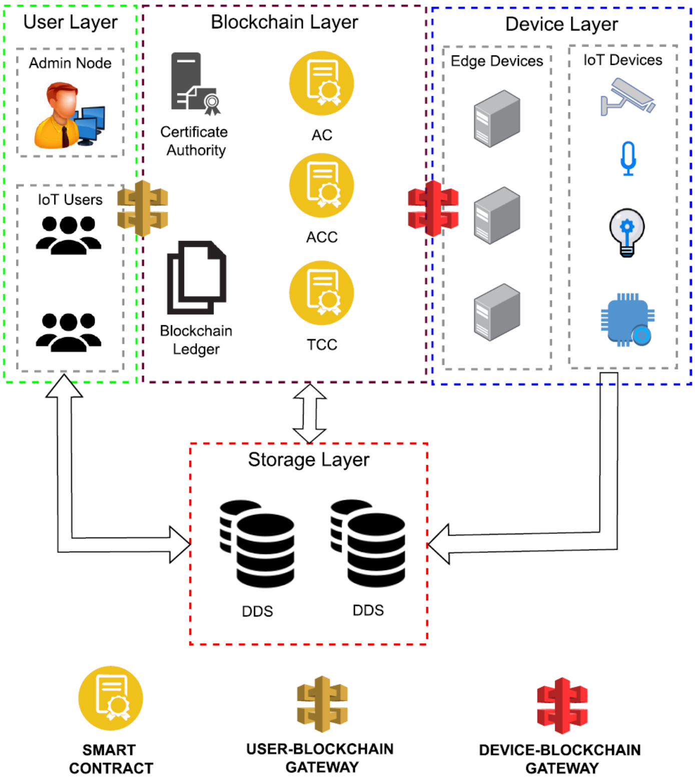


Figure 1. TABI architecture [20]

S. Seshadri et al. [21] present IoTCop, a blockchain-based monitoring framework designed to safeguard IoT devices. Unlike traditional servers, IoT devices are often geographically distributed and close to physical systems, leading to resource limitations despite the need for robust security measures. The study proposes leveraging blockchain technology to enforce security policies, allowing automatic isolation of compromised devices. By utilizing a permissioned blockchain (Hyperledger Fabric) and supplementary hardware modules, the framework delivers low latency and minimal workload while enabling seamless integration of existing IoT devices. Table 3 outlines various malicious attacks that IoTCop protects against.

Table 3. Common Attacks on IoT Networks

Attack	Category	Description	Research on defending attack using blockchain
Impersonation attack	Internal/External	An impersonation attack occurs when a device falsely assumes the identity or authorization of another device to gain unauthorized access to IoT networks.	[22], [23], [24]
Man-in-the-Middle attack	Internal/External	A Man-in-the-Middle (MitM) attack on IoT networks allows an attacker to intercept or alter communication between IoT devices.	[22], [24]
Bad-mouthing attack	Internal	A bad-mouthing attack is a type of cyber-attack on IoT networks where an attacker spreads false or misleading information to discredit other devices.	[20]
Replay attack	External	In a replay attack, a malicious actor attempts to gain unauthorized access to IoT networks by reusing or retransmitting recorded data.	[21], [22], [24]
Sybil attack	Internal/External	A Sybil attack in IoT networks involves an attacker infiltrating the network by generating numerous fake devices, each with a counterfeit identity	[20]

Table 3. (Continued)

Jamming attack	External	A jamming attack in IoT networks occurs when a malicious device or jammer floods radio frequencies, disrupting the communication between IoT devices.	[23]
Self-promoting attack	External	In IoT networks, a "self-promoting attack" involves an IoT device attempting to secretly join the network or gain unauthorized access by falsely identifying itself. The device typically infiltrates the network either directly or by exploiting existing vulnerabilities.	[20]
Eavesdropping attack	External	Eavesdropping is an attack on IoT networks where attackers monitor communication traffic to access sensitive information, creating security vulnerabilities and privacy breaches.	[24]
Ephemeral secret leakage (ESL) attack	Internal	In ESL IoT networks, an insider security breach occurs when unauthorized persons or devices leak temporary passwords without permission.	[22]
DDoS	Internal/External	A DDoS (Denial of Service) attack on IoT networks is a cyber attack where numerous IoT devices collectively send a massive volume of client traffic to a target, overwhelming it and causing a crash.	[32]

A review of literature on blockchain and IoT integration highlights several challenges that must be addressed, such as latency, scalability, and real-world applicability. Table 4 offers an in-depth look at these challenges and their potential impact on IoT security solutions.

Table 4. Challenges in Blockchain Integration with IoT

References	**Key areas**	**Challenges**
[21]	Delay	Consensus is reached within 1 to 10 minutes
	Resource Constraints	Resource-intensive blockchains may not be suitable for IoT devices
	Applicability	Assuming that all devices support the same blockchain framework is impractical
[19]	Efficiency and Scalability	IoT scalability is hindered by low blockchain throughput
	Privacy and Traceability	Balancing traceability and anonymity in blockchain transactions for IoT
	Device Management	Decentralized device management on blockchain faces challenges due to IoT mobility
[12]	Blockchain Attacks	Various Blockchain attacks may expose IoT devices to risks

3. The Proposed Method

This study combines Hyperledger Fabric, a blockchain platform, with OPNET, a network simulation tool. Hyperledger Fabric is used to demonstrate secure data transactions and management in IoT ecosystems, while OPNET simulates network dynamics and potential vulnerabilities in IoT infrastructures. Together, these tools provide a comprehensive framework for assessing the effectiveness of blockchain in IoT security.

Hyperledger Fabric is an open-source blockchain framework designed to meet the specific needs of businesses by offering high levels of privacy, security and scalability. Unlike public blockchains like Bitcoin and Ethereum, which operate on permissionless networks, Hyperledger Fabric offers a permissioned model, meaning that participants must be identified and verified before they are allowed to join the network [33].

OPNET is a powerful simulation tool used to model and analyze the performance of communication networks, protocols and devices. Originally developed by MIL 3, Inc. and later acquired by Riverbed Technology [34], OPNET allows users to test various network configurations, simulate traffic loads, and evaluate the impact of different network protocols before deploying them in real-world environments [35].

3.1. Hyperledger Fabric Network Setup and Data Operations

First, an outline of the network architecture was created, highlighting the roles and interactions of various components, including organizations, peers and orderer. Next, the Hyperledger Fabric network was run, followed by interactions through Postman. A user was registered and the corresponding token (key) was obtained. Using this token, vehicle registrations were added to the blockchain, each generating a unique transaction ID. The Fabcar.go code serves as an application that demonstrates how to interact with the blockchain network by adding and transacting on sample vehicle/car records [36].

The network diagram shown in Figure 2 illustrates the interaction between the two organizations, the peers each has, and the orderer, which plays a critical role in the consensus process. The process of confirming a transaction within the Hyperledger Fabric network, adding it to the blockchain, and adding new blocks to the existing blockchain is shown in Figure 2 [37].

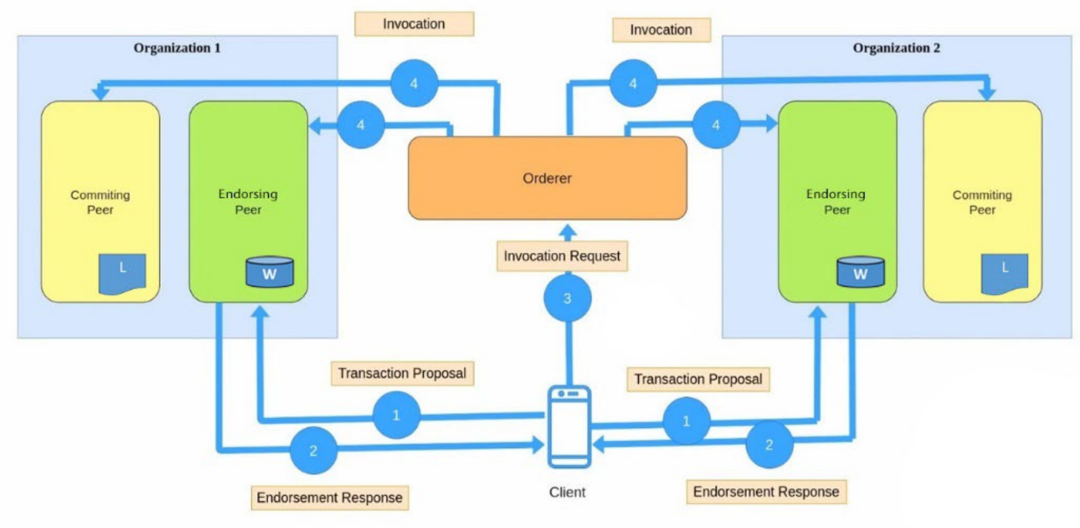


Figure 2. Transaction flow in Hyperledger Fabric [37]

In the Hyperledger Fabric network, organizations represent independent entities that participate in the blockchain network. Each organization operates its own peers, which are responsible for verifying and approving transactions. Approving peers show and approve transaction proposals. When a client submits a transaction proposal, the confirming peers access the world state database (W) to run the chaincode and indicate the transaction. They then generate an acknowledgment response. This response contains simulation results and a signature. The confirmation response is critical to the validity of the transaction and must comply with the network's confirmation policy. Committing peers do not keep a complete ledger; they only evaluate transactions using the current world state database. Committing peers are responsible for validating the confirmed transactions received from the orderer and committing them to the blockchain ledger (L). Once the transactions are verified, they are recorded in the ledger and the world state database is updated accordingly. Committing peers do not show transactions; instead, they perform verification and recording to maintain the integrity of the network and maintain an up-to-date ledger.

The client is the entity in the network that initiates the transaction process. It sends the transaction proposal to the network's confirming peers. After collecting the necessary approvals, it forwards the transaction request to the orderer. The client sends the transaction offer to the confirming peers. For the Confirmation Response, the confirming peers execute the trade offer, generate a confirmation response and send it back to the client.

To make an Invocation Request, the client collects the necessary confirmations and sends an invocation request to the orderer. The orderer queues transactions, creates blocks and distributes them to committing peers. Committing peers verify and commit transactions to the ledger, updating the world state accordingly.

The orderer plays a critical role in ensuring the consistency of the blockchain. It collects confirmed transactions, sorts them into blocks and distributes these blocks to committing peers. By ensuring that all peers receive the same order of transactions, the orderer maintains the integrity of the network. The Raft consensus algorithm is the primary consensus mechanism used in Hyperledger Fabric. Raft is a crash fault tolerant (CFT) consensus protocol that provides deterministic transaction ordering. Unlike Byzantine Fault Tolerant (BFT) algorithms, Raft focuses on scenarios where participants are generally trustworthy. It manages the process of ordering transactions by electing a leader among the orderers. In case the leader fails, a new leader is automatically elected, which ensures a continuous operation without downtime.

To initialize the Hyperledger Fabric network, a command is used that leverages Docker Compose, which helps manage multiple Docker containers. These containers represent different components of the Fabric network, such as peers and orderers, all of which are defined in a YAML configuration file. Channels provide data privacy and segregation by allowing certain participants to communicate and transact privately. Once the configuration of the channel is defined, it is created and peers from different organizations are instructed to join.

In Figure 3, there are two separate channels between organizations, Channel 1 and Channel 2. These channels allow specific organizations to communicate in a secure and private way. Once the channel is established, chaincode is distributed to all peers in the network, as shown in Figure 3. Chaincode is written in languages such as Go or JavaScript and governs how transactions are handled. The process involves packaging the chaincode, uploading it to the peers and obtaining approval from all relevant organizations. After approval, the chaincode becomes active and manages network transactions.

Each step ensures that the network, channels and smart contracts are properly configured and can interact securely and efficiently.

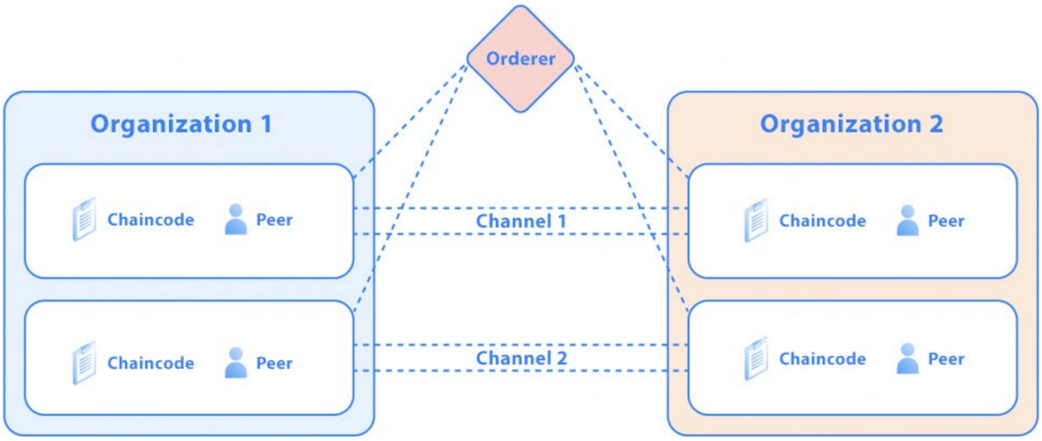
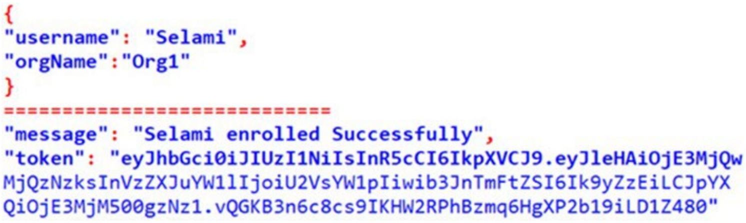


Figure 3. Channels and Chaincode [38]

A block named car is used as an example to add a new block to the system, but this process can be applied to any entity record. In Figure 5, the block named car is shown as an example, the type of data to be registered may vary depending on the system or the user. Clients can register users through the /users API endpoint. This requires providing a username and organization name. Upon successful registration, the system issues a JSON Web Token (JWT) to the client (see Figure 4). This token is required to perform other sensitive operations such as authentication, channel management and chaincode interactions.

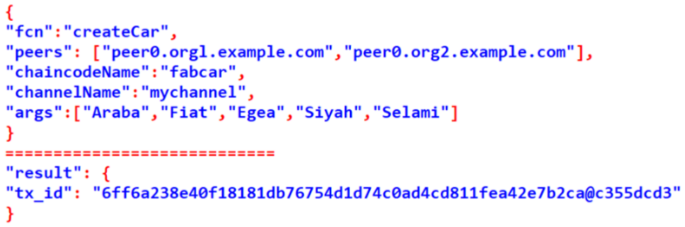


```
{
"username": "Selami",
"orgName":"Org1"
}
============================
"message": "Selami enrolled Successfully",
"token": "eyJhbGci0iJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MjQw
MjQzNzksInVzZXJuYW1lIjoiU2VsYW1pIiwib3JnTmFtZSI6Ik9yZzEiLCJpYX
QiOjE3MjM500gzNz1.vQGKB3n6c8cs9IKHW2RPhBzmq6HgXP2b19iLD1Z480"
```

Figure 4. User Enrolling and JWT Generation

This function starts by creating a unique ID for a block named “car” as an instance in the ledger. This instance represents a transaction specifically related to vehicle information. Using the Fabcar.go chaincode, specific data for each vehicle (make, model, color and owner) is recorded within a block.

In the first step, the client application sends a transaction proposal to the Hyperledger Fabric network. This includes a request to create a block for a specific vehicle data. Peer nodes approve this proposal and the transaction is forwarded to the orderer. The orderer merges the approved transaction with other transactions and adds a new block to the chain.



```
{
"fcn":"createCar",
"peers": ["peer0.orgl.example.com","peer0.org2.example.com"],
"chaincodeName":"fabcar",
"channelName":"mychannel",
"args":["Araba","Fiat","Egea","Siyah","Selami"]
}
============================
"result": {
"tx_id": "6ff6a238e40f18181db76754d1d74c0ad4cd811fea42e7b2ca@c355dcd3"
}
```

Figure 5. Creating A “car” Eecord

3.2. Network Simulation with OPNET

This section describes in detail the simulation of a ZigBee-based IoT network that mirrors a blockchain environment.

In the simulation scenarios, the ACK mechanism is enabled in many studies in the literature [39] to increase the reliability of the data packets, so it is done in the same way in this study. Also, the simulation time is set to 15 minutes in many studies [40], which is also used in this study.

The metrics selected to evaluate performance are as follows;

- End-to-End Latency (sec): Captures the total time it takes for a data packet to traverse the network from the source device to the destination. It is important for IoT applications, where fast response times are often required, to assess whether data delivery is timely [39].
- Data Traffic Sent (bits/sec): Measures the rate at which data is transmitted from a device in bits per second. Monitoring this metric is essential to understand how much data is being sent over the network. This helps to understand network efficiency and capacity utilization [41].
- Received Data Traffic (bits/sec): Indicates how much data is successfully received per second by ZigBee devices. It helps to evaluate network reliability and packet delivery success [41].
- Throughput (bits/sec): Throughput is the actual rate of successful data transmission over the network. It represents how efficiently the network is being utilized. Throughput is a key indicator of network performance as it reflects how well the network handles data transmission under load [41].

To simulate the IoT environment, the size of the blockchain packets was represented using OPNET's packet size adjustment feature. This has been done previously in the literature [42] [43] and [44]. This assumption is based on the average packet size in the literature, which is approximately 2500 bytes [42]. This size is reflected in OPNET's packet size feature to ensure consistency with simulated blockchain packet transmissions. By adjusting certain parameters, two different scenarios were realized by simulating different network conditions and the results were compared.

4. Experiments

In order to evaluate the performance of IoT networks transmitting packets encrypted using blockchain, four scenarios are realized in pairs. The first scenario tests the operation of the network under low load, while the second scenario examines the responsiveness of the network with more devices. Increasing the number of devices is very important in evaluating the scalability of IoT applications by affecting the performance metrics accepted in the literature such as latency, data traffic and throughput.

For comparison in the scenarios, packets were transmitted directly without the blockchain and encrypted and transmitted using the blockchain. The size of the packets assumed to be encrypted using the blockchain was set to 2500bytes (see Figure 6) based on previous work [42]. The size of packets transmitted without blockchain is set to 512bytes, which is the packet size in standard wireless networks [45].

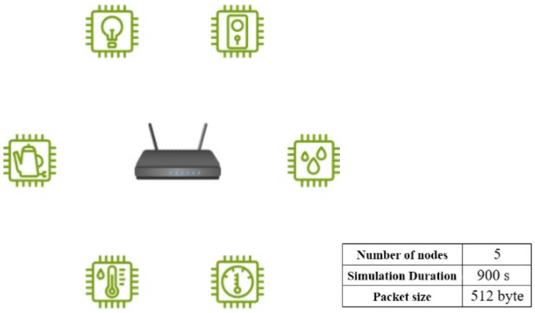
Figure 6. Blockchain-IoT environment

In the scenarios section, the network structure that transmits packets without blockchain is referred to as “standard” and the network structure that transmits packets encrypted using blockchain is referred to as “using blockchain”.

4.1. Scenario 1 and Scenario 2

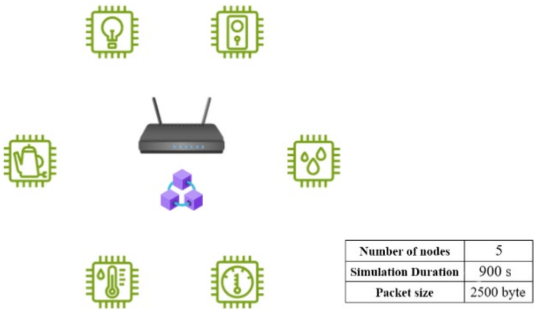
In Scenarios 1 and 2, five end devices and one central ZigBee coordinator, a total of six devices were deployed using OPNET as shown in Figure 6 and Figure 7. The number of devices was chosen as five based on [46] and [47]. Because in this section, it was chosen to test the performance of the network at low device density

In scenario 1, packets are transmitted directly on the wireless network without any processing, but in scenario 2, packets are transmitted after being encrypted using blockchain.



Number of nodes	5
Simulation Duration	900 s
Packet size	512 byte

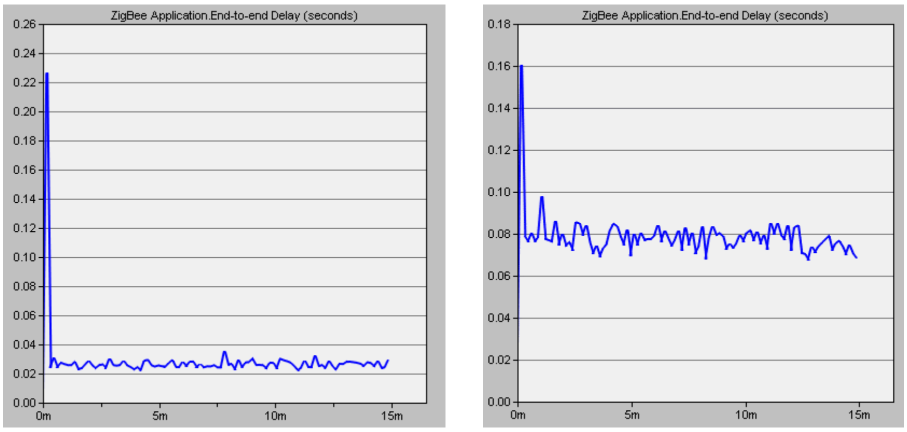
Figure 7. Topology of Scenario 1



Number of nodes	5
Simulation Duration	900 s
Packet size	2500 byte

Figure 8. Topology of Scenario 2

As shown in Figure 8, in scenario 1 the end-to-end delay is between 0.02 and 0.04 seconds, while in scenario 2 it is between 0.07 and 0.10 seconds.



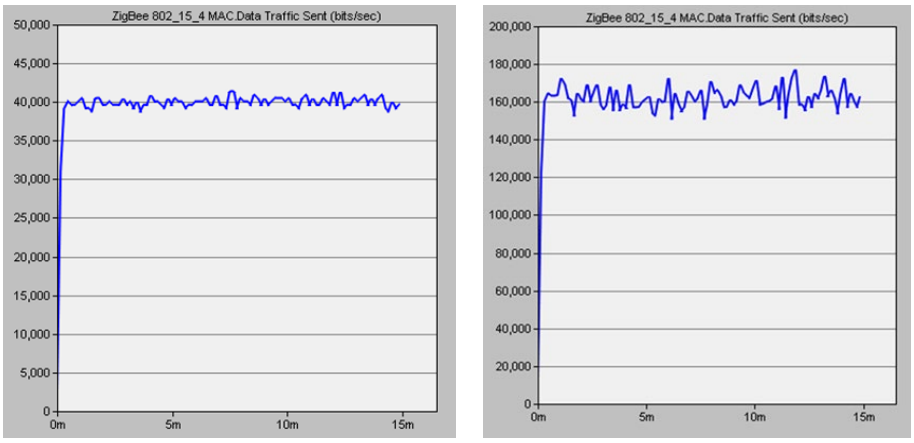
(a) (b)

Figure 9. End-to-End Delay

(a) Scenario 1 - “standard” (b) Scenario 2 - “using blockchain”

As shown in Figure 9, the transmitted data traffic varies between 39,000 bits/sec and 42,000 bits/sec in scenario 1. In scenario 2, these values vary between 150,000 and 180,000 bits/s.

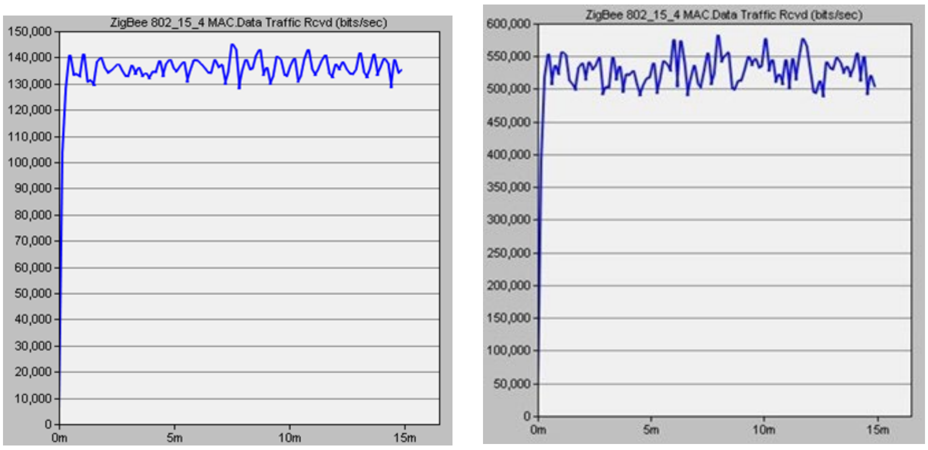
As seen in Figure 10, the received data traffic varies between 130,000 bit/s and 145,000 bit/s in scenario 1. In scenario 2, the received data traffic varies between 500,000 bit/s and 600,000 bit/s.



(a) (b)

Figure 10. Data Traffic Sent

a) Scenario 1 - “standard” (b) Scenario 2 - “using blockchain”

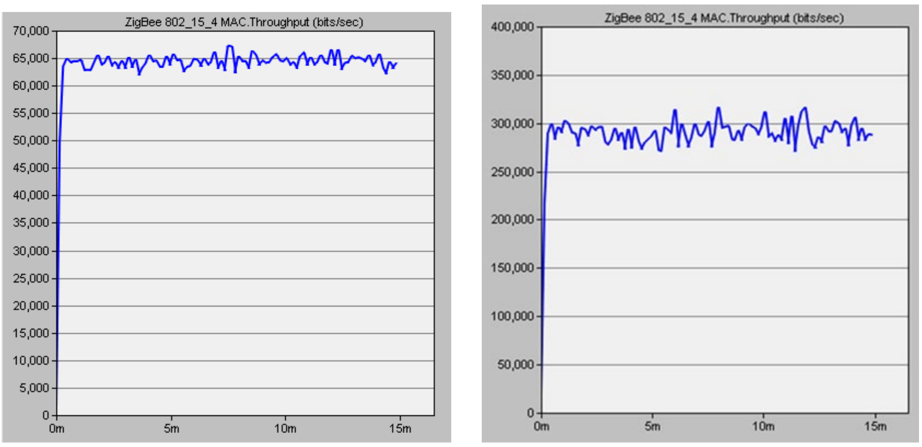


(a) (b)

Figure 11. Received Data Traffic

a) Scenario 1 - “standard” (b) Scenario 2 - “using blockchain”

As shown in Figure 11, the throughput fluctuates between 63,000 bits/sec and 68,000 bits/sec in scenario 1. In scenario 2, the throughput fluctuates between 270,000 and 320,000 bits/sec.



(a) (b)

Figure 12. Throughput

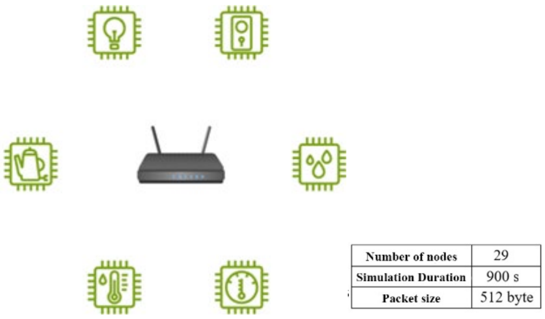
a) Scenario 1 - “standard” (b) Scenario 2 - “using blockchain”

4.2. Scenario 3 and Scenario 4

In Scenarios 3 and 4, as shown in Figure 12 and Figure 13, the network complexity is increased by adding 24 more devices, bringing the total to 29. The choice of the number of devices was based on [46] and [47] and was chosen as 29. This is because it is chosen in this section to test the performance of the network at medium device density.

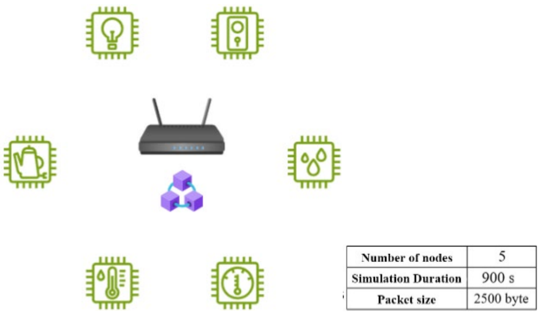
In scenario 3, packets are transmitted directly on the wireless network without any processing, but in scenario 4, packets are transmitted after being encrypted using blockchain.

In these scenarios, the packet size is set to 2500bytes for packets encrypted using blockchain and 512bytes for standard packets, as in the first two scenarios.



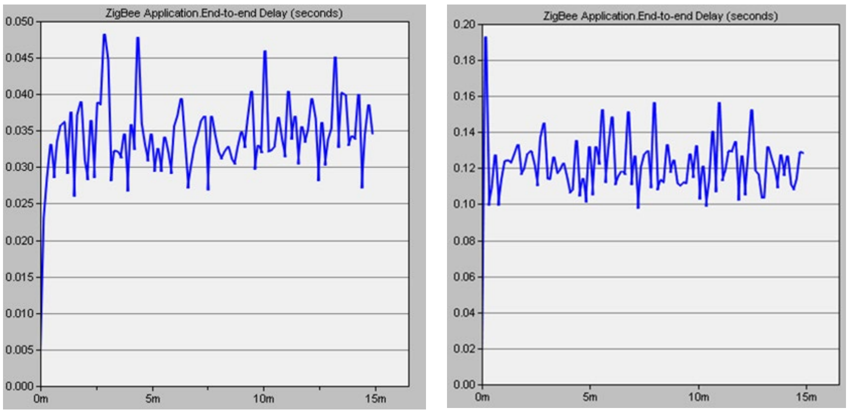
Number of nodes	29
Simulation Duration	900 s
Packet size	512 byte

Figure 13. Topology of Scenario 3



Number of nodes	5
Simulation Duration	900 s
Packet size	2500 byte

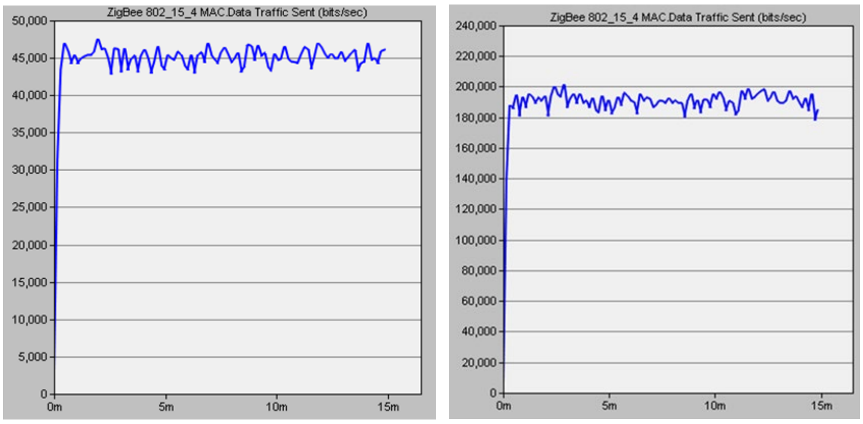
Figure 14. Topology of Scenario 4



(a) (b)

Figure 15. End-to-End Delay

a) Scenario 3 - “standard” (b) Scenario 4 - “using blockchain”



(a) (b)

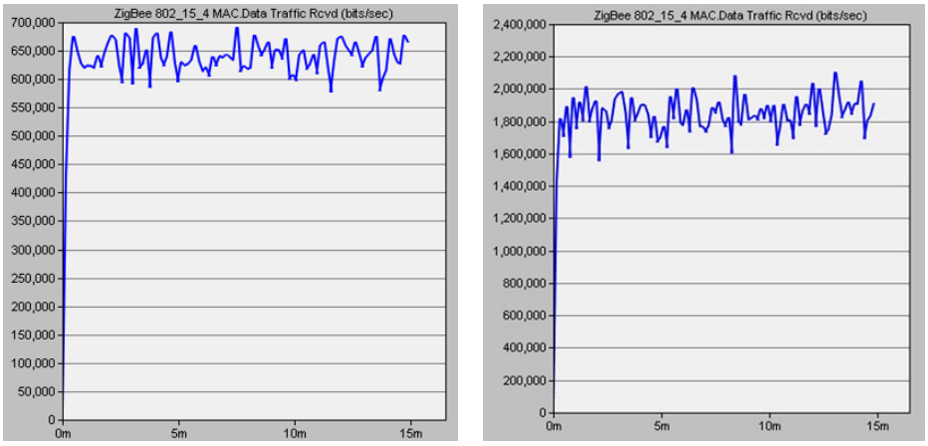
Figure 16. Data Traffic Sent

a) Scenario 3 - “standard” (b) Scenario 4 - “using blockchain”

In Figure 14, the end-to-end delay fluctuates between 0.026 and 0.048 seconds in scenario 3. In scenario 4, it fluctuates between 0.10 and 0.16 seconds.

As shown in Figure 15, the transmitted data traffic varies between 43,000 bits/sec and 47,000 bits/sec in scenario 3. In scenario 4, this value varies between 180,000 and 200,000 bits/s.

As shown in Figure 16, the received data traffic varies between 560,000 bit/s and 700,000 bit/s in scenario 3. In scenario 4, this value varies between 1,600,000 bit/s and 2,100,000 bit/s.

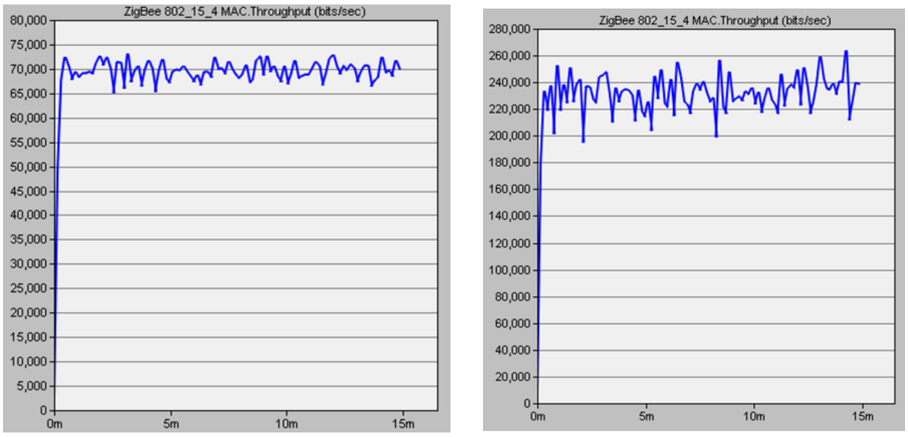


(a) (b)

Figure 17. Received Data Traffic

a) Scenario 3 - “standard” (b) Scenario 4 - “using blockchain”

As shown in Figure 17, throughput varies between 65,000 bits/sec and 73,000 bits/sec in scenario 3. In scenario 4, the throughput varies between 200,000 bits/sec and 260,000 bits/sec.



(a) (b)

Figure 18. Throughput

a) Scenario 3 - “standard” (b) Scenario 4 - “using blockchain”

4.3. Performance Evaluation

The main purpose of the scenarios is to compare the security and performance of packets transmitted directly without blockchain and encrypted and transmitted using blockchain in IoT networks. Blockchain offers an inherently secure system [48]. The security, integrity and confidentiality of the data are secured by the blockchain. However, these security features create some overhead in the performance of the network.

In the first two scenarios, the network that transmits packets encrypted using blockchain is compared with the network that transmits packets without blockchain using a small number of devices. In the first scenario, the networks transmitting packets without blockchain outperformed the networks transmitting packets without blockchain in terms of end-to-end delay, data sent and received, and throughput. For example, the end-to-end delay fluctuated between 0.02 and 0.04 seconds in scenario 1, while it fluctuated between 0.07 and 0.10 seconds in scenario 2 for networks transmitting packets encrypted using blockchain. Similarly, the data traffic sent varied from 39,000 bits/sec to 42,000 bits/sec in scenario 1 to 150,000 to 180,000 bits/sec in scenario 2.

In scenarios 3 and 4, the number of devices increased, and tests were performed with 29 devices in total. In Scenario 3, the end-to-end delay for networks transmitting packets without blockchain was between 0.026 and 0.048 seconds, while in Scenario 4, the delay for networks transmitting encrypted packets using blockchain was between 0.10 and 0.16 seconds. The data traffic sent and received increased substantially with more devices. In scenario 3, the received data traffic ranged from 560,000 bits/sec to 700,000 bits/sec, while in scenario 4 it ranged from 1,600,000 bits/sec to 2,100,000 bits/sec. This shows that networks transmitting packets encrypted using blockchain generate more data overhead. Similarly, throughput is also higher for networks transmitting packets without blockchain, fluctuating between 65,000 bits/sec and 73,000 bits/sec in scenario 3 and between 200,000 and 260,000 bits/sec in scenario 4.

As a result of these evaluations, it can be seen that networks that transmit packets encrypted using blockchain technology have some disadvantages in terms of performance despite their security advantages. However, in applications where the need for security is critical, the use of blockchain in transmitted packets can be preferred as a secure solution. Table 5 presents a comparison of the four scenarios in terms of key performance metrics.

Table 5. Comparison of Four Scenarios

Metric	**Scenario 1** “standard”	**Scenario 2** “using blockchain”	**Scenario 3** “standard”	**Scenario 4** “using blockchain”
Number of nodes	5	5	29	29
Simulation Duration	900 sec	900 sec	900 sec	900 sec
Packet size	512 bytes	2500 byte	512 bytes	2500 byte
End-to-end delay (average)	0,03 sec	0,08 sec	0,035 sec	0,12 sec
Data traffic sent (average)	40.000 bit/ sec	160.000 bit/ sec	45.000 bit/ sec	190.000 bit/ sec
Data traffic received (average)	137.000 bit/ sec	525.000 bit/ sec	650.000 bit/ sec	1.800.000 bit/ sec
Throughput (average)	64.000 bit/ sec	280.000 bit/ sec	68.000 bit/ sec	230.000 bit/ sec

5. Discussions and Conclusion

In this study, blockchain technology is used to improve IoT security. The process involves setting up a blockchain network, using IoT transactions through simulation, and verifying the results using real-world inspired scenarios. The practical potential of blockchain in IoT environments is to perform actions such as authenticating users, registering, adding and retrieving object records. This secure, decentralized approach strengthens data integrity, making it an ideal solution for IoT systems where sensitive information is frequently exchanged.

One of the findings of the effective use of blockchain is that it ensures transparency and traceability of data. In an IoT environment where devices communicate autonomously, ensuring data accuracy and preventing loss is critical. Blockchain's role as an immutable ledger for these interactions lays a strong foundation for secure IoT networks. Furthermore, the ability to retrieve transaction histories and exchange ownership records in a secure and authenticated manner supports the system's utility in sectors such as logistics, smart cities and connected devices.

In future work, the findings suggest that blockchain has great potential in improving IoT security. However, additional research and testing in more diverse settings is necessary to fully unlock its benefits. Developing a more efficient simulation model and incorporating real-world IoT data and conditions would allow for better performance evaluations. In addition, incorporating machine learning or AI-based approaches to further optimize the network's response to dynamic conditions in IoT networks could increase its applicability.

In conclusion, the combination of blockchain and IoT is advancing the field by offering robust security mechanisms and assurance of data integrity. With further optimization and scalability considerations, blockchain-enabled IoT systems could become the industry standard for secure device communication, data management and automated decision-making.

References

[1] “How Many IoT Devices Are There (2024-2032).” Accessed: Jul. 22, 2024. [Online]. Available: https://www.demandsage.com/number-of-iot-devices/

[2] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012, doi: 10.1016/J.ADHOC.2012.02.016.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet Things J*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.

[4] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Bus Horiz*, vol. 58, no. 4, pp. 431–440, Jul. 2015, doi: 10.1016/J.BUSHOR.2015.03.008.

[5] L. Woetzel, J. Remes, B. Boland, and K. Lv, “Smart city technology for a more liveable future | McKinsey,” Jun. 2018. Accessed: Sep. 13, 2024. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/smart-cities-digital-solutions-for-a-more-livable-future

[6] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Computer Networks*, vol. 76, pp. 146–164, Jan. 2015, doi: 10.1016/J.COMNET.2014.11.008.

[7] HP, “Internet of Things Research Study,” 2014.

[8] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer (Long Beach Calif)*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

[9] R. H. Weber and E. Studer, “Cybersecurity in the Internet of Things: Legal aspects,” *Computer Law and Security Review*, vol. 32, no. 5, pp. 715–728, Oct. 2016, doi: 10.1016/J.CLSR.2016.07.002.

[10] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.

[11] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008, Accessed: Sep. 13, 2024. [Online]. Available: www.bitcoin.org

[12] S. Singh, A. S. M. Sanwar Hosen, and B. Yoon, “Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network,” *IEEE Access*, vol. 9, pp. 13938–13959, 2021, doi: 10.1109/ACCESS.2021.3051602.

[13] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov. 2018, doi: 10.1016/J.FUTURE.2018.05.046.

[14] A. Bahga, V. K. Madisetti, A. Bahga, and V. K. Madisetti, “Blockchain Platform for Industrial Internet of Things,” *Journal of Software Engineering and Applications*, vol. 9, no. 10, pp. 533–546, Oct. 2016, doi: 10.4236/JSEA.2016.910036.

[15] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, “A Survey of IoT and Blockchain Integration: Security Perspective,” *IEEE Access*, vol. 9, pp. 156114–156150, 2021, doi: 10.1109/ACCESS.2021.3129697.

[16] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 557–564, Sep. 2017, doi: 10.1109/BIGDATACONGRESS.2017.85.

[17] E. Androulaki *et al.*, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, vol. 2018-January, Jan. 2018, doi: 10.1145/3190508.3190538.

[18] A. de Vries, “Bitcoin’s Growing Energy Problem,” *Joule*, vol. 2, no. 5, pp. 801–805, May 2018, doi: 10.1016/J.JOULE.2018.04.016.

[19] S. Basudan, “A Scalable Blockchain Framework for Secure Transactions in IoT-Based Dynamic Applications,” *IEEE Open Journal of the Communications Society*, 2023, doi: 10.1109/OJCOMS.2023.3307337.

[20] A. Pathak, I. Al-Anbagi, and H. J. Hamilton, “TABI: Trust-Based ABAC Mechanism for Edge-IoT Using Blockchain Technology,” *IEEE Access*, vol. 11, pp. 36379–36398, 2023, doi: 10.1109/ACCESS.2023.3265349.

[21] S. S. Seshadri *et al.*, “IoTCop: A Blockchain-Based Monitoring Framework for Detection and Isolation of Malicious Devices in Internet-of-Things Systems,” *IEEE Internet Things J*, vol. 8, no. 5, pp. 3346–3359, Mar. 2021, doi: 10.1109/JIOT.2020.3022033.

[22] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, “Designing blockchain-based access control protocol in iot-enabled smart-grid system,” *IEEE Internet Things J*, vol. 8, no. 7, pp. 5744–5761, Apr. 2021, doi: 10.1109/JIOT.2020.3030308.

[23] H. M. Buttar, W. Aman, M. M. U. Rahman, and Q. H. Abbasi, “Countering Active Attacks on RAFT-Based IoT Blockchain Networks,” *IEEE Sens J*, vol. 23, no. 13, pp. 14691–14699, Jul. 2023, doi: 10.1109/JSEN.2023.3274687.

[24] X. Yang *et al.*, “Blockchain-Based Secure and Lightweight Authentication for Internet of Things,” *IEEE Internet Things J*, vol. 9, no. 5, pp. 3321–3332, Mar. 2022, doi: 10.1109/JIOT.2021.3098007.

[25] G. Rathee, F. Ahmad, N. Jaglan, and C. Konstantinou, “A Secure and Trusted Mechanism for Industrial IoT Network Using Blockchain,” *IEEE Trans Industr Inform*, vol. 19, no. 2, pp. 1894–1902, Feb. 2023, doi: 10.1109/TII.2022.3182121.

[26] H. Liu, D. Han, and D. Li, “Fabric-iot: A Blockchain-Based Access Control System in IoT,” *IEEE Access*, vol. 8, pp. 18207–18218, 2020, doi: 10.1109/ACCESS.2020.2968492.

[27] J. Maeng, Y. Heo, and I. Joe, “Hyperledger Fabric-Based Lightweight Group Management (H-LGM) for IoT Devices,” *IEEE Access*, vol. 10, pp. 56401–56409, 2022, doi: 10.1109/ACCESS.2022.3177270.

[28] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, “An Attribute-Based Access Control Model for Internet of Things Using Hyperledger Fabric Blockchain,” *Wirel Commun Mob Comput*, vol. 2022, 2022, doi: 10.1155/2022/6926408.

[29] R. Kaur and A. Ali, “A Novel Blockchain Model for Securing IoT Based Data Transmission,” *International Journal of Grid and Distributed Computing*, vol. 14, no. 1, pp. 1045–1055, Apr. 2021.

[30] H. Zhang, X. Zhang, Z. Guo, H. Wang, D. Cui, and Q. Wen, “Secure and Efficiently Searchable IoT Communication Data Management Model: Using Blockchain as a New Tool,” *IEEE Internet Things J*, vol. 10, no. 14, pp. 11985–11999, Jul. 2023, doi: 10.1109/JIOT.2021.3121482.

[31] Z. Gong-Guo and Z. Wan, “Blockchain-based IoT security authentication system,” *Proceedings - 2021 International Conference on Computer, Blockchain and Financial Development, CBFD 2021*, pp. 415–418, 2021, doi: 10.1109/CBFD52659.2021.00090.

[32] D. Li, W. Peng, W. Deng, and F. Gai, “A blockchain-based authentication and security mechanism for IoT,” *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, vol. 2018-July, Oct. 2018, doi: 10.1109/ICCCN.2018.8487449.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 8, No. 1, 27-37, 2025
DOI: 10.35377/saucis...1525350

RESEARCH ARTICLE

Face Super Resolution Based on Identity Preserving V-Network

Ali Hüsameddin Ateş[1,2], Hüseyin Eski[1]

[1] Sakarya University, Department of Computer Engineering, Sakarya, Türkiye, ror.org/04ttnw109
[2] Sakarya University, Institute of Natural Sciences, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Ali Hüsameddin Ateş,
Department of Computer Engineering, Sakarya University
aliates@sakarya.edu.tr

Article History:
Received: 31.07.2024
Revised: 27.02.2025
Accepted: 20.03.2025
Published Online: 27.03.2025

ABSTRACT
Numerous super-resolution methods have been developed to restore and upsample low-resolution and low-detail images to higher resolutions. Specifically, face super-resolution studies aim to restore various degradations in facial images while enhancing their resolution and preserving details. This study proposes the VNet architecture, which consists of a deep learning-based convolutional network for converting low-resolution and degraded facial images into high-quality and detailed images, and a pre-trained FaceNet model to preserve identity (biometric) information. The architecture leverages the advantages of the Encoder-Decoder structure bidirectionally to maintain details and recover lost information. In the initial stage, the Encoder module compresses the image representation, filtering out unnecessary information. The Decoder module then reconstructs the high-resolution and restored image from the compressed representation. The use of residual connections in this process helps minimize information loss while preserving details. The final stage utilizes the identity loss feedback from the FaceNet model to enhance the image without deviating from the original identity context. Tests conducted on various facial datasets demonstrate that VNet achieves high metric performance in both super-resolution and restoration tasks. The results indicate that the proposed architecture is effective in producing realistic and high-quality versions of low-resolution and degraded facial images.

Keywords: Face super resolution, Face restoration, Super resolution, Deep learning

1. Introduction

The transformation of low-resolution and low-detail images into high-resolution, detailed, and sharp images is studied under the title of super-resolution (SR) in the field of digital image processing. Various algorithms and methods are used to achieve super-resolution. Super-resolution is generally utilized in many areas such as improving image quality in photographs or videos, enhancing the resolution of medical imaging devices like magnetic resonance imaging (MRI) and computed tomography (CT), and making satellite images more detailed and clearer. The main goal in super-resolution is to improve visual quality and increase the resolution of the image while preserving details. On the other hand, super-resolution can be used as an auxiliary tool in tasks such as object detection and segmentation. Additionally, when looking specifically at facial images, super-resolution techniques are employed to enhance the effectiveness of security cameras, especially under challenging conditions such as low light, noise, and blur, by increasing the resolution of low-resolution images to facilitate face detection [1].

While there are many interpolation-based and similar super-resolution techniques in image processing, studies have shown that deep learning-based SR techniques perform higher than traditional methods. Especially GAN and CNN-based approaches can successfully enhance details in low-resolution images, recovering fine details lost in low-resolution facial images. Therefore, deep learning-based SR techniques are more widely used compared to classical methods. However, training and applying deep learning models require high computational power and large datasets. Additionally, the performance of SR algorithms is directly related to the quality and diversity of the datasets used. Therefore, creating large and diverse datasets containing facial images captured under different conditions and training SR techniques on these datasets is crucial for developing more generalizable and high-performance SR models [2].

Super-resolution techniques used for image enhancement gained a new deep learning-based perspective with Dong et al. [3] using convolutional neural networks in super-resolution. SRCNN proposed a three-layer structure. The first layer extracts patches from the low-resolution image and maps each patch to a high-dimensional space, the second layer performs a non-linear mapping of these high-dimensional representations from low-resolution patch space to high-resolution patch space,

Cite as: A. H. Ateş and H. Eski, “Face super resolution based on identity preserving V-Network,” *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 27-37, 2025, doi: 10.35377/saucis...1525350

and the third layer combines high-resolution patches to produce the high-resolution image. To reduce the computational complexity of SRCNN, Dong et al. also proposed FSRCNN, which uses deconvolution for upsampling.

Kim et al. proposed the Very Deep Super Resolution – VDSR [4] network by further deepening the networks suggested by Dong. The proposed network is based on VGGNet used for ImageNet classification and has 20 weight layers. Feature extraction was performed using 64 filters (3x3) in each layer. Additionally, a residual learning connection was established between the first and last layer to improve the learning rate, which decreases as the network deepens.

To address the vanishing gradient problem, which occurs due to the deepening of networks leading to the activation functions approaching zero and the network being untrainable, He et al. proposed Residual Networks (ResNet) [5]. Ledig et al. adapted it to super-resolution as SRResNet (Super Resolution Residual Network), achieving superior performance compared to previous techniques due to feedback connections that feed the deepening network [6].

Lim et al. removed the normalization layers in the SRResNet network to reduce unnecessary memory and computation costs during training, increased the model's capacity using a deeper structure, and achieved higher PSNR and SSIM with more residual blocks [7].

With the proposal of Dense Blocks (DenseNet) [8] by Huang et al., Tong et al. proposed the SRDenseNet network to enhance information flow, backpropagate gradients more effectively, and leverage the advantages of residual learning, achieving high clarity while preserving details and edges [9].

Upon the proposal of Generative Adversarial Networks (GAN) by Goodfellow et al. [10], Lim et al. proposed the SRGAN (Super-Resolution using a Generative Adversarial Network) network along with SRResNet, achieving higher accuracy outputs by using generative networks as an alternative to classical linear networks [6].

Wang et al. built on the SRGAN network to propose ESRGAN, removing the normalization layer in the residual block and using Residual-in-Residual Dense Blocks, achieving higher visual quality with the Relativistic Average GAN, which learns whether one image is more realistic compared to another [11].

General super-resolution networks showed low performance due to the high structural complexity of facial images, leading to specialized studies for facial images. Zhou et al. proposed a bi-channel convolutional neural network (BCCNN) for face super-resolution (FSR). To obtain super-resolution output, one channel extracts features related to face regions from the input image, while the other appropriately combines the extracted features with the low-resolution input image, achieving higher success on facial images with Gaussian and motion blur compared to classical CNN architectures [12].

Yu et al. proposed the UR-DGN (Ultra-Resolution by Discriminative Generative Networks) network, the first GAN-based network for face image restoration. Here, the discriminator network learns significant components of the human face, while the generator network combines these components with the input image. A pixel-based ℓ_2 loss term was used in the generator model, and the discriminator network's feedback was used to make the upsampled face images more similar to the real ones, enabling upsampling of very small inputs up to eight times [13].

In classical convolutional networks, each channel is treated equally during feature extraction, despite some channels carrying more features than others. Zhang et al. achieved more efficient results in general super-resolution by using channel-focused feature extraction (Channel Attention) that assigns different importance levels (weights) to each channel [14].

Zhao et al. proposed the SAAN (Semantic Attention Adaptation Network for Face Super-Resolution) network, using a semantic channel attention feature extraction method to protect important regions such as eyes, nose, and mouth in face images and achieve more realistic and detailed restoration. This channel-focused mechanism assigns different weights to different regions of the input image, emphasizing important facial features, achieving more efficient results in face image super-resolution [15].

Lu et al. proposed the FSAAN (Face Hallucination via Split-Attention in Split-Attention Network) network, using split-attention feature extraction by splitting the input channels, focusing on both facial texture details and structural information, achieving structurally richer faces [16].

With the adaptation of the Transformer architecture used in natural language processing to image processing as Vision Transformers (ViT) by Dosovitskiy et al. [17], an alternative to networks that process images using convolution in the classical CNN architecture emerged. Subsequently, many super-resolution networks using Transformer architectures were proposed.

Wang et al. leveraged the strengths of both CNNs and Transformer structures to enhance face image reconstruction, proposing the TANet (A new Paradigm for Global Face Super-resolution via Transformer-CNN Aggregation Network) architecture based on CNN and Vision Transformer (ViT). The CNN part is used for restoring fine details and local features of the face, while the Transformer part is used to maintain the consistency of the overall face structure with the Global Attention mechanism, achieving higher fidelity and naturalness in reconstructed face images [18].

Gao et al. proposed the CTCNet (A CNN-Transformer Cooperation Network for Face Image Super-Resolution) architecture, a hybrid CNN-Transformer network. CTCNet, with a U-Net-like structure, uses a combination of Facial Structure Attention

Unit and Transformer blocks on the encoder side to capture both local facial textures and general facial structural information. In the intermediate transition, the Feature Refinement Module focuses on the essential facial structure among extracted features, and the Multi-scale Feature Fusion Units module on the decoder side combines and up-samples the extracted local and global features, resulting in high-resolution images. Thanks to its hierarchical U-Net-like architecture with interconnections and advanced modular structure, CTCNet achieved higher results compared to many architectures in various benchmarks [19].

In studies on restoring or upsampling face images with super-resolution, the quality of the obtained image, high metric performance, and visual enhancements are often prioritized, while identity fidelity, the extent to which the output image resembles the original in terms of identity, is considered secondary. In this study, a model is proposed that harmonizes high metric performance and visual enhancements without compromising the inherent similarity of the generated output image to the original. To achieve this, in this study, VNet architecture is proposed that uses encoder-decoder based convolutional neural networks to obtain high resolution restored images from low resolution corrupted images. In the architecture, the image is restored and upsampled with V-shaped blocks similar to U-Net, while the biometric verification of the person or fidelity is performed with the FaceNet model used, preserving personal characteristics.

2. Material and Methods

2.1. VNet Architecture

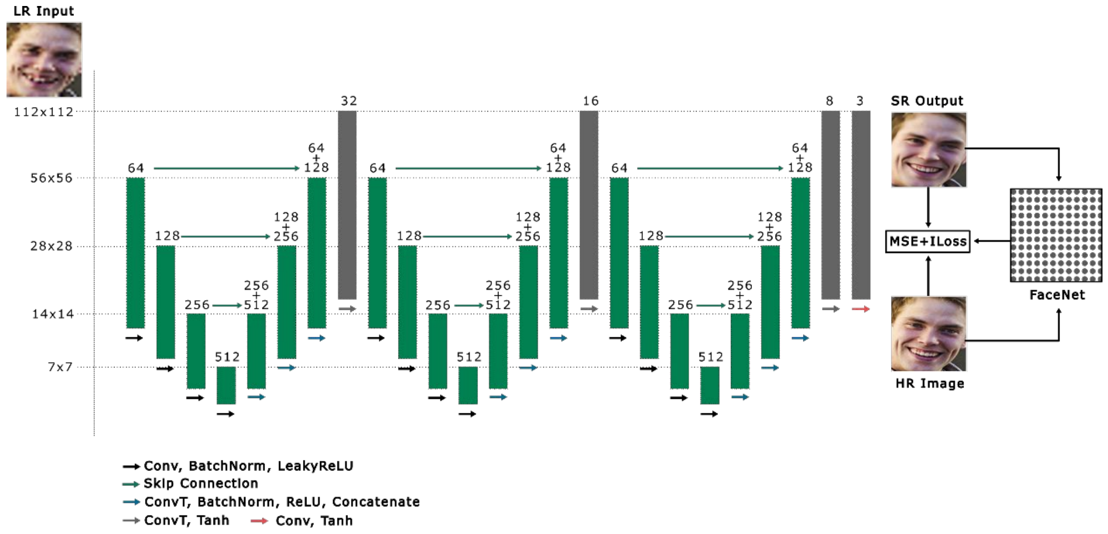


Figure 1. Proposed VNet Architecture

In Figure 1, the resolutions on the left indicate the output resolution of each convolution layer, the green and gray bars indicate the convolution layers, the numbers above the bars indicate the number of convolution filters, the jump links between the layers indicated by the green arrows indicate the merging process between the layers, the green bars indicate the encoder-decoder layers, the gray bars indicate the intermediate transition layers between the decoder and encoder, and the colored arrows indicate the operations performed in the layers.

Convolutional Neural Networks from deep learning architectures are used for the super-resolution of face images in the VNet network. The network, designed with an encoder-decoder architecture similar to U-Net, connects layers with residual learning shortcuts proposed in the ResNet architecture. Residual Learning aims to address the vanishing gradient problem by preventing the gradient function from converging to zero as the network deepens and layers increase.

In the VNet network, feature extraction from the image is performed through V-shaped green layers, and the extracted features are combined while maintaining the input resolution of the image through gray intermediate layers. The network, which has a total of 33 million trainable parameters, processes input images through a data preprocessing step. The network operates by feeding pairs of original (HR) and distorted (LR) images, where the original image is subjected to a distortion function to obtain an LR image. The distortion function reduces a 112x112 pixel image to a size of 28x28 pixels, then enlarges it back to its original size to produce the LR image.

No prior-based approach is used in architecture. The network takes a holistic approach to input images, learning the distortion function between HR-LR image pairs through various layers and connections, and attempts to predict the original image from the degraded image by learning the inverse of this function.

The overall structure of the network is shown in Figure 1, where the output resolution of the image is indicated after each layer in the left chart. The image resolution, which continuously halves after each layer, is restored to the input size with intermediate convolutions having 32, 16, and 8 filters, and finally, a super-resolved RGB image (SR output) is obtained with a 3-filter convolution.

The VNet network has a fixed input size, receiving inputs of 112x112 pixels. The input image progresses through the green blocks specified by the number of filters from left to right. The V-shaped green blocks consist of 4 encoder layers shown with black arrows, 3 decoder layers shown with blue arrows, and finally, 1 transposed convolution layer.

The input layer consists of 64 convolution filters, and in subsequent layers, the number of filters doubles in the encoder layers, reaching up to 512 filters as the network deepens. Then, in the decoder layers, the number of filters is halved, making the network shallower. All convolution filters used in the network are 4x4 in size.

In the encoder layers, convolution operations use a stride of 2 and same padding. This ensures that the image resolution is halved smoothly after each layer. Filter weights are initialized with a random starter value between 0 and 0.02. Batch normalization is used for optimization after convolution, and the activation function used is LeakyReLU.

In the decoder layers, the transposed convolutions also use a stride of 2, same padding, and random starter values between 0 and 0.02. Batch normalization is used for optimization, and the activation function is ReLU. The transposed convolution and same padding used ensure that the halved image resolution is smoothly doubled from layer to layer. The 512 filters used in the deepest part of the network are halved through the decoder layers; each layer's filter weights are combined with the filter weights (embeddings) of the corresponding layer with the same resolution as the image resolution in that layer. The merging process (green arrows in Figure 1) combines encoder and decoder features, refining feature maps and retaining important details from the original image.

In the gray layers with 32-16-8 filters following the decoder layers, the image resolution is restored to the input resolution through transposed convolutions, while feature maps are transferred between V-shaped blocks. The transposed convolution process uses a stride of 2, same padding, and a Tanh activation function.

In the final layer, the feature maps obtained from restoring and super-resolving the low-resolution and detailed input image through the network are combined in RGB channels with a 4x4 filter convolution and Tanh activation function, resulting in a high-resolution and detailed image.

As the loss function of the network, both Mean Squared Error (MSE) and Identity Loss (ILoss) functions are used together, as suggested by Khazaie et al. [20]. To obtain the Identity Loss, the super-resolved image produced by the network is paired with the original image and fed into a face recognition network, FaceNet [21]. The pre-trained FaceNet model [22], which was trained using the VGGFace2 dataset, is a successful network for obtaining filtered data (embeddings) related to human faces. The MSE function aims to numerically approach the original image by examining the pixel-based numerical difference between the original and predicted images. On the other hand, Identity Loss evaluates how similar the predicted image is to the original image of the person, providing feedback through similarity errors. This approach ensures that the output image not only has high visual quality and detail but also maintains similarity to the original person.

2.2. Dataset

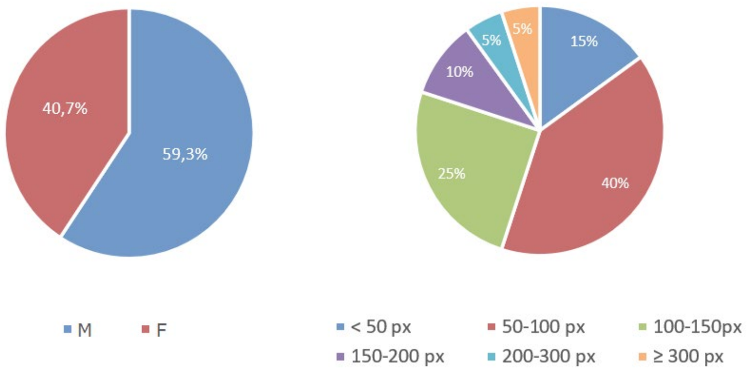


Figure 2. VGGFace2 Dataset Distribution [23]

The VGGFace2 dataset announced by Cao et al. [23] was used for training the proposed VNet network. This dataset contains 3.31 million face images of 9,131 individuals, obtained from Google Images, covering a wide range of poses, lighting conditions, ethnic backgrounds, races, ages, and professions (actors, athletes, politicians, etc.). With an average of 362 images per person, the dataset also includes facial bounding boxes, and the images vary in resolution and are loosely cropped.

In the data pre-preparation phase, for computational and time gain purposes, approximately 100 images were randomly selected from the images of each person from the dataset (9131 people in total), making a total of 863,732 images. The face images in the selected images were tightly cropped with a face detection tool, and the images with different resolutions were resized to 112x112 pixels.

2.3. Metrics

Several evaluation metrics are used to measure the similarity between images in terms of both quantity and quality [24]. These metrics, commonly used in the field of super-resolution, provide numerical information on how well the distorted image has been restored compared to the original image by establishing a similarity relationship using different methods and approaches between the two versions of the image. In this study, MAE (Mean Absolute Error), MSE (Mean Squared Error), PSNR (Peak Signal-To-Noise Ratio) metrics are used as pixel-based metrics and SSIM (Structural Similarity Index Measure) and LPIPS (Learned Perceptual Image Patch Similarity) metrics are used as perceptual metrics.

MAE and MSE focus on the numerical difference between the pixels of two images. As the images become more similar and the numerical difference between them decreases, these error values decrease. PSNR measures the error rate logarithmically with respect to the maximum pixel value between the two images, again using MSE. A high PSNR value indicates a high similarity between the two images. These pixel-based metrics focus on pixel-wise differences and do not provide information about perceptual similarity. These metrics range from a minimum value of 0 (perfect agreement) to a maximum value of ∞ (maximum error), with similarity approaching a maximum as the value approaches zero.

The SSIM structural similarity index measures perceptual similarity by comparing the brightness, contrast, and structural information between two images. The SSIM value ranges from 0 to 1. The SSIM value approaches 1 as the similarity between two images increases and 0 as the similarity decreases. LPIPS is a perceptual metric that measures image similarity by comparing features extracted from deep learning models and provides results closer to human visual perception. Although there is no exact range of values for the LPIPS metric, in practice, values close to 0 indicate a high similarity between two images, while values close to 1 or higher indicate that the similarity of these images is very low [25].

Table 1. VNet Model Hyperparameters

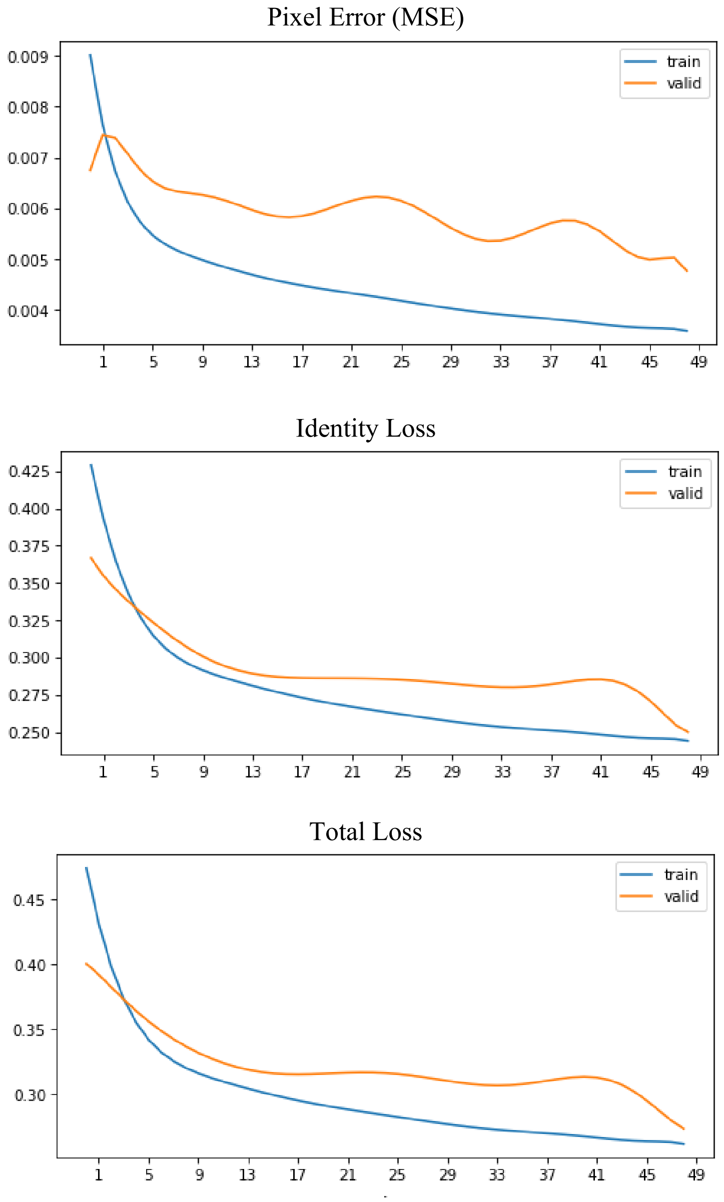
Hyperparameter	**Value**
Input Size	112x112
Input/Output Channels	3
Convolution Filter Size	4x4
Convolution Strides	2
Convolution Padding	Same
Use Bias	False
Apply Dropout	False
Activation Functions	Encoder Layers: LeakyReLU Decoder Layers: ReLU Gray Layers: Tanh
Loss Function	MSE + Identity Loss
Optimizer Function	Adam

3. Experiments

For training, the hardware used includes an Intel Core i9-7960X CPU and an Nvidia RTX 4090 24GB GPU, while the software includes Python 3.11.9 and TensorFlow 2.15.0. The VNet network was trained for 49 epochs using 863,732 images selected from the VGGFace2 dataset with the parameters shown in the table below (as seen in Table 1).

The number of images in the dataset, input size and epochs are kept minimum considering the hardware constraints, and more sample, input size and epochs will give better results. The error function of the network is calculated by the sum of MSE and Identity Loss errors and both pixel and perceptual based error control are performed. More efficient results can be obtained by using different weights while summing these two error functions. Since RGB images have 3 channels, the number of input and output channels must be 3. 4×4 filters keep the number of parameters and computational cost low compared to larger filters such as 5×5 or 7×7, while providing a wider detection area compared to 3×3 filters. Striding 2 enables learning hierarchical features by changing the resolution step by step in the encoder-decoder structure by downsampling and upsampling, halving the image size in each convolution and doubling it in deconvolution. In addition, equal padding is used to prevent the change in spatial size. Since Batch normalization is used as the normalization layer, no additional Bias usage was required. Dropout wasn't used to ensure that the data set was sufficiently diverse and to prevent unwanted visual artifacts. In the encoder layers, LeakyReLU prevents the vanishing gradient problem by keeping the negative region active with a small slope, while in the decoder layers ReLU encourages positive outputs and provides smooth outputs in image production. Using Tanh in gray layers allows the pixel values of the images to be pulled into a normalized range (-1,1). Thanks to the adaptive learning rates, momentum utilization and fast convergence features of the Adam optimization function, it enables the complex structure of the model and detailed parameter updates to be managed effectively and the fine details in the image to be learned more reliably.

The outputs of the Mean Squared Error (MSE), Identity Loss, and total loss functions used during training are shown in the graphs for the training and validation phases (as seen in Figure 3). The closely aligned training and validation loss outputs indicate the stability of the training process and the network.


Pixel Error (MSE)

Identity Loss

Total Loss

Figure 3. Train and Validation Losses

3.1. Comparative Results

After the training process, various evaluation metrics were used to quantitatively measure the difference between the original and super-resolved images during the testing phase, in order to assess the model's performance on the test data. The metrics used include MAE (Mean Absolute Error), MSE (Mean Squared Error), PSNR (Peak Signal-To-Noise Ratio), SSIM (Structural Similarity Index Measure), and LPIPS (Learned Perceptual Image Patch Similarity) [24].

For the test datasets, VGGFace2 [23], CelebA [26], UTKFace [27], FEIFace [28], and MultiPie [29] datasets were used, each consisting of face images obtained through different methods and purposes, with 2,000 images randomly selected from each. The evaluation metrics for the images obtained by enlarging 28x28 pixel resolution images by a factor of 4 are shown in Table 2.

VGGFace2 provides a large number of samples and people, a wide age range, demographic diversity and pose richness, while CelebA stands out in learning facial attributes by providing 40 different feature labels for each face. UTKFace has a relatively small dataset with images rich in age, gender and ethnicity and focuses on age in general, while FEIFace is useful in making sense of facial components thanks to its small scale but high-quality images. Multi-PIE, on the other hand, can prevent performance degradation against illumination and pose changes thanks to images of individuals taken from different angles, in various poses and lighting conditions.

Table 2. Test Datasets ×4 Test Scores

Test Dataset	**MAE↓**	**MSE↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**
VGGFace2	.02812	.00169	28.730	0.8806	0.0937
CelebA	.02757	.00172	28.271	0.8891	0.0835
UTKFace	**.02650**	**.00163**	**28.982**	**0.8999**	**0.0626**
FEIFace	.02717	.00165	28.769	0.8980	0.0747
MultiPie	.03984	.00362	25.884	0.8800	0.1131

When we look at the results obtained, the UTKFace dataset has the highest metric performance, while the other test datasets have similar values. The UTKFace dataset shows the highest metric performance due to its high-quality images that are standardized and well-aligned in terms of angle and exposure, while the MultiPie dataset shows lower metric performance due to its richness in different angles, lighting and exposure. On the other hand, the low LPIPS score (0.0626) obtained on the UTKFace dataset reflects superior perceptual similarity, consistent with high PSNR and SSIM values. The fact that the results obtained on the test datasets are close to each other is important in obtaining a balanced and generalizable model.

The images from the VGGFace2, CelebA, UTKFace, MultiPie, and FEIFace datasets, including their originals (HR), distorted versions reduced to 28x28 pixel resolution (LR), and the outputs obtained by upscaling the distorted images by a factor of 4 (SR), are shown sequentially in Figure 4.

When looking at the visual test outputs, it is seen that VNet shows a sampling performance suitable for the original image in the tests based on increasing the image size to 4 times super resolution, in the restoration of eyes, mouth, nose and other facial components that are not present in the distorted image; in the repair and upsampling of blurry, noisy and distorted textural regions. While doing this, as seen in detail in Figure 5, the low-resolution image passes through the filters in the encoder and decoder layers of the model. Each filter reconsiders the image with its learned weights. While the initial layers focus on low-level features such as edges and corners, emphasizing the contrasting regions of the image, simple lines and prominent edges; when the image reaches the mid-level layers, textures and facial features are addressed. When the network reaches the high-level layers, the output of the filters becomes more abstract as the network gets deeper. The filter outputs (as seen in Figure 5) may appear blurry and complex while being simple at the beginning; because the network has now learned high-level features that can distinguish and restore the entire face or certain key regions. This deep learning allows the model to create a rich feature representation. As a result, the model updates its weights to reproduce the face based on this information. Each filter capturing a feature of a specific part of the image, and the image given as input to the model is processed one by one by these filters to restore the image.

4. Conclusion

In this study, VNet architecture is proposed, which aims to protect biometric identity information in the restoration of facial images with super resolution. The identity preservation capability of the model is supported by the FaceNet-based identity loss function. In this way, critical facial components such as eyes, nose and mouth in images generated from low-resolution inputs are restored consistent with the biometric features of the original person. The model trained with the VGGFace2 dataset was tested on both VGGFace2 images not used in training and various datasets such as CelebA, UTKFace, FEIFace and MultiPie in 4-fold super-resolution tests. The obtained MAE, MSE, PSNR, SSIM and LPIPS metric scores revealed that the model exhibited stable and generalizable performance on different datasets. The reason for the relatively low SSIM value is that this metric considers brightness and contrast similarity as well as structural similarity when comparing images. Although the VNet model provides sufficient improvements structurally, it needs to be trained and optimized further in terms of brightness and contrast.

The UTKFace dataset achieved the highest metric score thanks to its standard and well-aligned high-quality images in terms of angular and exposure. On the other hand, relatively low metric performances were obtained in the tests conducted on the MultiPie dataset. The reason for this is that the images in the other test dataset were mostly taken from the front, while the MultiPie dataset consists of face images taken from many angles and with different exposures. In order to prevent such handicaps, the datasets used in training should be diversified, rich in terms of angular and exposure, and the number of samples should be as high as possible. Again, the integration of advanced loss functions to improve brightness and contrast restoration, attention mechanisms to improve global and local feature extraction, and the Transformer architecture are among the important improvements to present a more advanced and efficient architecture.

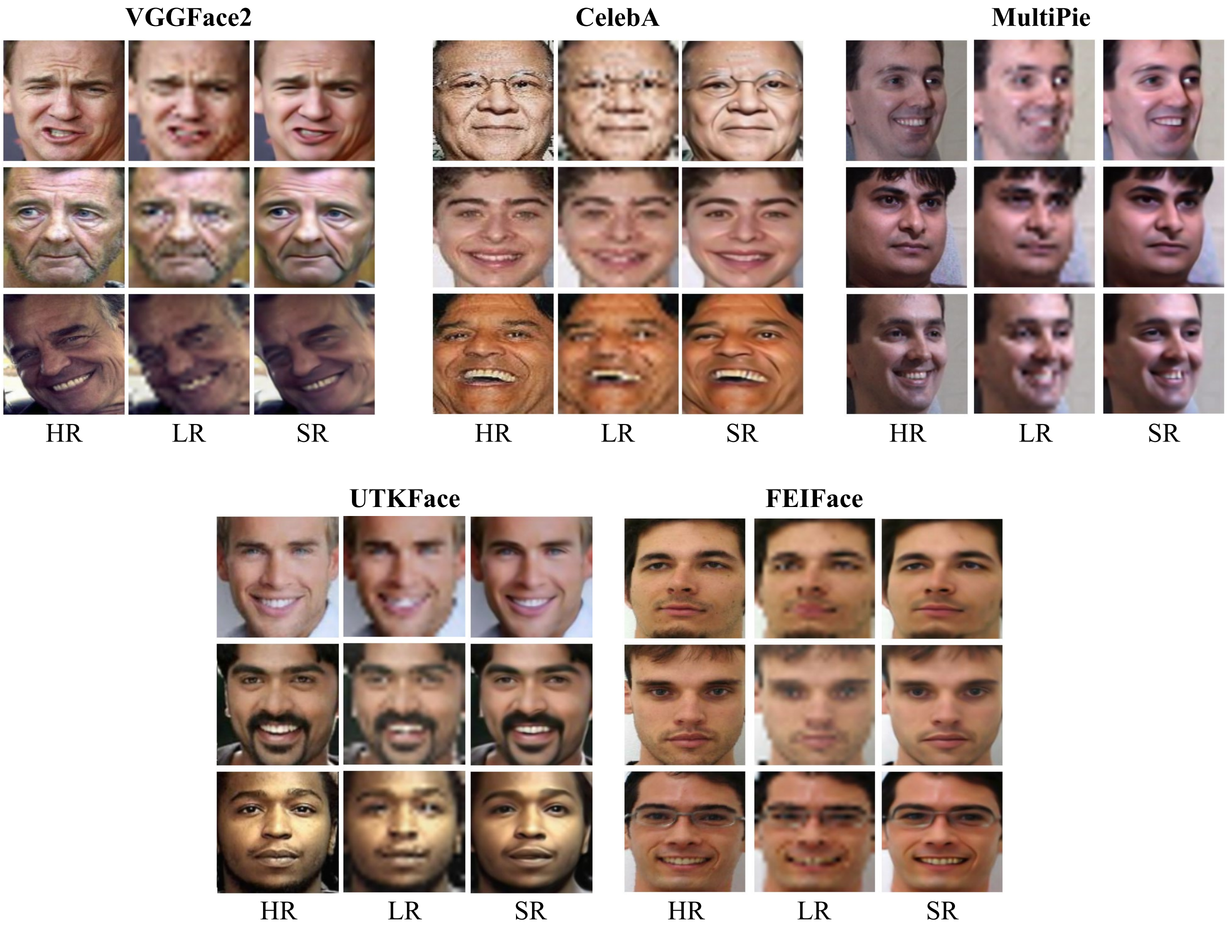


Figure 4. Test Datasets ×4 Outputs

This study offers a methodology that can be referenced especially in areas where identity verification is critical, such as security systems and forensic analysis. However, more advanced architectures are needed for images under difficult conditions. Expanding the capacity of the model with increased data diversity and hybrid architecture is of critical importance for the integration of the model to real-world conditions.

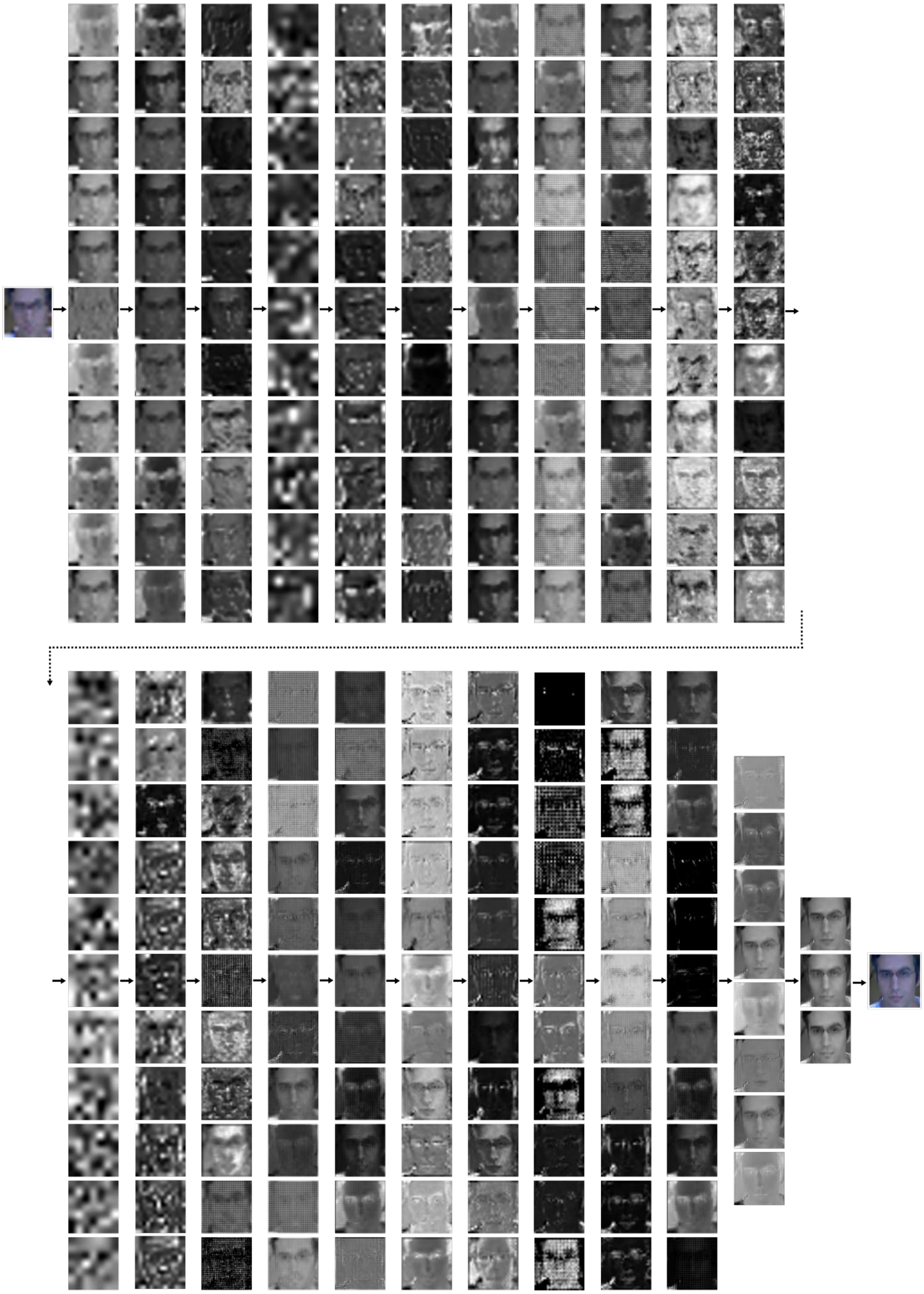


Figure 5. Sample Outputs of Model Layers (Layer Dimensions Shortened)

References

[1] N. Singh, S. S. Rathore, and S. Kumar, “Towards a super-resolution based approach for improved face recognition in low resolution environment,” *Multimed Tools Appl*, vol. 81, no. 27, pp. 38887–38919, Nov. 2022, doi: 10.1007/S11042-022-13160-Z/FIGURES/16.

[2] J. Jiang, C. Wang, X. Liu, and J. Ma, “Deep Learning-based Face Super-Resolution: A Survey,” *ACM Comput Surv*, vol. 55, no. 1, Jan. 2021, doi: 10.1145/3485132.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution Using Deep Convolutional Networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 2, pp. 295–307, Dec. 2014, doi: 10.1109/TPAMI.2015.2439281.

[4] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 1646–1654, Nov. 2015, doi: 10.1109/CVPR.2016.182.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.

[6] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 105–114, Sep. 2016, doi: 10.1109/CVPR.2017.19.

[7] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 1132–1140, Jul. 2017, doi: 10.1109/CVPRW.2017.151.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Aug. 2016, doi: 10.1109/CVPR.2017.243.

[9] T. Tong, G. Li, X. Liu, and Q. Gao, “Image Super-Resolution Using Dense Skip Connections”.

[10] I. J. Goodfellow *et al.*, “Generative Adversarial Nets”, Accessed: May 07, 2024. [Online]. Available: http://www.github.com/goodfeli/adversarial

[11] X. Wang *et al.*, “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11133 LNCS, pp. 63–79, Sep. 2018, doi: 10.1007/978-3-030-11021-5_5.

[12] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, “Learning face hallucination in the wild,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, in AAAI’15. AAAI Press, 2015, pp. 3871–3877.

[13] X. Yu and F. Porikli, “Ultra-resolving face images by discriminative generative networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9909 LNCS, pp. 318–333, 2016, doi: 10.1007/978-3-319-46454-1_20/TABLES/1.

[14] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image Super-Resolution Using Very Deep Residual Channel Attention Networks,” 2018.

[15] T. Zhao and C. Zhang, “SAAN: Semantic Attention Adaptation Network for Face Super-Resolution,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6. doi: 10.1109/ICME46284.2020.9102926.

[16] T. Lu *et al.*, “Face Hallucination via Split-Attention in Split-Attention Network,” in *Proceedings of the 29th ACM International Conference on Multimedia*, in MM ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 5501–5509. doi: 10.1145/3474085.3475682.

[17] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ICLR 2021 - 9th International Conference on Learning Representations*, Oct. 2020, Accessed: Jul. 10, 2024. [Online]. Available: https://arxiv.org/abs/2010.11929v2

[18] Y. Wang *et al.*, “TANet: A new Paradigm for Global Face Super-resolution via Transformer-CNN Aggregation Network,” Sep. 2021, Accessed: Jul. 10, 2024. [Online]. Available: https://arxiv.org/abs/2109.08174v1

[19] G. Gao, Z. Xu, J. Li, J. Yang, T. Zeng, and G.-J. Qi, “CTCNet: A CNN-Transformer Cooperation Network for Face Image Super-Resolution,” *IEEE Transactions on Image Processing*, vol. 32, pp. 1978–1991, Apr. 2022, doi: 10.1109/TIP.2023.3261747.

[20] V. R. Khazaie, N. Bayat, and Y. Mohsenzadeh, “Multi Scale Identity-Preserving Image-to-Image Translation Network for Low-Resolution Face Recognition,” *Proceedings of the Canadian Conference on Artificial Intelligence*, Oct. 2020, doi: 10.21428/594757db.66367c17.

[21] “davidsandberg/facenet: Face recognition using Tensorflow.” Accessed: Jul. 15, 2024. [Online]. Available: https://github.com/davidsandberg/facenet?tab=MIT-1-ov-file#readme

[22] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 815–823, Mar. 2015, doi: 10.1109/cvpr.2015.7298682.

[23] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,” in *International Conference on Automatic Face and Gesture Recognition*, 2018.

[24] T. Wang *et al.*, “A Survey of Deep Face Restoration: Denoise, Super-Resolution, Deblur, Artifact Removal,” Nov.

2022, Accessed: May 08, 2024. [Online]. Available: https://arxiv.org/abs/2211.02831v1

[25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–595, Jan. 2018, doi: 10.1109/CVPR.2018.00068.

[26] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," *CoRR*, vol. abs/1411.7766, 2014, [Online]. Available: http://arxiv.org/abs/1411.7766

[27] S. Y. Zhang Zhifei and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image Vis Comput*, vol. 28, no. 6, pp. 902–913, Jun. 2010, doi: 10.1016/J.IMAVIS.2009.11.005.

[29] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis Comput*, vol. 28, no. 5, pp. 807–813, May 2010, doi: 10.1016/J.IMAVIS.2009.08.002.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

Not applicable / or link

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 38-46, 2025
DOI: 10.35377/saucis...1627619

RESEARCH ARTICLE

Classification and Analysis of Employee Feedback with Deep Learning Algorithms

Gökhan Yiğidefe[1]*, Serap Çakar Kaman[1], Beyza Eken[1]

[1]Sakarya University, Faculty of Computer and Information Sciences, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Gökhan Yiğidefe,
Sakarya University,
Department of Computer Engineering
gokhan.yigidefe1@ogr.sakarya.edu.tr

Article History:
Received: 27.01.2025
Revised: 07.02.2025
Accepted: 27.02.2025
Published Online: 27.03.2025

ABSTRACT
This study aims to enhance organizational processes and support decision-making for managers by conducting an automated analysis of employee feedback through text classification of Turkish sentences. Employee satisfaction and motivation are critical factors that directly impact sustainability and efficiency goals. To overcome the challenges of manual feedback analysis, the study employs Temporal Convolutional Network (TCN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) algorithms. The dataset comprises feedback collected from meeting notes, internal surveys, and manager-employee interviews, with data synthesis and preprocessing steps including text cleaning, tokenization, and modelling. The study's findings reveal that the CNN algorithm achieved the best performance, with an accuracy of 99.12%, a test loss of 6.09%, precision of 99.12%, recall of 99.12%, and an F1 score of 99.11%. This research demonstrates the valuable contribution of automated classification models in effectively and efficiently analysing employee feedback.

Keywords: Employee feedback classification, TCN, CNN, LSTM, BERT, Deep learning for text analysis

1. Introduction

Employee satisfaction and motivation are fundamental to organizational success in today's business world. Effectively analyzing feedback from employees not only guides leadership processes but also informs organizations' strategic planning efforts. In processing large-scale datasets, machine learning and deep learning techniques accelerate the analysis process and provide cost advantages. Text classification algorithms play a crucial role by categorizing feedback into meaningful groups, thereby establishing decision-support mechanisms and contributing to developing innovative organizational strategies.

Text classification methods have broad applications, including customer relationship management, healthcare, education, law, and marketing. Examples such as categorizing customer complaints, classifying patient records, analyzing student feedback, organizing legal documents, and examining social media data highlight the functionality of this technology. Supported by artificial intelligence and deep learning methods, text classification technologies automate the transformation of unstructured data into actionable information, making processes more efficient and effective. Consequently, they serve as a critical tool in strategic decision-making processes.

The main purpose of this study is to contribute to organizational decision-making processes by analyzing employee feedback consisting of Turkish sentences. The ultimate objectives include deriving meaningful insights from feedback to enhance employee satisfaction, motivation, and engagement, transforming these insights into actionable information, and improving organizational efficiency. In the modern business world, such approaches, which serve the principles of transparency, rapid decision-making, and continuous improvement, play a strategic role in transforming business processes.

2. Literature Review

In literature, several machine learning methods have been employed for text classification. In the study by Kayakuş et al., 10,500 news articles collected from five news websites in Turkey were categorized into three classes: world, sports, and economy, using Naive Bayes and decision tree methods. Naive Bayes demonstrated better performance with an accuracy of 88.66% [1]. Bozkurt et al. classified Amazon food reviews using Random Forest (RF), CatBoost, and XGBoost algorithms, where RF achieved the highest performance with an accuracy of 90.22% [2]. Tuna et al. proposed a model for determining

Cite as: G. Yiğidefe, S. Çakar Kaman, and B. Eken, "Classification and analysis of employee feedback with deep learning algorithms," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 38-46, 2025, doi: 10.35377/saucis...1627619

target categories for Turkish texts, showing that the FastText model delivered the best performance in identifying target terms [3]. In their study on IMDB movie reviews, Öğe et al. found that Logistic Regression and SVM algorithms performed well when combined with the Word2Vec method [4]. In another study by Metin et al., human activity classification was conducted using gyroscope and accelerometer data, achieving 97% and 99% accuracy with TSA and ESA methods, respectively. The study also introduced a new dataset and software tools for human activity classification [5]. Aydemir et al. categorized Turkish news articles into eight distinct categories, where the RF algorithm achieved the best performance with an accuracy of 99.86% [6]. Akgümüş et al. demonstrated that the Multinomial Naive Bayes model achieved a 99% accuracy rate and effectively classified customer complaints in the banking sector [7].

Literature has increasingly utilized deep learning methods for text classification in recent years. In the study by Ertem et al., LSTM and feature selection techniques were used to detect COVID-19 vaccine opposition with an accuracy of 99.23%. Data imbalance was addressed using the SMOTE method and TF-IDF [8]. Demirbilek et al. conducted sentiment analysis on Google reviews of a university in Central Anatolia using LSTM and machine learning methods, where Amazon Comprehend demonstrated the best performance across all metrics [9]. Çataltaş et al. analyzed Turkish COVID-19 tweets, showing that a CNN-LSTM model achieved 76% and 84% accuracy for sentiment classification [10]. Güler et al. examined Turkish news articles and e-commerce reviews, where their KSA-based deep learning model achieved accuracies of 91.7% and 95.6%, respectively [11]. Yılmaz et al. classified 28,104 requests in a help desk system, achieving 97.60% accuracy with the LSTM model [12]. Budak et al. found that deep-learning methods performed better in analyzing airline reviews before and after COVID-19 [13]. In their study, Sel et al. employed BERT, LSTM, and CNN models to predict gender from Turkish Twitter posts, with BERT achieving the highest accuracy of 80.1% [14]. Acı et al. used Word2Vec and KSA methods for Turkish news articles, demonstrating that KSA provided 93.3% higher accuracy than classical methods [15]. Bişkin's study applied TCN to forecast COVID-19 cases in European countries, showing that TCN outperformed LSTM and GRU models in terms of lower computation time and higher prediction accuracy [16]. Kasapbaşı et al. aimed to convert Turkish Sign Language (TİD) gestures into text using CNN-based deep learning models, achieving a high accuracy of 98% [17]. Erol et al. conducted sales forecasting using models such as CNN, LSTM, and GRU, concluding that LSTM and its variants performed best, particularly on datasets with seasonality and trends [18]. Tuna et al. demonstrated that the DeepCusComp-1 model achieved 85.83% accuracy in classifying customer complaints, outperforming other methods [19]. Aydın's study compared LSTM and BERT-based models, revealing that BERT outperformed LSTM [20]. Arslan et al. showed the success of BERT-based models in stance detection using social media data [21]. Gür compared CNN, LSTM, and GRU models, finding that a hybrid CNN-LSTM-GRU model achieved the lowest error rates and the best R^2 values [22]. Demirbilek et al. compared AWS Comprehend with deep learning methods, noting that AWS Comprehend achieved the highest performance across all metrics [23]. Kahraman et al. identified BERT as an effective tool for classifying patent texts [24]. Aydın et al. found that BERT-based models were more efficient than LSTM models in processing time and accuracy [25]. Sel et al. highlighted that BERT achieved high accuracy even on short and unstructured texts, performing well in gender prediction through Twitter-based analyses [26].

This study was conducted to address several significant gaps in literature. First, while most existing research focuses on customer feedback or social media data, there is a limited number of studies analyzing organizational internal data, particularly employee feedback. Given the critical importance of intrinsic factors such as employee satisfaction and productivity for organizations, addressing this gap is essential. Second, while literature often focuses on the performance of a single model, this study provides a comparative analysis of different algorithms, including TCN, CNN, LSTM, and BERT, thereby highlighting the effectiveness of hybrid approaches. Lastly, although preprocessing steps such as data cleaning and tokenization receive limited emphasis in literature, this study thoroughly examines the impact of these steps, aiming to bridge gaps in Natural Language Processing (NLP). In this context, the study provides a valuable contribution to supporting decision-making processes in organizational settings.

3. Background Work

TCN, CNN, LSTM, and BERT are foundational architectures in modern deep learning. While TCN and LSTM excel in time series analysis, CNN dominates image processing, and BERT performs well in NLP tasks. These models extract data features through unique mechanisms, effectively solving complex problems. Notably, BERT stands out in NLP with its bidirectional context understanding, TCN and LSTM effectively model temporal dependencies, and CNN efficiently captures visual features.

TCN is a model designed to process time series data and learn long-term dependencies efficiently and hierarchically. Through 1D convolutional layers, pooling, and normalization processes, TCN captures temporal information. Its encoder-decoder architecture generates and expands compressed representations. TCN is recognized for its faster training than RNNs and LSTMs [27]. With its convolutional layers, pooling, and Rectified Linear Unit (ReLU) activation functions, CNN is prominent in image processing. The convolutional layers extract local features, while pooling reduces dimensionality and prevents overfitting. This structure performs highly in object detection, segmentation, and image classification tasks [28]. LSTM relies on cells equipped with input, forget, and output gates to selectively control information. The cell state retains critical information and addresses the vanishing gradient problem. This architecture is widely used in NLP, speech recognition, and time series forecasting due to its ability to learn both short- and long-term dependencies [29]. BERT is an NLP model with bidirectional context understanding based on the Encoder portion of Transformer architecture. It learns contextual relationships through tasks such as Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The Classification Token (CLS) and Separator Token (SEP) tokens are particularly effective for classification and sentence relationship tasks. With fine-tuning, BERT excels in sentiment analysis, question answering, and natural language understanding tasks [30].

4. Methodology

The flow chart in Figure 1 depicts the methodology we used in this study. It involves dataset construction, data preprocessing, construction of deep learning models, and training and evaluation using cross-validation. All these steps are elaborated in the following subsections.

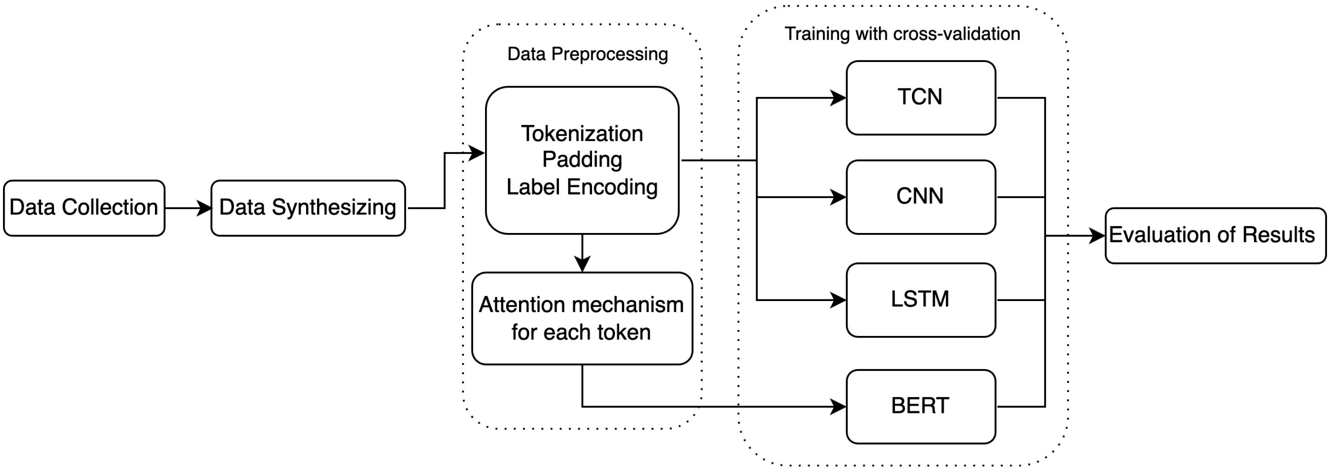


Figure 1. Methodology of the Study

4.1. Dataset

The 386 Turkish sentences obtained from meeting notes, internal surveys, and manager-employee discussions in a private company in Turkey were analyzed, identifying 14 categories. All sentences were anonymized and matched with the relevant categories. Since more data was needed for model training, the existing dataset was used to teach ChatGPT the sentence-category matching process. Subsequently, 500 synthetic Turkish sentences were generated for each category, resulting in 7,000 sentences. The length of these synthetic sentences was designed to be a maximum of 17 words, with an average of 11 words. The identified categories are based on common types of employee feedback regarding work processes, and each category was assigned a label number. Table 1 presents the categories in the dataset along with example sentences.

Table 1. Real and Synthetic Sentences by Categories

Label No	Category	Data Count	Real Sentence	Synthetic Sentence
1	Lack of Information	500	İş birimleri her talep açışında nasıl açılacağını soruyor. (Business units ask how to open requests every time they need to.)	Yöneticilerden gerekli detayları zamanında alamadığımız için projelerimiz aksıyor. (Projects are delayed because we cannot get the necessary details from managers on time.)
2	Work Environment	500	Campus gerçekten güzel. Yeni ofise beğenerek gidiyorum. (The campus is truly beautiful. I enjoy going to the new office.)	Çalışma alanlarının yetersizliği ekip içinde verimliliği olumsuz etkiliyor. (The lack of adequate workspaces negatively impacts team productivity.)
3	Change and Planning Management	500	IT tarafında hala ekipler arası iletişimin veya etkileşimin az olduğunu, birlikte çalışma ortamları için yeni planlamalar bekliyoruz. (We expect new plans to create collaborative environments as there is still limited interaction between IT teams.)	Planlamalar önceden paylaşılınca işler daha hızlı ilerliyor. (When plans are shared in advance, work progresses faster.)

Table 1. (Continued)

4	Training	500	İş biriminine ne zaman yeterli eğitim verilecek akış süreci için (When will sufficient training be provided to the business unit for the workflow process?)	Eğitimlerin yetersiz olması, çalışanların gelişimini olumsuz etkiliyor. (Insufficient training negatively affects employee development.)
5	Team Harmony	500	Ekip içi iletişim şahane herkes yardımsever. İyi ki bu ekibin bir parçasıyım. (Team communication is fantastic; everyone is helpful. I'm glad to be part of this team.)	Düzenli ekip toplantıları, iş birliğini artırıyor. (Regular team meetings enhance collaboration.)
6	Event Needs	500	Bir gün birlikte bir yerde çalışma yapsak. Gelsin hackathon. (Can we have a day of collaborative work somewhere? Let's organize a hackathon.)	Şirket piknikleri, çalışanların iş dışında da bağ kurmasını sağlar. (Company picnics help employees build bonds outside of work.)
7	Job Descriptions and Responsibilities	500	İş biriminin hiç bir şey söylemeden acil işleri kendileri yapmaları, sonra da ben bunu yapamadım IT bana yardım etsin deyip gece gündüz demeden aramaları hoş değil. Bu iş planlı olmalı herkes ona göre plan yapmalı. (The business unit takes over urgent tasks without informing anyone and later requests IT's help, which disrupts planning.)	Belirgin görev tanımları, çalışanların iş memnuniyetini artırır. (Clear job descriptions increase employee satisfaction.)
8	Welcome Kit	500	İşe başlamada hoşgeldin kiti olmaması üzücü. (It's disappointing not to have a welcome kit upon starting work.)	Hoşgeldin kitinin özenle hazırlanmış olması, yeni çalışanlara değer verildiğini hissettiriyor. (A thoughtfully prepared welcome kit makes new employees feel valued.)
9	Personal Requests	500	Personel avansı yada kredi için aksiyon alınmalı bir çok bankada olan bir süreç bizde neden yok. (Actions should be taken for employee advances or loans. Why don't we have this process like other banks?)	Yıllık izin günlerimizin arttırılmasını isterim. (I would like the number of annual leave days to be increased.)
10	Approval Processes	500	Paket oluşturma onaylar vs çok zaman alıyor ve yıpratıyor. (Package creation and approvals take too much time and are exhausting.)	Onay süreçlerinin dijitalleştirilmesi, zaman kazandırabilir. (Digitizing approval processes can save time.)
11	Staffing Shortages	500	İş birimleri kendi kaynak eksikliklerini IT deki kişileri kendi kaynakları gibi kullanarak çözmeye çalışıyorlar. (Business units try to address their staffing shortages by treating IT staff as their own resources.)	Personel eksikliği yüzünden zamanında sonuç alınamıyor. (Staffing shortages prevent timely results.)
12	Health Insurance	500	Sağlık sigortasının aileyi kapsayacak şekilde olmaması çok üzücü (It's disappointing that health insurance doesn't cover families.)	Çalışanlar için daha kapsamlı sağlık sigortası sunulmalı. (More comprehensive health insurance should be offered to employees.)
13	Salaries and Benefits	500	Zamlar çok yetersizdi, piyasanın altında kalmaya başladık. (The rises were insufficient; we're starting to fall behind the market.)	Çalışanlar için emeklilik fonları gibi yan haklar artırılmalıdır. (Benefits like retirement funds should be increased for employees.)
14	Efficient Work	500	Debug işini analistler de yapabilmeli. (Analysts should also be able to handle debugging tasks.)	Verimli çalışabilmek için iş yükü dengeli bir şekilde dağıtılmalıdır. (To work efficiently, workloads should be distributed evenly.)

4.2. Data Pre-processing

Similar preprocessing steps were applied for processing text data in TCN, CNN, and LSTM models. The texts were converted into numerical sequences using a tokenizer and padded to a fixed length of 17 using pad sequences. This ensured consistency by allowing the models to process input data of uniform length. Additionally, the categorical labels of the texts were encoded into numerical values using Label Encoder. These common preprocessing steps enabled the models to classify text data accurately. In contrast, the data preparation process for the BERT model differed from the other models. The texts were tokenized using Bert Tokenizer, padded to a fixed length 17, and subjected to truncation. Furthermore, an attention mask was created for each token. These steps facilitated BERT's ability to understand the text more accurately and effectively.

4.3. Model Training and Parameter Tuning

During the training process for TCN, CNN, LSTM, and BERT models, a stratified k-fold cross-validation method was applied, with the data validated across five folds. This approach enhanced the models' generalization capabilities and contributed to obtaining more consistent results. Additionally, the TCN, CNN, and LSTM models used categorical cross-entropy as the loss function, while the Adam optimizer was employed for optimization across all models.

Table 2. Parameter Values used in Models

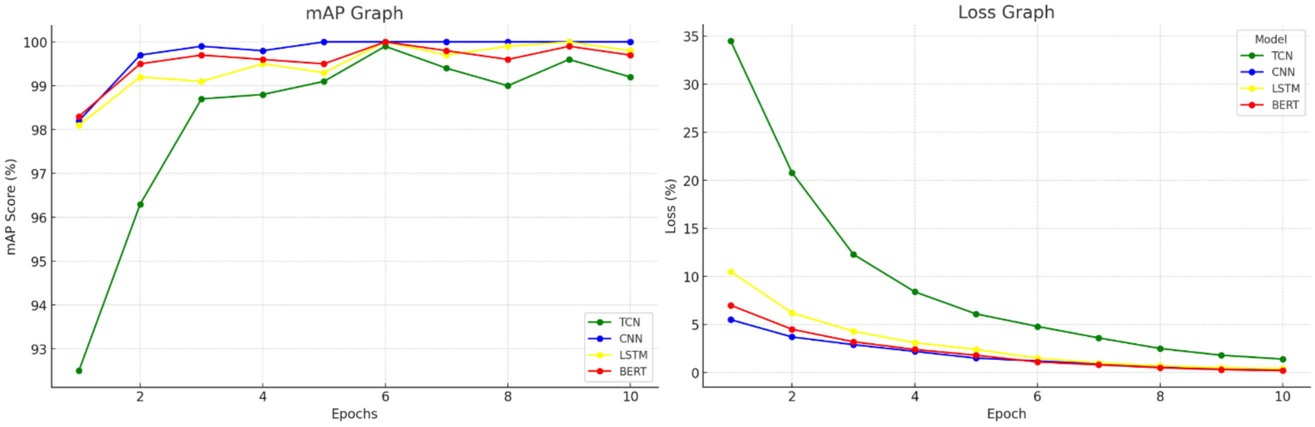
Parameter	TCN	CNN	LSTM	BERT
Embedding Dimension	128	128	128	128
Conv1D Filters (1st Layer)	64	128	-	-
Conv1D Filters (2nd Layer)	128	-	-	-
Kernel Size	3	3	-	-
Dropout Rate (1st Layer)	0.2	0.2	0.3	-
Dropout Rate (2nd Layer)	0.3	0.3	0.3	0.3
Dense Layer Units	128	128	128	-
Padding Sequences Max	17	17	17	17
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001
Loss Function	categorical_ crossentropy	categorical_ crossentropy	categorical_ crossentropy	CrossEntropyLoss
Batch Size	32	32	32	32
Epochs	10	10	10	10
Validation Splits	Stratified	Stratified	Stratified	-
Cross Validation Folds	5	5	5	5
Max Pooling Size	-	2	-	-
LSTM Units	-	-	128	-
Tokenizer	-	-		BertTokenizer (bert-base-uncased)
Pretrained Model	-	-		bert-base-uncased
Dropout Rate	-	-	-	0.3

The TCN model offered an architecture optimized for multi-class classification problems and was trained using the categorical cross-entropy loss function. Similarly, the CNN model followed a comparable training process but focused on extracting spatial features. In contrast, the LSTM model employed a specialized training process tailored for time-series data and sequential information. These three models generally shared similar loss functions and optimization methods during training.

Unlike the other models, BERT is a pre-trained language model, and therefore, smaller learning rates were used during its training. The cross-entropy loss function was employed for BERT, which is designed to optimize contextual language understanding. The operations performed on the models are outlined in Figure 1. All models were developed using Google Colab platform. The parameter configurations used in the models, determined based on the dataset size, are presented in Table 2.

4.4. Performance Evaluation

The performance of the models was evaluated using various metrics such as accuracy, precision, recall, and F1 score. Accuracy represents the proportion of correctly classified examples to the total number of examples. Precision and recall, respectively, indicate how accurately the model predicts a specific class and how effectively it identifies the actual instances of that class. The F1 score, as a balanced measure of precision and recall, comprehensively evaluates the model's classification performance. Test loss measures the model's performance on test data, where a lower loss indicates better generalizability. During the training of all models, 5-fold cross-validation was applied, and the values presented in the tables and matrices were calculated as averages across these folds. Additionally, during the model training process, the average loss and accuracy values for each epoch were calculated and presented as graphs in Figure 2. After training all models, the resulting performance metrics are presented in Table 3.



a) b)

Figure 2. Performance Graphics of the Models (a) mAP (b) Loss

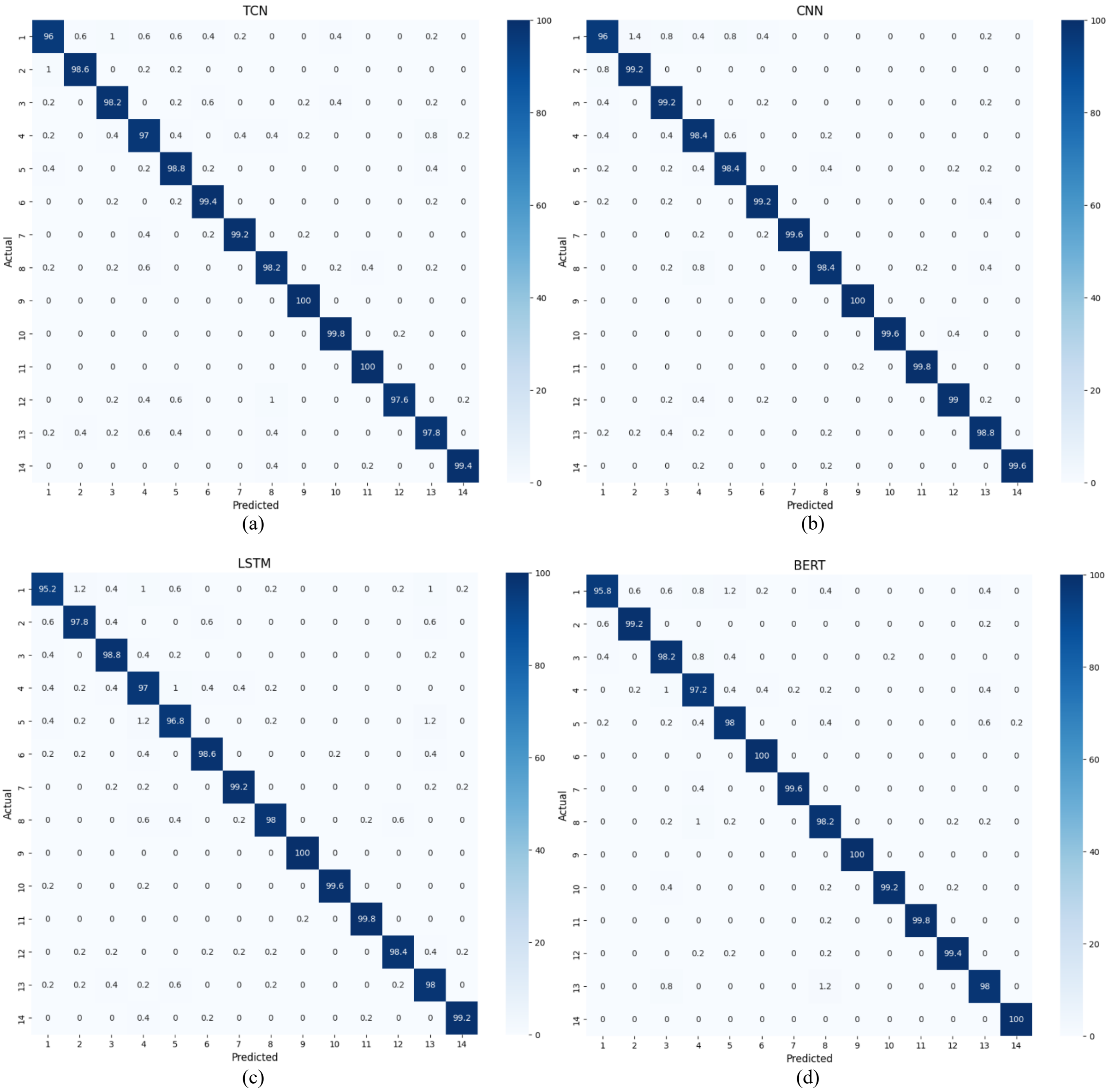
Table 3. Performance Values of the Models

Performance Metrics	TCN	CNN	LSTM	BERT
Test Loss (%)	33.79	6.09	10.55	6.35
Test Accuracy (%)	93.12	99.12	98.06	98.64
Precision (%)	90.78	99.12	98.11	98.69
Recall (%)	90.79	99.12	98.06	98.64
F1 Score (%)	90.68	99.11	98.06	98.64
Average Training Duration (sn)	100.63	185.16	646.39	418.59

According to the results in Table 3, the CNN model achieved the lowest test loss value of 6.09%, indicating a high generalization capability. BERT demonstrated a similar performance with a test loss of 6.35%. In contrast, the TCN model lagged with a test loss of 33.79%. Regarding accuracy, the CNN model attained the highest value at 99.12%, followed by BERT (98.64%) and LSTM (98.06%), showcasing strong performance. TCN ranked the lowest with an accuracy of 93.12%. For precision and recall metrics, CNN achieved the highest values, at 99.12%, followed by BERT and LSTM. When considering the F1 score, a balanced measure of precision and recall, CNN again led with 99.11%, with BERT (98.64%) and LSTM (98.06%) closely trailing. Regarding training time, TCN was the fastest, completing training in only 100.63 seconds. Although CNN required a longer training time of 185.16 seconds, it compensated for this with its superior accuracy. On the other hand, LSTM and BERT required significantly more resources, with training times of 646.39 seconds and 418.59 seconds, respectively. The average confusion matrices for the 5-fold cross-validation results for all four models are shown separately in Figure 3.

5. Conclusions

In this study, employee feedback data was classified using four different deep-learning models, and the models' performances and training times were compared. The CNN model emerged as the most successful, achieving the best results across all performance metrics, including test loss, accuracy, precision, recall, and F1 score. With its high accuracy, CNN proved an effective option for classification tasks. The BERT model, known for its ability to learn contextual language representations, demonstrated performance close to that of CNN. However, its longer training times made it more computationally expensive. The TCN model stood out with its fast-training time but fell behind the other models in performance metrics. While TCN offers advantages for time-series analysis, it did not deliver strong performance in the classification task of this study. Despite its ability to process sequential data, the LSTM model lagged behind CNN due to its longer training time and lower accuracy.



(a) (b)

(c) (d)

Figure 3. Confusion Matrices (a) TCN, (b) CNN, (c) LSTM, (d) BERT

References

[1] M. Kayakuş and F. Y. Açıkgöz, “Classification of news texts by categories using machine learning methods,” *Alphanumeric Journal*, vol. 10, no. 2, pp. 155–166, 2022, doi: 10.17093/alphanumeric.1149753.

[2] A. H. Bozkurt and N. Yalçın, “Topluluk öğrenmesi algoritmaları kullanarak Amazon yemek yorumları üzerine duygu analizi,” *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, vol. 11, no. 1, pp. 128–139, 2024, doi: 10.35193/bseufbd.1300732.

[3] M. F. Tuna, M. Polatgil, and O. Kaynar, “Restoran müşterilerinin geri bildirimleri üzerinde hedef kategorinin tespiti ve hedef tabanlı duygu analizi,” *Süleyman Demirel Üniversitesi Vizyoner Dergisi*, vol. 14, no. 40, pp. 1205–1221, 2023, doi: 10.21076/vizyoner.1208355.

[4] B. C. Öğe and F. Kayaalp, “Farklı sınıflandırma algoritmaları ve metin temsil yöntemlerinin duygu analizinde performans karşılaştırılması,” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 9, no. 6, pp. 406–416, 2021, doi: 10.29130/dubited.1015320.

[5] İ. A. Metin and B. Karasulu, “İnsanın günlük aktivitelerinin yeni bir veri kümesi: Derin öğrenme tekniklerini kullanarak sınıflandırma performansı için kıyaslama sonuçları,” *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, vol. 36, no. 2, pp. 759–778, 2021, doi: 10.17341/gazimmfd.772849.

[6] E. Aydemir, M. Işık, and T. Tuncer, "Türkçe haber metinlerinin çok terimli Naive Bayes algoritması kullanılarak sınıflandırılması," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 2, pp. 519–526, 2021, doi: 10.35234/fumbd.871986.

[7] M. M. Akgümüş and A. Boyacı, "Bankacılık sektörü için topluluk öğrenimini kullanan iki aşamalı bir müşteri şikayet yönetimi," *TBV Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 16, no. 1, pp. 45–52, 2023, doi: 10.54525/tbbmd.1163852.

[8] S. Ertem and E. Özbay, "Detection of COVID-19 anti-vaccination from Twitter data using deep learning and feature selection approaches," *Firat University Journal of Experimental and Computational Engineering*, vol. 3, no. 2, pp. 116–133, 2024, doi: 10.62520/fujece.1443753.

[9] M. Demirbilek and S. Ö. Demirbilek, "Google yorumları üzerinden makine öğrenme yöntemleri ve Amazon Comprehend ile duygu analizi: İç Anadolu'da bir üniversite örneği," *Üniversite Araştırmaları Dergisi*, vol. 6, no. 4, pp. 452–461, 2023, doi: 10.32329/uad.1383794.

[10] M. Çataltaş, B. Üstünel, and N. A. Baykan, "Sentiment classification on Turkish tweets about COVID-19 using LSTM network," *Konya Mühendislik Bilimleri Dergisi*, vol. 11, no. 2, pp. 341–353, 2023, doi: 10.36306/konjes.1173939.

[11] G. Alparslan and M. Dursun, "Konvolüsyonel sinir ağları tabanlı Türkçe metin sınıflandırma," *Bilişim Teknolojileri Dergisi*, vol. 16, no. 1, pp. 21–31, 2023, doi: 10.17671/gazibtd.1165291.

[12] M. Yılmaz and E. S. Günal, "Derin öğrenme temelli otomatik yardım masası sistemi," *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, vol. 30, no. 3, pp. 318–327, 2022, doi: 10.31796/ogummf.1038486.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint*, 2019, Available: https://arxiv.org/abs/1810.04805.

[14] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," *Lecture Notes in Computer Science*, Springer International Publishing, 2016, doi: 10.1007/978-3-319-49409-8_7.

[15] B. Ghojogh and A. Ghodsi, "Recurrent neural networks and long short-term memory networks: Tutorial and survey," *arXiv preprint*, 2023, Available: https://arxiv.org/abs/2304.11461.

[16] İ. Budak and A. Organ, "Veri ve metin madenciliği ile hava yolu işletmelerinin COVID-19 öncesi ve sonrası sosyal medya yorum ve skorlarının değerlendirilmesi," *Ömer Halisdemir Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 15, no. 4, pp. 998–1022, 2022, doi: 10.25287/ohuiibf.1149801.

[17] İ. Sel and D. Hanbay, "Ön eğitimli dil modelleri kullanarak Türkçe tweetlerden cinsiyet tespiti," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 2, pp. 675–684, 2021, doi: 10.35234/fumbd.929133.

[18] C. Aci and A. Çırak, "Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması," *Bilişim Teknolojileri Dergisi*, vol. 12, no. 3, pp. 219–228, 2019, doi: 10.17671/gazibtd.457917.

[19] O. T. Bişkin, "Multi-step forecasting of COVID-19 cases in European countries using temporal convolutional networks," *Mugla Journal of Science and Technology*, vol. 7, no. 1, pp. 117–126, 2021, doi: 10.22531/muglajsci.875414.

[20] A. Kasapbaşı and H. Canbolat, "İşitme engelli bireylerin hareketlerini sınıflandırmaya yönelik yapay zeka modelinin geliştirilmesi," *Black Sea Journal of Engineering and Science*, vol. 7, no. 5, pp. 826–835, 2024, doi: 10.34248/bsengineering.1477046.

[21] B. Erol and T. İnkaya, "Satış tahmini için derin öğrenme yöntemlerinin karşılaştırılması," *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, vol. 29, no. 2, pp. 535–554, 2024, doi: 10.17482/uumfd.1382971.

[22] M. F. Tuna and Y. Görmez, "Evrişimsel sinir ağları tabanlı derin öğrenme yöntemiyle müşteri şikayetlerinin sınıflandırılması," *Bingöl Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 8, no. 1, pp. 31–46, 2024, doi: 10.33399/biibfad.1362160.

[23] Ö. Aydın and H. Kantarcı, "Türkçe anahtar sözcük çıkarımında LSTM ve BERT tabanlı modellerin karşılaştırılması," *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 17, no. 1, pp. 9–18, 2024, doi: 10.54525/bbmd.1454220.

[24] S. Arslan and E. Fırat, "Stance detection on short Turkish text: A case study of Russia-Ukraine war," *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, vol. 24, no. 3, pp. 602–619, 2024, doi: 10.35414/akufemubid.1377465.

[25] Y. E. Gür, "Comparative analysis of deep learning models for silver price prediction: CNN, LSTM, GRU and hybrid approach," *Akdeniz İİBF Dergisi*, vol. 24, no. 1, pp. 1–13, 2024, doi: 10.25294/auiibfd.1404173.

[26] S. Y. Kahraman, A. Durmuşoğlu, and T. Dereli, "Ön eğitimli BERT modeli ile patent sınıflandırılması," *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, vol. 39, no. 4, pp. 2484–2496, 2024, doi: 10.17341/gazimmfd.1292543.

[27] E. Ülker and Ö. İnik, "Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri," *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 2017, Available: https://dergipark.org.tr/tr/pub/gbad/issue/31228/330663.

[28] Ö. Aydın and H. Kantarcı, "Türkçe anahtar sözcük çıkarımında LSTM ve BERT tabanlı modellerin karşılaştırılması," *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 17, no. 1, pp. 9–18, 2024, doi: 10.54525/bbmd.1454220.

[29] İ. Sel and D. Hanbay, "Ön eğitimli dil modelleri kullanarak Türkçe tweetlerden cinsiyet tespiti," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 2, pp. 675–684, 2021, doi: 10.35234/fumbd.929133.

[30] B. Ghojogh and A. Ghodsi, "Recurrent neural networks and long short-term memory networks: Tutorial and survey," *arXiv preprint*, 2023, Available: https://arxiv.org/abs/2304.11461.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 47-57, 2025
DOI: 10.35377/saucis...1517723

RESEARCH ARTICLE

Joint Detection and Removal of Specular Highlights using Vision Transformer with Multi-scale Patch Attention

Levent Karacan[1]

[1] Gaziantep University, Department of Computer Engineering, Gaziantep, Türkiye, ror.org/020vvc407

Corresponding author:
Levent Karacan, Department of Computer Engineering, Gaziantep University, Gaziantep, Türkiye
lkaracan@gantep.edu.tr

Article History:
Received: 17.07.2024
Revised: 19.12.2024
Accepted: 22.02.2025
Published Online: 27.03.2025

ABSTRACT
Specular highlights play a pivotal role in comprehending scenes within developed visual environment. Nevertheless, their presence can adversely affect the efficacy of solutions in various computer vision tasks. Current methodologies typically use Convolutional Neural Network (CNN)-based Unet architectures for specular highlight detection. However, CNNs exhibit limitations in capturing global contextual information, despite excelling in local context analysis. To utilize global context information, it is proposed a novel network architecture leveraging Vision Transformers (ViTs) to jointly detect and remove specular highlights for a given image. Developed model incorporates a multi-scale patch-based self-attention mechanism to effectively capture global context, alongside a CNN-based feed-forward network for local contextual cues. Experimental results with both quantitative and qualitative evaluations demonstrate that the proposed approach achieves state-of-the-art performance.

Keywords: Specular highlight detection, Specular highlight removal, Vision transformers, Convolutional neural networks

1. Introduction

Specular highlights are visual phenomena that appear on smooth and shiny surfaces. They are essential in helping the human visual system to interpret the environment by conveying information about light sources and surface materials. However, these highlights pose significant challenges in computer vision tasks such as image segmentation, text detection, object recognition, and scene understanding. They appear as bright and intense regions in images, as shown in Figure 1, and can degrade the performance of computer vision algorithms. Researchers have proposed various methods to detect and remove specular highlights to mitigate their impact on these tasks. Additionally, detecting these regions can be useful for light source detection and intrinsic image decomposition [1].

Early methods for detecting specular highlights [2] - [8] relied on the assumption that specular highlight regions contain the brightest pixels. These methods defined a threshold process to identify specular highlights. However, the bright pixel assumption does not hold for complex cases. In the context of removing specular highlights, traditional methods [9], [10], [11] - [14] predominantly depend on color values derived from the Dichromatic Reflection Model [2]. While previous highlight removal techniques have demonstrated success, they are constrained in their ability to handle large-scale removal tasks due to their reliance on prior information regarding material, color, or lighting conditions.

Learning-based highlight detection and removal methods [15] - [20] leverage the convolutional neural networks (CNNs) to train a highlight detection or removal model on carefully curated datasets. Recent research [18], [21] has shown that joint highlight detection and removal models produce more effective results than single detection or removal models. The CNN-based methods achieved effective results by leveraging the convolution operation to model local hierarchical image context. However, they are not well suited to model global context, which is important to model scene illumination.

Wu et al. [21] defined a ViT-based network to remove specular highlights, incorporating global context information into joint detection and removal models. However, they use a CNN-based network to detect specular highlights in this joint model, similar to previous methods.

Cite as: L. Karacan, “Joint detection and removal of specular highlights using vision transformer with multi-scale patch attention,” *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 47-57, 2025, doi: 10.35377/saucis...1517723

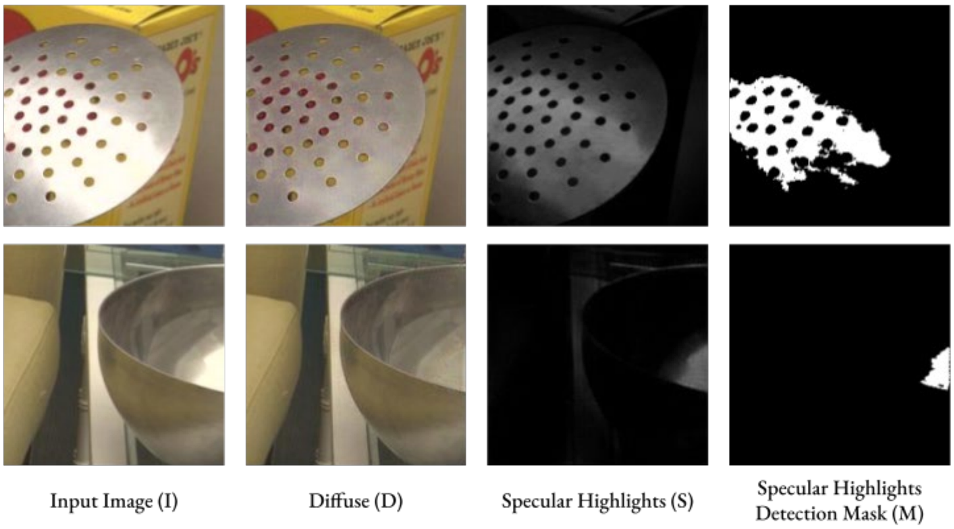


Figure 1. Specular Highlights Detection and Removal. The Proposed Model Detects and Remove Specular Highlights as Shown

The primary purpose of this study is to address the challenges posed by specular highlights in images, which can significantly degrade the performance of computer vision systems by affecting tasks like object detection, segmentation, and recognition. To tackle this issue, a novel network architecture is proposed that jointly detects and removes specular highlights in a given color image. At the core of the proposed network is a modified vision transformer, which employs a multi-scale patch-based self-attention mechanism to reduce scale dependency. The developed approach ensures a unified and efficient framework using a shared transformer-based backbone for detection and removal processes. Quantitative and qualitative results demonstrate that the proposed model achieves state-of-the-art performance on a standard dataset for both tasks. Additionally, an ablation study is conducted to analyze the multi-scale patch-based attention mechanism's effectiveness, further validating the developed method's robustness.

The manuscript is constructed as follows: Section 2 reviews the literature on specular highlight detection and removal. Section 3 provides a detailed description of the proposed method, encompassing its architectural framework, training protocol, and experimental configuration. Section 4 presents and critically examines the experimental results. Finally, Section 5 discusses the findings and potential future directions for the conclusion.

2. Related Works

Table 1 provides a detailed overview of significant studies on specular highlight detection and removal. It highlights the methods, key contributions, and tasks each work addresses. The initial approaches to highlight detection were largely based on color constancy models [2] - [4], which employed thresholding techniques to identify specular pixels. Incorporating the dark channel prior into their optimization scheme, Kim et al. [10] employed a different approach to that of Liu et al. [12], who estimated specular highlight reflection ratio and tuned saturation of the input image to remove highlights. Park et al. [5] proposed using two images in a least-squares regression scheme. These images are captured under distinct illumination conditions. One image was captured with specularities to be detected, while the other served as a reference, largely devoid of specularities, to generate a threshold map for image pixels. Building on the dichromatic reflection model [2], Meslouhi et al. [6] applied a specularity condition in the CIE-XYZ color space. Despite their successes, these methods were constrained by their reliance on specific assumptions and inability to cope with complex illumination scenarios or images with textured colors.

Learning-based approaches for specular highlight detection and removal offer more generalized solutions but necessitate diverse and extensive datasets. Fu et al. [17] introduced a real-world highlight dataset with annotated ground-truth masks, covering various material categories by providing different highlight shapes and appearances. They also trained a convolutional neural network that is used on this dataset for specular highlight detection. However, since this dataset lacks ground-truth diffuse images, it is unsuitable for training models aimed at removing specular highlights.

Table 1. Summary of Related Works on Specular Highlight Detection and Removal. The Table Lists the Reference, Authors, Title, Approach, Key Contributions, Publication, and Task Categories, Including Specular Highlight Detection, Removal, or Both Detection and Removal

Ref.	Authors	Title	Task	Approach	Key Contributions	Publication
[29]	Shen et al.	Chromaticity-based separation of reflection components in a single image	Specular Highlight Detection	Chromaticity-based method	Separates reflection components in single images using chromaticity information.	Pattern Recognit., 2008
[14]	Yamamoto et al.	Efficient improvement method for separation of reflection components based on an energy function	Specular Highlight Detection	Energy function-based optimization	A method to separate reflection components efficiently using an energy function.	IEEE ICIP, 2017
[8]	Zhang et al.	Improving shadow suppression for illumination robust face recognition	Specular Highlight Removing	Chromaticity-based method	Enhances shadow suppression for better face recognition under varying illumination conditions.	IEEE Trans. Pattern Anal. Mach. Intell., 2018
[7]	Li et al.	Specular reflection removal for endoscopic image sequences with adaptive-RPCA decomposition	Specular Highlight Removing	Adaptive-RPCA decomposition	Removal of specular reflections in endoscopic image sequences using adaptive Robust Principal Component Analysis (RPCA).	IEEE Trans. Med. Imaging, 2019
[30]	Lin et al.	Deep multi-class adversarial specularity removal	Specular Highlight Removing	Convolutional Neural Networks	A deep learning model for multi-class specularity removal with adversarial training.	SCIA 2019, Springer
[16]	Muhammad et al.	Spec-Net and Spec-CGAN: Deep learning models for specularity removal from faces	Specular Highlight Removing	Convolutional Neural Networks Generative Adversarial Networks	Proposes two deep learning models for removing specularity from facial images.	Image Vis. Comput., 2020
[17]	Fu et al.	Learning to Detect Specular Highlights from Real-world Images	Specular Highlight Detection	Convolutional Neural Networks	Highlights detection in real-world images using a deep learning approach.	ACM Multimedia, 2020
[18]	Fu et al.	A multi-task network for joint specular highlight detection and removal	Joint Specular Highlight Detection and Removing	Convolutional Neural Networks	Joint detection and removal of specular highlights using a multi-task network.	IEEE/CVF CVPR, 2021
[21]	Wu et al.	Joint specular highlight detection and removal in single images via Unet-Transformer	Joint Specular Highlight Detection and Removing	Vision Transformers	Proposes a joint detection and removal approach using a hybrid Unet-Transformer model.	Comput. Vis. Media, 2023

Shi et al. [23] proposed a CNN model comprised of encoder-decoder architectures to train on a large-scale object intrinsic database for the intrinsic image decomposition. Funke et al. [15] proposed a Generative Adversarial network (GAN)--based deep learning model capable of removing specular highlights from an endoscopic image. They trained this model on a small image patches dataset. The small patch images are extracted from the endoscopic video. Muhammad et al. [16]

proposed a facial specular highlight removal dataset and presented two alternative models, Spec-Net and Spec-CGAN. A real-world dataset with a pair of color images and ground-truth diffuse components was collected by Wu et al. [19] to train the specular highlights removal model. They proposed a GAN-based approach to remove specular highlights considering polarization theory. [20] introduced a specular highlight removal network comprising three stages in which the image is first decomposed into albedo, shading, and specular components. Subsequently, refinement and tone correction networks are employed to obtain decent specular-free images.

Recently, joint highlight detection and removal models [18] and [21] have been proposed using a multi-task network trained on a large dataset with corresponding diffuse and highlight components. Fu et al. [18] proposed the SHIQ dataset, a new large-scale specular highlight detection and removal dataset including ground-truth diffuse and specular highlights components of each image sample. They also introduced a multi-task CNN-based Unet architecture trained on the SHIQ dataset to detect and remove specular highlights jointly. Wu et al. [21] proposed a Swin Transformer-based [24] highlight removal model taking input image and specular highlight mask predicted by a CNN-based Unet detection model. They trained both removal and detection models jointly in an end-to-end manner. While Wu et al. demonstrate the efficacy of vision transformers for highlight removal tasks, it is asserted that a separate detection model is unnecessary. Hence, Fu et al. [18] used the same architecture backbone for detection and removal. Rather than relying solely on a CNN-based architecture like in Fu et al., a novel joint specular highlight detection and removal model that combines both CNN and ViT [25] architectures is proposed to leverage local and global context information. Moreover, while vanilla ViT employs the same patch scale across the different heads, the proposed MsPAT operates each scale in different heads.

3. Method

The Dichromatic Reflection Model (DFM) [2] defines a color image as the following composition of diffuse and specular highlight components:

$$I = D + S \tag{1}$$

The diffuse D, represents light uniformly scattered across an object's surface. In contrast, the specular highlight component S, accounts for the concentrated reflections of light from smooth surfaces. The model follows Fu et al. [18] to jointly detect and remove specular highlights for a given image. More clearly, it adopted the generalized version of DFM:

$$I = D + S \times M \tag{2}$$

where M denotes the pixel-wise binary mask indicating the location of the specular highlights. The point-wise multiplication of S and M provides to restrict the specular highlight removal in the masked regions.

Given a dataset of input RGB images I with ground-truth D, S, and M for each sample. Main goal is to train the proposed ViT-based model G to predict $\hat{D}$, $\hat{S}$, and $\hat{M}$ from a given single RGB image I:

$$[\hat{D}, \hat{S}, \hat{M}] = G(I) \tag{3}$$

where $\hat{D}$, $\hat{S}$, and $\hat{M}$ are predicted diffuse, specular highlights, and binary masks, respectively.

3.1. Network Architecture

The proposed model's overall structure is illustrated in Figure 2. The model includes an Encoder, multiple Multi-scale Patch Attention Transformers (MsPAT), and a Decoder. As shown in the Figure 2, an input RGB image $\boldsymbol{I} \in \boldsymbol{R}^{\boldsymbol{H} \times \boldsymbol{W} \times \boldsymbol{3}}$ with a height $\boldsymbol{H}$ and width $\boldsymbol{W}$is first passed through three convolutional blocks that reduce its spatial dimensions by half twice. The first convolutional block comprises a $\mathbf{7} \times \mathbf{7}$ convolution layer, batch normalization (BN), and ReLU activation layer, respectively. The next two convolutional blocks consist of $\mathbf{3} \times \mathbf{3}$ convolutions with stride $\mathbf{2}$, BN, and ReLU activation layers. Consequently, 256-dimensional feature representation $\boldsymbol{f} \in \boldsymbol{R}^{\boldsymbol{H}/\boldsymbol{4} \times \boldsymbol{W}/\boldsymbol{4} \times \boldsymbol{256}}$ is obtained, where the spatial dimensions are a quarter of the input image's dimensions. Subsequently, the feature representation is passed on to residually connected to multiple MsPAT modules. The transformed features $\boldsymbol{f}$ processed by the MsPAT modules are forwarded to the decoder network. The decoder network comprises two blocks, each containing two $\mathbf{4} \times \mathbf{4}$ transposed convolution layers with a stride of 2, followed by batch normalization (BN) and ReLU activation layers. These operations double the input feature maps' spatial resolution, restoring them to their original dimensions. Subsequently, the features are directed to three distinct prediction heads, each composed of $\mathbf{7} \times \mathbf{7}$convolutional layers. As shown in Figure 2, the predicted specular highlight mask highlight mask $\hat{\boldsymbol{M}}$ is concatenated with the feature maps before entering the specular highlight prediction head. Similarly, the predicted specular highlight mask $\hat{\boldsymbol{M}}$ and the specular highlight $\hat{\boldsymbol{S}}$ are concatenated with the feature maps before passing the residual prediction head.

3.2. Multi-scale Patch Attention Transformer (MsPAT)

The proposed Multi-scale Patch Attention Transformer (MsPAT) is illustrated in Figure 3. The MsPAT employs three basic steps: Embedding, Matching, and Attending.

3.2.1. Embedding

The feature maps $f \in R^{H/4 \times W/4 \times 256}$ gathered from the encoder or previous transformer are first embedded into query Q, key K, and value V features $Q, K, V \in R^{H/4 \times W/4 \times 256}$ using 1×1 convolution with a stride of 1. These features are then decomposed into multi-scale non-overlapping patches in parallel. As a result, it is obtained c-dimensional $(c = 256/s)$ patch embeddings at different scales s, $(q_i^s, k_i^s, v_i^s) \in R^{h^s \times w^s \times c}$, where i denotes the index of patches of height h^s and width w^s in each scale. In contrast to the vanilla Vision Transformer (ViT) [24], where patches are embedded into one-dimensional tokens, it is maintained the patch embeddings as two-dimensional. Additionally, while the vanilla ViT processes the same scale across different heads, the proposed MsPAT assigns each head to operate on different scales, enhancing its multi-scale processing capability.

3.2.2. Matching

In the matching step, the patch embeddings are initially flattened to the one-dimensional vector, and then patch similarities between query and key patches are calculated by dot product as follows:

$$S_{i,j}^s = \frac{q_i^s . k_j^s}{\sqrt{h^s \times w^s \times c}} \quad (4)$$

where i and j are indices within the N^s patches at each scale $(1 \le i, j \le N^s)$, h^s and w^s denote the height and width of the patch at scale s, and c denotes the dimension of the feature embedding. Based on this similarity, the weights in the attention map are calculated using the expression given below:

$$A_{i,j}^s = \frac{exp\left(S_{i,j}^s\right)}{\sum exp\left(S_{i,n}^s\right)} \quad (5)$$

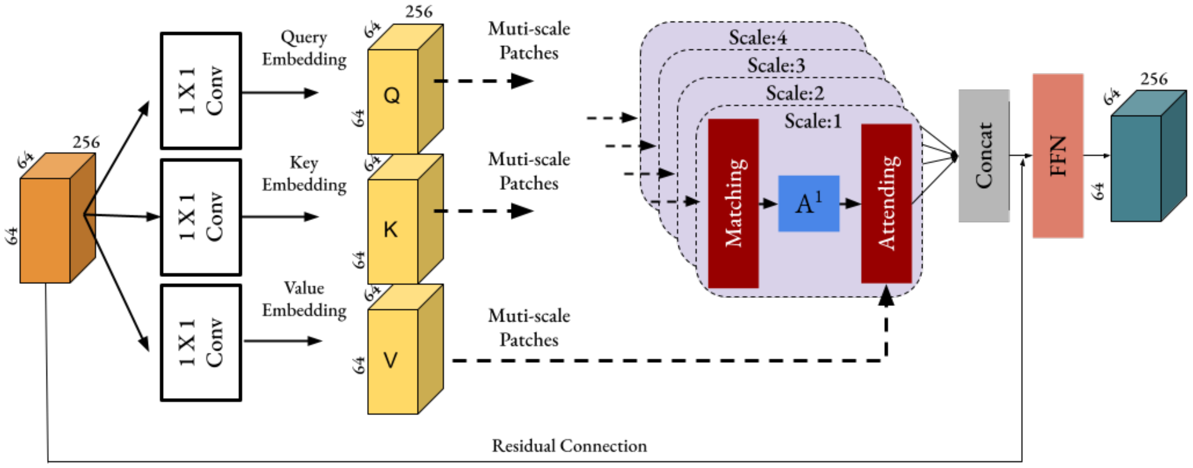


Figure 2. Proposed Multi-scale Patch Attention Transformer Module. This Module Applies Patch-Based Attention to Provide Global Connectivity. It Effectively Captures Global Dependencies by Incorporating Attention Mechanisms at Multiple Patch Scales

3.2.3. Attending

After the attention map $A_{i,j}^s$ is calculated for each scale s i.e. head, the output feature is obtained as the weighted summation of the value patches using the $A_{i,j}^s$:

$$f_i^s = \sum A_{i,j}^s \, v_j^s \quad (6)$$

The output feature patches f_i^s are reshaped to 2D and then recomposed to the input feature dimensions. Lastly, the output features $f^s \in R^{H/4 \times W/4 \times c}$ from each head are concatenated along the feature dimension:

$$f = [f^1, f^2, \dots, f^s] \quad (7)$$

Lastly, the transformed feature $f \in R^{H/4 \times W/4 \times 256}$ is given to a feed-forward network and then passed to the subsequent transformer block or decoder. Note that there is a residual connection from the transformer block's input before passing to the feed-forward network.

As for the feed-forward network (FFN), it adopted the LocalViT [26], [27] that consists of convolutional layers of ReLU activation functions and squeeze-excitation (SE) module [28] to enrich the local context information in the transformer block.

3.3. Loss Function

To jointly detect and remove specular highlights, the proposed model is trained using a hybrid loss function comprising binary cross-entropy (BCE) loss, dice loss, and mean squared error (MSE) loss:

$$L = L_{BCE}(M, \hat{M}) + L_{Dice}(M, \hat{M}) + L_{MSE}^{S}(S, \hat{S}) + L_{MSE}^{D}(D, \hat{D}) \quad (8)$$

The BCE loss $\boldsymbol{L_{BCE}}$ and dice loss $\boldsymbol{L_{Dice}}$ is employed for the specular highlight detection as a binary mask segmentation. The BCE loss is commonly used for segmentation and mask prediction for specular highlight detection. The loss function between the predicted value $\boldsymbol{\hat{M}_p}$ and the actual value $\boldsymbol{M_p}$ for each pixel $\boldsymbol{p}$ in the image is the sum of the binary cross-entropies for all pixels:

$$L_{BCE} = -\sum \left[M_p log(\hat{M}_p) + (1 - M_p) log(1 - \hat{M}_p) \right] \quad (9)$$

The regions of specular highlight are relatively small compared to the non-specular regions. To alleviate the effect of this imbalance, it is employed Dice loss [29], which measures the overlap between the predicted binary mask and the ground truth mask for each pixel $\boldsymbol{p}$:

$$L_{Dice} = 1 - \frac{1 + 2 \times \sum M_p \hat{M}_p}{1 + \sum(M_p + \hat{M}_p)} \quad (10)$$

To provide the model to remove specular highlight, it is defined the mean squared error (MSE) loss on the ground truth $\boldsymbol{S}$ and predicted $\boldsymbol{\hat{S}}$ specular highlights:

$$L_{MSE}^{S} = \frac{1}{N} \sum (S_p - \hat{S}_p)^2 \quad (11)$$

Similarly, it is employed mean squared error loss for the ground truth diffuse image $\boldsymbol{D}$ and the predicted diffuse image $\boldsymbol{\hat{D}}$ to predict the diffuse image as follows:

$$L_{MSE}^{D} = \frac{1}{N} \sum (D_p - \hat{D}_p)^2 \quad (12)$$

4. Experiments

4.1. Dataset and Implementation Details

It is trained the proposed model on the SHIQ dataset [18], which includes input images and their corresponding ground-truth binary masks M, specular highlight components S, and diffuse images D. The SHIQ dataset consists of 9825 training and 1000 test samples, with 200×200 image dimensions. It contains challenging examples of highly reflective objects like metal, plastic, and glass.

It is implemented with the proposed model network using PyTorch. The training was conducted over 60 epochs by using the Adam optimizer with a learning rate of 2×10^{-5}, and parameter settings of $\beta_1 = 0.5$ and $\beta_2 = 0.999$. During training, random horizontal flip data augmentation was applied.

4.2. Evaluation Metrics

It is used accuracy (Acc) and Balanced Error Rate (BER) metrics to evaluate the specular highlight detection results by following the previous works. Acc and BER can be defined as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

$$BER = \frac{1}{2}\left(\left(\frac{FP}{TN + FP}\right) + \left(\frac{FN}{TN + TP}\right)\right) \tag{14}$$

where TP, FP, TN, and FN refer to pixel-wise true positives, false positives, true negatives, and false negatives, respectively. Better detection results are indicated by a higher accuracy value and a lower BER value. Three commonly used metrics are utilized to evaluate the specular highlight removal performance: mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM). Higher values of SSIM and PSNR, as well as lower MSE, imply improved performance.

4.2. Quantitative Results

It quantitatively compared specular highlight detection results with traditional methods NMF [7] and ATA [8], as well as deep learning-based methods SHDN [17], JSHDR [18], and Unet-Trans [21], using accuracy (Acc) and balanced error rate (BER) metrics commonly used for this task. Table 2 presents the average accuracy (Acc) and balanced error rate (BER) metric results obtained on the SHIQ test set. According to the results, the developed method produces the best results in terms of both accuracy and balanced error rate.

Table 2. Quantitative Specular Highlight Detection Results with Comparisons on the SHIQ Dataset. ↑ Denotes Better Performance with Higher Values, while ↓ Indicates Better Performance with Lower Values

Method	**Accuracy ↑**	**BER ↓**
NMF [7]	0,700	18,80
ATA [8]	0,710	24,40
SHDN [17]	0,910	6,180
JSHDR [18]	0,930	5,920
Unet-Trans [21]	0,970	5,920
Developed Model	**0,980**	**5,260**

Table 3. Quantitative Specular Highlight Removal Results with Comparisons on the SHIQ Dataset. ↑ Denotes Better Performance with Higher Values, while ↓ Indicates Better Performance with Lower Values

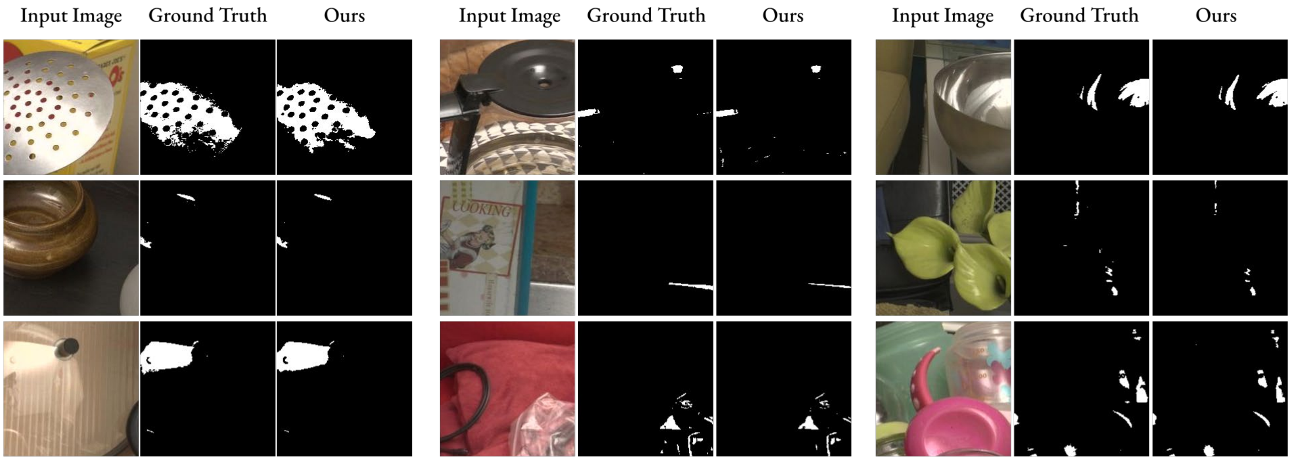
Method	**MSE ↓**	**SSIM ↑**	**PSNR ↑**
Shen et al. [29]	5,44	0,4596	19,2
Yamamoto et al. [14]	12,86	0,2945	9,15
Multi-class GAN [30]	0,3375	0,9103	24,49
Spec-CGAN [16]	0,6575	0,9154	23,53
JSHDR [18]	0,2525	0,9614	28,19
Trans-Unet [21]	0,1425	**0,9669**	29,85
Developed Model	**0,1353**	0,9523	**30,82**

For specular highlight removal, it is compared results with both traditional Shen et al. [29] and Yamamoto et al. [14] and state-of-the-art learning-based methods Multi-class GAN [30], Spec-CGAN [16], JSHDR [18], and Trans-Unet [21] as shown in Table 2. The developed approach outperforms the others in mean squared error (MSE) and peak signal-to-noise ratio (PSNR) while delivering a highly competitive structural similarity index (SSIM) score in comparison to the latest methods.

4.3 Visual Results

Figure 4 presents the proposed method's visual specular highlight detection results and the corresponding ground truth. As shown, the proposed method provides highly accurate detection performance. Additionally, Figure 5 compares detection performance with the recent Unet-Trans method. It is worth noting that Unet-Trans employs a Unet architecture for specular highlight detection while utilizing a transformer architecture for specular highlight removal. In contrast, the method employs the proposed transformer-based approach for joint specular highlight detection and removal. As demonstrated in Figure 5, detection masks are more accurate than those produced by Unet-Trans.

Visual results of specular highlight removal are presented in Figure 6, alongside the corresponding input images and ground truth specular highlight-free images. The developed method demonstrates a high capability in accurately removing specular highlights and closely approximating the ground truth images. This highlights the robustness and effectiveness of the approach in preserving underlying image details while eliminating unwanted specular highlights. The method of comparison with the Unet-Trans approach is shown in Figure 7 to evaluate it further. The comparison reveals that it surpassed Unet-Trans in terms of visual quality or achieved competitive performance. The accuracy and consistency of the removal process are evident, showcasing cleaner and more natural-looking results. This can be attributed to innovative transformer-based architecture for detection and removal tasks, providing a cohesive and powerful framework for handling specular highlights.



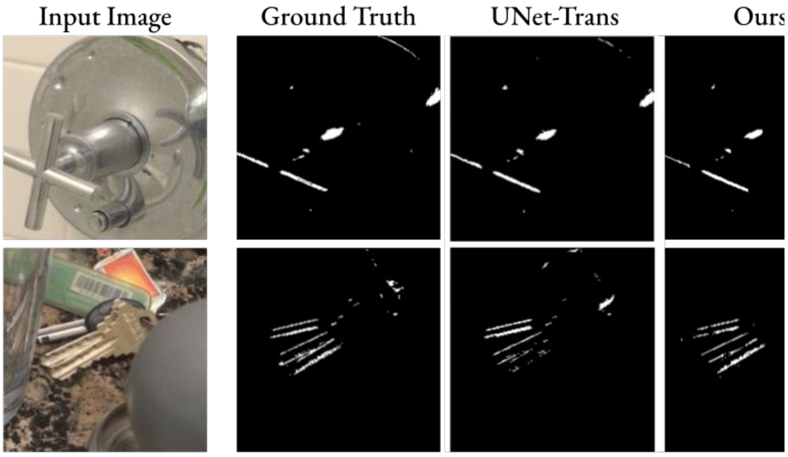


Figure 3. Visual Comparison of Specular Highlight Detection Results with A Recent Unet-Based Method Unet-Trans

4.4. Patch Scale Analysis on Attention

Table 4 analyzes the impact of the multi-scale approach on patch-based self-attention, where each head processes different scales. It compares multi-scale attention with various single-scale patch-based multi-head self-attention methods, where each head processes the same scale. The observations indicate that different scales contribute to different metrics. Overall, multi-scale patch attention provides a balanced compromise across metrics. A trade-off between detection metrics, Accuracy (Acc), and Balanced Error Rate (BER) is also found. While the smallest scale provides the best BER score, it deteriorates the Accuracy. In this regard, the multi-scale approach offers a compromise, balancing these metrics. For specular highlight removal, the multi-scale approach achieves the best scores for MSE and PSNR, slightly trailing behind the largest single-scale attention for SSIM.

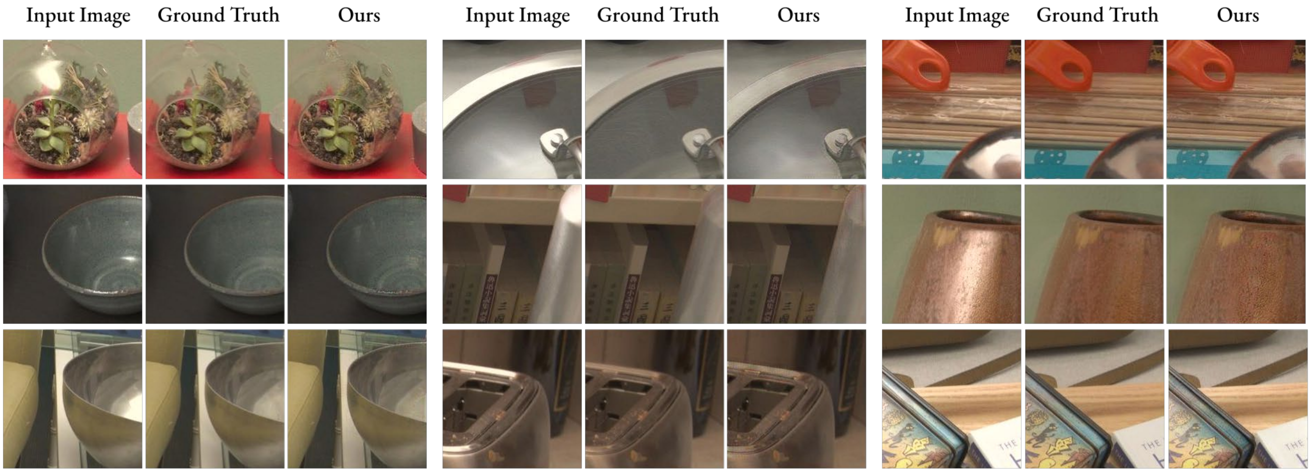


Figure 6. Visual Specular Highlight Removal Results

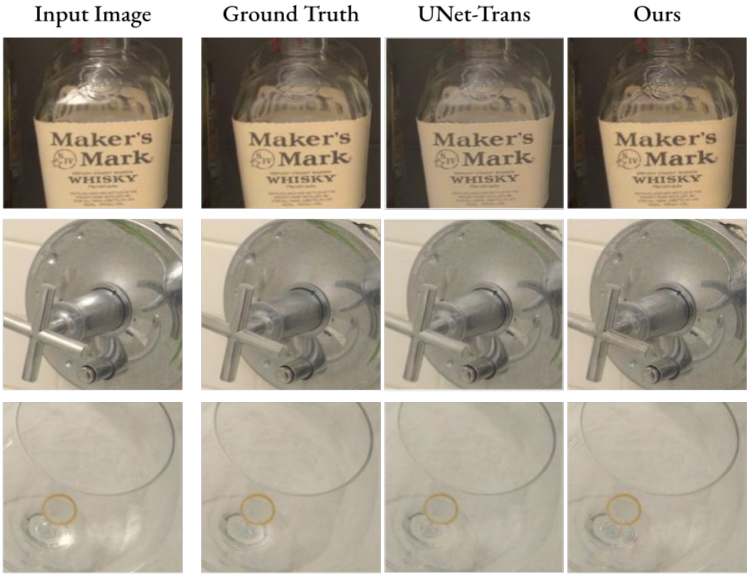


Figure 5. Visual Comparison of Specular Highlight Removal Results with A Recent Method Unet-Trans

Table 4. Patch Scale Analysis on Multi-Head Self-Attention. ↑ Denotes Better Performance with Higher Values, while ↓ Indicates Better Performance with Lower Values

Scale	Acc ↑	BER ↓	MSE ↓	SSIM ↑	PSNR ↑
Multi-scale	**0,9825**	5,26	**0,1353**	0,9523	**30,8200**
Patch Scale: (56 × 56)	0,9775	4,69	0,1624	**0,9541**	29,7693
Patch Scale: (28 × 28)	0,9813	5,41	0,1445	0,9513	30,5848
Patch Scale (14 × 14)	0,9794	5,32	0,159	0,9475	30,3146
Patch Scale: (7 × 7)	0,9725	**3,55**	0,2129	0,9434	28,9624

5. Conclusion

This study proposes a new ViT-based model architecture to jointly detect and remove specular highlights in a given image, defining a multi-scale patch self-attention in the transformer block. The proposed model demonstrated superior performance in specular highlight detection tasks, achieving the best accuracy and balanced error rate (BER) results. This approach enhances detection accuracy and removal quality. The experiments showed that multi-scale attention outperforms single-scale attention, particularly in MSE and PSNR metrics, while maintaining competitive SSIM scores. The multi-scale patch attention mechanism allows the model to process different scales within each attention head, leading to a comprehensive understanding of the image features.

Several enhancements can be explored for future work to improve the model's performance. Incorporating overlapping patches could provide better coverage and finer granularity in detection and removal processes. Exploring new transformer models with advanced architecture might yield additional performance gains. Utilizing pre-trained models on large-scale datasets could offer a strong initial foundation, reduce training time, and improve generalization. These directions offer promising opportunities to refine and enhance the effectiveness of specular highlight detection and removal.

References

[1] S. Jiddi, P. Robert, and E. Marchand, “Detecting specular reflections and cast shadows to estimate reflectance and illumination of dynamic indoor scenes,” *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 2, pp. 1249–1260, 2020.

[2] S. A. Shafer, “Using color to separate reflection components,” *Color Res. Appl.*, vol. 10, no. 4, pp. 210–218, 1985.

[3] L. T. Maloney and B. A. Wandell, “Color constancy: a method for recovering surface spectral reflectance,” in *Readings in Computer Vision*, Elsevier, 1987, pp. 293–297.

[4] Osadchy and Ramamoorthi, “Using specularities for recognition,” in *IEEE ICCV*, IEEE, 2003, pp. 1512–1519.

[5] J. B. Park and A. C. Kak, “A truncated least squares approach to detecting specular highlights in color images,” in *IEEE ICRA*, IEEE, 2003, pp. 1397–1403.

[6] O. El Meslouhi, M. Kardouchi, H. Allali, T. Gadi, and Y. A. Benkaddour, “Automatic detection and inpainting of specular reflections for colposcopic images,” *Cent. Eur. J. Comput. Sci.*, vol. 1, pp. 341–354, 2011.

[7] R. Li, J. Pan, Y. Si, B. Yan, Y. Hu, and H. Qin, “Specular reflections removal for endoscopic image sequences with adaptive-RPCA decomposition,” *IEEE Trans. Med. Imaging*, vol. 39, no. 2, pp. 328–340, 2019.

[8] W. Zhang, X. Zhao, J.-M. Morvan, and L. Chen, “Improving shadow suppression for illumination robust face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 611–624, 2018.

[9] Q. Yang, S. Wang, and N. Ahuja, “Real-time specular highlight removal using bilateral filtering,” in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, Springer, 2010, pp. 87–100.

[10] H. Kim, H. Jin, S. Hadap, and I. Kweon, “Specular reflection separation using dark channel prior,” in *IEEE CVPR*, 2013, pp. 1460–1467.

[11] Q. Yang, J. Tang, and N. Ahuja, “Efficient and robust specular highlight removal,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1304–1311, 2014.

[12] Y. Liu, Z. Yuan, N. Zheng, and Y. Wu, “Saturation-preserving specular reflection separation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3725–3733.

[13] J. Suo, D. An, X. Ji, H. Wang, and Q. Dai, “Fast and high quality highlight removal from a single image,” *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5441–5454, 2016.

[14] T. Yamamoto, T. Kitajima, and R. Kawauchi, “Efficient improvement method for separation of reflection components based on an energy function,” in *2017 IEEE international conference on image processing (ICIP)*, IEEE, 2017, pp. 4222–4226.

[15] I. Funke, S. Bodenstedt, C. Riediger, J. Weitz, and S. Speidel, “Generative adversarial networks for specular highlight removal in endoscopic images,” in *Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling*, SPIE, 2018, pp. 8–16.

[16] S. Muhammad, M. N. Dailey, M. Farooq, M. F. Majeed, and M. Ekpanyapong, “Spec-Net and Spec-CGAN: Deep learning models for specularity removal from faces,” *Image Vis. Comput.*, vol. 93, p. 103823, 2020.

[17] G. Fu, Q. Zhang, Q. Lin, L. Zhu, and C. Xiao, “Learning to Detect Specular Highlights from Real-world Images,” in *ACM Multimedia*, 2020, pp. 1873–1881.

[18] G. Fu, Q. Zhang, L. Zhu, P. Li, and C. Xiao, “A multi-task network for joint specular highlight detection and removal,” in *IEEE/CVF CVPR*, 2021, pp. 7752–7761.

[19] Z. Wu, C. Zhuang, J. Shi, J. Xiao, and J. Guo, “Deep specular highlight removal for single real-world image,” in *SIGGRAPH Asia 2020 Posters*, 2020, pp. 1–2.

[20] G. Fu, Q. Zhang, L. Zhu, C. Xiao, and P. Li, “Towards High-Quality Specular Highlight Removal by Leveraging Large-Scale Synthetic Data,” in *IEEE/CVF ICCV*, 2023, pp. 12857–12865.

[21] Z. Wu, J. Guo, C. Zhuang, J. Xiao, D.-M. Yan, and X. Zhang, "Joint specular highlight detection and removal in single images via Unet-Transformer," *Comput. Vis. Media*, vol. 9, no. 1, pp. 141–154, 2023.

[22] J. Shi, Y. Dong, H. Su, and S. X. Yu, "Learning non-lambertian object intrinsics across shapenet categories," in *IEEE CVPR*, 2017, pp. 1685–1694.

[23] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.

[24] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2020.

[25] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *ArXiv Prepr. ArXiv210405707*, 2021.

[26] L. Karacan, "Multi-image transformer for multi-focus image fusion," *Signal Process. Image Commun.*, vol. 119, p. 117058, 2023.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE CVPR*, 2018, pp. 7132–7141.

[28] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, Springer, 2017, pp. 240–248.

[29] H.-L. Shen, H.-G. Zhang, S.-J. Shao, and J. H. Xin, "Chromaticity-based separation of reflection components in a single image," *Pattern Recognit.*, vol. 41, no. 8, pp. 2461–2469, 2008.

[30] J. Lin, M. El Amine Seddik, M. Tamaazousti, Y. Tamaazousti, and A. Bartoli, "Deep multi-class adversarial specularity removal," in *Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings 21*, Springer, 2019, pp. 3–15.

Author(s) Contributions

The article was authored and prepared solely by the author. The author was responsible for all aspects of the research, including conceptualization, methodology, implementation, analysis, and writing of the manuscript.

Conflict of Interest Notice

The author declares that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 58-75, 2025
DOI: 10.35377/saucis...1579003

RESEARCH ARTICLE

Enhanced Classification of Ear Disease Images Using Metaheuristic Feature Selection

Furkancan Demircan[1*], Murat Ekinci[2], Zafer Cömert[1], Eyüp Gedikli[3]

[1*] Software Engineering, Faculty of Engineering and Natural Sciences, Samsun University, Türkiye, ror.org/02brte405
[2] Computer Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Türkiye, ror.org/03z8fyr40
[3] Software Engineering, of Faculty of Technology, Karadeniz Technical University, Trabzon, Türkiye, ror.org/03z8fyr40

Corresponding author:
Furkancan Demircan, Software Engineering, Faculty of Engineering and Natural Sciences, Samsun University, Samsun, Türkiye
furkancan.demircan@samsun.edu.tr

Article History:
Received: 04.11.2024
Revised: 22.02.2025
Accepted: 24.02.2025
Published Online: 27.03.2025

ABSTRACT
Ear diseases are characterized by various symptoms, including balance disturbances, delayed speech development in children, headaches, fever, and hearing loss. To prevent further complications, these conditions must be diagnosed and treated promptly. The traditional diagnostic method has been an otoscope examination by otolaryngologists. However, the accuracy of this approach is contingent upon the clinician's expertise and the quality of the equipment used, which can render it susceptible to misdiagnosis. Incorrect diagnoses may result in the administration of antibiotics unnecessarily, disease progression, and other adverse consequences. This study aims to evaluate the efficacy of computationally efficient machine learning models in classifying ear disease images. To enhance classification accuracy, a Histogram of Oriented Gradients (HOG) was employed for feature extraction and optimization algorithms were utilized for feature selection. The Whale Optimization Algorithm (WOA) effectively selected informative features for the k-Nearest Neighbors (kNN) model, achieving a classification accuracy of 92.6%. Furthermore, the Support Vector Machine (SVM) model achieved an accuracy of 92% using a feature map comprising features selected by a range of optimization algorithms. The experimental findings emphasize the potential of strategic feature selection in enhancing the performance of classical machine learning models for ear disease classification. By employing computationally efficient techniques such as HOG and optimization algorithms, these models can attain classification accuracies that are on par with those of more resource-intensive deep learning approaches. Such developments facilitate the creation of accessible and efficient diagnostic tools, particularly beneficial in resource-constrained clinical settings. The findings of this study provide a basis for further research to enhance the diagnostic precision of machine learning-based techniques in medical imaging.

Keywords: Machine learning, Optimization, Feature extraction, Feature selection

1. Introduction

The human ear is a highly sophisticated organ consisting of three sections—the outer, middle, and inner ear—that work in harmony to capture, amplify, and convert sound waves into neural signals, which are then transmitted to the brain for auditory perception. Despite its advanced structure and functionality, the ear is prone to various disorders that can obstruct the transmission of sound waves and lead to hearing loss. Diseases affecting the eardrum, such as otitis media (OM), cerumen impaction, and myringosclerosis, can reduce the eardrum's elasticity or even cause it to rupture, resulting in diminished hearing capabilities. Additionally, certain conditions can affect the ear canal, distorting or reducing the sound waves transmitted to the eardrum[1].

Beyond hearing impairments, ear disorders can also lead to systemic effects on the body, such as fever, nausea, and itching in the ear. Common ailments like otitis externa (swimmer's ear) and otitis media, if left untreated, may escalate into serious complications, including balance disorders, developmental delays in speech among children, and a significant reduction in the quality of life. These challenges highlight the importance of timely and accurate diagnosis and treatment to prevent health complications and improve overall well-being[1].

Traditionally, the diagnosis of ear diseases has relied on examining the ear canal and eardrum using a medical instrument known as an otoscope performed by trained otolaryngologists. This process typically involves waiting for the disease to reach a certain stage before a diagnosis can be confirmed. The treatment approach is implemented following the diagnosis, and its effectiveness is monitored based on the patient's response. Depending on the observed outcomes, the treatment may be adjusted or continued[1].

Cite as: F. Demircan, M. Ekinci, Z. Cömert, and E. Gedikli, "Enhanced Classification of Ear Disease Images Using Metaheuristic Feature Selection," *Sakarya University Journal of Computer and Information Sciences*, vol. 8, no. 1, pp. 58-75, 2025, doi: 10.35377/saucis...1579003

While this method is widely used, its accuracy depends on the clinician's expertise and the quality of the otoscope used. This reliance introduces significant subjectivity into the diagnostic process, increasing the potential for human error. Misdiagnoses can lead to delayed or inappropriate treatments, sometimes resulting in irreversible complications, such as disease progression or unnecessary interventions. These limitations highlight the need for more objective, accurate, and reliable diagnostic tools to mitigate human error risks and improve patient outcomes[1].

Integrating autonomous decision support systems into the diagnostic process has demonstrated significant potential for improving the accuracy and efficiency of identifying ear diseases. By leveraging advanced artificial intelligence (AI) techniques, these systems can analyze otological images with precision, identifying patterns and abnormalities that might be overlooked by human observers. This technology provides objective, data-driven insights, reducing clinicians' reliance on subjective interpretations. Moreover, AI-powered systems can incorporate patient history and other contextual data to support more nuanced and individualized diagnoses[1].

A key advantage of this approach is its ability to minimize the overuse of antibiotics, a common issue arising from misdiagnoses or precautionary prescriptions. By providing clinicians with reliable diagnostic support, these systems can help differentiate between conditions requiring antibiotic treatment and those not, promoting responsible antibiotic usage. This benefits individual patient care and contributes to broader public health efforts to combat antibiotic resistance. Adopting AI-driven decision support systems represents a step toward more accurate, consistent, and efficient diagnostics, particularly in resource-limited clinical settings where access to specialist expertise may be constrained[2].

Previous studies have focused on classifying ear diseases using machine learning and deep learning techniques. However, deep learning often demands significant computational resources and prolonged processing times. This study adopts a more efficient approach to address these challenges by utilizing the Histogram of Oriented Gradients (HOG) algorithm for feature extraction, a traditional yet effective method. Conventional machine learning algorithms are employed for the classification step, providing a less resource-intensive alternative to deep learning. Additionally, feature selection techniques are integrated into the workflow to enhance the model's performance by ensuring that only the most relevant features contribute to the classification process.

2. Literature Reviews

Artificial intelligence and deep learning have led to significant advances in diagnosing ear diseases. Many studies have employed convolutional neural networks (CNNs) to enhance the accuracy of diagnoses across a range of ear disease types.

Wu et al. proposed a mobile application for efficient monitoring and classification of pediatric otitis media using Xception and MobileNet-V2 convolutional neural network (CNN) architectures pre-trained on ImageNet for background classification. The proposed method's training and testing were applied to 12203 images containing three types of otitis media diseases obtained from Shenzhen Children's Hospital between January 2016 and December 2019. As a result of the study, Xception and MobileNet-V2 architectures achieved 97.82% and 96.04% accuracy in the test images on the system and 90.85% and 88.89% accuracy in the test images on the mobile application, respectively[3].

Sundgaard et al. proposed that Inception V3 architecture be pre-trained on ImageNet to diagnose otitis media in otoscope images automatically. The proposed method was trained and tested on 1136 images of three types of otitis media from 519 patients at the Kamide ENT clinic in Shizuoka, Japan. Five different loss functions were compared in the study, and the best success was obtained from the triple loss function with an accuracy of 85%[4].

Alhudhaif et al. proposed a hybrid model based on CNN architecture to support the diagnosis of five types of otitis media diseases. In the training and testing of the proposed architecture, an open-access dataset of 956 otoscope images from Private Van Akdamar Hospital between October 2018 and June 2019, which were classified into eight types of diseases by experts, was used. The decision mechanism of the CNN architecture was developed using a combination of channel and spatial model (CBAM), residual blocks and hyper-column technique. On the same dataset, AlexNet, VGG-16, VGG-19, GoogleNet, ResNet-50 and the proposed method showed 81.40%, 81.40%, 84.88%, 80.23%, 80.23%, 80.23% and 98.26% accuracy for four classes respectively[5].

Tran et al. proposed an automatic diagnostic system for pediatric otitis media images using a level set-based active counter method for segmentation, color, HOG and SSIM features for feature extraction and joint sparse representation algorithm for disease classification. For training and testing the proposed method, 1230 digital otoscope images of children in General Cathay Hospital were used, and experts under two types of diseases diagnosed them. The proposed method showed 91.41% accuracy on the test dataset[6].

Myburgh et al. proposed a mobile application using cloud-based decision trees and neural network structure to diagnose five types of otitis media diseases automatically. In the training and testing of the proposed method, 562 video otoscope images containing five types of otitis media diseases obtained from different sources were used. The proposed decision tree and neural network showed 81.58% and 86.84% accuracy, respectively[7].

Mohammed et al. proposed a system that can automatically diagnose otitis media using CNN-EfficientNetB0 feature extractor, Bayesian optimization hyperparameter selector and CNN-BiLSTM classifier. The combination of CNN-BiLSTM architectures showed higher performance and worked faster than the methods based on CNN architecture alone. In the training and testing of the proposed method, 880 otoscope images with four types of diseases were obtained from the Department of Otorhinolaryngology of the Clinical Hospital of Universidad de Chile. The proposed architecture showed 100% accuracy[8].

Wang et al. proposed the MESIC architecture for classifying intra-ear diseases. The dataset consists of 998 images of three classes obtained from CT scans of patients in Xiangya Hospital. Preprocessing is performed before using the images in the proposed architecture. In the preprocessing, a region of interest with a size of 100x100 pixels is extracted. The proposed architecture extracts features from the images with the VGG-16 model. Three new feature maps are obtained from the extracted features with the attention block. A full-connected layer and softmax are used to classify the feature maps. The proposed architecture showed 90.1% accuracy on the dataset[9].

Cha et al. compared the classification performance of different models to diagnose ear diseases automatically. The dataset was 10544 otoendoscopic images of six classes obtained from patients at Severance Hospital. The images are divided into 80% and 20% for training and testing, respectively. AlexNet, GoogLeNet, ResNet, Inception-V3, Inception-ResNet-V2, SqueezeNet and MobileNet-V2 were selected from open deep learning models. As a result of the application of the dataset on the models, the two models with the highest success were selected. A probability selector is proposed that combines the outputs of the two models. The probability selector classifies the image according to the label with the highest score from the two models. As a result of the study, the proposed model shows 93.67% accuracy[10].

Chen et al. proposed an AI-assisted mobile algorithm that can classify diseases inside the ear. 2161 images of 10 classes obtained from Taipei Veteran General Hospital were used as a dataset. The images are divided into 80% and 20% for training and testing, respectively. "Augmentation" was applied to the images as preprocessing. VGG19, VGG19, Xception, InceptionV3, NASNetLarge and ResNet50 were selected as deep-learning models. The feature maps of the models were extracted using CAM (Class Activation Map). The InceptionV3 model showed the highest accuracy, with 98%. The weight values obtained from the InceptionV3 model were given to the NasNetMobile, MobileNetV2 and DenseNet201 models for use in the mobile system. At this study's end, the mobile system's highest success rate was obtained from the MobileNetV2 model at 97.6%. The proposed study shows that the mobile system can diagnose as successfully as experts[11].

Sundgaard et al. proposed an architecture that uses broadband tympanometry (WBT) signal values to classify otitis media. Broadband tympanometry measures the acoustic properties of the ear canal using frequency signals. The healthy ear canal showed higher absorption values than the diseased one. The study used 1014 WBT measurements obtained from patients aged 2 months to 12 years at the Kamide ENT clinic in Japan as the data set. The measurements were performed using the Titan system. Three data classes were obtained: 488 healthy (NOE), 372 OME and 154 AOM. The data set was divided into 80% for training and 20% for testing; the training data set was again divided into 20% and used for validation. First, a model inspired by the AlexNet architecture is proposed for two classes (healthy and OM). The proposed model is compared with the pre-trained VGG, ResNet and InceptionV3 available in the PyTorch library. Since the pre-trained models are trained image-based, their accuracy is low. The accuracy of the proposed model is 92.6%. In the second approach, OM disease was tried to differentiate but could not be differentiated[12].

Tsutsumi et al. proposed a web-based deep-learning model for automatically classifying ear diseases. The dataset used in the study was a combination of images available on the internet and Google Images images. The dataset comprises 400 images, including 116 AOM, 44 otitis externa, 23 CSOM, and 21 earwax impactions. The dataset is divided into two different classes: categorical and dichotomous. The dataset was divided into 75% for training and 25% for testing. CNN architectures ResNet50, InceptionV3, InceptionResNetV2 and MobileNetV2 were selected for classification. As a result of the study, the highest accuracy success for the binary class was obtained from the MobileNetV2 model with 90%. For the categorical class, the MobileNetV2 model achieved 91% accuracy[13].

Wang et al. proposed a deep-learning model using CT images for COM disease detection. Images from 672 CT scans of 562 patients were used as the dataset. The images were obtained from a 128-channel multi-detector SOMATOM Definition Edge CT scanner. The axial view of the temporal bone allowed both ears to be included in the images, resulting in 1344 ear images. After experts examined the two ears, a final dataset of 1147 images belonging to 3 classes, CSOM, normal and cholesteatoma, was extracted. In the dataset, 85% of the images were used for training and validation and 15% for testing. Faster R-CNN model was used to extract the ear region in the dataset. The number of images increased by applying augmentation to the images in the dataset. The inceptionV3 model was used for feature extraction, and the softmax function was used for classification. The performances of the proposed model and six experts on the test data are compared. The average accuracy of the proposed model on two classes (CSOM and Normal) is 86%. The average accuracy of the experts is 82%[14].

Zeng et al. investigated the potential of deep learning (DL) for identifying key features of otitis media with effusion (OME), a middle ear infection. Within the scope of the study, a data set consisting of 6393 images was used. Their study focused on detecting atelectasis (collapsed tissue) and attic retraction pocket (a specific indentation) in otoscopic images (ear canal images) collected from multiple centers. The researchers developed and validated a DL model (InceptionV3 with CAM) that achieved high accuracy in detecting both features, with 89% accuracy for attic retraction and 79% for atelectasis. The model highlighted the most relevant image regions for identifying these features[15].

In a study exploring the generalizability of deep learning for ear disease detection, Habib et al. investigated how well these algorithms performed on images from different locations. They used a dataset of over 1800 otoscopic images collected across diverse geographic regions: Van (Turkey), Santiago (Chile), and Ohio (USA). The images were classified as normal or abnormal using four deep-learning models: ResNet-50, DenseNet-161, VGG16, and Vision Transformer. Five-fold cross-validation assessed the models' performance on the original dataset and unseen external images. The Vision Transformer model obtained the highest accuracy value of 92%. While the algorithms achieved high accuracy when tested on the initial dataset (internal images), their performance dropped significantly when evaluated on external images from new locations[16].

In a 2021 study, Pham et al. proposed a novel deep-learning approach called EAR-UNet. This model tackles the challenge of segmenting tympanic membranes from otoscopic images. Built upon the foundation of the UNet architecture, EAR-UNet leverages three key components: EfficientNet for efficient feature extraction in the encoder, an Attention gate for improved information flow in the skip connections, and Residual blocks within the decoder to enhance feature learning. To assess its performance, the researchers evaluated EAR-UNet on a dataset containing otoscopic images from 1024 patients, both diagnosed with and without otitis media. Notably, the model achieved an impressive average Dice Similarity Coefficient (DSC) of 0.929, demonstrating its effectiveness in segmentation without requiring additional pre-processing or post-processing steps[17].

Cömert et al. proposed a computerized otoscopy image-based AI model for diagnosing otitis media (OM). The dataset consisted of 880 otoscope images from a publicly available Ear Imagery dataset, categorized into four classes: chronic otitis media, myringosclerosis, earwax plug, and normal conditions. The method involved using Vision Transformers (ViT) for deep feature extraction, combined with Support Vector Machines (SVM) optimized through grid search. The best-performing model, Optimizable SVM, achieved an accuracy of 99.37%, showcasing exceptional classification performance[18].

Demircan et al. proposed a computer-aided diagnosis system for classifying ear diseases using otoscope images. The dataset comprised 880 eardrum images categorized into four classes: chronic suppurative otitis media (CSOM), earwax, myringosclerosis, and normal conditions. The method utilized a Vision Transformer (ViT) as a feature extractor, followed by classification using machine learning algorithms, including k-Nearest Neighbors (kNN), Support Vector Machine (SVM), and Random Forest. The best-performing model, ViT with kNN, achieved an accuracy of 99.00%, demonstrating high classification performance[19].

3. Material and Methods

The study consists of three phases, each addressing a specific component of the proposed methodology. In the first phase, features extracted using the Histogram of Oriented Gradients (HOG) algorithm were classified using traditional machine learning algorithms to establish baseline performance metrics. The second phase investigated the influence of feature selection on classification accuracy by employing metaheuristic optimization algorithms. These algorithms were applied to identify the most relevant features, reducing the dimensionality of the feature space while preserving essential information. In the final phase, the features selected in the previous steps were combined to analyze their collective impact on classification performance. To ensure the reliability of the integration process, duplicate features were eliminated. The results from all three phases were systematically compared, highlighting improvements in classification performance achieved through the proposed methodology.

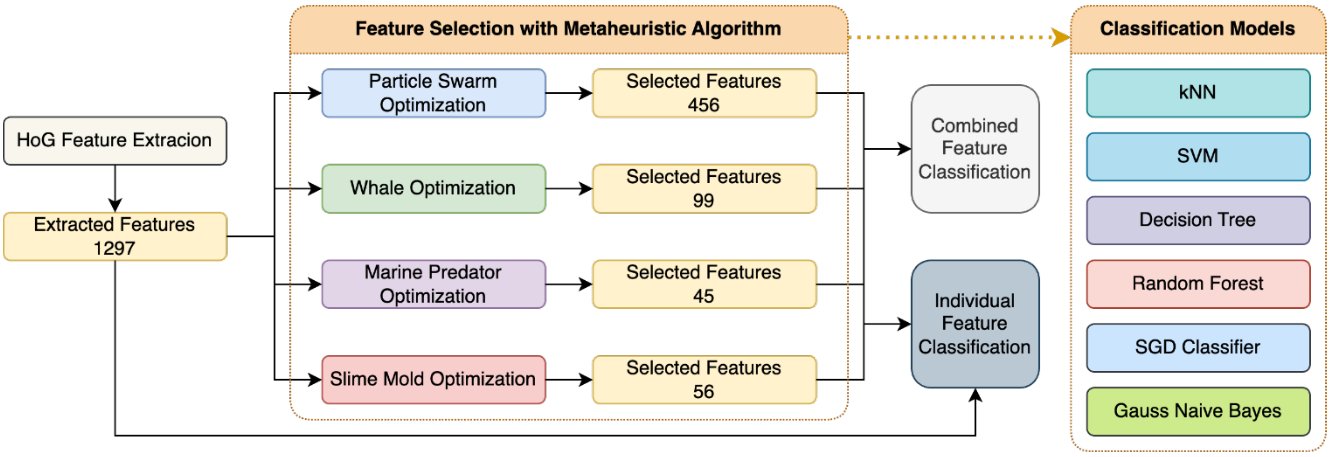


Figure 1. Overview of Feature Extraction, Metaheuristic Optimization, And Machine Learning Classifiers for Automated Ear Disease Diagnosis

Metaheuristic optimization algorithms, including the Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), Marine Predators Algorithm (MPA), and Slime Mold Algorithm (SMA), were integral to the feature selection process. These algorithms iteratively explore the feature space to identify subsets of features that maximize classification accuracy while minimizing redundancy. The selection process is guided by a well-defined fitness function incorporating classification error and the number of selected features, ensuring a balance between accuracy and parsimony. By optimizing feature sets, metaheuristic algorithms significantly enhance the efficiency and effectiveness of the classification process. This structured approach underscores feature selection's critical role in improving the proposed methodology's overall performance for ear disease classification.

3.1. Dataset

The experiment uses a dataset of 880 eardrum images from the Hospital Clínico of Universidad de Chile, categorized into four classes: CSOM, earwax blockage, myringosclerosis, and normal. Images are in JPG format with a 420x380 resolution [7]. Figure 2 presents sample images from each category in the dataset.

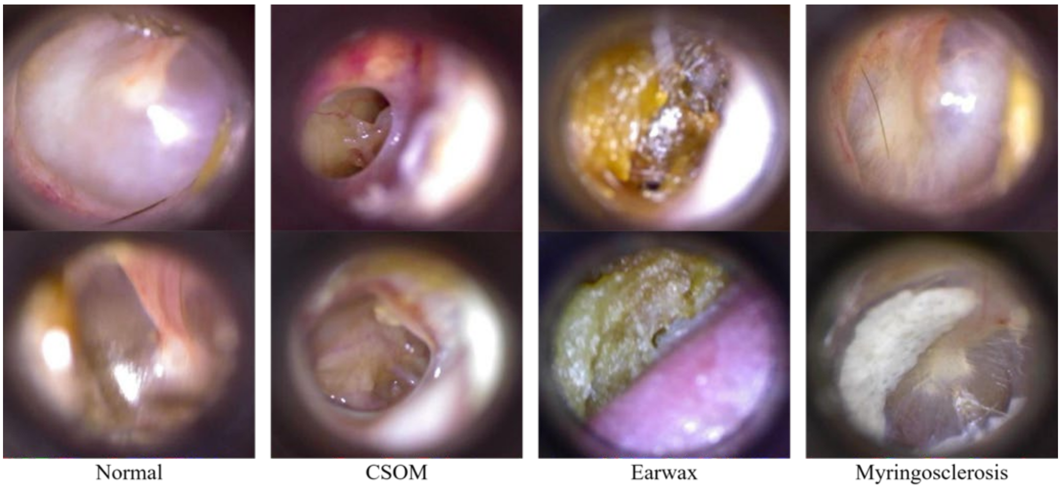


Figure 2. Representative Samples from the Eardrum Image Dataset, Categorized by Disease Type

3.2. Histograms of Oriented Gradients (HOG) Feature Extraction Algorithm

The Histogram of Oriented Gradients (HOG) feature extraction process captures local shape information by effectively analyzing pixel gradients to represent an image’s structure through a feature vector. This widely used technique involves four main steps: gradient computation, orientation binning, descriptor block formation, and feature vector construction. First, image intensity gradients are calculated using convolution to highlight edges and intensity changes, resulting in gradient magnitudes and orientations for each pixel. The image is then divided into small cells (e.g., 8×8 pixels), and a histogram of gradient orientations is constructed for each cell, where each pixel contributes to a bin based on its gradient magnitude and orientation. To enhance robustness against illumination changes, these histograms are normalized within larger blocks composed of multiple cells (e.g., 2×2 cells). Finally, the normalized histograms from all blocks are concatenated to form a comprehensive feature vector [20].

HOG’s success is attributed to its ability to capture detailed local intensity gradients and edge orientations while maintaining resilience to geometric and photometric transformations. This makes it a crucial component in computer vision applications, particularly object detection [20].

3.3. Metaheuristic Optimization Algorithms

Applying metaheuristic optimization techniques enables effective resolution of complex problems through a balanced approach that integrates exploration and exploitation. In contrast to deterministic algorithms, which guarantee optimal solutions but at significant computational expense, metaheuristic methods employ strategies inspired by natural processes to escape local optima and identify high-quality solutions. The versatility of these methods renders them valuable across various disciplines, including engineering, logistics, and machine learning.

3.3.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a metaheuristic algorithm inspired by the collective behavior of swarms, such as flocks of birds or schools of fish. It addresses optimization problems by iteratively refining a group of candidate solutions, called particles, within a defined search space. Each particle represents a potential solution and possesses a position and velocity that determine its movement through the search space [21].

The particles adjust their trajectories by considering both their best-known positions and the best-known positions of the entire swarm. This collective sharing of information enables the swarm to converge toward optimal solutions over time. An objective function evaluates each particle’s position's fitness, updating personal and global best positions based on these evaluations.

The algorithm continues this process until a termination criterion is met, such as reaching a maximum number of iterations or achieving a desired fitness level. PSO effectively balances search space exploration with the exploitation of the best-found solutions. Three key parameters influence this balance: the inertia weight, which affects a particle's tendency to continue in its current direction; and two acceleration coefficients, which control the influence of the particle's own best position (cognitive component) and the swarm's global best position (social component) [21].

3.3.2. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA) is a metaheuristic optimization technique inspired by the unique hunting behavior of humpback whales, particularly their bubble-net feeding strategy. This algorithm seeks optimal solutions by simulating the social and cooperative behaviors observed in whale pods during hunting. It operates through three main phases: encircling prey, bubble-net attacking, and searching for prey [22].

In the encircling prey phase, the algorithm assumes that the current best candidate solution represents the prey, and all other candidate solutions (whales) adjust their positions relative to this best solution. This mimics how whales surround their prey in nature. Each whale updates its position by moving closer to the best-known position, effectively narrowing the search area around promising solutions. The bubble-net attacking phase simulates the bubble-net feeding mechanism of humpback whales using two strategies: shrinking encircling and spiral updating. The shrinking encircling strategy involves gradually reducing the distance between the whales and the prey, akin to whales tightening their circle around the prey. The spiral updating strategy models the helical movement of whales as they spiral upward toward the surface while creating bubbles to trap the prey. This combination allows the algorithm to exploit the search space around the current best solution more effectively, refining candidate solutions and enhancing optimization. In the search for prey phase, the algorithm emphasizes exploration by moving whales toward randomly selected candidate solutions rather than the current best one. This simulates the whales' behavior when searching for new prey and helps the algorithm avoid local optima by diversifying the search process. By considering random positions in the search space, the algorithm increases the likelihood of finding the global optimum [22].

3.3.3. Marine Predator Algorithm (MPA)

The Marine Predator Algorithm (MPA) is a modern metaheuristic optimization technique inspired by the hunting strategies of marine animals like fish, sharks, and whales. It efficiently explores and exploits the solution space to find optimal solutions. The MPA process consists of three phases: exploration, transition, and exploitation, which emulate the adaptive behaviors of marine predators and balance global exploration with local exploitation [23].

The algorithm begins by generating a population of candidate solutions, termed "prey," randomly distributed in a multidimensional search space. In the exploration phase, the algorithm mimics the wide-ranging search for prey, using a probabilistic movement pattern known as Lévy flight to explore large areas of the search space. This phase ensures thorough investigation and prevents premature convergence on suboptimal solutions. The transition phase serves as a bridge between exploration and exploitation, fine-tuning the movement of candidate solutions based on their proximity to the best-known solution, referred to as the "best predator." This phase gradually shifts the focus from broad exploration to more targeted exploitation of promising regions while introducing random factors to maintain diversity and avoid local optima. In the exploitation phase, the algorithm intensifies the search for the best solutions, simulating the focused hunting behavior of predators when prey is nearby. Candidate solutions move based on Brownian motion, allowing for precise refinement of solutions in promising regions. The fitness of all candidate solutions is then evaluated, and the best-known solution is updated if a superior solution is found [23].

3.3.4. Slime Mold Optimization Algorithm (SMA)

The Slime Mold Optimization Algorithm (SMA) is a bio-inspired metaheuristic algorithm modeled after the foraging behavior of slime molds. It efficiently explores and exploits the solution space to solve complex optimization problems. The process begins by initializing a population of candidate solutions, or "slime molds," randomly distributed within a multidimensional search space. Each solution is evaluated using a fitness function that determines its quality based on the specific problem [24].

SMA updates the positions of the slime molds by simulating their movement toward better solutions, much like how slime molds move toward nutrient-rich areas. It also incorporates a mechanism to avoid poorer solutions, guiding the slime molds away from suboptimal regions of the search space. To prevent the algorithm from getting stuck in local optima, a random movement component is added to encourage broader exploration [24].

Additionally, SMA mimics the collective behavior of slime molds aggregating around nutrient sources by considering both the best-known solution and the average position of the population. This behavior helps refine the search process and directs the slime molds toward more promising areas. The algorithm iteratively updates the positions and evaluates the fitness of the slime molds until a termination condition, such as a set number of iterations or a desired fitness level, is reached. This balance of exploration and exploitation allows SMA to navigate complex optimization problems and find optimal or near-optimal solutions effectively [24].

3.4. Machine Learning Algorithms

Machine learning, a branch of artificial intelligence (AI), transforms many fields by enabling computers to learn from data and improve their performance without explicit instruction. It develops algorithms that autonomously recognize patterns and make data-driven decisions. Its impact ranges from personalized recommendations and search engine improvements to medical predictions and industrial automation. As the volume of data grows, machine learning's potential to drive innovation and efficiency increases, fundamentally changing how we interact with technology. This integration offers unprecedented opportunities for advances in scientific discovery, business optimization, and societal progress.

3.4.1. k-Nearest Neighbors (kNN)

The k-Nearest Neighbors (kNN) algorithm is a nonparametric classification method based on proximity. It stores the entire data set during training and calculates the distance from a new data point to all training instances to find the k nearest neighbors. The class label for the new point is assigned based on the majority class of these neighbors. The effectiveness of kNN depends on the chosen distance metric, the value of k, and the dimensionality of the feature space[25].

3.4.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning model for classification tasks. Its goal is to find the optimal hyperplane that maximally separates data points of different classes. The data points closest to the hyperplane, called support vectors, are crucial for defining the decision boundary. To handle non-linearly separable data, SVM uses kernel functions to project the data into a higher-dimensional space[25].

3.4.3. Stochastic Gradient Descent (SGD) Classifier

Stochastic Gradient Descent (SGD) is an iterative optimization algorithm that trains linear classifiers. It updates model parameters by processing small batches of data, making it efficient for large datasets. By minimizing a loss function, such as hinge or logistic loss, SGD adjusts model weights to improve predictive accuracy and can incorporate regularization techniques to prevent overfitting[26].

3.4.4. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem that assumes features are conditionally independent given the class label for efficient computation. It computes the posterior probability for each class based on the observed features and assigns the class with the highest probability. While this independence assumption simplifies the model, it can hinder performance when significant feature dependencies exist[26].

3.4.5. Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive accuracy. Each tree is constructed on a random subset of the data, and feature selection at each node is randomized. By aggregating the predictions of individual trees through majority voting, random forests effectively reduce overfitting and enhance generalization performance [25].

3.4.6. Decision Tree

A decision tree is a supervised learning algorithm that creates a tree-like model of decisions and their possible consequences. It partitions the data into subsets based on feature values, aiming to maximize information gain or minimize impurity at each split. Decision trees can handle both categorical and numerical data and are interpretable, but they can be prone to overfitting [25].

3.5. Metaheuristic Feature Selection

The application of metaheuristic algorithms for feature selection in this study was designed to optimize the classification performance by reducing the dimensionality of the feature space and selecting the most informative attributes. This process was guided by a mathematical fitness function that balanced classification accuracy and model simplicity. The fitness function, denoted as $F(X)$, was formulated as a weighted combination of the classification error rate and the cardinality of the selected feature set, as shown in Equation (1):

$$F(X) = \alpha \cdot Error(X) + \beta \cdot (\frac{\|X\|}{\|X_{total}\|}) \tag{1}$$

In this equation, α and β are weighting factors, $Error(X)$ represents the classification error rate, $\|X\|$ is the number of selected features, and $\|X_{total}\|$ is the total number of available features. The optimization aimed to minimize $F(X)$, thus achieving a trade-off between maximizing classification accuracy and minimizing the number of selected features to enhance computational efficiency. The error value, denoted as $Error(X)$, is calculated using the formulation provided in Equation (2). This equation measures the proportion of incorrect predictions made by the model relative to the total number of predictions.

Table 1. Selected Parameters and Values for Machine Learning Algorithms

Machine Learning Algorithm	Parameter	Value	Explanation
kNN	Metric	Minkowski	The distance metric is used for neighboring computation. Minkowski is a generalization of Euclidean and Manhattan distances.
	Neighbors	Labels Size (4)	Several nearest neighbors used for classification correspond to the number of class labels.
	Weights	Uniform	Uniform weight means all points in the neighborhood are equally weighted.
SVM/SVC	Regularization Parameter	1	Controls the trade-off between achieving a low error and a large margin.
	Kernel	Radial Basis Function (RBF)	The kernel function is used to transform data into a higher-dimensional space.
	Gamma	1 / (Features * X)	Kernel coefficient for 'rbf' kernel; smaller values mean a wider influence.
SGD Classifier	Loss	Hinge	Specifies the loss function; hinge is used for linear Support Vector Machines.
	Penalty	L2	L2 regularization adds squared penalty terms to prevent overfitting.
	Max Iterations	10	Maximum number of iterations for the algorithm to converge.
	Learning Rate	Optimal	The learning rate schedule adjusts step size based on data and iterations.
Gaussian Naïve Bayes	Variance Smoothing	1,00E-09	A portion of the largest variance was added for numerical stability.
	Priors	None	Prior probabilities of classes; if None, they are inferred from data.
Random Forest Classifier	Number of Trees	Labels Size (4)	Number of decision trees built in the forest. More trees improve accuracy but increase computation.
	Criterion	Gini	Function to measure split quality: Gini impurity is a common measure.
	Bootstrap	TRUE	Whether bootstrap samples are used to build each tree.
Decision Tree Classifier	Strategy	Best	Choosing the best-split strategy based on impurity.
	Criterion	Gini	The function is to measure the quality of a split; Gini impurity is used.
	Max Depth	None	Maximum depth of the tree; if None, nodes are expanded until all leaves are pure.

$$Error(X) = 1 - \frac{\sum_{i=1}^{N} g(\hat{y}_i, y_i)}{N} \tag{2}$$

$$g(\hat{y}_i, y_i) = \begin{cases} 1, & \hat{y}_i = y_i \\ 0, & \hat{y}_i \neq y_i \end{cases} \tag{3}$$

Where, $\boldsymbol{N}$ represents the total number of samples, $\boldsymbol{\hat{y}_i}$ denotes the predicted value for the i-th sample, and $\boldsymbol{y_i}$ corresponds to the true value. The Equation (3), $\boldsymbol{g(\hat{y}_i, y_i)}$ acts as an indicator function that returns a value of 1 when the predicted value matches the true value (indicating a correct prediction) and 0 otherwise (indicating an incorrect prediction).

4. Results

The experimental setup comprised a desktop computer with an Intel i9 13900k processor and an NVIDIA RTX 4080 graphics card with 16GB of GDDR6X memory. The CUDA toolkit enhanced GPU performance, and the PyTorch library was the main framework. The scikit-learn library was also utilized for data preprocessing and implementing baseline machine-learning models.

The initial parameters and their corresponding values for the machine learning algorithms employed in this study are presented in Table 1. These parameters were carefully chosen based on their default settings, which are widely accepted in the literature and serve as a reliable baseline for performance evaluation. Adjustments to these parameters can be considered in future work to optimize results for specific datasets or tasks.

This study extracted HOG features from in-ear images, resulting in a 1297-dimensional feature matrix using a 32x32 filter. The initial experiment classified these features using a variety of algorithms, including kNN, SVM, DT, RF, SGD, and GNB. The performance metrics from the experiments are presented in Table 2.

Table 2. Performance Metrics of the First Experimental Study (The Best-Performing Model Highlighted in Bold)

Models	Accuracy	Precision	Recall
kNN	0,886	0,889	0,886
SVM	**0,914**	**0,914**	**0,914**
SGDC	0,869	0,878	0,869
GNB	0,767	0,768	0,767
RF	0,812	0,813	0,812
DT	0,761	0,768	0,761

The SVM outperformed all other models in classifying ear diseases, achieving the highest accuracy, precision, and recall with minimal false positives and negatives. The kNN algorithm closely followed, demonstrating strong classification capabilities. The SGDC showed moderate performance, while GNB and RF models exhibited balanced but lower effectiveness. The DT model was the least effective among those evaluated.

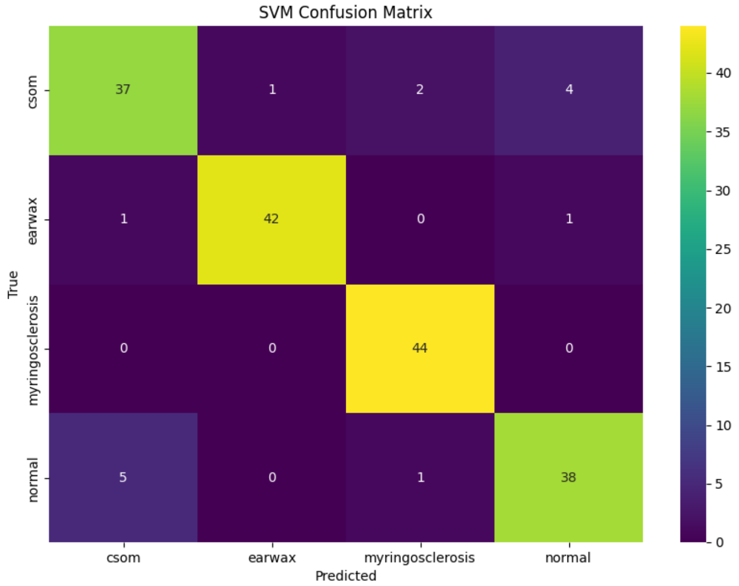


Figure 3. Confusion Matrix of SVM Algorithm

Table 3. Initial Parameters for Metaheuristic Optimization Algorithms

Optimization Algorithm	Parameters	Values
Whale Optimization Algorithm	C Constant	1
	Lower Bond	0
	Upper Bond	1
	Threshold	0,5
	B constant	1
	Population	100
Slime Mold Optimization Algorithm	Lower Bond	0
	Upper Bond	1
	Threshold	0,5
	Control local and global	0,03
	Population	100
Marine Predator Algorithm	Lower Bond	0
	Upper Bond	1
	Threshold	0,5
	Levy Component	1,5
	Constant (P)	0,5
	Fish Aggregating Devices Effect	0,2
	Population	100
Particle Swarm Optimization	Lower Bond	0
	Upper Bond	1
	Threshold	0,5
	Cognitive Factor	2
	Social Weight	2
	Inertia Weight	0,9
	Maximum Velocity	$\frac{(Upper\ Bond\ -\ Lower\ Bond)}{2}$
	Population	100

These findings highlight the reliability and accuracy of the SVM and kNN models for ear disease classification. This analysis can guide the selection of machine learning algorithms based on specific classification needs and resource considerations. The confusion matrix for the SVM algorithm, which demonstrated the highest accuracy, is presented in Figure 3.

The subsequent experimental investigation used metaheuristic optimization algorithms to select pertinent features from the HOG-derived feature maps. The algorithms employed were MPA, SMA, PSO, and WOA. Each algorithm was executed 50 times over 1,000 iterations with consistent parameter settings to facilitate comparison.

The initial parameters selected for the metaheuristic optimization algorithms utilized in this study are presented in Table 3. These parameters have been chosen based on their default values, as commonly recommended in the literature, to ensure a standardized and unbiased evaluation of the algorithms' performance.

The generated feature maps were evaluated using the kNN classifier to identify features with the highest classification accuracy. Figure 4 illustrates the mean fitness values attained by each algorithm.

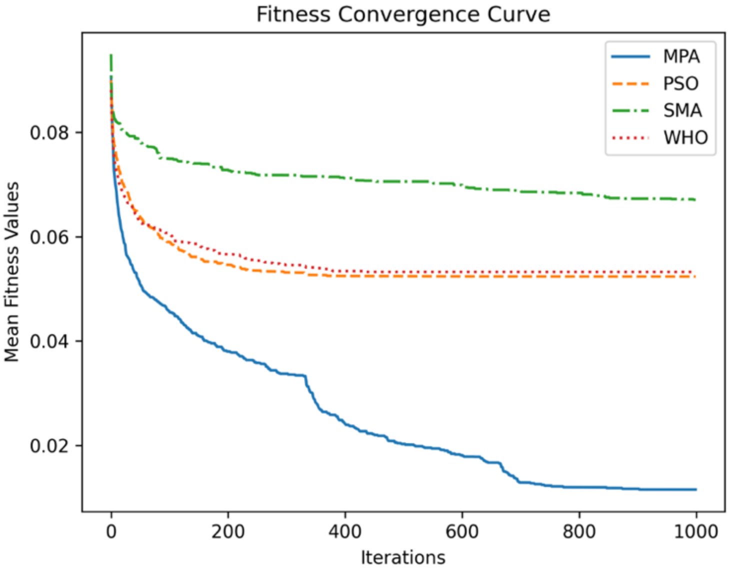


Figure 4. Average Fitness Values of the MPA, PSO, SMA, and WOA Algorithms Over 50 Runs with 1000 Iterations

Figure 4 displays the convergence behaviors of four optimization algorithms: WOA, SMA, PSO, and MPA. The x-axis denotes the number of iterations, while the y-axis represents the fitness value, indicating each algorithm's performance over time. WOA (red line) and SMA (green line) show rapid initial decreases in fitness value but plateau after approximately 20 iterations, limiting further improvement. PSO (orange line) also converges quickly but stabilizes at a higher fitness level, suggesting less optimal solutions. In contrast, MPA (blue line) exhibits a slower initial improvement with continuous enhancement throughout the iterations, ultimately achieving the lowest fitness value among all algorithms.

These findings highlight the superior optimization performance of the MPA algorithm due to its sustained improvement and ability to reach the most optimal solution. The results underscore the importance of considering initial convergence rates and long-term performance when selecting optimization algorithms for scenarios requiring optimal solutions.

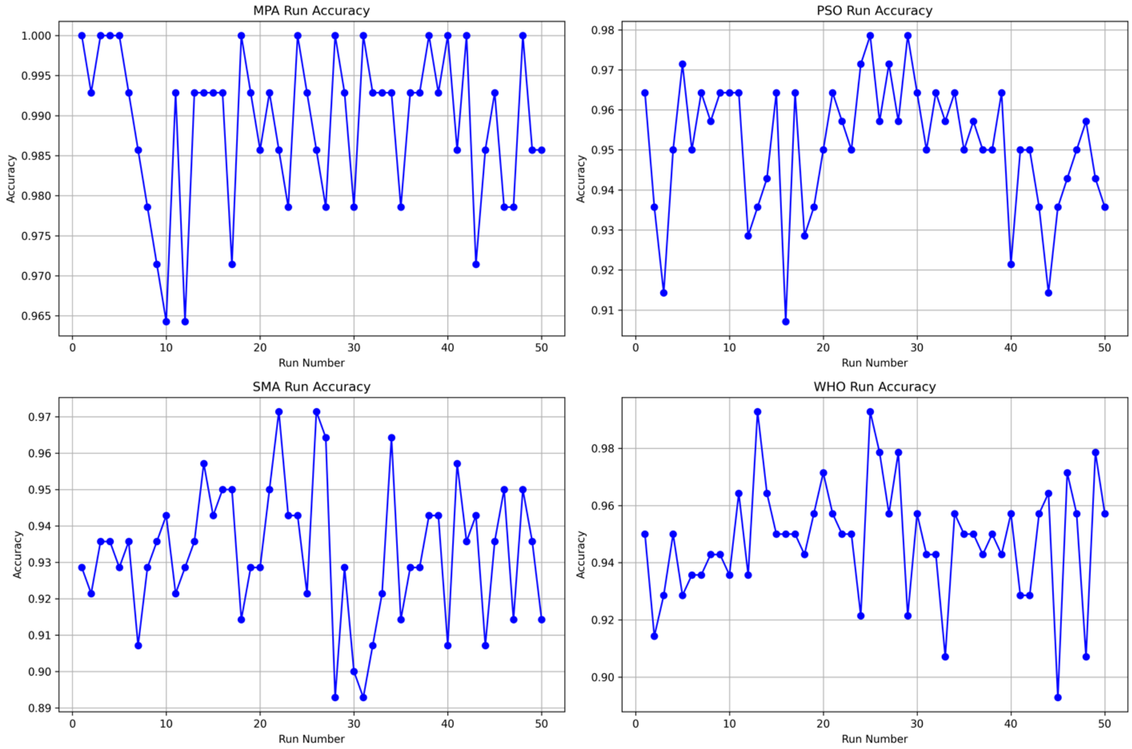


Figure 5. The Accuracy of 50 Iterations for the Metaheuristic Optimization Algorithm

Figure 5 displays the classification accuracy achieved by the kNN algorithm when trained and tested on feature sets selected using various metaheuristic optimization algorithms. The metrics presented are based on 50 iterations and reflect the kNN classifier's performance in evaluating different feature maps to identify those that yield the highest accuracy. By employing the features selected by each optimization algorithm to calculate performance metrics, the study aims to determine the most effective feature selection strategy and to identify the feature set with the highest overall classification accuracy.

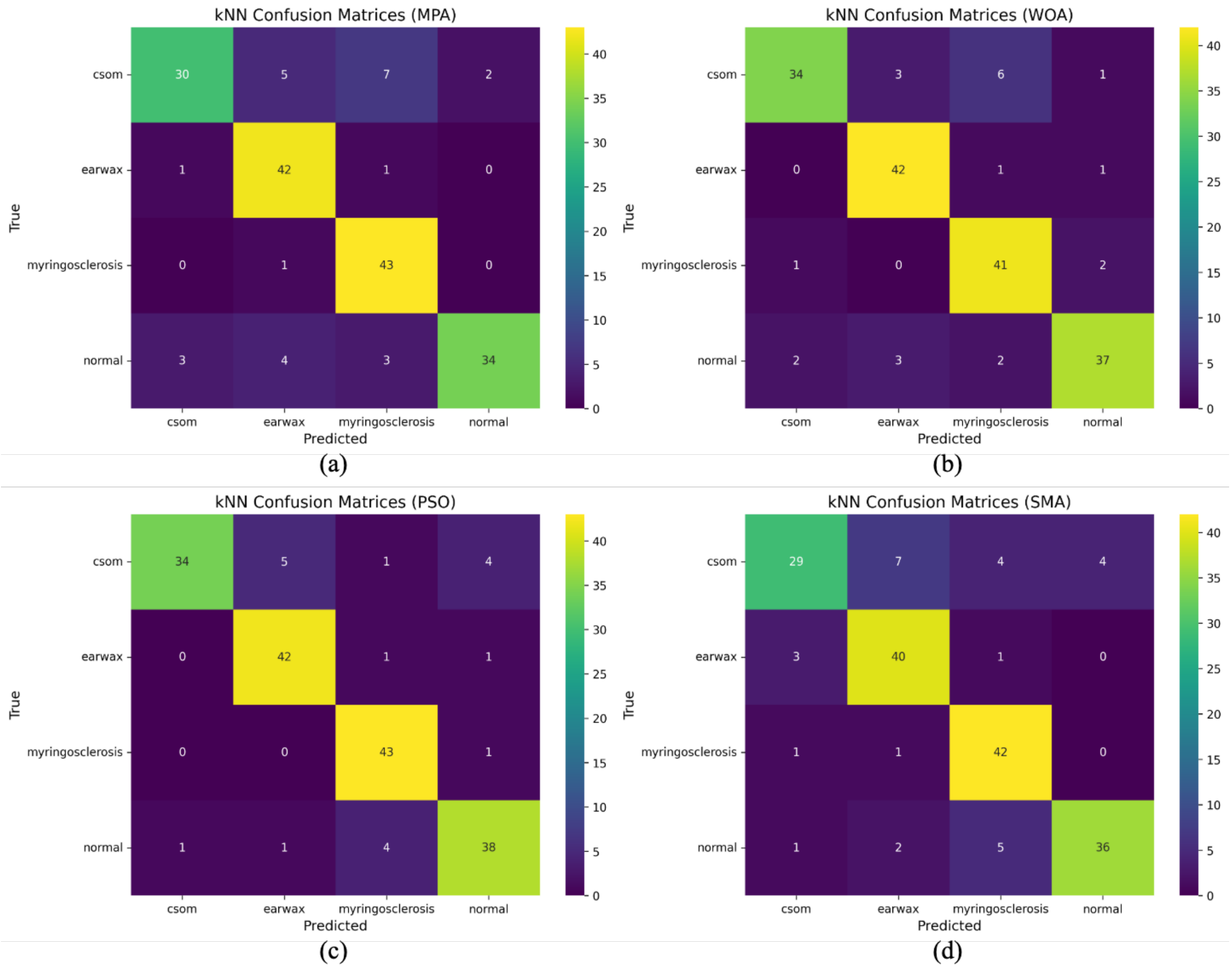


Figure 6. Confusion Matrices Obtained from the First Run with Corresponding Classification Accuracies: MPA - 84.66% (a), WOA - 87.50% (b), PSO - 89.20% (c), and SMA - 83.52% (d)

This study selected feature maps from HOG extraction using four metaheuristic optimization algorithms: WOA, MPA, SMA, and PSO. Table 4 shows the number of features selected by each. Figure 6 presents the confusion matrices obtained from the first run, where the kNN algorithm achieved a classification accuracy of 89.20% using features selected by the PSO algorithm. This initial run provides insight into the model's classification performance based on the selected feature subset. The experiment was repeated for 50 independent runs to ensure a robust evaluation. The highest accuracy achieved by the kNN algorithm across these runs, along with the corresponding run number and the number of selected features, is summarized in Table 4. The "Best Run" column in Table 4 represents the highest accuracy obtained from the classification results of the kNN algorithm using the features selected by the optimization algorithms. The selected features from this best-performing run were subsequently used to generate feature maps for the later stages of the study.

Table 4. Number of Features Selected by Metaheuristic Algorithms and Corresponding Iterations

Metaheuristic Algorithm	Best Run (From 50)	Number of Selected Feature
PSO	29	456
MPA	3	45
SMA	22	56
WOA	25	99

Table 5 displays the performance metrics from the feature selection process for each optimization algorithm, showing the classification accuracy of machine learning models trained on their selected feature sets.

Table 5. Summary of Classification Algorithm Performance (The Best-Performing Model is Highlighted in Bold)

Models	**MPA**			**SMA**			**PSO**			**WOA**		
	Acc	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**
kNN	0,88	0,887	0,88	0,863	0,878	0,863	0,88	0,889	0,88	**0,926**	**0,929**	**0,926**
SVM	0,892	0,895	0,892	0,875	0,88	0,875	0,914	0,914	0,914	0,897	0,899	0,897
SGDC	0,426	0,771	0,426	0,607	0,721	0,607	0,806	0,835	0,806	0,778	0,794	0,778
GNB	0,642	0,668	0,642	0,596	0,63	0,596	0,778	0,781	0,778	0,715	0,724	0,715
RF	0,789	0,79	0,789	0,801	0,803	0,801	0,795	0,793	0,795	0,778	0,785	0,778
DT	0,801	0,811	0,801	0,772	0,779	0,772	0,772	0,785	0,772	0,778	0,785	0,778

In this study, we evaluated the effectiveness of several metaheuristic optimization algorithms for feature selection to enhance machine learning model performance. The results indicated that the WOA algorithm was the most effective feature selection method, achieving the highest performance metrics, particularly when used with the kNN and RF classifiers. Additionally, the PSO algorithm demonstrated significant efficacy, especially in improving the performance of the SVM and SGDC classifiers. These findings underscore the crucial role of selecting appropriate feature selection algorithms to optimize the efficacy of machine learning models. The confusion matrix for the SVM model exhibiting the greatest performance is presented in Figure 7.

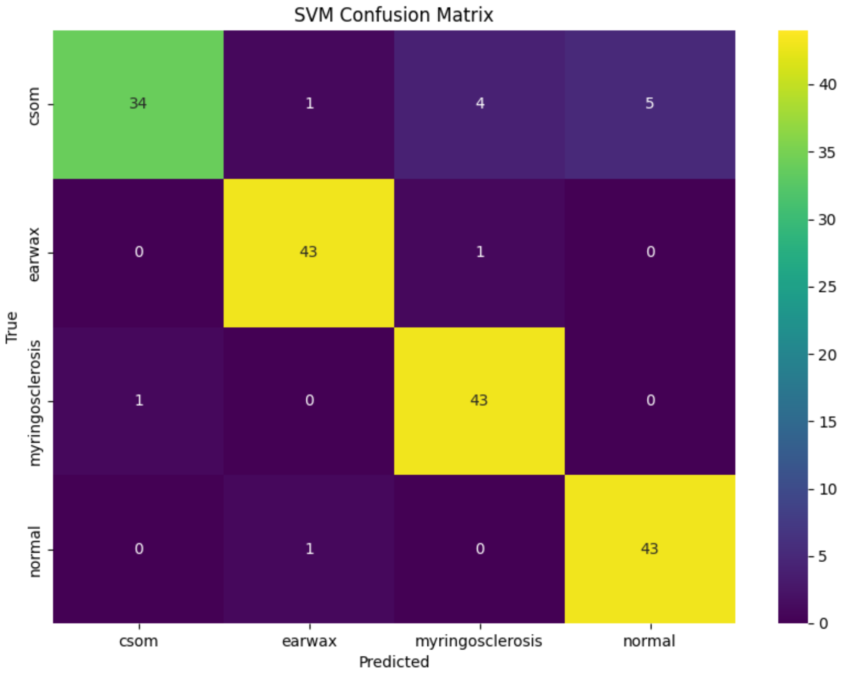


Figure 7. Confusion Matrix of the kNN Algorithm with Feature Selection

In the final experiment, a feature map was created by combining selected features from the optimization algorithms, resulting in 578 unique features. Performance metrics from this study are shown in Table 6.

Table 6. Performance Metrics of Algorithm for Combined Features (The Best-Performing Model Highlighted in Bold)

Models	Accuracy	Precision	Recall
kNN	0,892	0,895	0,892
SVM	**0,92**	**0,919**	**0,92**
SGDC	0,892	0,894	0,892
GNB	0,778	0,784	0,778
RF	0,789	0,79	0,789
DT	0,812	0,815	0,812

To better illustrate the comparative performance of the models, a graphical representation of accuracy values from Table 6 is provided in Figure 8.

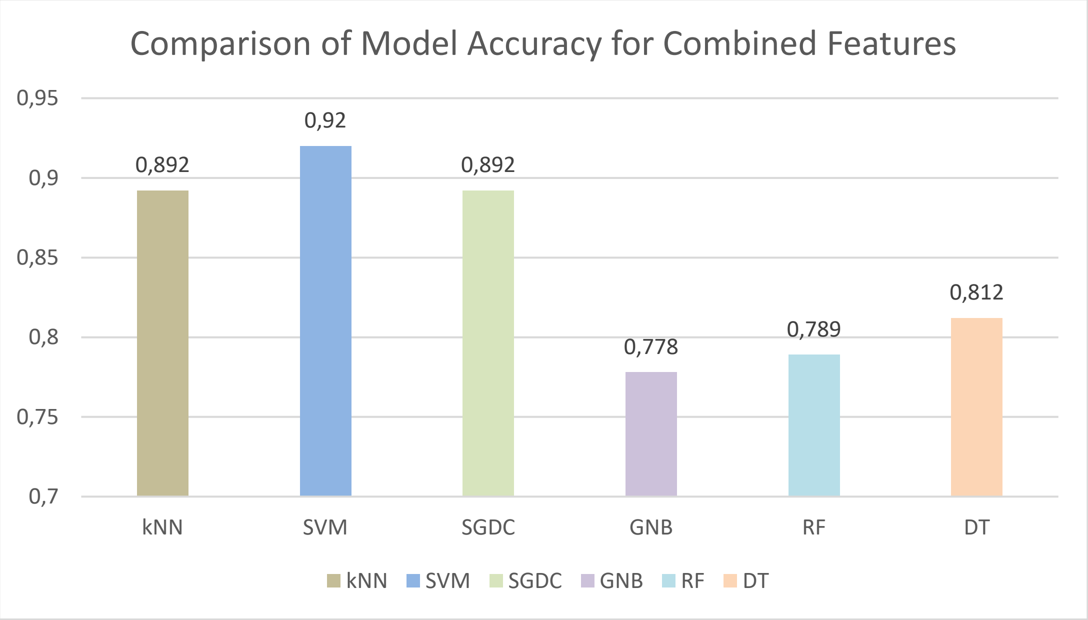


Figure 8. Accuracy Values of Different Models Based on the Combined Features, as Presented in Table 6

A comparative analysis was conducted on six machine learning algorithms to evaluate their performance in classifying ear diseases based on accuracy, precision, and recall metrics. The SVM model demonstrated superior performance, achieving an accuracy of 0.92, precision of 0.919, and recall of 0.92, indicating its effectiveness in minimizing false positives and false negatives. Both kNN and SGD performed well with accuracy scores of 0.892; however, kNN exhibited slightly higher precision at 0.895. GNB and RF showed moderate performance levels, while the DT model had the lowest accuracy at 0.812, highlighting its limited effectiveness for applications requiring high precision. In conclusion, the SVM model was the most effective classifier, outperforming the other algorithms evaluated.

In conclusion, the SVM model was the most effective classifier, outperforming other algorithms. Model selection should consider specific classification requirements. The confusion matrix for SVM is shown in Figure 9.

The model accurately classified 37 out of 44 cases of Chronic Suppurative Otitis Media (CSOM). Still, some misclassifications occurred, primarily labeling CSOM cases as normal or other pathologies, suggesting feature overlap between CSOM and normal cases. It demonstrated high proficiency in identifying earwax impaction, correctly classifying 43 out of 44 instances, and achieving perfect accuracy for all 44 instances of myringosclerosis. For normal cases, 38 out of 44 were correctly identified, with some misclassified as CSOM or myringosclerosis. These findings highlight the need to improve the model’s ability to distinguish between CSOM and normal cases to enhance classification accuracy and diagnostic reliability.

A comprehensive investigation assessed the efficacy of diverse feature selection techniques and machine learning algorithms for ear disease classification to identify optimal combinations for enhanced accuracy. Combining WOA with the kNN method yielded the highest classification accuracy of 92.6%, thereby demonstrating the efficacy of the WOA in improving the selection of features. Furthermore, the SVM demonstrated a noteworthy performance, attaining a 92% accuracy rate by utilizing features selected by multiple optimization algorithms. The SVM achieved 91.4% accuracy without explicit feature selection, illustrating its robustness. These findings highlight the crucial importance of strategic feature selection in enhancing the efficacy of machine learning for otitis media diagnosis.

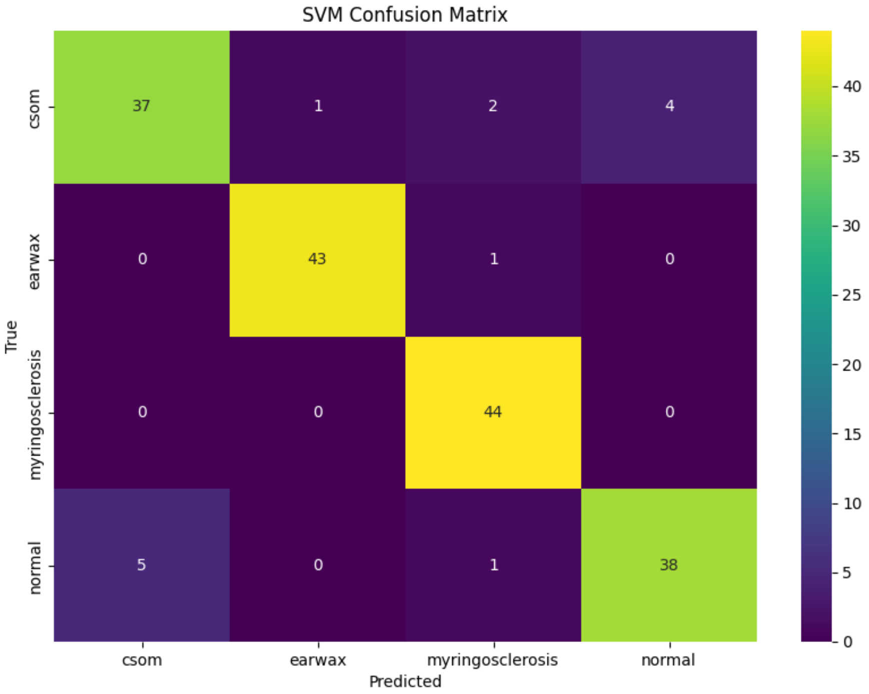


Figure 9. The Confusion Matrix of the SVM Algorithm

5. Discussion

The classification of visual representations of ear diseases has been the subject of investigation in several studies. Table 7 summarizes the datasets utilized, the number of diseases included, and the classification models developed in these studies.

The table highlights various approaches to ear disease classification using different machine learning models, datasets, and feature extraction techniques, showing significant variation in accuracy. Deep learning models generally performed well, with Wu et al. achieving 97.82% accuracy using Xception on a three-class dataset and Alhudhaif et al. improving on this with 98.26% accuracy by combining CNN, CBAM and other techniques on a four-class dataset. However, Sundgaard et al. reported a lower accuracy of 85% using InceptionV3, indicating that model performance can vary based on architecture and dataset. Mohammed et al. reached 100% accuracy by combining CNN with BiLSTM on a four-class dataset, showing hybrid models' potential for capturing spatial and temporal features. Similarly, Uçar et al. achieved 99.06% accuracy using SURF with VGG16, demonstrating that traditional feature extraction methods can enhance deep learning models. Other studies also showed strong results with lightweight architectures. Chen et al. achieved 97.6% accuracy with MobileNetV2 on a ten-class dataset, and Byun et al. reached 97.18% using ResNet18 with a Shuffle Attention Module, showing the benefit of attention mechanisms. However, Wang et al. reported lower accuracies of 86%, using the Faster R-CNN model, suggesting that model selection and dataset complexity can influence outcomes. Cloud-based platforms like Google Cloud AutoML, used by Livingstone et al., achieved 90.9% accuracy on a fourteen-class dataset, demonstrating the effectiveness of automated tools. Tsutsumi et al. achieved 90-91% accuracy using MobileNetV2 on datasets with varying class sizes, further showcasing the adaptability of lightweight models.

The study in this paper presents a computationally efficient approach combining HOG feature extraction with WOA for feature selection and kNN for classification, achieving 92.6% accuracy on a four-class ear imagery dataset. While slightly lower than some deep learning models, this result demonstrates that traditional models, combined with effective feature engineering, can offer competitive performance with lower computational requirements, making them valuable for resource-limited applications.

Table 7. The Previous Studies on Ear Disease Classification

Authors	Datasets	Class Size	Models	Accuracy (%)
Wu et al.[3]	Private	3	Xception	97,82
Sundgaard et al.[4]	Private	3	InceptionV3	85
Alhudhaif et al.[5]	Tympanic Membrane	4	CNN, CBAM, Residual block, and Hypercolumn Technique	98,26
Tran et al.[6]	Private	2	Multitask Joint Sparse Representation Algorithm	91,41
Myburgh et al.[7]	Private	5	NN	86,84
Mohammed et al.[8]	Ear Imagery	4	CNN-BiLSTM	100
Uçar et al.[27]	Ear Imagery	4	SURF + VGG16	99,06
Wang et al.[14]	Private	3	MESIC	90,1
Cha et al.[10]	Private	6	Models + Probability Selectors	93,67
Chen et al.[11]	Private	10	MobileNetV2	97,6
Sundgaard et al.[12]	Private	2	AlexNet Base Model	92,6
Tsutsumi et al.[13]	Mixed Dataset	2, 4	MobileNetV2	90, 91
Wang et al.[9]	Private	2	FasterRCNN + InceptionV3	86
Byun et al.[26]	Private	4	ResNet18+Shuffle Attention Module	97,18
Livingstone et al.[28]	Mixed Dataset	14	Google Cloud AutoML	90,9
Zeng et al.[15]	Private	2	DL model	89
Habib et al.[16]	Private	2	Vision Transformer	92
Pham et al.[17]	Private	2	EAR-UNet	92.9
This paper	Ear Imagery	4	HOG Feature Extraction + WOA Feature Selection + kNN Classification	92.6

6. Conclusions

This study introduces a novel approach to ear disease classification by employing computationally efficient machine learning models in conjunction with advanced feature engineering techniques. The research aims to achieve high classification accuracy by optimizing feature extraction and selection while reducing computational resource demands, facilitating practical clinical implementation and integrating the HOG for feature extraction and the WOA for feature selection to enhance model performance significantly. Notably, the kNN model combined with WOA-selected features achieved an impressive accuracy of 92.6%. The SVM classifier also performed well, attaining 92% accuracy with optimized features and 91.4% without explicit feature selection. These findings demonstrate that traditional machine learning models can perform comparably to computationally intensive deep learning architectures when paired with effective feature engineering. The results contribute to developing efficient diagnostic systems for ear diseases, paving the way for future enhancements in accuracy and accessibility.

References

[1] W. H. Organization, *Primary ear and hearing care training manual*. Genève, Switzerland: World Health Organization, 2023.

[2] Institute of Electrical and Electronics Engineers and Manav Rachna International Institute of Research and Studies, *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: trends, perspectives and prospects: COMITCON-2019: 14th-16th February 2019*.

[3] Z. Wu *et al.*, “Deep Learning for Classification of Pediatric Otitis Media,” *Laryngoscope*, vol. 131, no. 7, pp. E2344–E2351, Jul. 2021, doi: 10.1002/lary.29302.

[4] J. V. Sundgaard *et al.*, “Deep metric learning for otitis media classification,” *Med Image Anal*, vol. 71, Jul. 2021, doi:

10.1016/j.media.2021.102034.

[5] A. Alhudhaif, Z. Cömert, and K. Polat, "Otitis media detection using tympanic membrane images with a novel multi-class machine learning algorithm," *PeerJ Comput Sci*, vol. 7, pp. 1–22, 2021, doi: 10.7717/PEERJ-CS.405.

[6] T. T. Tran, T. Y. Fang, V. T. Pham, C. Lin, P. C. Wang, and M. T. Lo, "Development of an automatic diagnostic algorithm for pediatric otitis media," *Otology and Neurotology*, vol. 39, no. 8, pp. 1060–1065, 2018, doi: 10.1097/MAO.0000000000001897.

[7] H. C. Myburgh, S. Jose, D. W. Swanepoel, and C. Laurent, "Towards low cost automated smartphone- and cloud-based otitis media diagnosis," *Biomed Signal Process Control*, vol. 39, pp. 34–52, Jan. 2018, doi: 10.1016/j.bspc.2017.07.015.

[8] K. K. Mohammed, A. E. Hassanien, and H. M. Afify, "Classification of Ear Imagery Database using Bayesian Optimization based on CNN-LSTM Architecture," *J Digit Imaging*, vol. 35, no. 4, pp. 947–961, Aug. 2022, doi: 10.1007/s10278-022-00617-8.

[9] Y. M. Wang *et al.*, "Deep Learning in Automated Region Proposal and Diagnosis of Chronic Otitis Media Based on Computed Tomography," *Ear Hear*, pp. 669–677, 2020, doi: 10.1097/AUD.0000000000000794.

[10] D. Cha, C. Pae, S. B. Seong, J. Y. Choi, and H. J. Park, "Automated diagnosis of ear disease using ensemble deep learning with a big otoendoscopy image database," *EBioMedicine*, vol. 45, pp. 606–614, Jul. 2019, doi: 10.1016/j.ebiom.2019.06.050.

[11] Y.-C. Chen *et al.*, "Smartphone-based artificial intelligence using a transfer learning algorithm for the detection and diagnosis of middle ear diseases: A retrospective deep learning study," *EClinicalMedicine*, vol. 51, no. 201, p. 101543, 2022, doi: 10.1016/j.

[12] J. V. Sundgaard *et al.*, "A Deep Learning Approach for Detecting Otitis Media from Wideband Tympanometry Measurements," *IEEE J Biomed Health Inform*, vol. 26, no. 7, pp. 2974–2982, Jul. 2022, doi: 10.1109/JBHI.2022.3159263.

[13] K. Tsutsumi *et al.*, "A Web-Based Deep Learning Model for Automated Diagnosis of Otoscopic Images," *Otol Neurotol*, vol. 42, no. 9, pp. e1382–e1388, Oct. 2021, doi: 10.1097/MAO.0000000000003210.

[14] Z. Wang *et al.*, "Structure-aware deep learning for chronic middle ear disease," *Expert Syst Appl*, vol. 194, May 2022, doi: 10.1016/j.eswa.2022.116519.

[15] J. Zeng *et al.*, "A deep learning approach to the diagnosis of atelectasis and attic retraction pocket in otitis media with effusion using otoscopic images," *European Archives of Oto-Rhino-Laryngology*, vol. 280, no. 4, pp. 1621–1627, Apr. 2023, doi: 10.1007/s00405-022-07632-z.

[16] A. R. Habib *et al.*, "Evaluating the generalizability of deep learning image classification algorithms to detect middle ear disease using otoscopy," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-31921-0.

[17] V. T. Pham, T. T. Tran, P. C. Wang, P. Y. Chen, and M. T. Lo, "EAR-UNet: A deep learning-based approach for segmentation of tympanic membranes from otoscopic images," *Artif Intell Med*, vol. 115, May 2021, doi: 10.1016/j.artmed.2021.102065.

[18] Z. Cömert, A. Sbrollini, F. Demircan, and L. Burattini, "Computerized otoscopy image-based artificial intelligence model utilizing deep features provided by vision transformer, grid search optimization, and support vector machine for otitis media diagnosis," *Neural Comput Appl*, vol. 36, no. 36, pp. 23113–23129, Dec. 2024, doi: 10.1007/s00521-024-10457-y.

[19] F. Demircan, M. Ekinci, and Z. Cömert, "Enhancing intra-aural disease classification with attention-based deep learning models," *Neural Comput Appl*, Jan. 2025, doi: 10.1007/s00521-025-10990-4.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 886–893 vol. 1. doi: 10.1109/CVPR.2005.177.

[21] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015, doi: https://doi.org/10.1016/j.chemolab.2015.08.020.

[22] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[23] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert Syst Appl*, vol. 152, p. 113377, 2020, doi: https://doi.org/10.1016/j.eswa.2020.113377.

[24] T. T. Nguyen, H. J. Wang, T. K. Dao, J. S. Pan, J. H. Liu, and S. Weng, "An Improved Slime Mold Algorithm and its Application for Optimal Operation of Cascade Hydropower Stations," *IEEE Access*, vol. 8, pp. 226754–226772, 2020, doi: 10.1109/ACCESS.2020.3045975.

[25] B. ERGEN and M. E. SERTKAYA, "Alzheimer Hastalığının Erken Teşhisinin Çoklu Değişken Kullanarak Tespiti," *European Journal of Science and Technology*, Mar. 2022, doi: 10.31590/ejosat.1082297.

[26] H. Byun *et al.*, "An assistive role of a machine learning network in diagnosing middle ear diseases," *J Clin Med*, vol. 10, no. 15, Aug. 2021, doi: 10.3390/jcm10153198.

[27] M. Uçar, K. Akyol, Atila, and E. Uçar, "Classification of Different Tympanic Membrane Conditions Using Fused Deep Hypercolumn Features and Bidirectional LSTM," *IRBM*, vol. 43, no. 3, pp. 187–197, Jun. 2022, doi: 10.1016/j.irbm.2021.01.001.

[28] D. Livingstone and J. Chau, "Otoscopic diagnosis using computer vision: An automated machine learning approach," *Laryngoscope*, vol. 130, no. 6, pp. 1408–1413, Jun. 2020, doi: 10.1002/lary.28292.

Authors Contributions

Furkancan Demircan: Conceptualization, Methodology, Writing – Original Draft, Software
Eyüp Gedikli: Conceptualization, Methodology, Writing – Original Draft
Murat Ekinci: Methodology, Writing – review & editing
Zafer Cömert: Methodology, Writing – review & editing

Conflict of Interest Notice

Authors declare no conflict of interest related to this paper.

Ethical Approval and Informed Consent

This article contains no data or other information from studies or experimentation involving human or animal subjects.

Availability of data and material

Availability of data and materials: Data used in this study is openly available and free for research.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 76-88, 2025
DOI: 10.35377/saucis...1564138

RESEARCH ARTICLE

Turkish Stance Detection on Social Media Using BERT Models: A Case Study of Stray Animals Law

Selma Alav[1], Kristin S. Benli[2*]

[1] Software Engineering, Institute of Science, Fırat University, Elazığ, Türkiye, ror.org/05teb7b63
[2]Software Engineering, Faculty of Engineering and Natural Sciences, Üsküdar University, İstanbul, Türkiye, ror.org/02dzjmc73

Corresponding author:
Kristin S. Benli, Üsküdar University,
Software Engineering, Üsküdar, Türkiye
kristin.benli@uskudar.edu.tr

Article History:
Received: 09.10.2024
Revised: 12.01.2025
Accepted: 05.03.2025
Published Online: 27.03.2025

ABSTRACT
Recently, social media has transformed into an essential platform for information dissemination, allowing individuals to articulate their opinions and apprehensions on a wide array of subjects. Stance detection, which refers to the automated examination of text to ascertain the author's perspective regarding a specific proposition or subject, has emerged as a significant area of research. Within the scope of this study, a Turkish-labeled dataset was created to determine the stances of social media users regarding the Stray Animals Law and various pre-trained BERT models were fine-tuned on this dataset, four of which were Turkish (BERTurk 32k and 128k, ConvBERTurk and ConvBERTurk mC4), one multilingual (mBERT) and one base (BERT-Base). The BERTurk 128k model outperformed other BERT models by achieving a remarkable accuracy rate of 87.10%, along with 87.11% precision, 87.10% recall, and 87.10% F1 score. In conclusion, this study has accomplished a contribution in the limited field of Turkish stance detection research by comparing various BERT models in the context of Turkish texts that has not been previously undertaken to our knowledge. The promising results that were obtained from this and similar studies could contribute to the automatic extraction of public opinions, thereby assisting policymakers in formulating efficient policies.

Keywords: Stance detection, BERT, Text mining, Social media analysis, Turkish dataset

1. Introduction

The use of digital platforms like social media, news portals, and discussion forums has grown with the expansion of the Internet. People use these platforms to stay updated on topics such as politics, economy, sports, and social issues and share their thoughts and concerns. As a result, these platforms have become a crucial source of information. These data sources allow decision-makers to understand public opinion about the goals of interest. For example, those in decision-making positions seek to assess the reactions of the populace—whether in support or opposition to trending issues—and can utilize the information gathered from these platforms. However, manually analyzing the enormous amount of data is time-consuming and costly. This is where a growing interest in research on processing and analyzing textual data comes into play. Various fields focusing on automated content analysis include sentiment analysis, emotion recognition, sarcasm/irony detection, rumor detection, and fake news detection [1]. Stance detection, which is closely associated with sentiment analysis, has also gained attention as a recent area of research.

Stance detection is the automatic analysis of text to determine whether the author's perspective on a particular proposition or target is in favor, against, or neutral. The target of this analysis may include a variety of topics, such as individuals, organizations, government policies, movements, or products [2]. Based on application scenarios, stance detection can be divided into four subcategories: single-target, multi-target, cross-target and zero-shot [3].

In this study, we will focus on single target stance which can be expressed as Equation 1:

$$S_i = f(d_i, T) \ where \ S_i = \{Favor, Against, None\} \tag{1}$$

where T stands for the specified target, d_i represents the comment, which is made by user U_i, and S_i indicates the stance [3].

Research in Stance Detection has predominantly focused on English texts. In contrast, investigations into stance detection in Turkish are relatively scarce, and there is a lack of sufficient Turkish stance datasets for researchers interested in this topic.

Cite as: S. Alav and KS. Benli, "Turkish stance detection on social media using BERT models: A case study of stray animals law," *Sakarya University Journal of Computer and Information Sciences*, vol. 8, no. 1, pp. 76-88, 2025, doi: 10.35377/saucis...1564138

This study employs various Bidirectional Encoder Representations from Transformers (BERT) models, which are based on the transformer architecture and are recognized for their exceptional performance across various natural language processing applications, to analyze stances based on user comments concerning the Stray Animals Law written in Turkish. The lack of comparable studies that examine different BERT models, specifically in Turkish stance detection, increases the original value of our research. The study presents the following contributions:

- Development of a Turkish-Labeled Dataset: A novel and manually labeled Turkish dataset was developed, focusing on the Stray Animals Law as a target.
- Assessment of BERT Models' Performance: The study examines how BERT models (four Turkish, one multilingual and one base model) perform on the recently established Turkish dataset.
- Analysis of models: Examining the words and phrases most significantly contributes to models' decision-making using the LIME (Local Interpretable Model-Independent Explanations) method.

The remaining sections of this paper are organized in the following manner. Section 2 provides an overview of related studies, while Section 3 details the materials and methods that are employed in the research. The study's findings are presented in Section 4, and qualitative analysis is given in Section 5. Finally, Section 6 evaluates the results and discusses potential directions for future research.

2. Related Work

The following paragraphs present recent research studies that employed the BERT model for stance detection and the limited number of studies on the Turkish language.

Küçük and Can [4] conducted a study on stance detection from Turkish tweets about two prominent sports clubs in Turkey, Galatasaray (target-1) and Fenerbahçe (target-2). Researchers created three versions of the Turkish tweet dataset and labeled the stance information as either Favor or Against. They evaluated the performance of the Support Vector Machine (SVM) classifier using different feature sets such as unigrams, bigrams, hashtags, external links, emoticons and named entities. The results showed that a combination of unigrams, hashtags, and named entities performed better than other combinations of features. The study also revealed that named entities benefited from the Turkish stance detection analysis.

Ghosh et al. [5] examined seven stance detection models. They successfully implemented six of them and employed the publicly available code for the remaining model. Their initial focus was on assessing the reproducibility of these models. They applied them to two distinct datasets: 1) the SemEval dataset, which contains microblog data about topics such as atheism, climate change, the feminist movement, Hillary Clinton, and the legalization of abortion, and 2) the Multi Perspective Consumer Health Query (MPCHI) Data, which addresses five specific claims: MMR vaccination may lead to autism, E-cigarettes are less harmful than traditional cigarettes, Hormone Replacement Therapy is advisable for women after menopause, Vitamin C is effective in preventing the common cold, and exposure to sunlight can result in skin cancer. In addition to exploring current stance detection techniques, they utilized a pre-trained BERT (Large-Uncased) model. The results indicated that the performance metrics of the BERT model significantly surpassed those of other competing models. Furthermore, it was noted that the Convolutional Neural Network (CNN) model demonstrated effective performance with shorter tweets, specifically those containing 5 to 10 words, whereas BERT excelled with longer tweets.

Cotfas et al. [6] analyzed public stances towards COVID-19 vaccination using social media posts. Initially, they gathered a dataset of English tweets expressing various stances on COVID-19 vaccination. A subset of this dataset was manually labeled as Neutral, in Favor, or Against vaccination for training the stance classification model. Four different approaches were explored for text representation and classification: 1) Bag-of-Words with classical machine learning, 2) Word embeddings with classical machine learning, 3) Word embeddings with deep learning, and 4) BERT. The BERT model demonstrated better performance compared to other models.

Grimminger and Klinger [7] conducted a study focused on stance detection in political tweets, specifically examining whether supporters of Trump and Biden, who were the candidates in the 2020 US Presidential Elections, engaged in hateful and offensive speeches in their online communications. They developed an annotation task that combined the detection of hateful or offensive speech and stance detection. In addition to the established categories of Favorable and Against opinions, the analysis incorporated Mixed and Neutral positions and instances where a candidate was referenced without any expressed opinion. A pre-trained BERT base model was employed, revealing that the model successfully identified support for a candidate; however, determining an individual's opposition to a candidate proved more challenging.

Polat et al. [8] developed a Turkish stance dataset that contained comments related to different targets such as working from home, mask, e-book, vegan, e-cigarette, and vaccine. Comments were collected from the Ekşi Sözlük online forum, which was also used in this study, and tagged with Favor, Against and None stance classes. They conducted stance detection experiments using various machine learning methods, ensemble learning methods, and CNNs. Texts were represented using Bag-of-Words and Term Frequency-Inverse Document Frequency (TF-IDF) models for machine learning and ensemble learning techniques. For the deep learning approach, Word embedding was utilized for text representation. The results of the experiments showed that, despite target-based variations, the highest performances were observed with XGBoost and CNN models.

Küçük and Arıcı [9] introduced Turkish datasets related to COVID-19 vaccination for sentiment analysis and stance detection purposes. They gathered tweets from two distinct time frames (December and July). They categorized them into Favor, Against, and None stance classes—the feature set comprised unigrams, hashtag use, and emoticon use. Two different machine learning methods, SVM and Random Forest (RF), were utilized for training and testing, and the evaluation was conducted using a 10-fold cross-validation approach. The process resulted in relatively lower performance rates. The results showed that the SVM outperformed the RF in sentiment analysis and stance detection tasks. In their subsequent research, Küçük and Arıcı [10] assessed the capabilities of BERTurk and ChatGPT utilizing an enhanced version of the same dataset. The findings indicated that ChatGPT demonstrated better performance in stance detection, whereas BERTurk was more successful in sentiment analysis.

Zengin et al. [11] conducted a study on Turkish stance detection, examining how the performance of a fine-tuned BERT model was influenced by training data that was cross-target, cross-domain, and cross-lingual. They developed datasets encompassing football, health, economics, and politics. BERTurk was utilized to process Turkish data, while M-BERT was employed to process English data and cross-lingual experiments. The researchers reached multiple conclusions, notably that the integration of data for different targets within the same domain led to higher performance, manually annotated datasets outperformed automatically assessed datasets, the presence of training data that was aligned with the domain of the test data was a vital element in attaining higher classification performance and training exclusively on Turkish data produced better outcomes than training with a combination of Turkish and English data.

Arslan and Fırat [12] created a labeled dataset in Turkish to analyze user stances on the Russia-Ukraine conflict through social media posts. They categorized the tweets as either Favor or Against and experimented with machine-learning techniques using GloVe and FastText word embeddings. Additionally, they employed the 128K uncased BERT for the Turkish (BERTurk) model. They utilized both undersampling and oversampling techniques to address the imbalance in the dataset. The findings revealed that BERT-based models surpassed all other approaches, with LSTM and GRU yielding comparable results.

3. Materials and Methods

The phases of the study are illustrated in Figure 1. A dataset was developed comprising comments related to the "Stray Animals Law," which sparked considerable debate and commentary in Turkey, using the Ekşi Sözlük platform. The comments were scraped using BeautifulSoup, and the initially dirty data was cleaned. Subsequently, the data was manually categorized into "Favor" and "Against" labels. Next, six pre-trained BERT models were fine-tuned to detect stance in Turkish comments. Experiments were conducted on Google Colab and Drive. The models were assessed using four common metrics: accuracy, precision, recall, and F1 score. Furthermore, the words that had a considerable impact on the predictions made by the models were examined with LIME.

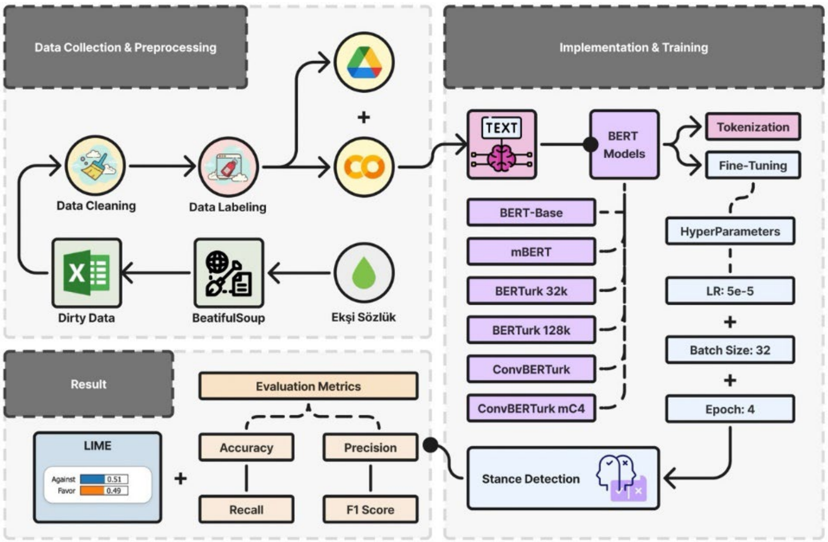


Figure 1. Workflow of the study

3.1. Data Collection and Pre-processing

While developing the dataset, comments on Ekşi Sözlük (a Turkish blog platform) were used. User comments were collected from 29 open headings regarding the "Stray Animals Law." The specific subject headings that were utilized in this study are listed below.

- "29 temmuz 2024 sokak hayvanları yasası değişikliği" (29 July 2024 stray animals law amendment)
- "sokak hayvanları yasasının komisyondan geçmesi" (passing the stray animals law from the commission)
- "sahipsiz hayvanlara yönelik kanun teklifi" (law proposal for homeless animals)
- "sokak hayvanları uyutulacak" (stray animals will be put to sleep).
- "yasayı geri çek" (withdraw the law)

The dataset contained comments where the authors explicitly expressed their viewpoints. The comments were extracted using the BeautifulSoup web scraping library and urllib.request package. In the data cleaning phase, all characters were converted to lowercase, while URLs, symbols, and punctuation marks were removed, and any unnecessary whitespaces were deleted. These operations were performed using string, re and nltk libraries. For the target "Stray Animals Law," each comment was manually labeled as either "Favor" or "Against." Table 1 presents sample target-comment pairs from the Turkish stance-tagged dataset.

Table 1. Target-Comment Pairs from the Turkish Stance Tagged Dataset

Target	**Comment**	**Stance**
Stray Animals Law	**(TR[a])** yetkili mercilerden resmi kurumlardan belediyelerden veterinerlerden avukatlardan bunu nasıl engelleyebileceğimize dair acil ve etkili bir yönlendirme bekliyoruz yalvarıyorum [13] **(EN[b])** we expect urgent and effective guidance from competent authorities, official institutions, municipalities, veterinarians, and lawyers on how we can prevent this. I beg	Against
Stray Animals Law	**(TR[a])** başıboş sokak köpeklerinin toplanması gerekiyor gereğinin yapılmasını bekliyoruz [14] **(EN[b])** stray dogs need to be collected; we expect the necessary action to be taken	Favor

[a]Turkish, [b]English

Upon completing the labeling process, there were 5000 comments in the dataset concerning the Stray Animals Law, with 2500 in Favor and 2500 Against. Table 2 presents an in-depth examination of the word count metrics for the comments, such as the shortest, average, and longest word counts within the comments related to the target.

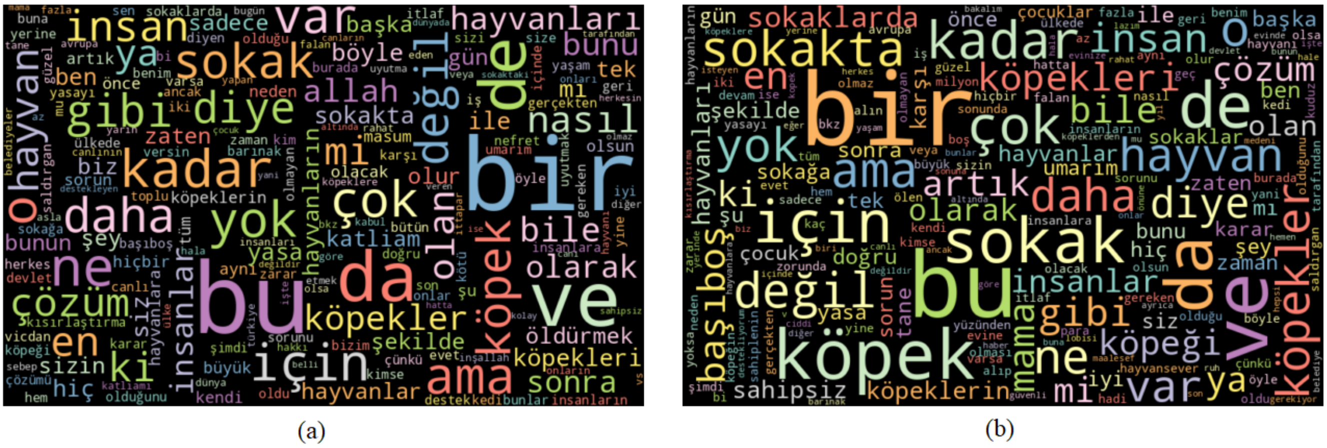
Table 2. Word Count Statistics for the Target "Stray Animals Law"

Stance	**Word Count**		
	Min	***Max***	***Average***
Against	1	163	41
Favor	1	170	36

The word cloud technique was employed to represent the words in the dataset visually. The visuals were generated utilizing the matplotlib, pandas, and wordcloud modules. Parts (a) and (b) of Figure 2 illustrate the prominent words in the against and favor classes, respectively. The font size of each word in the cloud indicates its frequency or significance within the text. Generally, a word that occurs more often in the text will be displayed larger in the word cloud. While the word clouds associated with the two labels exhibited a general similarity, a closer examination of the smaller font sizes revealed distinct differences. The word cloud for the Against class featured terms such as "katliam" (massacre), "öldürmek" (killing), "masum" (innocent) and "karşı" (against). In contrast, the Favor class included words like "sahiplenin" (adopt), "destekliyorum" (support), "kısırlaştırma" (neuter), "zarar" (harm), "kuduz" (rabies) and "saldırgan" (aggressive).

3.2. BERT Models

BERT, which was created by Google [15] for the field of natural language processing (NLP), represents a significant advancement in language modeling. The architecture of BERT is almost identical to a multilayer bidirectional transformer encoder, as found in research by Vaswani et al. [16]. Unlike its predecessors, this model analyzes text bi-directionally, allowing it to grasp the context more effectively by considering both the preceding and following text.



(a) (b)

Figure 2. Word Cloud Layouts (a) Against and (b) Favor

BERT has two pre-training tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The task of MLM is to predict the next word based on a given sequence of words. In each sequence, 15% of the words are randomly masked. The words “masked” are not always substituted with the actual [MASK] token. For instance, the i-th token

- replaced with the [MASK] token 80% of the time
- replaced with a random token 10% of the time
- remained unchanged 10% of the time

The model tries to predict the actual values of the masked words with the help of the remaining unmasked words in the sequence.

The NSP task involves understanding the relationship between two sentences or predicting the next sentence in a pair. In this task, the input typically comprises two sentences, A and B. In the %50 of the time, second sentence B directly follows the first sentence A. So, they are related to each other. In the other half, the second sentence, B, is randomly chosen from the dataset, and there is no connection with the first sentence, A.

This research involves experiments with six distinct BERT case models: four Turkish, one multilingual, and one base model. A brief introduction to them will be given in the following lines.

- ***BERT-Base:*** BERT-Base is the foundational model for all BERT variants, having undergone pre-training with an English dataset. It features a vocabulary comprising 30,000 tokens [15]. However, it is limited by a relatively small training dataset and a masking process that occurs only once, which can lead to errors since the masked data remains unchanged across all training epochs [17]. Consequently, it is designed to be fine-tuned and adaptable for further enhancements. Many new models have emerged from this pre-trained version [18].
- ***BERT Multilingual (mBERT):*** BERT includes two multilingual models: one that is cased and another that is uncased. The cased model has been trained in 104 languages, while the uncased model has been trained in 102 languages.
- ***BERTurk:*** BERTurk operates similarly to BERT, being a variant that has been pre-trained on a corpus of 35GB, which includes the Oscar Corpus, Opus Corpora, and a Wikipedia dump. There are variations of BERTurk models that differ in vocabulary size, offering options of 32k and 128k, with both cased and uncased versions available.
- ***ConvBERTurk:*** The ConvBERT Base model has been improved with additions to facilitate using BERT in less challenging tasks [19]. This model requires fewer parameters for operation. ConvBERTurk, on the other hand, is pre-trained in the Turkish language for over 1 million steps with a sequence length of 512, employing a methodology that differs from the conventional approach [20]. ConvBERTurk mC4, a variant of the ConvBERTurk model, is developed utilizing the C4 dataset.

3.3. Tokenization

BERT’s pre-trained models utilize a Tokenizer, and each model may require a specific one. For optimal performance of the BERT model, it is essential to divide the text into tokens, which are defined as small text segments. BERT models accept a maximum input of 512 tokens by default [21]. This number also includes special tokens such as [CLS], which is prepended for classification, and [SEP], which indicates where the token belongs [22, 23]. In this study, the text lengths within the dataset were adjusted to not exceed 512 tokens in the BERT models, and the same dataset was utilized across all models. Table 3 presents each model’s maximum, minimum, and average token counts.

Table 3. Token Counts of the Dataset as Per the Model Tokenization

Method	**Stance**	**Token Count**		
		Min	***Max***	***Average***
BERT-Base	Against	3	505	134
	Favor	3	497	119
mBERT	Against	2	350	94
	Favor	2	368	84
BERTurk 32k	Against	1	242	66
	Favor	2	268	58
BERTurk 128k	Against	1	210	55
	Favor	1	225	48
ConvBERTurk	Against	1	213	56
	Favor	1	268	50
ConvBERTurk mC4	Against	1	213	56
	Favor	1	268	50

Table 4 illustrates the tokenization and token count of the BERT models for the sample comment, "başıboş sokak köpeklerinin toplanması gerekiyor gereğinin yapılmasını bekliyoruz [24]" (stray dogs need to be collected we expect the necessary to be done). This comment comprises eight words and is categorized under the "Favor" class in the dataset.

Table 4. Tokenization of Bert Models in Detail

Method	**Tokenization**	**Token Count**
BERT-Base	['b', '##as', '##ı', '##bos', 'so', '##ka', '##k', 'k', '##ope', '##kle', '##rini', '##n', 'top', '##lan', '##mas', '##ı', 'g', '##ere', '##ki', '##yo', '##r', 'g', '##ere', '##gin', '##in', 'ya', '##p', '##ı', '##lma', '##s', '##ı', '##n', '##ı', 'be', '##k', '##li', '##yo', '##ru', '##z']	39
mBERT	['bas', '##ıb', '##os', 'sok', '##ak', 'kop', '##ek', '##lerinin', 'top', '##lanması', 'ger', '##eki', '##yor', 'ger', '##egi', '##nin', 'ya', '##pı', '##lması', '##nı', 'be', '##kli', '##yor', '##uz']	24
BERTurk 32k	['bası', '##bos', 'sokak', 'kop', '##ekler', '##inin', 'toplanması', 'gerekiyor', 'ger', '##eg', '##inin', 'yapılmasını', 'bekliyoruz']	13
BERTurk 128k	['bası', '##bos', 'sokak', 'kopek', '##lerinin', 'toplanması', 'gerekiyor', 'geregi', '##nin', 'yapılmasını', 'bekliyoruz']	11
ConvBERTurk	['başı', '##bo', '##ş', 'sokak', 'köpekler', '##inin', 'toplanması', 'gerekiyor', 'gereğini', '##n', 'yapılmasını', 'bekliyoruz']	12
ConvBERTurk mC4	['başı', '##bo', '##ş', 'sokak', 'köpekler', '##inin', 'toplanması', 'gerekiyor', 'gereğini', '##n', 'yapılmasını', 'bekliyoruz']	12

The "##" symbols represent the splitting of words into smaller pieces of the words. Each tokenization has a specific splitting strategy based on the natural language in which it is pre-trained. The BERT-Base model performed with the highest number of tokenizations, with the mBERT model following closely behind. The Turkish BERT models also demonstrated considerable effectiveness in segmenting sentences into tokens. Also, it was observed that ConvBERTurk and ConvBERTurk mC4 models tokenized the given samples in the same way. In their studies, Kaya and Tantuğ [25] stated that a tokenizer working in English made 2.5 times more word splitting when tokenizing in Turkish.

3.4. Evaluation Metrics

Performance metrics are derived from a confusion matrix, as outlined in Table 5. The prediction results can lead to one of four possible outcomes defined by the confusion matrix: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN represent the regions where the model makes accurate predictions, whereas FP and FN denote the regions where the model's predictions are inaccurate.

Table 5. Confusion Matrix

		Predicted Label	
		Positive (Favor)	Negative (Against)
Actual Label	Positive (Favor)	True Positive (TP)	False Negative (FN)
	Negative (Against)	False Positive (FP)	True Negative (TN)

Metrics, including accuracy rate, precision, recall, and F1 score, were employed to assess the performances of the models. The formulas for these performance metrics are presented in Equations 2 through 5.

$$Accuracy\ Rate = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

$$F1\ score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

4. Experimental Results

The experiments were conducted on a Tesla T4 GPU via Google Colab. Key hyperparameters, including learning rate, batch size, number of epochs, and learning optimizer, were set the same across all models. All models were trained for four epochs at a batch size of 32 and a learning rate of 5e-5 with the AdamW optimizer. A total of six different BERT case models, four Turkish, one multilingual and one base, were fine-tuned. As Grimminger and Klinger [7] did in their study, the dataset was partitioned into 80% for training and 20% for testing purposes. The TensorFlow and Transformers libraries were employed during the language model training process. All models were trained in approximately 24 minutes. The results are presented in Table 6, including accuracy, precision, recall, and F1 score.

It was observed that every model, except for BERT-Base, attained a success rate exceeding 80%. The best performance on the stance detection problem was obtained through the BERTurk 128k model with 87.10% accuracy, 87.11% precision, 87.10% recall and 87.10% F1 score. BERTurk32k emerged as the second-best model, attaining an accuracy rate of 86.50%.

The findings also revealed that the models accurately classified "Favor" examples more than "Against" examples, except for mBERT and ConvBERTurk. This result aligned with the research conducted by Grimminger and Klinger [7], who observed that the BERT-Base classifier achieved higher accuracy in identifying the "Favor" class. In contrast, detecting the "Against" class proved more challenging. Additionally, in the "Against" class, the BERT-Base model achieved the lowest accuracy rate of 68.80%, while the BERTurk 128k model attained the highest accuracy rate of 86.20%. In the "Favor" class, the mBERT model recorded the lowest accuracy rate at 83%, whereas the BERTurk32k model achieved the highest accuracy rate of 88.60%. Another notable finding is that the BERT base model outperformed the ConvBERTurk and ConvBERTurk mC4 models by achieving a success rate of 88% in classifying the Favor class.

Table 6. Stance Classification Results

Method	Stance	Accuracy	Precision	Recall	F1 score
BERT-Base	Against	68.80	85.15	68.80	76.11
	Favor	88.00	73.83	88.00	80.29
	Weighted Avg.	78.40	79.49	78.40	78.20
mBERT	Against	83.40	83.07	83.40	83.23
	Favor	83.00	83.33	83.00	83.17
	Weighted Avg.	83.20	83.20	83.20	83.20
BERTurk 32k	Against	84.40	88.10	84.40	86.21
	Favor	88.60	85.03	88.60	86.78
	Weighted Avg.	86.50	86.57	86.50	86.49
BERTurk 128k	Against	86.20	87.78	86.20	86.98
	Favor	88.00	86.44	88.00	87.22
	Weighted Avg.	**87.10**	**87.11**	**87.10**	**87.10**
ConvBERTurk	Against	84.80	84.13	84.80	84.46
	Favor	84.00	84.68	84.00	84.34
	Weighted Avg.	84.40	84.40	84.40	84.40
ConvBERTurk mC4	Against	82.60	85.16	82.60	83.86
	Favor	85.60	83.11	85.60	84.34
	Weighted Avg.	84.10	84.13	84.10	84.10

Research on Turkish stance detection has been carried out utilizing datasets from diverse fields, including sports, war, and vaccination. The studies incorporating the BERTurk model are presented alongside the current research in Table 7.

The summary table indicates that Küçük and Arıcı [10] attained an F1 score of 69.93% when focusing on Covid-19 vaccination, while Zengin et al. [11] recorded the highest F1 score of 68.90%, with Trabzonspor target under experimental conditions where the same target was utilized in both the training and testing datasets. Also, the F1 score values Arslan and Fırat [12] reported were 78.7% for the Russia target and 87.0% for the Ukraine target. In our study, both BERTurk 32k and 128k models attained notable levels of success in F1 scores and obtained values of 86.49% and 87.10%, respectively. Additionally, Arslan and Fırat [12] reported a classification accuracy of 78.4% when the target was Russia and 87.2% when the target was Ukraine in their research. The highest accuracy observed in the present study was 87.10%, which closely aligned with the findings of Arslan and Fırat.

5. Qualitative Analysis

LIME [26] was employed to gain insights into the predictive mechanisms of the models and to determine which words were most influential for the predicted labels. Table 8 illustrates the prediction results of each model for a sample text belonging to the "Favor" class. The text that is used for the assessment of this case is: "desteklediğim karar artık sokaklar güvenli olacak herkes için [27]" (the decision I support will now make the streets safe for everyone). Words highlighted in orange signify support (Favor) for the corresponding predicted label, while those in blue indicate opposition (Against). A darker color denotes a greater level of impact.

All models accurately classified the text that was designated as Favor. Moreover, it was observed that the word "desteklediğim" (I support), which indicated that the commenter supported this law, significantly influenced the decision-making processes of the models. Additionally, the word "güvenli" (safe) was another prominent word in decision-making processes and was associated with the "Favor" class.

Table 9 presents the prediction results of each model for a sample text categorized under the "Against" class. The text that is referenced for the evaluation of this case is: "başka çareler mümkünken en basit yola başvurulması üzücü [28]" (it is heartbreaking that the simplest method was used when other solutions were possible).

Table 7. Comparison with Similar Studies

Study	Dataset Size	Method	Results		
			Target	Accuracy	F1 score
Küçük and Arıcı [10]	830	BERTurk 32k	Covid-19 Vaccination	-	69.93%
Zengin et al. [11]	830 Galatasaray : 353 Fenerbahçe : 276 Beşiktaş : 100 Trabzonspor : 101	BERTurk 32k	Galatasaray	-	63.30%
			Fenerbahçe	-	58.90%
			Beşiktaş	-	57.60%
			Trabzonspor	-	68.90%
			*Single target, same target in the train and test set results		
Arslan and Fırat [12]	8215 Ukraine: 3264 Russia : 4951	BERTurk 128k	Russia	78.4%	78.7%
			Ukraine	87.2%	87.0%
Current Study	5000	BERTurk 32k	Stance Animals Law	86.50%	86.49%
		BERTurk 128k	Stance Animals Law	87.10%	87.10%

Among the models, only the BERTurk 128k model correctly associated the sample text with the Against class. The word "üzücü" (it is sad), which indicated the commentator's disagreement with this law, was understood by all models as a sign of support for the "Against" label. Nevertheless, this interpretation did not provide the models with enough information to make a correct final decision. Moreover, the word "çareler" (remedies) had a considerable impact on the models that performed "Favor" class prediction.

In addition, the decision-making processes of the BERTurk 128k model, which achieved the highest performance on the dataset, were analyzed under various cases, including TP, TN, FP, and FN, using LIME. The findings are illustrated in Figures 2-5.

- **Case 1: True Positive**

Figure 3 illustrates a case where both the actual and predicted class of the text is Favor. The sample text that is used during the examination of this case is: “sonuna kadar desteklediğim uygulamadır [29]” (this is an application that I fully support).

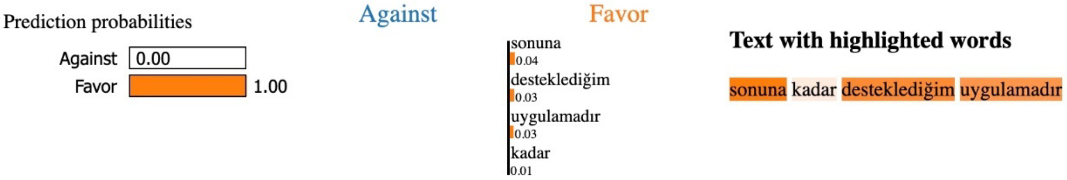
Table 8. LIME Results of Various Fine-Tuned BERT Models for Favor Class Sample Text

Method	Sample Text	Prediction Probabilities
BERT-Base	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.03 Favor 0.97
mBERT	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.03 Favor 0.97
BERTurk 32k	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.00 Favor 1.00
BERTurk 128k	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.00 Favor 1.00
ConvBERTurk	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.01 Favor 0.99
ConvBERTurk mC4	desteklediğim karar artık sokaklar güvenli olacak herkes için	Against 0.01 Favor 0.99

Table 9. LIME Results of Various Fine-Tuned BERT Models Against Class Sample Text

Method	Sample Text	Prediction Probabilities
BERT-Base	başka çareler mümkünken en basit yola başvurulması üzücü	Against 0.06 Favor 0.94
mBERT	başka çareler mümkünken en basit yola başvurulması üzücü	Against 0.34 Favor 0.66
BERTurk 32k	başka çareler mümkünken en basit yola başvurulması üzücü	Against 0.44 Favor 0.56
BERTurk 128k	başka çareler mümkünken en basit yola başvurulması üzücü	Against 1.00 Favor 0.00
ConvBERTurk	başka çareler mümkünken en basit yola başvurulması üzücü	Against 0.26 Favor 0.74
ConvBERTurk mC4	başka çareler mümkünken en basit yola başvurulması üzücü	Against 0.15 Favor 0.85

The word “desteklediğim” (I support), which reflected a supportive attitude, contributed to the classification of this text as Favor.

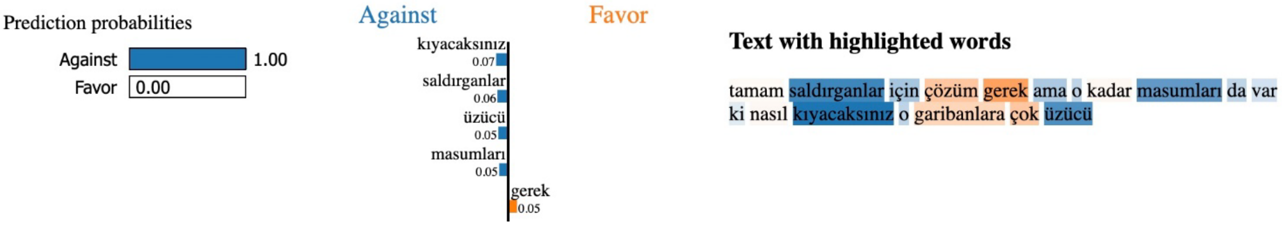


sonuna kadar desteklediğim uygulamadır

Figure 3. Result of Using LIME on Sample Text where the Actual Label and Predicted Label are Favor

- **Case 2: True Negative**

Figure 4 presents a case where both the actual and predicted class of the text is Against. The text that is utilized as a sample in the evaluation of this case is: "tamam saldırganlar için çözüm gerek ama o kadar masumları da var ki nasıl kıyacaksınız o garibanlara çok üzücü [30]" (ok a solution is needed for the aggressive ones but there are so many innocent ones how can you kill those poor ones it is very sad.). The words "kıyacaksınız" (you will kill), "üzücü" (sad) and "masumları" (innocents) were crucial in guiding the classifier towards an Against prediction.

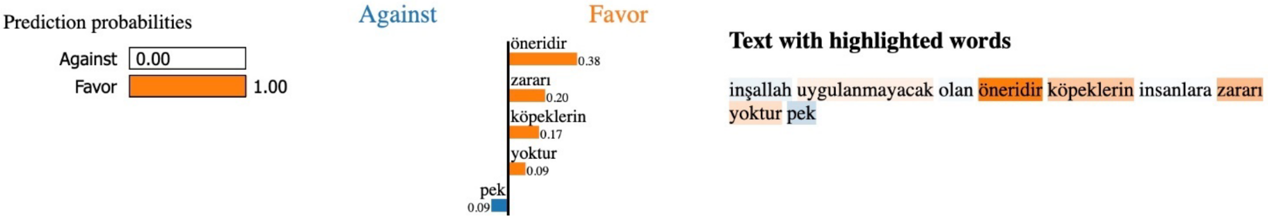


tamam saldırganlar için çözüm gerek ama o kadar masumları da var ki nasıl kıyacaksınız o garibanlara çok üzücü

Figure 4. Result of Using LIME on Sample Text where the Actual Label and Predicted Label are Against

- **Case 3: False Positive**

Figure 5 shows a case where the actual class of the sample text is Against, while the predicted class is Favor. The text that is employed for the analysis of this case is: "inşallah uygulanmayacak olan öneridir köpeklerin insanlara zararı yoktur pek [31]" (I hope this is a suggestion that will not be applied dogs do not harm people much). The words "zararı" (harm) and "yoktur" (there is no) played significant roles in leading the classifier to make a Favor prediction.

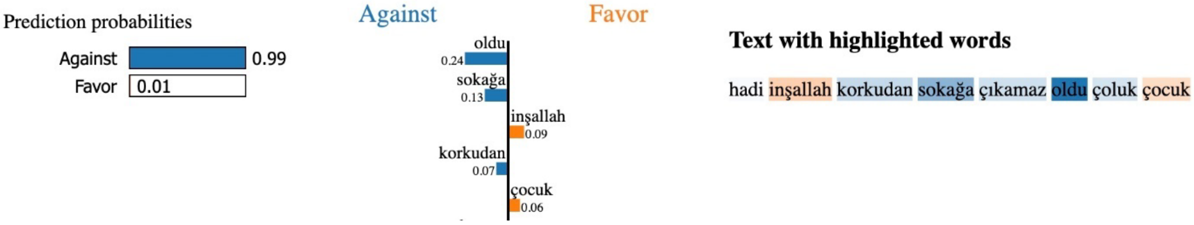


inşallah uygulanmayacak olan öneridir köpeklerin insanlara zararı yoktur pek

Figure 5. Result of Using LIME on Sample Text where the Actual Label is Against but the Predicted Label is Favor

- **Case 4: False Negative**

Figure 6 depicts a case where the actual class of the sample text is Favor, whereas the predicted class is Against. The text that is referenced during the assessment of this case is: "hadi inşallah korkudan sokağa çıkamaz oldu çoluk çocuk [32]" (hope so children cannot go out on the street because of fear). The words "oldu" (happened) and "korkudan" (fear) were significant in determining the classification of this text as Against.



hadi inşallah korkudan sokağa çıkamaz oldu çoluk çocuk

Figure 6. Result of Using LIME on Sample Text where the Actual Label is Favor but the Predicted Label is Against

6. Conclusion

In recent times, social media has evolved into a crucial medium for the distribution of information, enabling individuals to express their views and apprehensions on various subjects. Stance detection studies, which involve the automated examination of text to ascertain the author's stance on a specific proposition or topic, have attracted the attention of researchers and have become a significant focus of research.

This study aims to identify the most effective BERT model for the Turkish stance detection task. To our knowledge, this subject has not been addressed in existing literature. In this context, user comments concerning the Stray Animals Law, which has generated significant discussions and commentaries in Turkey, were collected from Ekşi Sözlük. A total of 5000 comments were manually classified, with 2500 labeled as "Favor" and 2500 as "Against." The performances of fine-tuned language models, including BERT-Base, mBERT, BERTurk 32k, BERTurk 128k, ConvBERTurk and ConvBERTurk mC4, were evaluated in terms of accuracy, precision, recall and F1 score. The experimental findings indicated that BERTurk 128k outperformed all other BERT models. It achieved an accuracy of 87.10%, precision of 87.11%, recall of 87.10% and F1 score of 87.10%. Additionally, most models were more successful in correctly predicting comments labeled as "Favor" than "Against."

This research presents a novel instance within the scarce Turkish stance detection studies. It highlights the effectiveness of BERT models, particularly those tailored for the Turkish language, in stance detection.

Research of this nature and others like it could play a crucial role in the automated extraction of public opinions, enabling governments to formulate policies on animal rights, vaccination, and climate change efficiently and cost-effectively.

As a future work, this study could be expanded to use alternative tokenization techniques and various strategies to improve the performance of BERT models on larger Turkish texts.

References

[1] D. Küçük and F. Can, "Stance detection: A survey," ACM Computing Surveys (CSUR), vol. 53, no.1, pp. 1-37, 2020.

[2] S. Mohammad et al., "Semeval-2016 task 6: Detecting stance in tweets," Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). 2016.

[3] R. Cao et al., "Stance detection for online public opinion awareness: An overview," International Journal of Intelligent Systems, vol. 37 pp. 11944-11965, 2022.

[4] D. Küçük and F. Can, "Stance detection on tweets: An svm-based approach," arXiv preprint arXiv:1803.08910, 2018.

[5] S. Ghosh et al., "Stance detection in web and social media: a comparative study," Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings 10. Springer International Publishing, 2019.

[6] L-A. Cotfas et al., "The longest month: analyzing COVID-19 vaccination opinions dynamics from tweets in the month following the first vaccine announcement," Ieee Access, vol. 9, pp. 33203-33223, 2021.

[7] L. Grimminger and R. Klinger, "Hate towards the political opponent: A Twitter corpus study of the 2020 US elections on the basis of offensive speech and stance detection," arXiv preprint arXiv:2103.01664, 2021.

[8] K.K. Polat, N. G. Bayazıt, and O. T. Yıldız, "Türkçe duruş tespit analizi," Avrupa Bilim ve Teknoloji Dergisi vol. 23, pp.99-107, 2021

[9] D. Küçük, and N. Arıcı, "Sentiment analysis and stance detection in Turkish tweets about COVID-19 vaccination," Handbook of research on opinion mining and text analytics on literary works and social media. IGI Global, 371-387, 2022.

[10] D. Küçük, and N. Arıcı, "Deep learning-based sentiment and stance analysis of Tweets about Vaccination," International Journal on Semantic Web and Information Systems (IJSWIS), vol. 19, no.1, pp.1-18, 2023.

[11] M. S. Zengin, B. U. Yenisey, and M. Kutlu, "Exploring the impact of training datasets on Turkish stance detection," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 31, no.7, pp.1206-1222, 2023.

[12] S. Arslan and E. Fırat, "Stance Detection on Short Turkish Text: A Case Study of Russia-Ukraine War," Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi, vol. 24, no. 3, pp. 602-619, 2024.

[13] Ekşi Sözlük, "29 temmuz 2024 sokak hayvanları yasası değişikliği," Available at https://eksisozluk.com/entry/166760652 (Accessed Date: 08.08.2024)

[14] Ekşi Sözlük, "29 temmuz 2024 sokak hayvanları yasası değişikliği," Available at https://eksisozluk.com/entry/166761830 (Accessed Date: 08.08.2024)

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[16] A. Vaswani et al., "Attention is all you need." Advances in neural information processing systems, 30, 2017.

[17] P. Savci and B. Das, "Comparison of pre-trained language models in terms of carbon emissions, time and accuracy in multi-label text classification using AutoML," Heliyon, vol. 9, issue. 5, 2023.

[18] Hugging Face, "Models," Available at https://huggingface.co/models?other=bert (Accessed Date: 12.08.2024)

[19] Z-H. Jiang et al., “Convbert: Improving bert with span-based dynamic convolution,” Advances in Neural Information Processing Systems, vol. 33, pp. 12837-12848, 2020.

[20] Hugging Face, “dbmdz Turkish ConvBERT model,” Available at https://huggingface.co/dbmdz/convbert-base-turkish-cased (Accessed Date: 12.08.2024)

[21] X. Chen, P. Cong and S. Lv, “A Long-Text Classification Method of Chinese News Based on BERT and CNN,” Ieee Access, vol. 10, pp. 34046-34057, 2022.

[22] Medium, “Handle Long Text Corpus for Bert Model,” Available at https://medium.com/@priyatoshanand/handle-long-text-corpus-for-bert-model-3c85248214aa (Accessed Date: 13.08.2024)

[23] Medium, “Fine-tuning BERT model for arbitrarily long texts, Part 1,” Available at https://medium.com/mim-solutions-blog/fine-tuning-bert-model-for-arbitrarily-long-texts-part-1-299f1533b976 (Accessed Date: 13.08.2024)

[24] Ekşi Sözlük, “29 temmuz 2024 sokak hayvanları yasası değişikliği”, Available at https://eksisozluk.com/entry/166761830 (Accessed Date: 29.09.2024)

[25] Y. B. Kaya and A. C. Tantuğ, “Effect of Tokenization Granularity for Turkish Large Language Models,” Journal of Intelligent Systems with Applications, vol. 21, 2024.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin, “ Why should I trust you?” Explaining the predictions of any classifier, In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135-1144, 2016.

[27] Ekşi Sözlük, “sahipsiz hayvanlara yönelik kanun teklifi,” Available at https://eksisozluk.com/entry/166275172 (Accessed Date: 23.09.2024)

[28] Ekşi Sözlük, “sokak hayvanları uyutulacak,” Available at https://eksisozluk.com/entry/164628545 (Accessed Date: 23.09.2024)

[29] Ekşi Sözlük, “sokak hayvanları uyutulacak,” Available at https://eksisozluk.com/entry/164641295 (Accessed Date: 11.01.2025)

[30] Ekşi Sözlük, “sahipsiz hayvanlara yönelik kanun teklifi,” Available at https://eksisozluk.com/entry/166318051 (Accessed Date: 11.01.2025)

[31] Ekşi Sözlük, “14 günde sahiplenilmeyen köpeklerin uyutulması,” Available at https://eksisozluk.com/entry/159777438 (Accessed Date: 11.01.2025)

[32] Ekşi Sözlük, “sokak hayvanları uyutulacak”, Available at https://eksisozluk.com/entry/164651341 (Accessed Date: 11.01.2025)

Author(s) Contributions

Selma Alav: Data curation, Investigation, Software, Visualization, Writing-Original draft preparation

Kristin S. Benli: Conceptualization, Methodology, Investigation, Software, Visualization, Writing-Original draft preparation.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

The data that support the findings of this study are available from the authors upon reasonable request.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 89-111, 2025
DOI: 10.35377/saucis...1560377

RESEARCH ARTICLE

Comparative Analysis of Data Visualization and Deep Learning Models in Air Quality Forecasting

Damla Mengus[1*], **Bihter Daş**[2]

[1]Marmara University, Computer Engineering, İstanbul, Türkiye, ror.org/02kswqa67
[2]Fırat University, Elazığ, Türkiye, ror.org/05teb7b63

Corresponding author:
Damla Mengus
Computer Engineering,
Marmara University, İstanbul, Türkiye
damla.mengus@marmara.edu.tr

Article History:
Received: 03.10.2024
Revised: 13.12.2024
Accepted: 09.01.2025
Published Online: 27.03.2025

ABSTRACT
This study utilizes air pollution data from the Continuous Monitoring Center of the Ministry of Environment, Urbanization, and Climate Change in Turkey to predict various pollutants using three advanced deep learning approaches: LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and RNN (Recurrent Neural Network). Missing data in the dataset were imputed using the K-Nearest Neighbor (K-NN) algorithm to ensure data completeness. Furthermore, a data fusion technique was applied to integrate multiple pollutant enhancing the richness and reliability of the input features for modeling. The increasing air pollution issue, driven by factors such as population growth, urbanization, and industrial development, is a major environmental concern. The study evaluates these models to estimate pollutant concentrations and selects the most accurate, RNN, for forecasting air pollution over the next three years. Each prediction was assessed using performance metrics such as MAE, RMSE, and R² to ensure robust model evaluation. Visualization of the data and forecast results was achieved through methods like Box Plots, Violin Plots, and Point Scatter Graphs, making air quality information more accessible to general audiences. In terms of model performance, CNN achieved an R² of 0.88 for PM10 and 0.93 for SO2, while LSTM demonstrated an R² of 0.94 for PM10 and 0.95 for SO2. However, RNN emerged as the most accurate model, achieving an R² of 0.97 for both PM10 and SO2 forecasts. This model allows for forecasts of pollutant levels over a three-year period. The findings indicate that predictive modeling, combined with data fusion and visualization techniques, could significantly contribute to mitigating future uncertainties and enhance the comprehension of air quality patterns for non-expert audiences.

Keywords: Data prediction, CNN, RNN, LSTM, Data visualization

1. Introduction

Air pollution occurs when harmful substances are present in the atmosphere at levels that can negatively impact human health and the environment's equilibrium. Factors such as increasing population, rapid urbanization, and accelerated industrial growth contribute significantly to the degradation of air quality [1-3]. These trends have led to a rise in fossil fuel consumption, increased vehicular traffic, and expanded industrial activities. Major pollutants that contribute to air pollution include gases such as carbon monoxide (CO), nitrogen dioxide (NO2), sulfur dioxide (SO2), ozone (O3), and particulate matter (PM). Additionally, nitrogen oxides (NOx), particularly the interaction of nitric oxide (NO) and nitrogen dioxide (NO2), are central to atmospheric chemical reactions that further reduce air quality [4-6]. Particulate matter, especially PM10 and PM2.5, is among the most concerning pollutants due to its severe impact on human health. These particles are so small that they can bypass respiratory defenses, penetrating deep into the lungs, where prolonged exposure can lead to severe health issues, including cancer. When inhaled, SO2 is another dangerous pollutant that irritates the respiratory system and poses a considerable health risk [7-9]. Air quality is typically assessed using an index that reflects the level of pollution based on meteorological conditions. However, the air quality index may not always capture the true impact of air pollution due to measurement inaccuracies or insufficient sensor data, which can result in delayed responses. Although the air quality index is widely used to evaluate the harmful effects of pollution on public health, the sensitivity and precision of the measurements need improvement. Moreover, current measurements are often performed daily and cannot provide predictive insights, making it difficult to implement timely preventive measures. As the measured air quality index rises, it indicates worsening air conditions with increasing risks to human health. The index categorizes air quality from 0-50 as "good," 51-100 as "moderate," 101-150 as "sensitive," 151-200 as "unhealthy," 201-300 as "very unhealthy," and above 300 as "hazardous." Once the index surpasses 151, the likelihood of health problems increases, and outdoor activity becomes risky [10,11]. Figure 1 shows the air quality index.

Cite as: D. Mengus and B. Das, "Comparative analysis of data visualization and deep learning models in air quality forecasting," *Sakarya University Journal of Computer and Information Sciences*, vol. 8 no. 1, pp. 89-111, 2025, doi: 10.35377/saucis...1560377

In this study, missing data is systematically filled to address the issues of sensitivity and accuracy in air quality measurements, and comparative estimations are performed using three distinct deep learning approaches: RNN, CNN, and LSTM. The analysis identifies the model that provides the most accurate predictions, and this model is then used to forecast air quality over the next three years. The results are presented through visualizations to enhance clarity. One of the gaps in the current literature is the lack of comparative studies using these methods and the absence of hybrid models. This study seeks to bridge that gap by offering more visualization-based insights. The proposed solution provides a viable method to overcome existing air quality measurement limitations and generate future forecasts. Air quality data is collected from continuous monitoring stations (CMS) at 39 locations across Istanbul, making it highly effective for tracking air quality trends throughout the city and generating data-driven predictions. The location of Başakşehir district is depicted in Figure 2.

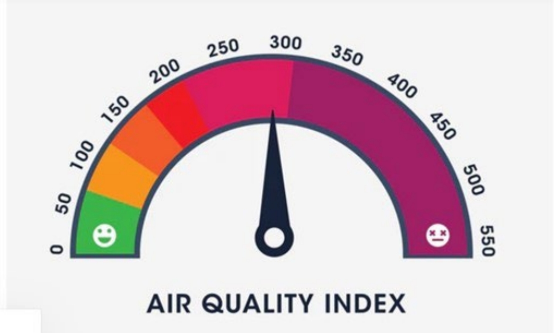


Figure 1. Air Quality Index

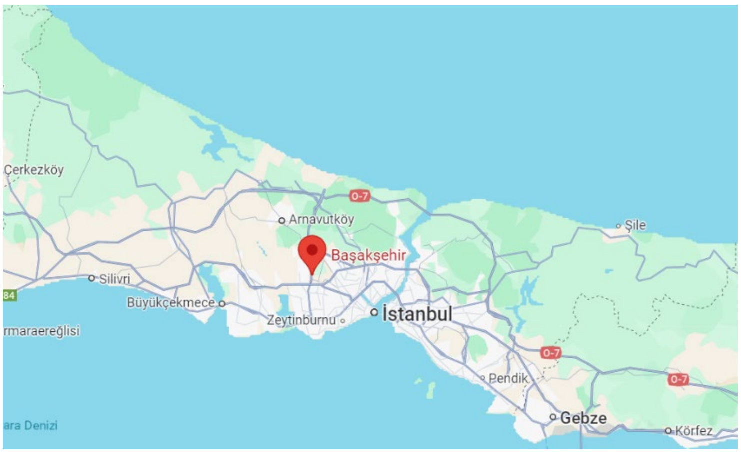


Figure 2. Location of Başakşehir District

1.1. Main Contributions

The key contributions of this study are as follows:

- LSTM, RNN, and CNN methods were used to predict air quality data, and these three methods were analyzed comparatively. As a result of the comparison, the method with the highest accuracy was determined, and air quality predictions for the next 3 years were made using this method. This study provides a comprehensive approach to visualizing large data sets and comparing the performances of different deep learning models, making significant contributions to environmental data analysis and prediction processes.
- Future air quality prediction will be made with RNN, which was determined as the best method. Thus, it is aimed at obtaining longer-term and more accurate predictions.
- By using Box Plots, Violin Plots, and Point Scatter Plots as visualization methods, non-experts helped interpret large data. This study aims to increase the effectiveness of data visualization and prediction methods. Thus, even non-expert users can interpret complex data sets.

1.2. Related Studies

Many methodologies and techniques have been applied in the literature to predict and visualize air quality. Deep learning and machine learning methods are especially prominent in air quality prediction. These methods have great potential in predicting the concentrations of various air pollutants and monitoring air quality. Below, a summary of important studies in this field is presented. In the article written by C.-J. Huang et al. aim to develop a model that combines deep learning methods LSTM and CNN networks to predict PM2.5 particulate matter. The model is applied to monitor and predict air quality in smart cities. It is a valuable study highlighting the effectiveness of hybrid deep learning models in air quality prediction for smart cities [12]. Similarly, R. Janarthanan et al. used deep learning approaches to predict the air quality index in a major metropolitan area. Their work highlights how deep learning can improve air quality forecasting in large urban settings [13]. S. Masmoudi et al. propose a machine learning framework that includes multi-objective regression and feature selection methods to predict the concentration of multiple air pollutants. This paper is an exemplary study that analyzes the application of multi-objective regression techniques in air pollution prediction [14]. Another study uses an attention mechanism model to predict air pollutant levels in the Yangtze River Delta during frequent heatwaves, improving accuracy in extreme weather conditions. The study provides a valuable approach by integrating an attention mechanism, offering improved air pollutant predictions during heatwaves, which could enhance environmental monitoring in climate-sensitive regions [15]. Another paper proposes a novel hybrid optimization model that considers other air pollutants and meteorological conditions to predict PM2.5 concentrations. It presents an approach highlighting hybrid models' importance in air pollution prediction [16]. P. A. Rani et al. developed a novel artificial intelligence algorithm that analyzes the levels of air pollutants to predict air quality in Tamil Nadu. This study investigates the innovative use of artificial intelligence techniques in air quality prediction [17]. Q. Wu and H. Lin have developed a new, highly efficient hybrid model for predicting the daily air quality index, incorporating air pollution factors. factors into account. This study is important research examining the effectiveness of hybrid model strategies in air quality prediction [18]. B. Zhang et al. predict the dispersion trends of air pollutants in specific seasons using deep learning. They examine the example of Northern China. They investigate the potential of deep learning applications in targeted seasonal analysis [19]. In another study, J. Luo and Y. Gong consider the prediction of air pollutants using a combination of ARIMA, Whale Optimization Algorithm (WOA), and LSTM models. It is aimed to increase the accuracy of air pollution prediction by integrating multiple algorithms [20]. G. I. Drewil and R. J. Al-Bahadili perform air pollution prediction using a combination of LSTM deep learning and metaheuristic algorithms. It is innovative research examining the integration of various algorithms in air pollution prediction [21]. Another study analyzes the thermal comfort indexes specific to the summer months in Istanbul and evaluates the effects of environmental variables. A comprehensive study has examined the effects of local thermal comfort and climate change [22]. In their paper, A. Kshirsagar and M. Shah deeply analyze using neural networks, regression, and hybrid models in air quality prediction. They present a comprehensive review of air quality prediction by comparing different model structures [23]. Another study reviews the existing literature on deep learning techniques for air pollutant concentration prediction and evaluates the application areas of these techniques. It has been a comprehensive review covering the advanced applications of deep learning techniques in air pollution prediction [24.] T. D. Akinosho et al. This paper uses a deep learning-based multi-objective regression model for traffic-related air pollution prediction. The model aims to improve its performance by simultaneously making predictions for multiple pollutants. This study demonstrates the potential of multi-objective deep learning models in traffic-related air pollution prediction [25]. M. Yılmaz et al. Their study detected the characteristics of heat waves in Istanbul and performed a regional analysis. The study provides a comprehensive analysis to understand the effects of climate change on cities. The study has been important research contributing to investigating urban climate change effects at the local level [26]. J. González-Pardo et al. This paper uses data mining models to predict changes in air pollutant levels in urban traffic areas in Spain during COVID-19 lockdown measures. It provides a valuable case study analyzing the changes in air pollution levels during the pandemic [27]. X. Shi et al. This paper evaluates the potential and additional benefits of reducing emissions of CO2 and other air pollutants from mobile sources in Shanghai. It is a case study investigating emission reduction strategies' environmental and economic benefits [28]. Another paper uses machine learning methods to predict air quality parameters and determine their spatial distribution. It emphasizes the importance of machine learning applications for spatial analysis in air quality prediction [29]. S. Ünaldi and N. Yalçin present a case study using machine learning methods for air pollution prediction in Başakşehir, Istanbul. It is a study examining the applicability of machine learning in local air quality prediction [30]. In the article, P. Aksak et al. examine the urban heat island effects and related climate parameters in Istanbul using remote sensing techniques. The study demonstrates the use of remote sensing data to understand the effects of the urban heat island phenomenon [31].

Y.-C. Lin et al.'s study uses Bayesian networks and deep learning models to evaluate the effects of meteorological and traffic factors on air pollutants. It analyzes using Bayesian networks and deep learning methods to model the factors affecting air quality [32]. I. H. Fong et al.'s study uses transfer learning and recurrent neural networks (RNN) to predict the concentration levels of air pollutants. It has been valuable research examining the role of transfer learning and RNN models in air pollution prediction [33]. Another study explores using a neural transfer learning approach to improve the prediction of various air pollutants. The model aims to enhance accuracy and efficiency in air quality forecasting by transferring knowledge across different pollutants. The application of transfer learning in air pollution prediction presents a promising method for improving the accuracy of forecasts across multiple pollutants, potentially leading to better environmental management strategies [34]. Another paper uses a one-dimensional multi-scale CNN-LSTM model that considers spatial-temporal features to predict the concentrations of air pollutants in the case of China. It investigates deep learning approaches that integrate spatial and temporal factors in air quality prediction [35]. An alternative study utilizes the LSTM-based neural network model for predicting air pollutant levels. LSTM is recognized for its capability to identify and learn long-term dependencies in time series data. The study evaluates the accuracy and effectiveness of LSTM for air quality prediction and highlights the importance of predictive modeling for air pollution management strategies. It has been a comprehensive study that demonstrates the accuracy and advantages of LSTM models in air pollution prediction [36]. B. Das et al. Their paper discusses using deep learning methods to predict air pollutants in a large metropolitan city. The study applies various deep learning algorithms to analyze air quality dynamics and improve prediction accuracy. It has been an important and valuable study that examines the performance of deep learning methods in air quality assessment in large cities [37]. Another study applies a deep learning-based recurrent neural network (RNN) to predict air pollutants like SO2 and PM10 levels in industrial cities like Sakarya. This research underscores the critical role of predicting air pollution in industrial zones and highlights the capabilities of RNN models for this purpose. The study demonstrates the efficiency and potential of RNN models in forecasting air quality in industrial settings [38]. Furthermore, J. Yang et al. explore spatial and temporal predictions of airborne particle (PM) levels by incorporating data related to traffic and weather conditions. The study aims to improve air quality forecasting by combining different data sources. It highlights the advantages of integrating traffic and weather data to enhance the precision of air pollution predictions [39]. Another study investigates how short- and long-term exposure to air pollutants affects plant phenology and leaf traits. It evaluates the ecological consequences of air pollution on plant health and development. The study provides a review that examines both short- and long-term effects of air pollution on plant biology [40]. In their paper, S. A. Ajayi et al. examine the effects of traffic mobility measures on vehicle emissions under heterogeneous traffic conditions in Lagos. The study analyzes the environmental impacts of traffic regulations and suggests improvement strategies. A significant study has evaluated the emission reduction potential of urban traffic regulations [41]. S. Arslankaya et al. applied machine and deep learning techniques to forecast stock prices. The study evaluates the effectiveness of financial data analysis and forecasting models. It reviews the applications of artificial intelligence techniques for data analysis and forecasting in financial markets [42]. Another study presented a comparative analysis of k-nearest Neighbor (K-NN), Gaussian Naive Bayes (GNB), Support Vector Machines (SVM), Random Forest (RF), and XGBoost models using air quality data from 23 cities in India. The experimental results show that XGBoost performs the best [43]. Baran et al. used an adaptive network-based fuzzy inference system (ANFIS), support vector regression (SVR), classification and regression trees (CART), random forest (RF), K-NN, and extreme learning machine (ELM) methods for the prediction of PM10 and PM2.5 components in Sıhhiye region. The ANFIS model was more successful in predicting PM10 values than other methods [44]. In another study, Dokuz et al. investigated the use of deep learning methods such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) along with traditional classification algorithms such as LASSO regression, Support Vector Machines (SVM), Random Forest (RF) and K-NN for the prediction of air quality parameters [45]. These studies reveal the potential of various methods and models in air quality prediction and evaluate the effectiveness of different approaches. The existing literature shows that hybrid models, deep learning, and machine learning techniques have a wide application in air quality predictions. These techniques have great potential to increase prediction accuracy. However, each method and model has its limitations and uncertainties, and future studies need to develop more advanced modeling techniques and data integration strategies to address these shortcomings.

In summary, unlike the existing studies in literature, the study includes many innovative and original elements such as comparative analysis of deep learning models, long-term forecasting capacity, use of advanced visualization techniques, local application, use of missing data completion techniques, and detailed explanation of the methods. This highlights the study's important contributions and differences that distinguish it from other studies in the literature.

To provide a more structured comparison of previous studies and highlight this research's innovative contributions, Table 1 summarizes the key studies in the literature. The table focuses on the methods, performance metrics, dataset characteristics, and missing data imputation techniques used. Unlike most previous studies that evaluate a single model or use basic imputation techniques, this study performs a comparative analysis of three deep learning models (RNN, CNN, LSTM) using multiple performance metrics (MAE, RMSE, R^2). The dataset spans 10 years and incorporates advanced missing data imputation via the K-Nearest Neighbor algorithm, offering a more comprehensive approach to air quality prediction.

As seen in Table 1, while previous studies have provided significant insights into air quality prediction, most focus on a single model or limited pollutant types, often without addressing missing data comprehensively. This study distinguishes itself by directly comparing three deep learning models on the same dataset, implementing advanced imputation techniques, and offering long-term forecasts. These methodological advancements are expected to contribute significantly to air quality management and planning.

Table 1. Comparison of Some Air Quality Forecast Methods in Literature

Studies	Model	Pollutant	Performance Metrics	Missing Data Completion	Comparative Analysis
Kurnaz ve Demir [46]	RNN	SO_2	R^2: 0.67	Missing Information	No
Bernardino et al. [47]	Random Forest	SO_2	R^2: -0.0231	No	No
Cerezuela-Escudero [48]	DNN	PM_{10}	R^2: 0.61	Average Filling	No
Kim et al. [49]	LSTM	PM_{10}	R^2: 0.57	No	No
This Study	RNN, LSTM, CNN	PM_{10}, SO_2	R^2: 0.97, 0.97 MAE: 3.52, 1.26 RMSE: 2.47, 0.86	K-Nearest Neighbor	Yes

1.3. Problem Statement and Motivation

Several important points stand out from the articles in the literature in this study.

- Three different deep learning methods for air quality prediction, namely RNN, CNN, and LSTM, were comparatively analyzed on the same dataset. In literature, the performance of a single model is usually evaluated, or the use of a model as a hybrid with another model is prominent. However, in this study, the direct comparison of these three models on the same dataset reveals the advantages and disadvantages of different methods.
- The second difference is that the study used the RNN model to make air quality predictions for the next three years and visualize these predictions. While most studies in the literature make short-term predictions, this approach offers a longer-term forecast. This important contribution can be made when making strategic decisions for air pollution management and planning.
- Another contribution is that it used various advanced visualization methods (such as Box Plot, Violin Plot, and Scatter Plot) to increase the understandability of the prediction results. This approach makes it easier for non-experts to understand and interpret the prediction results. Such techniques are generally limited in literature and usually limited to only basic graphics (such as line graphs).
- Again, many studies in the literature usually make air quality predictions in wide geographical areas or at a general level. However, this study offers a more localized and detailed prediction model by focusing on a specific local area, such as the Başakşehir district of Istanbul. This provides more directly applicable data for local governments to monitor air quality and develop effective intervention strategies.
- In the study, using the K-Nearest Neighbor (K-NN) algorithm to solve the missing data problem is a subject that is missing in many studies in the literature. While the missing data problem is usually approached with simple filling methods (e.g., filling with average), using a more sophisticated method such as K-NN in this study makes a significant difference in increasing the integrity and accuracy of the data.
- In the study, more than one metric (MAE, RMSE, R^2) was used to evaluate the model performance, and the results of these metrics were discussed in detail. While some studies in literature usually focus on a single performance metric, using more than one metric in this study allows a more comprehensive evaluation of the performance of the models.
- Finally, the application steps of each deep learning model used for data preparation, model architecture, and the parameters used are explained in great detail. Such details are usually briefly mentioned in the literature, and it may be difficult for the reader to understand the subject fully. However, this approach reveals how the model works and why it was chosen in this way.

The remainder of the paper is structured as follows: Section 2 presents general information about data visualization methods, missing data completion techniques, and deep learning models used in air quality data collection. Section 3 details the proposed methodology, explaining the dataset, preprocessing steps, the K-Nearest Neighbor (K-NN) algorithm used for missing data completion, and the prediction processes performed with deep learning models, CNN, RNN, and LSTM. Section 4 presents experimental results and discussions. The success rates of CNN, RNN, and LSTM models in air pollution predictions were evaluated using metrics like MAE, RMSE, and R², and the most successful model was determined. Graphical results are presented with methods such as violin plots, box plots, and scatter plots. In addition, the next 3-year prediction results are discussed. Section 5 represents a general evaluation of the study and the conclusions. It was concluded that the RNN model provided the most successful results in air quality predictions by learning both short-term and long-term dependencies. At the end of the study, forward-looking suggestions were made, and it was stated that model performance could be improved by adding meteorological data.

2. Background

This part of the study offers an overview of data visualization techniques, methods for handling missing data, and deep learning approaches.

2.1. Data Collection

Accurate and reliable data collection forms the basis of the research and analysis processes. The data used in this study are air quality indices (AQI) data collected from various environmental sensors. At certain intervals, the sensors measure various air quality parameters (PM2.5, PM10, SO2, NO2, CO, O3). This collected data can be recorded in databases and made available for analysis. However, incomplete data may be encountered during the collection for reasons such as sensor failures, connection problems, or environmental factors. This incomplete data must be completed and processed with appropriate methods to perform accurate analysis.

2.2. Filling in Missing Data Method

One of the most frequent challenges when working with data is handling missing data. Missing values can negatively impact the accuracy and validity of analysis and visualization. Various algorithms and methods can be applied to fill in missing data to address this. K-NN is a method commonly used for classification and regression in supervised learning. In the context of missing data imputation, K-NN replaces the missing value with the average of the 'k' nearest neighbors' values.

In this study, the K-Nearest Neighbor approach was chosen to address missing data issues due to its ability to maintain the local structure of the dataset and preserve critical interdependencies among features. Air quality datasets, which often include PM2.5, PM10, SO2, and NOx, inherently exhibit complex multivariate relationships and spatial-temporal dependencies. Unlike simpler imputation techniques such as mean or median substitution, which assume a uniform distribution of missing values and fail to account for these dependencies, K-NN uses similarity-based criteria to identify and use the most relevant neighboring data points for imputation. This ensures that the imputed values align with the dataset's statistical properties and reflect the localized patterns and trends inherent in the data. Additionally, preliminary tests conducted during the data preprocessing phase demonstrated that K-NN outperformed simpler methods in terms of both error metrics and downstream predictive model performance. Specifically, K-NN-imputed data resulted in lower mean absolute error (MAE) and root mean square error (RMSE) values than mean substitution. In contrast, models trained on K-NN-imputed datasets exhibited better accuracy and generalization capabilities. This empirical evidence supports the notion that K-NN provides a more robust and contextually appropriate solution for missing data imputation, particularly in applications involving air quality estimation, where data quality directly influences the reliability of forecasts [43].

2.3. Data Fusion and Gaussian Filter

Data fusion means using air quality data from sensors to make more comprehensive and reliable air quality estimates. Data fusion increases data accuracy and predictive power by integrating data from different values. This method contributes to more reliable analysis results. Gaussian filter is a data filtering method used to reduce the noise of environmental data and make the signal smoother. Gaussian filter averages the data by weighing it according to a certain distribution, thus obtaining a more stable and regular data set. This study reduced the noise in air quality data, and more accurate estimates were obtained using the Gaussian filtering method.

2.4. Data Visualization Methods

The process of simplifying large and intricate data sets to make them easier to understand and meaningful by visually representing them through graphs is known as data visualization. Data visualization converts raw data from challenging numerical sets into visual formats where trends, relationships, and patterns can be easily identified. This study used three data visualization techniques to analyze and interpret the predicted data better.

One of these, the Point Scatter Plot, is a technique utilized to analyze the relationship between two different numerical data. This graph visualizes possible correlations or distribution trends between the data, allowing researchers to understand the data more deeply. This method visualizes the relationship between two variables and can show many values in a data set through different colors or sizes. Point scatter plots determine possible correlations or patterns between data and make complex data more understandable [50].

Box Plot is a method used to visualize the distribution, central tendency, and possible data outliers. It makes it easier to understand the general distribution of the data by showing the minimum, maximum, median, first, and third quartiles in a data set. The box plot is especially useful for visualizing comparisons between different data groups and effectively determining possible extreme values in the data set.

The violin plot illustrates the data distribution like the box plot, but it also presents the density distribution of the data. A violin plot provides more comprehensive details regarding the data's form and distribution by visualizing the data's probability density function. This approach enables us to analyze the symmetry and distribution of the data sets in more detail [51].

3. Material and Methods

This study gathered a 10-year dataset from 2014 to 2024 from the Continuous Monitoring Center, affiliated with Turkey's Ministry of Environment, Urbanization, and Climate Change. Afterward, missing data were identified and imputed using K-Nearest Neighbor (K-NN) algorithms. Finally, three deep learning methods, CNN, RNN, and LSTM, were applied to the data completed with k-nn. The results were compared, and the most accurate values were obtained using RNN. After this process, the RNN method was selected, and the next 3 years were estimated. Then, a point distribution graph, box plot, and violin graph were used in the data, respectively. The diagram showing the application steps is shown in Figure 3.

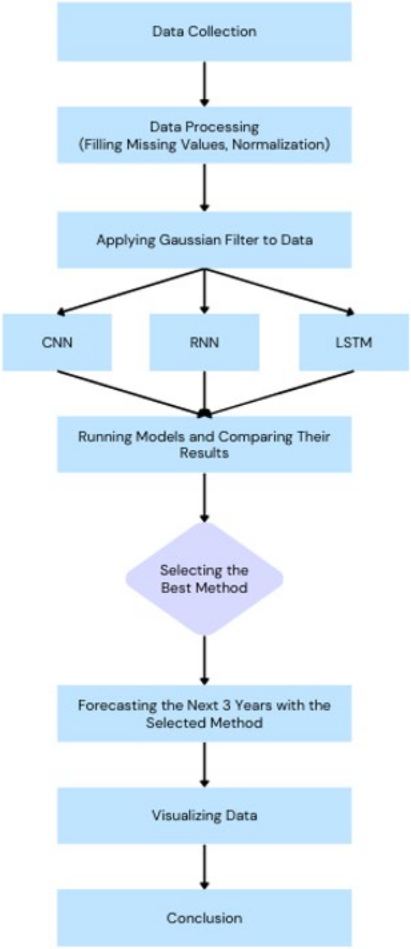

Data Collection
Data Processing (Filling Missing Values, Normalization)
Applying Gaussian Filter to Data
CNN
RNN
LSTM
Running Models and Comparing Their Results
Selecting the Best Method
Forecasting the Next 3 Years with the Selected Method
Visualizing Data
Conclusion

Figure 3. Diagram Outlining the Study

3.1. Dataset

This study obtained data from the Continuous Monitoring Center, made publicly available by the Ministry of Environment, Urban Planning, and Climate Change of the Republic of Turkey [52]. This dataset, covering the years 2014-2024, served as the foundation for the research. The dataset includes relevant air pollutant values and date information for each day and is presented ready for analysis. The first 10 rows of the original data collected are given as an example in Table 2.

The dataset comprises 3655 rows of records in total, and the missing data rates are remarkable:

- Missing PM10 Data: 246 rows
- Missing SO2 Data: 364 rows

- Missing CO Data: 376 rows
- Missing NO2 Data: 224 rows
- Missing NOX Data: 542 rows
- Missing NO Data: 335 rows
- Missing O3 Data: 255 rows

The main reason for choosing this comprehensive 10-year dataset is to train models more effectively and powerfully with deep learning methods. A large dataset enhances the model's sensitivity and significantly improves the accuracy of predictions in environmental analyses. Large data sets improve the model's performance and provide an important advantage for increasing the accuracy of the analysis. For the experimental analysis, 64% of the data was allocated for training, 16% for validation, and 20% was reserved for testing. This division ensures that the model is well-trained in sufficient data, properly validated during training, and adequately evaluated on unseen data to assess its performance effectively.

Table 2. The First 10 Rows of the Original Dataset

Date	PM10(µg/m3)	SO2(µg/m3)	NO2(µg/m3)	CO(µg/m3)	NO(µg/m3)	NOX(µg/m3)	O3(µg/m3)
2014-03-04 00:00:56	70,57	8,54	49,56	460,87	nan	nan	35,23
2014-03-05 00:00:56	135,89	23,23	65,74	756,52	nan	nan	19,28
2014-03-06 00:00:56	53,92	4,50	37,21	217,39	nan	nan	47,35
2014-03-07 00:00:56	52,29	nan	26,60	273,91	nan	nan	62,34
2014-03-08 00:00:56	21,05	nan	15,78	352,17	nan	nan	56,39
2014-03-09 00:00:56	23,71	nan	8,89	369,57	nan	nan	63,58
2014-03-10 00:00:56	21,43	nan	15,80	343,48	nan	nan	59,63
2014-03-11 00:00:56	26,06	nan	14,01	527,27	nan	nan	67,79
2014-03-12 00:00:56	nan	nan	nan	nan	nan	nan	nan
2014-03-13 00:00:56	nan	nan	nan	nan	nan	nan	nan

3.2. Preprocessing

The K-Nearest Neighbor (K-NN) algorithm offers a highly effective approach to complete missing data. This algorithm estimates a value with missing data by looking at its similarities with other values in the dataset. The K-NN algorithm determines the 'k' nearest neighbors to complete a missing value in the dataset and estimates the missing values using the average of these neighbors. The K-NN algorithm not only completes the data but also increases the accuracy and integrity of the dataset, allowing it to work effectively in large datasets.

K-NN was chosen for this study because, in our previous work, we used the mean to address missing data [43]. However, we found that using the mean was insufficient for accurate prediction after completing the missing data. As a result, we opted for K-NN, which demonstrated superior performance in handling missing values and improving predictive accuracy. Using K-NN offers an effective solution, especially in estimating missing values in the dataset with variable values, such as the 10-year air quality in our study. Table 3 below shows the first 10 rows of the dataset filled with K-NN.

Table 3. Dataset Filled with K-NN

Date	**PM10(µg/m3)**	**SO2(µg/m3)**	**NO2(µg/m3)**	**CO(µg/m3)**	**NO(µg/m3)**	**NOX(µg/m3)**	**O3(µg/m3)**
2014-03-04 00:00:56	70.57	8.54	49.56	460.87	29.41000	100.948	35.23
2014-03-05 00:00:56	135.89	23.23	65.74	756.52	67.178	198.724	19.28
2014-03-06 00:00:56	53.92	4.5	37.21	217.39	21.892	66.328	47.35
2014-03-07 00:00:56	52.29	28.74	26.6	273.91	10.978	40.892	62.34
2014-03-08 00:00:56	21.05	21.534	15.78	352.17	6.196	24.17	56.39
2014-03-09 00:00:56	23.71	3.304	8.89	369.57	4.672	20.01	63.58
2014-03-10 00:00:56	21.43	12.302	15.8	343.48	6.51	25.16	59.63
2014-03-11 00:00:56	26.06	2.686	14.01	527.27	3.196	19.578	67.79
2014-03-12 00:00:56	48.78043	7.12803	28.35337	517.94021	14.91725	52.35370	56.82346
2014-03-13 00:00:56	48.78043	7.12803	28.35337	517.94021	14.91725	52.35370	56.82346

3.3. Methods Used and Analysis

This paper used three deep learning methods, namely CNN, RNN, and LSTM, to predict air pollutant data. During the application of these methods, the data was first organized with a Gaussian filter, and predictions were made for each method. The outcomes were assessed based on R² scores, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). Finally, these studies were supported with graphics.

MAE represents the average of the absolute differences between predicted and actual values. This metric measures the magnitude of prediction errors and assigns equal weight to all errors. The Formula 1 for MAE is as follows:

$$MAE = \left(\frac{1}{n}\right) \Sigma \, |y_{\mathrm{i}} - \bar{y}_{\mathrm{i}}| \qquad \text{Formula 1}$$

Where:

- y_{i}: Actual value,
- $\bar{y}$: Predicted value,
- n: Total number of data points.

MAE is simple to interpret and directly measures the average error size. However, it does not emphasize larger deviations, making it less sensitive to extreme prediction errors than other metrics.

RMSE calculates the square root of the average squared differences between predicted and actual values. Squaring the errors assigns greater importance to larger deviations, offering a better understanding of error distribution. In Formula 2 for RMSE is:

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \Sigma \left(y_{\mathrm{i}} - \bar{y}_{\mathrm{i}}\right)^2} \qquad \text{Formula 2}$$

Where:

- y_{i}: Actual value,
- $\bar{y}_{\mathrm{i}}$: Predicted value,
- n: total number of data points.

RMSE is particularly useful when large prediction errors are critical to the analysis. It provides a more comprehensive view of the model's performance by emphasizing significant deviations. However, it is slightly more complex to interpret compared to MAE.

In Formula 3, R² measures the proportion of variance in the dependent variable that the model explains. It provides an overall evaluation of how well the model captures the data's variability. The formula for R² is:

$$R^2 = 1 - \left(\Sigma \left(y_{\mathrm{i}} - \bar{y}_{\mathrm{i}}\right)^2\right) / \left(\Sigma \left(y_{\mathrm{i}} - \bar{\bar{y}}\right)^2\right) \qquad \text{Formula 3}$$

Where:

- y_{i} : Actual value,
- $\bar{y}_{\mathrm{i}}$: Predicted value,
- $\bar{\bar{y}}^{-}$: Mean of actual values,

R² values typically range from 0 to 1, where 1 indicates perfect prediction, and 0 indicates no explanatory power. Negative R² values suggest the model performs worse than a simple mean-based prediction. This metric is particularly useful for assessing the model's overall fit to the data. These three metrics collectively provide a holistic evaluation of model performance. MAE focuses on the average error magnitude, offering a straightforward interpretation. RMSE emphasizes larger errors, making it more informative for analyzing error distribution and model reliability. R² evaluates the model's ability to explain variance in the target variable, providing a summary measure of its overall effectiveness.

This study employs MAE, RMSE, and R² to compare the predictive performance of CNN, RNN, and LSTM models. MAE and RMSE quantify prediction errors, while R² assesses the models' explanatory power. This combination of metrics ensures a comprehensive analysis of the models' accuracy and generalization capacity for forecasting PM10 and SO2 levels.

For air pollutant modeling, performance metrics like R² , MAE, and RMSE are crucial indicators of model accuracy. An R² value between 0.9 and 1 is considered ideal, indicating the model explains most of the variability in the data, while values between 0.8 and 0.9 are still acceptable. For MAE, values below 5 µg/m³ are ideal, with 5–10 µg/m³ being acceptable, as lower errors mean predictions are closer to actual values. Similarly, RMSE should ideally be below 7 µg/m³, with values between 7 and 12 µg/m³ still considered good. These thresholds ensure reliable predictions for pollutants like PM10 and SO_2, aligning with air quality standards.

A Gaussian filter is a data filtering application that reduces the noise of environmental data and makes the signal smoother. This application takes the average of the data by weighting it according to a certain distribution, and thus, a more stable data set is obtained.

Convolutional Neural Networks (CNNs), while primarily utilized for image processing tasks, have proven to be highly effective for time series data analysis [53]. Their ability to extract localized features and learn short-term dependencies makes them suitable for sequential data. In this study, the CNN architecture, illustrated in Figure 4, has been optimized for forecasting PM10 and SO2 levels.

The model begins with a data input layer, where raw time series data is fed into the network. The first layer is a one-dimensional convolutional (Conv1D) with 64 filters and a kernel size of 2. This layer slides a filter over the data, extracting local features and patterns essential for understanding the structure of the time series. Following this, a MaxPooling1D layer with a pooling size of 2 reduces the dimensionality of the data. The model retains critical information while discarding unnecessary details, creating a more compact representation.

A dropout layer with a rate of 0.2 is included to prevent overfitting. This layer randomly disables some neurons during training, ensuring the model generalizes well to new data. The data then passes through a flattening stage, where the multi-dimensional representation is converted into a one-dimensional vector, preparing it for the dense layers.

The dense layers form the final stage of the model. A dense layer with 50 neurons learns high-level representations of the data. Subsequently, an output layer with two neurons generates predictions for the target variables, PM10 and SO2. The model is optimized using the Adam optimizer, with a learning rate of 0.001, and employs the mean squared error loss function.

This CNN architecture is especially adept at processing time series data, leveraging its capability to learn and predict patterns effectively. This study demonstrates that it provides a robust framework for tasks like forecasting air quality parameters.

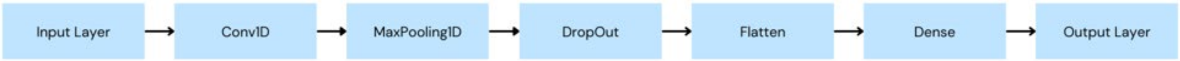


Figure 4. CNN Model Architecture

Recurrent Neural Networks (RNNs) are specialized artificial neural networks designed to model sequential data, such as time series [54]. Their primary advantage lies in their ability to retain information from previous time steps in a hidden state, enabling them to capture temporal dependencies effectively. This makes RNNs highly suitable for tasks like forecasting and sequential data analysis.

In this study, the RNN architecture, depicted in Figure 5, is tailored for predicting PM10 and SO2 levels. The model starts with an input layer where the sequential data is received and prepared for processing. The first layer is a SimpleRNN layer with 50 neurons. This configuration ensures that the outputs of the first RNN layer are passed to the next layer, allowing the model to learn dependencies across multiple time steps. Following the first RNN layer, a dropout layer with a rate of 0.2 is applied. This layer randomly deactivates neurons during training, reducing the risk of overfitting and improving the model's generalization ability.

The architecture continues with a second SimpleRNN layer containing 50 neurons, which deepens the model’s ability to capture sequential patterns. Another dropout layer with a rate of 0.2 is added after this RNN layer to enhance the model's robustness and prevent over-learning.

Finally, the data is processed by an output layer of two neurons, generating predictions for PM10 and SO2 concentrations. The model is trained using the Adam optimizer with a learning rate of 0.001, and the mean squared error loss function is employed to minimize prediction errors.

This RNN architecture leverages its capacity to store and utilize past information, making it a powerful tool for time series prediction. The model achieves accuracy and generalization in its predictions by combining sequential learning with dropout regularization.

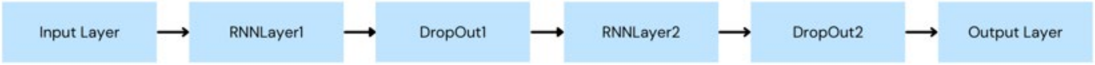


Figure 5. RNN Model Architecture

Long Short-Term Memory (LSTM) networks are a specialized variant of Recurrent Neural Networks (RNNs) that excel at capturing both short-term and long-term dependencies in sequential data [55]. Their unique cell state and gating mechanisms allow them to effectively learn patterns across extended time steps, making them particularly suitable for time series forecasting tasks.

In this study, the LSTM architecture, illustrated in Figure 6, is designed to predict PM10 and SO2 concentrations. The model begins with an input layer where the sequential data is introduced. The first processing stage involves an LSTM layer with 50 neurons. This configuration enables the first LSTM layer to output sequential data passed to the next layer for further processing. Following this, a dropout layer with a rate of 0.2 is applied to randomly deactivate neurons, reducing the risk of overfitting and improving generalization.

The second stage consists of another LSTM layer with 50 neurons, which deepens the model's ability to learn complex dependencies in the data. Similar to the first stage, a dropout layer with a rate of 0.2 is applied after the second LSTM layer to enhance robustness further. Finally, the output layer, containing two neurons, generates predictions for the target variables, PM10 and SO2.

The model is optimized using the Adam optimizer with a learning rate of 0.001, and the mean squared error loss function is used to minimize prediction errors. During training, EarlyStopping is implemented to monitor the validation loss, with a patience of 100 epochs. This ensures that training halts if performance does not improve, preventing unnecessary computations and overfitting. Additionally, ModelCheckpoint is utilized to save the best-performing model for future use.

This LSTM architecture effectively captures temporal dependencies in the data, leveraging its advanced structure to deliver accurate and generalizable predictions for time series forecasting tasks.

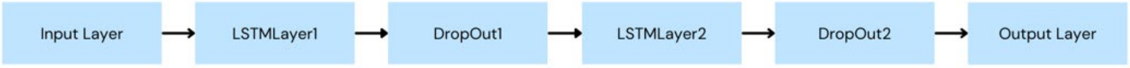


Figure 6. LSTM Model Architecture

The performance of the models is evaluated using three key metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the R² score. These metrics collectively provide a comprehensive assessment of the models' predictive capabilities. Mean Absolute Error (MAE) represents the average of the absolute differences between the predicted and actual values. It serves as a straightforward measure of prediction accuracy by quantifying the magnitude of errors, without considering their direction. Root Mean Square Error (RMSE) is calculated as the square root of the average squared differences between predicted and actual values. This metric gives greater weight to larger errors, making it particularly sensitive to significant deviations in predictions. R² Score evaluates the proportion of the variance in the dependent variable that is explained by the model. It provides an overall indication of how well the model captures the variability in the target data, with a higher R² score indicating better predictive performance. By comparing these metrics across the CNN, RNN, and LSTM architectures, the study identifies the most effective model for forecasting PM10 and SO2 time series data. Each metric contributes unique insights into the models' performance, enabling a thorough evaluation of their strengths and limitations.

Table 4. Comparison of Hyperparameters for CNN, RNN, and LSTM Models

Hyperparameter	**CNN**	**RNN**	**LSTM**
Model Nodes	64 CNN Filters	64 RNN Nodes	2 LSTM Nodes
Epoch	20, 30, 40	20, 30, 40	20, 30, 40
Batch Size	16	16	16
Interpolate Method	N/A	N/A	Linear
Train Data	64% Dataset	64% Dataset	64% Dataset
Validation Data	16% Dataset	16% Dataset	16% Dataset
Test Data	20% Dataset	20% Dataset	20% Dataset
Optimizer	SGD	ADAM	ADAM
Learning Rate	0.0001	0.001	0.001
Dense Layer	64	N/A	N/A

The hyperparameter Table 4 of the study is given above. This table shows the comparatively determined parameters of CNN, RNN, and LSTM models. While the CNN model exhibits a structure suitable for image processing, it worked with 64 filters and the SGD optimization method. RNN and LSTM models are designed for stronger performance on time series data and trained with ADAM optimizer. LSTM successfully captured long-term dependencies using the "linear" interpolation method, while RNN was more effective in learning short-term and long-term relationships. All models were run on the same data rates (64% training, 16% validation, 20% test) and similar epoch values. These hyperparameters were carefully selected to evaluate how the models respond to different data structures.

4. Experimental Results and Discussion

In this study, analyses were conducted on PM10 and SO2 pollutants using CNN, LSTM and RNN as air quality prediction models. In the study, the performance of these models was evaluated with metrics such as MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and R². According to the results, the RNN model showed the best performance with the lowest error rates. The capacity of RNN to learn short- and long-term dependencies enabled the model to be more successful in predictions. While LSTM was successful in capturing long-term dependencies, it did not perform as well as RNN. The CNN model had higher error rates and had difficulty in capturing high variances, especially in time series data. According to the experimental results, the RNN model achieved the highest accuracy in PM10 and SO2 predictions, while CNN exhibited lower performance with higher error values. The differences between the modeled predictions and real values were visualized with violin plot and scatter plot graphics, and higher air pollution was predicted in certain years, especially in PM10 predictions.

Among the models used in this study, accuracy rates of 0.88 for PM10 and 0.93 for SO2 were achieved with CNN; 0.94 for PM10 and 0.95 for SO2 were achieved with LSTM. The most successful model, RNN, provided an accuracy rate of 0.97 for PM10 and SO2, and forecasts are made for the next 3 years using this model. The values and visualizations resulting from the study are given in detail below. The CNN model results shown in Table 5 provide reasonable accuracy in PM10 and SO2 forecasts. However, the MAE and RMSE values are higher than the other models, especially for PM10, indicating that the errors are larger. The R² scores are at the levels of 0.88 and 0.93, indicating that the model's forecast performance is generally good, but it has higher error rates.

Table 5. CNN Model Performance Metrics

Pollutants	**MAE**	**RMSE**	**R²**
PM10	4.68	6.44	0.88
SO2	1.34	2.07	0.93

The RNN model in Table 6 has the lowest MAE and RMSE values in both PM10 and SO2 predictions, showing the best performance. This shows that the RNN model can successfully learn both short-term and long-term dependencies and make more accurate predictions. The R^2 scores are also at 0.97, which shows that the model explains the variance very well and the predictions are very close to the actual values.

Table 6. RNN Model Performance Metrics

Pollutants	**MAE**	**RMSE**	**R²**
PM10	2.47	3.52	0.97
SO2	0.86	1.26	0.97

The LSTM model in Table 7 performs slightly lower than the RNN but still shows a good performance. The MAE and RMSE values are higher than the RNN but lower than the CNN. The R^2 scores are 0.94 and 0.95, demonstrating that the model explains a significant portion of the dataset's variability. While the LSTM performs well in identifying and capturing long-term dependencies in time series data, it is less effective than the RNN in this particular scenario.

Table 7. LSTM Model Performance Metrics

Pollutants	**MAE**	**RMSE**	**R²**
PM10	3.67	4.66	0.94
SO2	1.07	1.85	0.95

As a result, the LSTM model gives a result close to RNN in terms of its performance on time series data, but it does not perform as well as RNN. Nevertheless, it made better predictions compared to CNN. The fact that the CNN model has higher error values indicates that its capacity to capture serial dependencies in time series data is lower than other models. The RNN model has the lowest error values (MAE and RMSE) and the highest R^2 scores in both PM10 and SO2 predictions and stands out as the most successful model for this data set and problem. The study provides better results than other studies conducted before [56-58]. In Figure 7, which represents the performance of the CNN model for predicting PM10 and SO_2 concentrations, there is a noticeable difference between the actual values (blue and orange lines) and the predicted values (cyan and red lines). These differences provide insights into the model's strengths and weaknesses.

For PM10 predictions, the model generally captures the overall trends of the real data, especially during periods of lower concentration. However, significant deviations are observed during peaks, particularly in high pollution events. These deviations suggest that the CNN model struggles to accurately predict extreme values, which could be due to the limited ability of CNNs to model abrupt changes or anomalies in time series data. Despite this, the model successfully tracks seasonal and periodic fluctuations.

For SO_2 predictions, the CNN model performs better at capturing the general trends compared to PM10. The predicted values (red line) closely align with the actual values (orange line) during periods of stability. However, similar to PM10, the model shows weaknesses in predicting sudden spikes or drops in SO_2 levels. This may indicate that while CNNs are effective at identifying overall patterns, they may require additional features or architectural modifications to handle abrupt changes more effectively.

Additionally, the graph highlights that for both pollutants, the predicted values show a slightly smoother pattern compared to the real values. This smoothing effect is common in CNN models, as they prioritize extracting dominant trends rather than capturing noise or highly localized variations. While this improves the generalization of the model, it can reduce its ability to capture sharp fluctuations accurately. In summary, the CNN model demonstrates the ability to follow the overall patterns and seasonal variations of both PM10 and SO_2 levels.

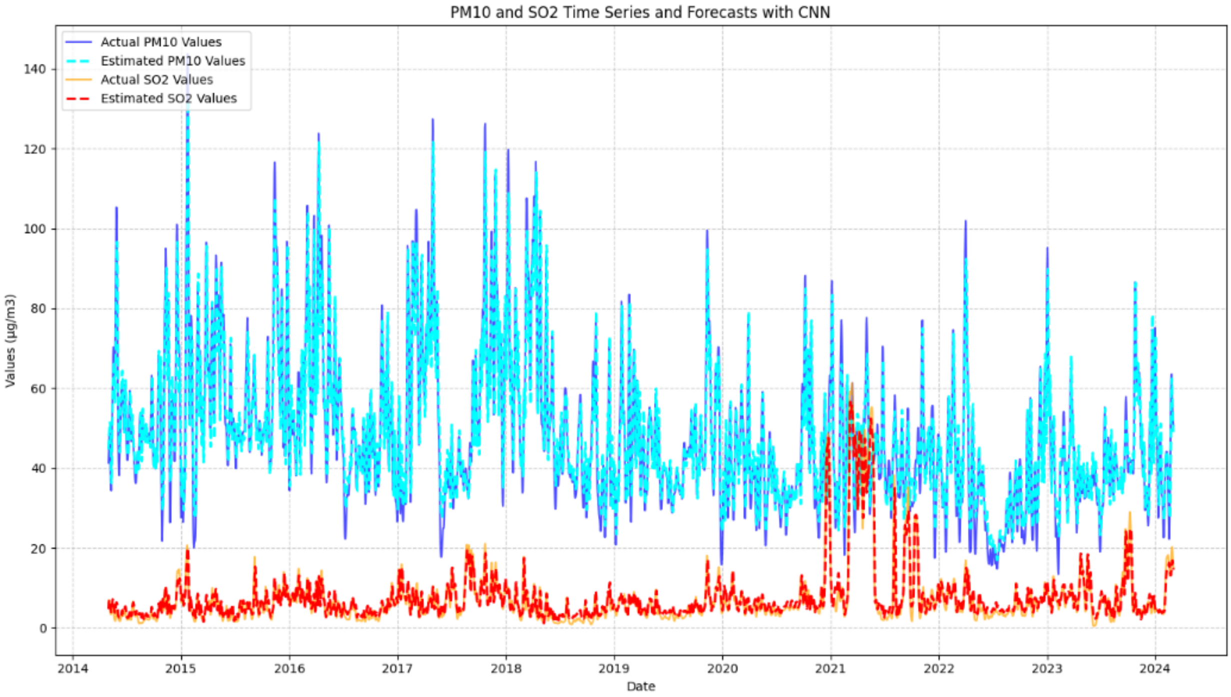


Figure 7. Prediction with CNN Model

In the graph in Figure 8, the performance of the RNN model is demonstrated, showing an impressive alignment between the predicted values (cyan and red lines) and the actual values (blue and orange lines). The RNN model excels at capturing the variations and fluctuations in the time series data for both PM10 and SO_2, with minimal differences between predictions and real values compared to other models.

For PM10 predictions, the predicted values (cyan line) closely follow the actual data (blue line), successfully capturing both gradual and sudden changes. This is especially evident in high pollution events, where the RNN model handles the peaks better than other models, resulting in smaller deviations during extreme conditions. The alignment in seasonal and periodic patterns further demonstrates the model's strength in learning temporal dependencies.

For SO_2 predictions, the predicted values (red line) show an even stronger agreement with the actual values (orange line). The model accurately tracks both the general trends and the sharp fluctuations in SO_2 concentrations. Sudden increases and decreases are effectively modeled, showcasing the RNN's ability to handle dynamic changes in time series data.

The graph also reveals the RNN model's ability to generalize across different periods. Unlike other models, the prediction lines do not exhibit significant smoothing, indicating that the model effectively preserves the detailed patterns and variability present in the data. This allows the RNN to achieve the lowest MAE and RMSE values among the compared models, making it the most accurate in predicting both PM10 and SO_2 levels. Overall, the RNN model's capability to capture both short-term and long-term dependencies in the time series data is evident. This makes it a reliable choice for air quality prediction tasks, especially when accurate forecasting of abrupt changes and complex patterns is required.

In Figure 9, the LSTM model's performance is depicted, showing strong agreement between the predicted values (cyan and red lines) and the actual values (blue and orange lines) for both PM10 and SO_2 levels. This indicates that the LSTM model is effective in capturing temporal dependencies and producing reliable predictions.

For PM10 predictions, the model demonstrates a good fit, closely following the actual data. The predicted values (cyan line) align well with the actual values (blue line) during both periods of stability and fluctuations. While the model captures seasonal patterns effectively, it occasionally underestimates extreme peaks. However, these deviations are relatively small, and the model retains a good balance between generalization and precision, making it comparable to the RNN in performance.

For SO_2 predictions, the LSTM model also shows a strong alignment with the actual values (orange line). The predicted values (red line) follow the overall trends effectively, accurately modeling gradual changes and periodic behaviors. However, the model struggles slightly with sudden spikes or sharp drops in SO_2 levels, leading to minor deviations. Despite these occasional mismatches, the model maintains consistency in capturing broader trends. Overall, the LSTM model handles both pollutants effectively, with its performance close to the RNN model. While the MAE and RMSE values are slightly higher, LSTM excels in capturing long-term dependencies within the time series data. Its ability to balance accuracy and generalization makes it a strong candidate for modeling and forecasting air pollutant levels, particularly in scenarios requiring the identification of broader trends over extended periods.

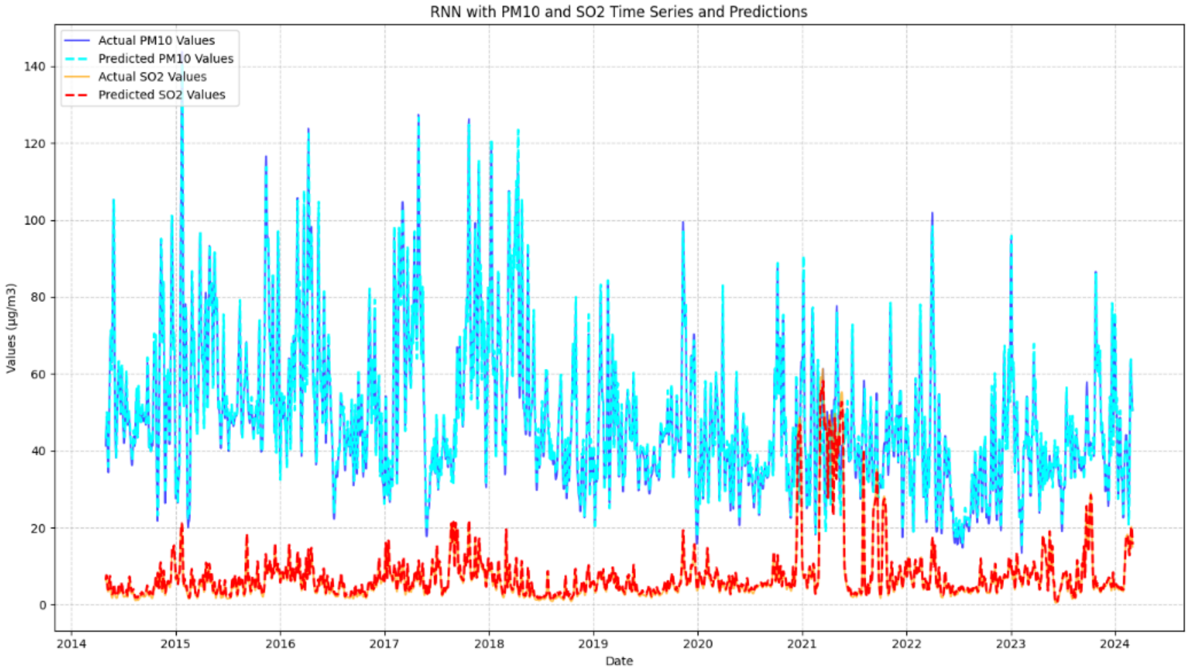


Figure 8. Prediction with RNN Model

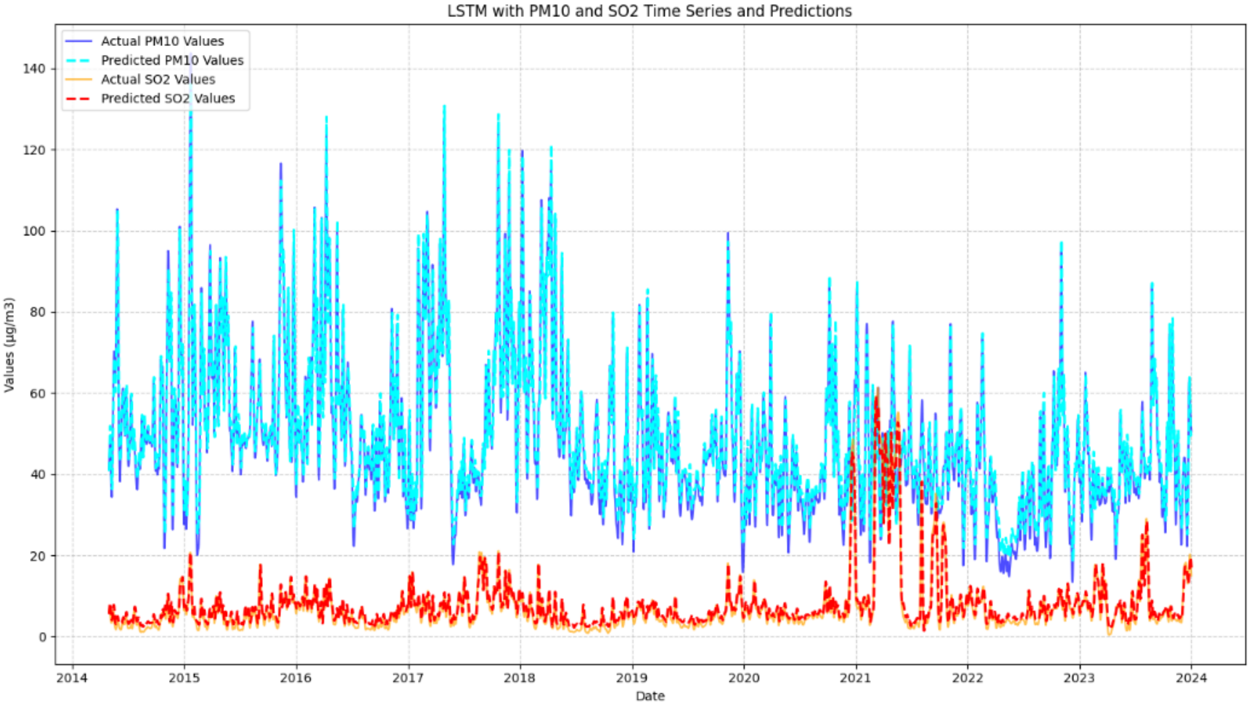


Figure 9. Prediction with CNN Model

As a result, the model that showed the best performance was RNN. The discrepancies in the differences between the actual and predicted values are minimal and the model effectively learns both short and long-term relationships or dependencies within the data. In the second place, LSTM has a good performance and is close to RNN. It has been successful in capturing long-term dependencies in the time series. CNN has a lower performance compared to other models. It has difficulty in capturing the high variance of PM10 values in particular. CNN, which is more suitable for image processing, is not as effective as other models in time series data. Since our study is the most successful RNN model among these three methods, we continue with this method when making future predictions. The output of the performance metrics for the next 3 years is shown in Table 8 below. According to this table, it indicates that the model's predictions are not perfect compared to the actual data but show a moderate performance. It shows that the model can make larger errors in some of its predictions, but it is generally at a reasonable level of accuracy.

Table 8. Performance Metrics in Future Prediction with RNN Model

Pollutants	**MAE**	**RMSE**	**R²**
PM10	11.60	17.15	0.63
SO2	2.39	3.99	0.82

In the graphical output of the model in Figure 8, the future predictions for PM10 are flatter and exhibit lower fluctuations. This indicates that the model may have limited capacity to predict future changes. For SO2, the predictions are relatively lower variance and follow a flatter course. Flatter and lower fluctuations are observed for both PM10 and SO2, indicating that the model may have limited capacity to manage uncertainties and may have difficulty predicting more complex events.

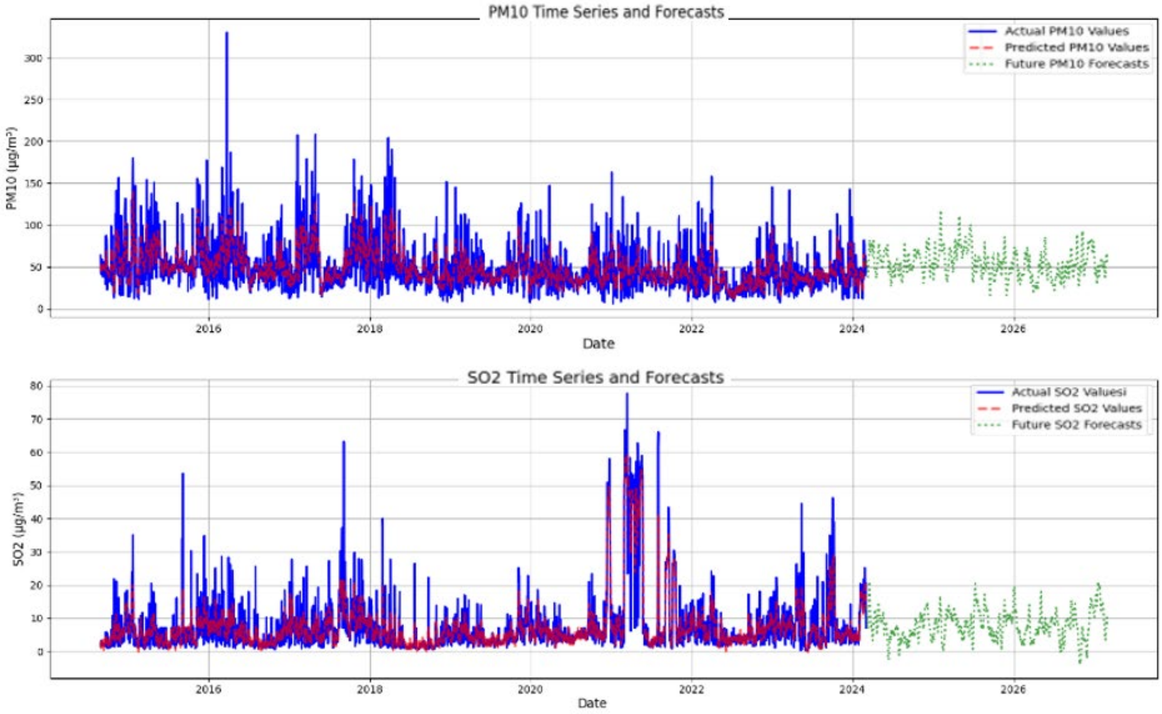


Figure 8. Next 3 Years Prediction Graph with RNN

As a result, the model performs moderately for PM10, while it is more successful in SO2 predictions. The fact that future predictions are relatively flat and have low variance reveals some limitations of the model's predictive capacity. These results can be used for future air quality management and planning, but they indicate that the predictions should be evaluated carefully. Further developments and different model structures can increase the prediction accuracy.

In the final stage of the study, efforts were made to make this large dataset more manageable and complex data easier and more understandable. For this purpose, data visualization was made in the study. To make comparison easier, available data and data obtained with future predictions were used.

Figure 9 summarizes the changes in PM10 and SO2 levels over the years and their future predictions. While PM10 levels showed a wider distribution in 2021 and 2024, indicating that air pollution increased in these years, they concentrated in narrower and lower concentrations in 2022 and 2023, indicating relatively better air quality. PM10 projections for 2024-2027 show a wider distribution than in previous years, indicating that the model has uncertainty in predicting future PM10 levels. SO2 levels, on the other hand, generally show consistent distribution, but in some years, such as 2021, they show a wider distribution, indicating periods of poorer air quality. Low concentrations of SO2 in 2020, 2022, and 2024 indicate cleaner air conditions in these years. Future SO2 projections, although generally concentrated at low levels, show a wider distribution in 2025 and 2027, indicating that the model has more uncertainty for these years.

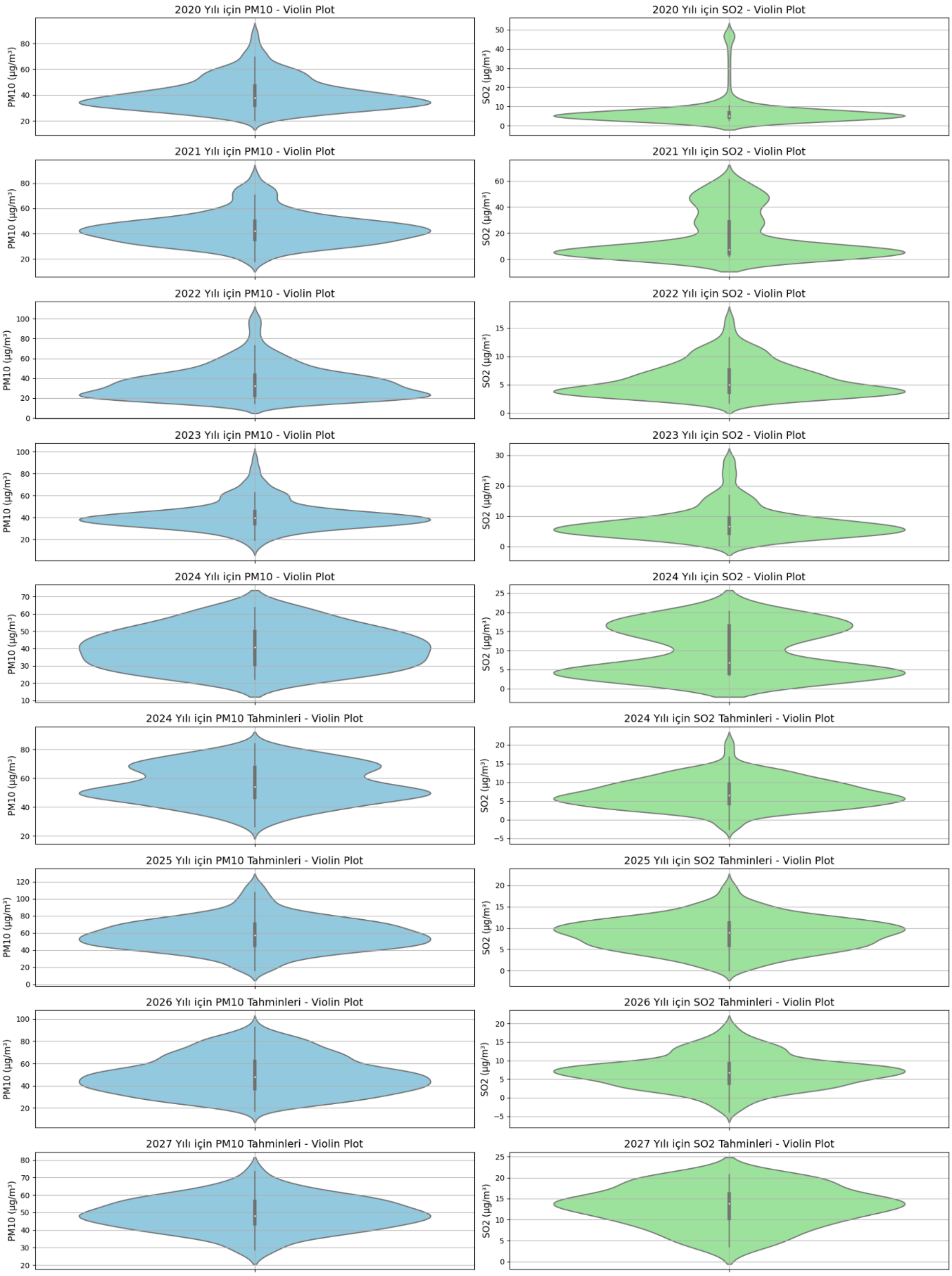


Figure 9. Visualization with Violin Plot

When the actual data between 2020-2024 in Figure 10 is analyzed, it becomes evident that PM10 levels contain more extreme values in some years (especially 2021 and 2023) and air pollution events are more frequent and intense in these years. It is understood that the median values for PM10 are generally concentrated between 20-40 µg/m³ and the variance between years is generally similar. SO2 levels generally remain in a low range (2-10 µg/m³), but more variance and some extreme values are noted in 2021 and 2022. This shows that air quality varies especially in these years and that unexpectedly high SO2 concentrations are experienced in some periods. When looking at future projections, the distribution of PM10 and SO2 levels predicted by the model between 2024-2027 seems generally consistent with previous years but contains some uncertainties and variances. For PM10 estimates, the wider bins (IQR) indicate that the model predicts more variance in future PM10 levels and therefore expects higher pollution values in some periods. For SO2 estimates, consistent intensity and relatively narrow distribution are observed at low concentration levels, reflecting the model's expectation that future SO2 levels will generally remain low. However, a few outliers in some years indicate that possible future unexpected events should also be taken into account.

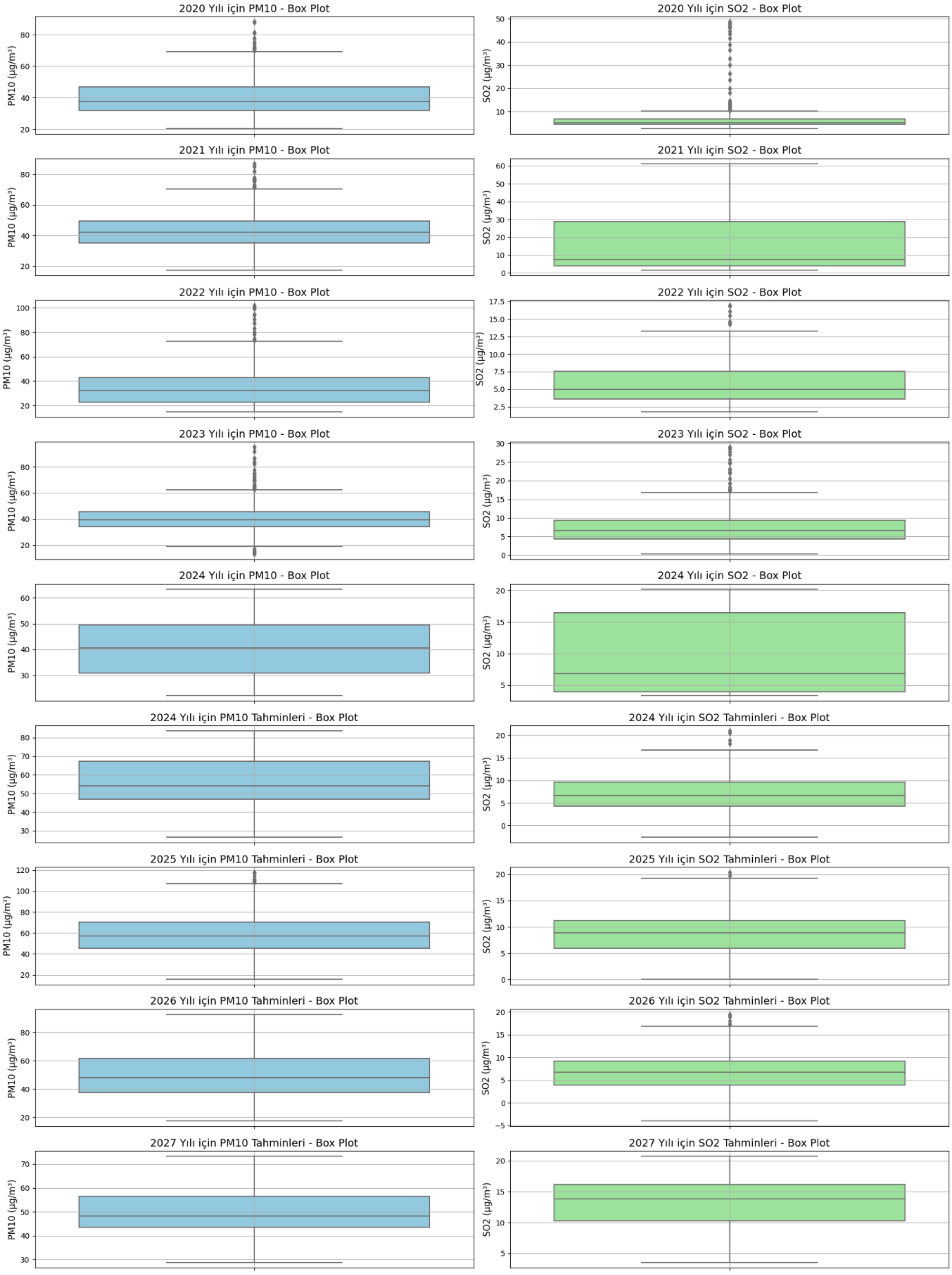


Figure 10. Visualization with Box Plot

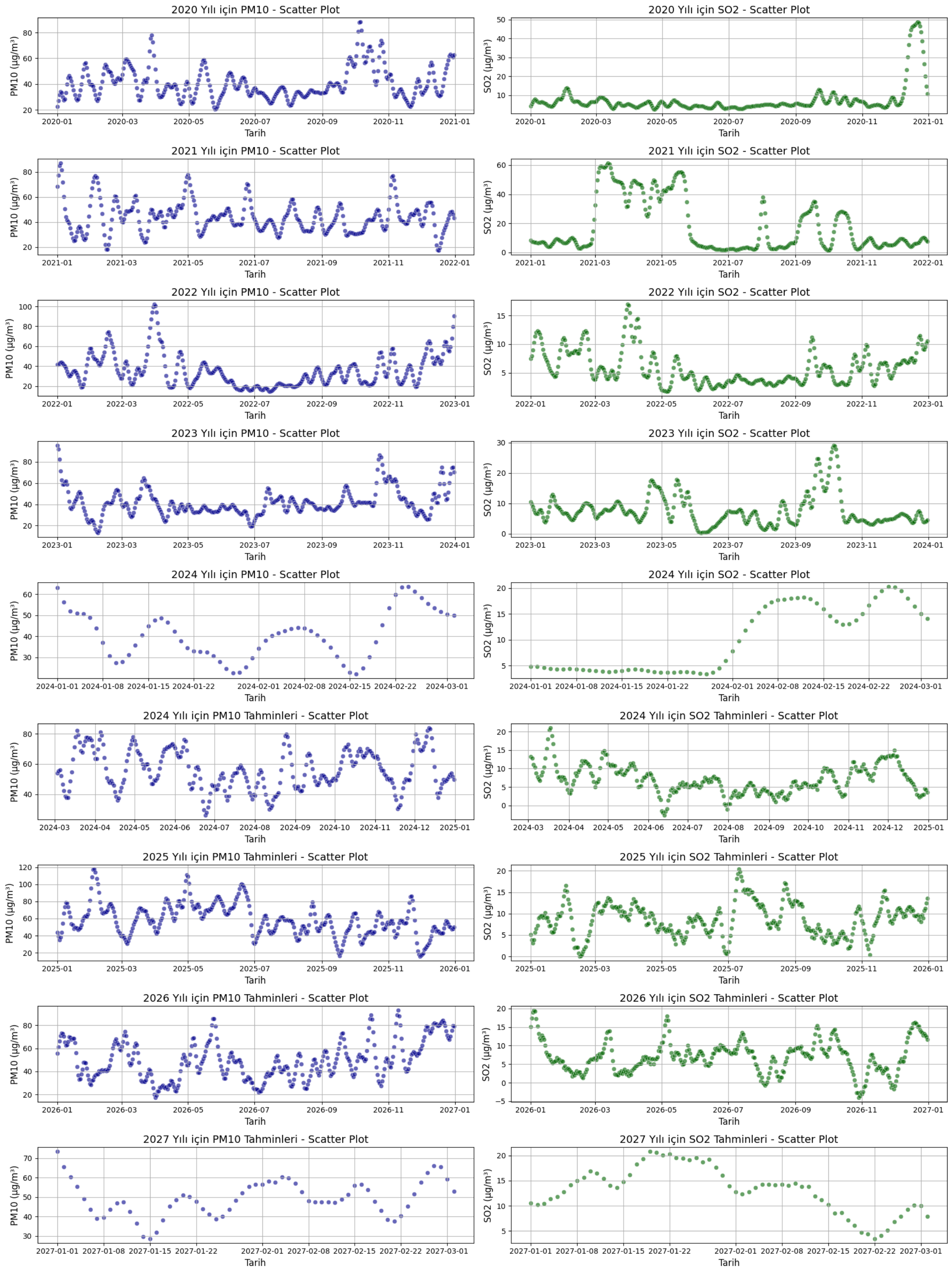


Figure 11. Visualization with Point Scatter Plot

When we look at the actual data between 2020-2024 in Figure 11, a significant fluctuation is observed in PM10 levels. There are frequent and high peaks in PM10 levels, especially in 2021 and 2023; this shows that air pollution events were more frequent and intense in these years. In 2022 and 2024, it is understood that PM10 levels were more stable and at lower levels, and sudden increases were less common. This shows that air quality was relatively better in these years. SO2 levels, on the other hand, follow a more regular course over the years and generally remain at low concentrations. However, some sudden increases were observed in SO2 levels in 2021, which shows that this year was more variable in terms of air quality. It is seen that SO2 levels were lower and stable in 2022 and 2023, and some fluctuations occurred again in 2024. When the future estimates in Figure 11 are examined, significant fluctuations and high peaks are predicted in PM10 levels, especially in 2025 and 2026. This shows that the model predicts serious changes in PM10 levels and potential pollution events in these years. For 2027, PM10 levels are expected to be lower and less variable, reflecting a better air quality expectation for this year. SO2 estimates generally remain at low levels and do not show major fluctuations. Some increases in SO2 levels are expected in 2025 and 2026, but these levels generally remain at low concentrations. SO2 levels are predicted to be quite stable in 2027. In general, these scatter plot graphics are useful in visualizing changes in PM10 and SO2 levels over time and possible future trends. Despite some uncertainties in the estimates, it can be concluded that the model successfully captures seasonal and yearly variations in air quality.

The results of this study highlight both the strengths and weaknesses of deep learning models in air quality prediction. While the RNN model demonstrated superior performance in capturing short- and long-term dependencies, its predictive accuracy in future forecasts, particularly for PM10, raises questions about the limitations of relying solely on historical data for complex, multi-faceted environmental phenomena. The relatively flat future predictions indicate that the model may struggle to represent extreme events or sudden changes, which are crucial for proactive air quality management. This underscores the importance of integrating external factors such as meteorological variability, policy interventions, or socio-economic changes to improve the robustness of predictions. Furthermore, the comparative performance of CNN and LSTM models revealed that while these architectures have potential, their limitations in handling time-series need to be addressed, perhaps through hybrid or ensemble approaches.

The visualization techniques employed, including scatter plots, violin plots, and box plots, played a significant role in interpreting and communicating the results. These tools provided an intuitive understanding of air quality patterns, seasonal variations, and model discrepancies, making the findings accessible even to non-expert audiences. However, static visualizations inherently limit real-time applications, highlighting the potential of integrating dynamic, interactive visualization systems for monitoring and forecasting. These graphical insights also revealed areas where models underperformed, such as higher variance in PM10 predictions, emphasizing the need for tailored visual analytics to complement predictive models. Addressing these challenges through adaptive modeling techniques, richer datasets, and advanced visualization frameworks could significantly enhance the applicability of such studies, ensuring that the tools developed are not only accurate but also actionable for air quality management and policymaking.

5. Conclusion

Air pollution remains a pressing environmental and health issue for individuals living in densely populated urban areas and industrialized regions. It contributes to respiratory and cardiovascular diseases, and various forms of cancer, and has broader ecological impacts, including its role in climate change through ecosystem damage. This underscores the importance of continuous monitoring and accurate prediction of air quality to mitigate its harmful effects. In this study, the air quality of the Başakşehir district in Istanbul was evaluated, and future air quality levels were predicted using deep learning methods: CNN, RNN, and LSTM. A 10-year dataset (2014–2024) was utilized to compare the performances of these models based on metrics such as Root Mean Square Error (RMSE), R^2 scores, and Mean Absolute Error (MAE). Among the models, the RNN demonstrated the highest accuracy, with the lowest error rates and the highest R^2 scores for both PM10 and SO_2 predictions. Its ability to learn both short-term and long-term dependencies in time series data made it the most effective model for predicting air pollutant levels in the near future. The LSTM model showed performance close to the RNN, especially in capturing long-term dependencies, but it fell slightly short in predictive accuracy. Meanwhile, the CNN model, while capable of capturing some patterns, struggled with the sequential and dynamic nature of the data, resulting in relatively higher error rates compared to RNN and LSTM. Using the RNN model, PM10 and SO_2 levels were forecasted for the next three years. The results revealed a reasonable level of accuracy for short-term predictions; however, the model displayed limitations in capturing complex, long-term trends. For instance, PM10 predictions showed a flatter trend with lower variance, indicating the model's difficulty in forecasting significant future changes. Similarly, while SO_2 predictions were generally concentrated at low levels, occasional variability suggested the potential for unexpected fluctuations. For longer-term predictions, hybrid models that integrate multiple methods and consider external factors may provide a more effective approach. External influences, such as meteorological variables (e.g., wind speed, temperature, precipitation), socio-economic shifts (e.g., changes in fossil fuel usage, industrial activity), and policy interventions (e.g., emission regulations, green energy incentives), play a critical role in shaping air quality over time. These factors introduce complexities that single models like RNN may not fully capture when forecasting extended periods. While hybrid models are advantageous for long-term predictions, this study highlights that single models such as RNN excel in near-future forecasts due to their ability to capture immediate temporal dependencies efficiently. Incorporating hybrid modeling approaches for long-term forecasts, alongside single models for short-term predictions, can offer a balanced and comprehensive framework for air quality prediction. Furthermore, including additional environmental and meteorological variables in future datasets would enhance predictive accuracy by accounting for the broader range of factors influencing air pollution. In conclusion, the findings demonstrate the effectiveness of deep learning models, particularly RNN, for short-term air quality predictions. Future research should focus on hybrid model development for long-term forecasting while continuing to explore ways to integrate external influences into predictive frameworks. Such advancements will ensure more reliable predictions and contribute to more effective strategies for managing and mitigating air pollution.

References

[1] R. Kaur and P. Pandey, "Air pollution, climate change, and human health in Indian cities: A brief review," Frontiers in Sustainable Cities, vol. 3, p. 705131, 2021. doi: 10.3389/frsc.2021.705131.

[2] M. B. Khan, S. Setu, N. Sultana, S. Gautam, B. A. Begum, and M. A. Salam, "Street dust in the largest urban agglomeration: Pollution characteristics, source apportionment and health risk assessment of potentially toxic trace elements," Stochastic Environmental Research and Risk Assessment, pp. 1–20, 2023. doi: 10.1007/s00477-023-02432-1.

[3] O. Isinkaralar, K. Isinkaralar, and T. N. T. Nguyen, "Spatial distribution, pollution level, and human health risk assessment of heavy metals in urban street dust at neighbourhood scale," International Journal of Biometeorology, pp. 1–13, 2024. doi: 10.1007/s00484-024-02729-y.

[4] Y. Liu, T. Jin, S. Yu, and H. Chu, "Pollution characteristics and health risks of heavy metals in road dust in Ma'anshan, China," Environmental Science and Pollution Research, vol. 30, no. 15, pp. 43726–43739, 2023. doi: 10.1007/s11356-023-25303-2.

[5] A. Lai, "Analysis of air pollution from vehicle emissions for the contiguous United States," Journal of Geovisualization and Spatial Analysis, 2022.

[6] G. D. Oreggioni, O. Mahiques, F. Monforti-Ferrario, E. Schaaf, and M. Muntean, "The impacts of technological changes and regulatory frameworks on global air pollutant emissions from the energy industry and road transport," Energy Policy, vol. 168, 2022. doi: 10.1016/j.enpol.2022.113021.

[7] T. Li, Y. Yu, and Z. Sun, "A comprehensive understanding of ambient particulate matter and its components on the adverse health effects based from epidemiological and laboratory evidence," Particle and Fibre Toxicology, vol. 19, p. 67, 2022. doi: 10.1186/s12989-022-00507-5.

[8] A. Garcia, E. Santa-Helena, and A. De Falco, "Toxicological effects of fine particulate matter (PM2.5): Health risks and associated systemic injuries—Systematic Review," Water Air and Soil Pollution, vol. 234, p. 346, 2023. doi: 10.1007/s11270-023-06278-9.

[9] A. Unnikrishnan and S. Rajeswari, "Forecasting daily air quality index and early warning system for estimating ambient air pollution on road networks using Gaussian models," Tehnički Glasnik, vol. 18, no. 4, pp. 549–559, 2024.

[10] P. Perez, C. Menares, and C. Ramírez, "PM2.5 forecasting in Coyhaique, the most polluted city in the Americas," Urban Clim., vol. 32, p. 100608, 2020, doi: 10.1016/j.uclim.2020.100608.

[11] A. Aggarwal and D. Toshniwal, "A hybrid deep learning framework for urban air quality forecasting," J. Clean. Prod., vol. 329, Art. no. 129660, 2021, doi: 10.1016/j.jclepro.2021.129660.

[12]C.-J. Huang and P.-H. Kuo, “A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities,” Sensors, vol. 18, no. 7, Art. no. 7, Jul. 2018, doi: 10.3390/s18072220.

[13] R. Janarthanan, P. Partheeban, K. Somasundaram, and P. Navin Elamparithi, “A deep learning approach for prediction of air quality index in a metropolitan city,” Sustainable Cities and Society, vol. 67, p. 102720, Apr. 2021, doi: 10.1016/j.scs.2021.102720.

[14] S. Masmoudi, H. Elghazel, D. Taieb, O. Yazar, and A. Kallel, “A machine-learning framework for predicting multiple air pollutants’ concentrations via multi-target regression and feature selection,” Science of The Total Environment, vol. 715, p. 136991, May 2020, doi: 10.1016/j.scitotenv.2020.136991.

[15] B. Liu, M. Lai, P. Zeng, and J. Chen, “Air pollutant prediction based on a attention mechanism model of the Yangtze River Delta region in frequent heatwaves,” *Atmospheric Research*, vol. 311, p. 107701, Dec. 2024, doi: 10.1016/j.atmosres.2024.107701.

[16] H. Yang, Z. Liu, and G. Li, “A new hybrid optimization prediction model for PM2.5 concentration considering other air pollutants and meteorological conditions,” Chemosphere, vol. 307, p. 135798, Nov. 2022, doi: 10.1016/j.chemosphere.2022.135798.

[17] P. A. Rani and Dr. V. Sampathkumar, “A novel artificial intelligence algorithm for predicting air quality by analysing the pollutant levels in air quality data in tamilnadu,” e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 5, p. 100234, Sep. 2023, doi: 10.1016/j.prime.2023.100234.

[18] Q. Wu and H. Lin, “A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors,” Science of The Total Environment, vol. 683, pp. 808–821, Sep. 2019, doi: 10.1016/j.scitotenv.2019.05.288.

[19] B. Zhang et al., “Air pollutant diffusion trend prediction based on deep learning for targeted season—North China as an example,” Expert Systems with Applications, vol. 232, p. 120718, Dec. 2023, doi: 10.1016/j.eswa.2023.120718.

[20] J. Luo and Y. Gong, “Air pollutant prediction based on ARIMA-WOA-LSTM model,” Atmospheric Pollution Research, vol. 14, no. 6, p. 101761, Jun. 2023, doi: 10.1016/j.apr.2023.101761.

[21] G. I. Drewil and R. J. Al-Bahadili, “Air pollution prediction using LSTM deep learning and metaheuristics algorithms,” Measurement: Sensors, vol. 24, p. 100546, Dec. 2022, doi: 10.1016/j.measen.2022.100546.

[22] “Analysis of the summer thermal comfort indices in İstanbul | International Journal of Biometeorology.” Accessed: May 10, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s00484-024-02669-7

[23] A. Kshirsagar and M. Shah, “Anatomization of air quality prediction using neural networks, regression and hybrid models,” Journal of Cleaner Production, vol. 369, p. 133383, Oct. 2022, doi: 10.1016/j.jclepro.2022.133383.

[24] B. Zhang et al., “Deep learning for air pollutant concentration prediction: A review,” Atmospheric Environment, vol. 290, p. 119347, Dec. 2022, doi: 10.1016/j.atmosenv.2022.119347.

[25] T. D. Akinosho, M. Bilal, E. T. Hayes, A. Ajayi, A. Ahmed, and Z. Khan, “Deep learning-based multi-target regression for traffic-related air pollution forecasting,” Machine Learning with Applications, vol. 12, p. 100474, Jun. 2023, doi: 10.1016/j.mlwa.2023.100474.

[26] M. Yılmaz, Y. Kara, H. C. Çulpan, G. Can, and H. Toros, “Detection and regional analysis of heatwave characteristics in İstanbul,” Sustainable Cities and Society, vol. 97, p. 104789, Oct. 2023, doi: 10.1016/j.scs.2023.104789.

[27] J. González-Pardo, S. Ceballos-Santos, R. Manzanas, M. Santibáñez, and I. Fernández-Olmo, “Estimating changes in air pollutant levels due to COVID-19 lockdown measures based on a business-as-usual prediction scenario using data mining models: A case-study for urban traffic sites in Spain,” Science of The Total Environment, vol. 823, p. 153786, Jun. 2022, doi: 10.1016/j.scitotenv.2022.153786.

[28] X. Shi, Z. Huang, Y. Dai, W. Du, and J. Cheng, “Evaluating emission reduction potential and co-benefits of CO2 and air pollutants from mobile sources: A case study in Shanghai, China,” Resources, Conservation and Recycling, vol. 202, p. 107347, Mar. 2024, doi: 10.1016/j.resconrec.2023.107347.

[29] Y. Dokuz, A. Bozdağ, and B. Gökçek, “HAVA KALİTESİ PARAMETRELERİNİN TAHMİNİ VE MEKANSAL DAĞILIMI İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KULLANILMASI,” Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, Jan. 2020, doi: 10.28948/ngumuh.654092.

[30] S. Ünaldi and N. Yalçin, “Hava Kirliliğinin Makine Öğrenmesi Tabanlı Tahmini: Başakşehir Örneği Prediction of Air Pollution based on Machine Learning Methods: A Case Study for Başakşehir, İstanbul,” p. 10.

[31] P. Aksak, Ş. K. Öztürk, and Ö. Ünsal, “Kentsel Isı Adasının İklim Parametreleri ve Uzaktan Algılama Üzerinden İncelenmesi: İstanbul Kenti Örneği,” ECD, vol. 32, no. 1, Art. no. 1, Jun. 2023, doi: 10.51800/ecd.1266060.

[32] Y.-C. Lin, Y.-T. Lin, C.-R. Chen, C.-Y. Lai, and Y.-B. Wang, “Meteorological and traffic effects on air pollutants using Bayesian networks and deep learning,” Journal of Environmental Sciences, Feb. 2024, doi: 10.1016/j.jes.2024.01.057.

[33] I. H. Fong, T. Li, S. Fong, R. K. Wong, and A. J. Tallón-Ballesteros, “Predicting concentration levels of air pollutants by transfer learning and recurrent neural network,” Knowledge-Based Systems, vol. 192, p. 105622, Mar. 2020, doi: 10.1016/j.knosys.2020.105622.

[34] I. Jairi, S. Ben-Othman, L. Canivet, and H. Zgaya-Biau, “Enhancing air pollution prediction: A neural transfer learning approach across different air pollutants,” *Environmental Technology & Innovation*, vol. 36, p. 103793, Nov. 2024, doi: 10.1016/j.eti.2024.103793.

[35] H. Dai, G. Huang, J. Wang, H. Zeng, and F. Zhou, “Prediction of Air Pollutant Concentration Based on One-Dimensional Multi-Scale CNN-LSTM Considering Spatial-Temporal Characteristics: A Case Study of Xi’an, China,” Atmosphere, vol. 12, no. 12, Art. no. 12, Dec. 2021, doi: 10.3390/atmos12121626.

[36] Z. Wu, Y. Tian, M. Li, B. Wang, Y. Quan, and J. Liu, “Prediction of air pollutant concentrations based on the long short-term memory neural network,” Journal of Hazardous Materials, vol. 465, p. 133099, Mar. 2024, doi: 10.1016/j.jhazmat.2023.133099.

[37] B. Das, Ö. O. Dursun, and S. Toraman, “Prediction of air pollutants for air quality using deep learning methods in a metropolitan city,” Urban Climate, vol. 46, p. 101291, Dec. 2022, doi: 10.1016/j.uclim.2022.101291.

[38] “Prediction of SO2 and PM10 air pollutants using a deep learning-based recurrent neural network: Case of industrial city Sakarya | Elsevier Enhanced Reader.” Accessed: Dec. 05, 2022. [Online]. Available: https://reader.elsevier.com/reader/sd/pii/S2212095521002819?token=4F8C7E774AFEF62F87DEEDFCC66D0CD0E6760E1DD3BA1655B309B6F55BDB090842B65A1488055ED705E638D6699E9847&originRegion=eu-west-1&originCreation=20221203124930

[39] J. Yang, L. Shi, J. Lee, and I. Ryu, “Spatiotemporal prediction of particulate matter concentration based on traffic and meteorological data,” Transportation Research Part D: Transport and Environment, vol. 127, p. 104070, Feb. 2024, doi: 10.1016/j.trd.2024.104070.

[40] S. Jochner, I. Markevych, I. Beck, C. Traidl-Hoffmann, J. Heinrich, and A. Menzel, “The effects of short- and long-term air pollutants on plant phenology and leaf characteristics,” Environmental Pollution, vol. 206, pp. 382–389, Nov. 2015, doi: 10.1016/j.envpol.2015.07.040.

[41] S. A. Ajayi, C. A. Adams, G. Dumedah, A. O. Adebanji, and W. Ackaah, “The impact of traffic mobility measures on vehicle emissions for heterogeneous traffic in Lagos City,” Scientific African, vol. 21, p. e01822, Sep. 2023, doi: 10.1016/j.sciaf.2023.e01822.

[42] S. Arslankaya and Ş. Toprak, “Using Machine Learning and Deep Learning Algorithms for Stock Price Prediction,” Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi, vol. 13, no. 1, pp. 178–192, Jan. 2021, doi: 10.29137/umagd.771671.

[43] D. Mengüş and B. Daş, “Actionable Data Visualization for Air Quality Data in the Istanbul Location,” *Balkan Journal of Electrical and Computer Engineering*, vol. 10, no. 4, Art. no. 4, Oct. 2022, doi: 10.17694/bajece.1180676.

[44] R. Rabie, M. Asghari, H. Nosrati, M. Emami Niri, and S. Karimi, “Spatially resolved air quality index prediction in megacities with a CNN-Bi-LSTM hybrid framework,” *Sustainable Cities and Society*, vol. 109, p. 105537, Aug. 2024, doi: 10.1016/j.scs.2024.105537.

[45] D. Saravanan and K. Santhosh Kumar, “Improving air pollution detection accuracy and quality monitoring based on bidirectional RNN and the Internet of Things,” *Materials Today: Proceedings*, vol. 81, pp. 791–796, Jan. 2023, doi:

10.1016/j.matpr.2021.04.239.

[46] G. Kurnaz and A. S. Demir, "Prediction of SO2 and PM10 air pollutants using a deep learning-based recurrent neural network: Case of industrial city Sakarya," *Urban Climate*, vol. 41, p. 101051, Jan. 2022, doi: 10.1016/j.uclim.2021.101051.

[50] T. Bernardino, M. A. Oliveira, and J. N. Silva, "Using remotely sensed data for air pollution assessment," Feb. 04, 2024, *arXiv*: arXiv:2402.06653. doi: 10.48550/arXiv.2402.06653.

[51] E. Cerezuela-Escudero, J. M. Montes-Sanchez, J. P. Dominguez-Morales, L. Duran-Lopez, and G. Jimenez-Moreno, "A systematic comparison of different machine learning models for the spatial estimation of air pollution," *Appl Intell*, vol. 53, no. 24, pp. 29604–29619, Dec. 2023, doi: 10.1007/s10489-023-05109-y.

[49] H. S. Kim *et al.*, "Development of a daily PM_{10} and $PM_{2.5}$ prediction system using a deep long short-term memory neural network model," *Atmospheric Chemistry and Physics*, vol. 19, no. 20, pp. 12935–12951, Oct. 2019, doi: 10.5194/acp-19-12935-2019.

[46] B. Cui, M. Liu, S. Li, Z. Jin, Y. Zeng, and X. Lin, "Deep learning methods for atmospheric PM2.5 prediction: A comparative study of transformer and CNN-LSTM-attention," *Atmospheric Pollution Research*, vol. 14, no. 9, p. 101833, Sep. 2023, doi: 10.1016/j.apr.2023.101833.

[47] X. Qi, G. Mei, S. Cuomo, C. Liu, and N. Xu, "Data analysis and mining of the correlations between meteorological conditions and air quality: A case study in Beijing," *Internet of Things*, vol. 14, p. 100127, Jun. 2021, doi: 10.1016/j.iot.2019.100127.

[48] D. Qu, X. Lin, K. Ren, Q. Liu, and H. Zhang, "AirExplorer: visual exploration of air quality data based on time-series querying," *J Vis*, vol. 23, no. 6, pp. 1129–1145, Dec. 2020, doi: 10.1007/s12650-020-00683-6.

[52] SIM Air Quality- Station Data Download Continuous Monitoring Center. https://sim.csb.gov.tr/STN/STN_Report/StationDataDownloadNew (202 0) (accessed 2 March 2024)

[53] A. Utku, "Hindistan'daki Turistik Şehirlerin İklim Değişkenlerinin Tahminine Yönelik Hibrit ConvGRU Modeli Hybrid ConvGRU Model for Prediction of Climate Variables of Touristic Cities in India".

[54] B. Baran, "Sıhhiye Bölgesi Hava Kalitesi İndeksinin Aşırı Öğrenme Makineleri ve Yapay Sinir Ağları ile Tahmini," 2022.

[55] Y. Dokuz, A. Bozdağ, and B. Gökçek, "HAVA KALİTESİ PARAMETRELERİNİN TAHMİNİ VE MEKANSAL DAĞILIMI İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KULLANILMASI," *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, Jan. 2020, doi: 10.28948/ngumuh.654092.

[56] K. Oğuz and M. A. Pekin, "Makine Öğrenme Algoritmaları ile PM10 Konsantrasyon Tahmini," *Journal of Advanced Research in Natural and Applied Sciences*, vol. 8, no. 2, pp. 201–213, Jun. 2022, doi: 10.28979/jarnas.981202.

[57] B. Kotan and A. Erener, "PM10, SO2 hava kirleticilerinin çoklu doğrusal regresyon ve yapay sinir ağları ile sezonsal tahmini," *Geomatik*, vol. 8, no. 2, Art. no. 2, Aug. 2023, doi: 10.29128/geomatik.1158565.

[58] B. Gökçek, N. Şaşa, Y. Dokuz, and A. Bozdağ, "PM10 Parametresinin Makine Öğrenmesi Algoritmalari ile Mekânsal Analizi, Kayseri İli Örneği," *DEUFMD*, vol. 24, no. 70, Art. no. 70, Jan. 2022, doi: 10.21205/deufmd.2022247008.

Author(s) Contributions

Damla Mengus: Data collection, software, performing analysis, writing.
Bihter Das: Supervision, writing, review, and editing.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 112-122, 2025
DOI: 10.35377/saucis...1488149

RESEARCH ARTICLE

Harnessing AI for Leadership Development: Predictive Model for Leadership Assessment

Adel AlOamiri[1], **Abdullahi Abdu Ibrahim**[1]

[1]Electrical and Computer Engineering, Altinbas University, Istanbul, Türkiye, ror.org/0145w8333

Corresponding author:
Adel AlOmairi, Electrical and Computer Engineering, Istanbul, Türkiye
adel.alomairi10@gmail.com

Article History:
Received: 22.05.2024
Revised: 07.03.2025
Accepted: 13.03.2025
Published Online: 27.03.2025

ABSTRACT
The present paper has been devoted to the study conducted with the purpose of examining the possibility of applying Machine Learning techniques in classifying leadership based on structured survey data. The objective was to create a predictive model that would allow classifying leadership into three groups – Low, Medium, and High – based on behavior scores. The model was expected to offer a reliable tool for improving leadership development programs and recruitment processes by providing a precise and scalable leadership classification, The study illustrates the potential of advanced ML techniques for rethinking the traditional approaches to the assessment of leadership. Due to the use of advanced ensemble modeling, it was possible to ensure the high accuracy of 93.3% in leadership predicting. Such outcomes can generate considerable advantages for organizational development strategies. The use of ensemble machine learning in the domain of organizational behavior studies can be considered as a valuable academic contribution as it has demonstrated the capacity of determining the application of ensemble techniques for enhancing leadership studies. at the same time, it offers a useful instrument to develop more sophisticated and data-driven practices for leadership development.

Keywords: Machine learning, Leadership style classification, Ensemble learning, Predictive Analytics in HR, Quantitative leadership evaluation, AI in organizational development

1. Introduction

The development of advanced machine learning methods significantly impacted leadership assessment in organizational behavior. Leadership is one of the most critical concepts in examining organizational processes because of its impact on various organizational outcomes [1], such as employee engagement, team productivity, overall corporate prosperity, and even the presence of factors associated with routine organizational culture. Personality assessment is one of the most complicated challenges in psychology because of the subjective character of personality judgment. Nevertheless, leadership is one of the most researched and examined roles in psychology, and this fact allows for the application of various objectively validated questionnaires to assess leaders' behavior [2].

In the industrial and organizational psychology of the past decade, research in leadership assessment has been significantly developed. One of the latest trends in the field is machine learning, which categorizes latent constructs impacted by this or that aspect of leaders' behavior. In the present paper, the authors report on developing a powerful machine-learning tool capable of classifying leadership. For this purpose, the authors used a leadership questionnaire dataset containing information on various leadership indicators, such as transformational leadership or intellectual stimulation. The dataset was divided into three groups according to the level of behaviors, with two cut-off points, and the machine learning model was trained on these data.

The methodology is based on the use of several sophisticated machine learning models that have been investigated in several studies, including such types as Random Forest [3], Support Vector Machines [4], K-nearest neighbor [5] and Gradient Boosting [6]. These models favored the above specifics, with the high-dimensional ability and robust characteristics in classification challenges. Additionally, to refine and enhance the accuracy and viability of the prediction, the Stacking Ensemble approach was investigated and adopted [7] - [9]. As far as it is known, given each model's strengths and capacities, the Stacking Ensemble technique is based on combining them in favor of an individually better performance. Stacking uses a meta-model, Logistic Regression, in the present study to integrate and convert the predictions of each model into the final, optimal classification [10]. This benefits accuracy and reduces errors or poor performance of the final classification generalization.

Cite as: A. AlOmairi and A. Abdu Ibrahim, "Harnessing AI for leadership development: Predictive model for leadership assessment," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 112-122, 2025, doi: 10.35377/saucis...1488149

First, the specificity of this approach is reduced potential and opportunities for human bias. One of the most crucial drawbacks of human assessment is the heavy influence of those in favor and against a particular leader being objectively assessed. If the developed methodology heavily relies on purely quantitative data and objective algorithms, no room for such biased implementation can ruin the conceptual framework. In this way, the reliance on machine learning eliminates the potential for such a negative impact and does it in favor by reducing biasedness.

Secondly, methodology benefits from the machine learning characteristics in terms of being more scalable in leadership assessment. In this framework, more scalability indicates that, unlike the risks of losing quality in assessments of bigger populations, the machine models may easily operate and provide accurate leadership assessments for bigger datasets. This is not the case for human manual evaluation. Lastly, the overall predictive nature of the outcomes of these steps provides a way to predict future leadership success.

With the evolution of artificial intelligence, its adoption in leadership assessment is also likely to increase, and future research can potentially focus on utilizing even more advanced AI mechanisms, such as deep learning and neural networks [11]. The advancements of such AI mechanisms can make assessments even more accurate, reliable, and applicable in various contexts, eventually improving the effectiveness of leadership and the outcomes of organizations.

2. Literature Review

There are different algorithms and statistical methods used in leadership assessment, and depending on the nature of the assessment instrument and the desired outcomes, several algorithms and statistical methodologies are employed to evaluate leadership practices. Common algorithms and techniques for evaluating leaders include:

Factor analysis is a statistical technique for extracting underlying structures from data collection. Factor analysis is commonly used in leadership evaluation to isolate the most crucial features or behaviors [12]. Regression analysis establishes a connection between a set of independent variables and a set of predictor variables. Regression analysis may be used in leadership evaluation to determine which characteristics, skills, or context variables most indicate future performance as a leader [13]. Item Response Theory (IRT) estimates the latent skill or trait being tested; this statistical technique analyzes answers to multiple-choice questions. It is possible to gauge a leader's efficacy using IRT by analyzing their replies to a questionnaire [14]. Cluster analysis is a statistical technique for classifying data or cases into clusters with comparable features. Cluster analysis may be used in leadership evaluation to categorize candidates into teams based on shared leadership traits [15]. Content analysis is a technique used to dissect text or speech for underlying structures like themes or categories. Content analysis may be used to extract the most salient points from leadership narratives and interviews for use in evaluation [16].

Various artificial intelligence algorithms and techniques have been used in business management and leadership, such as natural language processing, NLP, decision trees, neural networks, support vector machines, and Bayesian Networks.

Researchers focus on how machine learning affects several aspects and recommend continuous investigation using the lenses of the digital era to approach leadership. Several studies have been conducted, such as automating leadership assessments [2] and understanding leadership traits in machine learning [17].

The NPL is a subset of AI that focuses on understanding and processing human language. To determine a leader's success, NLP may examine text data from various sources, including emails, social media, and performance reviews. In [18], NLP has been used to distinguish the employees' views regarding the organization's leaders. At the same time, in [19], the AI approaches address human resource management HRM and how it can enhance their work.

Another AI algorithm is the decision tree, a machine learning algorithm for classification and prediction. Decision trees may be used to forecast leadership performance and isolate the variables that most affect it in leadership assessments; Bekensiene and Hoskova [20] used them to identify the effective indicators of leadership. Then, in 2019, they implemented this model in Lithuania army forces, where their research helped to understand how troops might be categorized not just by military rank but also concerning how they respond to leadership indications [21].

In [22], the leadership styles have been addressed. How can they be predicted by using a localized multiple kernel learning method (LMKL)? Various NFs extracted from different modalities were utilized to detect leadership styles. However, the suggested technique was tested on two distinct types of data. To our knowledge, no in-the-wild dataset exists for leadership style prediction.

Inclusive leadership has been addressed using machine learning in [23]; their results show that the path of inclusive leadership's evolution is most clearly seen in its gradual transition from the field of education, where it primarily focuses on schools, students, diversity, and equity, to that of organizational behavior, where it primarily focuses on leaders, employee participation, the workplace, and situational factors. The current and future focus of inclusive leadership studies will be on the role of contextual variables.

Regarding automated assessments, research was conducted to assess and identify the leadership in college students [24]; this study presents LAIGA, a machine learning-based methodology for passively and automatically analyzing and selecting college student leaders based on their academic profiles and conduct in the LMS. (LMS). The proposed method deals with the issues of leadership evaluation and identification. The suggested technique is validated by a case study of graduating IT majors. It produced a MAPE of 5.87% in the leadership assessment task and an F1 of 83.2% in the leadership identification test. Consequently, the suggested strategy can predict leadership evaluation and identification post-graduation with as little information as available during the first year of college. Understanding the factors that contribute to the development of leadership skills and designing mechanisms that promote the growth of student leaders is made possible by the findings of this study. This study also sheds light on the ability to use machine learning technologies to assess and identify leadership in college students automatically. If grades and LMS activities can be obtained from students, the suggested technique may be easily applied to various academic disciplines.

3. Methodology

3.1. Dataset

To address the research questions, the study utilizes a dataset that combines behavioral assessments and structured surveys scored equally by several organizational leaders. Each instance in the dataset represents the evaluation of each respondent across multiple categories of leadership behavior. These categories are Core Transformational Leadership – modeling the way, encouraging the heart, challenging the process, inspiring a shared vision and ennobling the spirit. High-Performance Expectations – focusing on goals, setting high standards and not letting standards slip. Supportive Leader Behavior – understanding team members, acting supportive when required, and creating a friendly climate. Intellectual Stimulation – urging issues to be perceived from another angle, creating an environment to be imaginative and novel, and discussing things philosophically. The survey responses were provided on a Likert scale from 1 to 10, which allows for a quantitative analysis. The questionnaire used for collecting data is shown in Table 1.

Table 1. Questionnaire of Transformational Leadership

No.	Items
➢	***Core Transformational Leadership Behavior (CT)***
1	My leader defines the vision and mission of the organization clearly.
2	My leader sets a suitable role model.
3	My leader encourages the attainment of common goals.
➢	***High-Performance Expectation (HP)***
4	My leader encourages his subordinates to contribute to the organization.
5	My leader encourages his subordinates to perform at their best
6	My leader encourages his subordinates to achieve their best goals.
➢	***Supportive Leader Behavior (SL)***
7	My leader shows respect for his subordinates.
8	My leader shows concern for his subordinates.
9	My leader is concerned about the welfare of his subordinates.
10	My leader considers suggestions from subordinates before acting.
➢	***Intellectual Stimulation (IS)***
11	My leader encourages his subordinates to solve old problems in the right way.
12	My leader encourages subordinates to think about the best way to do the job.
13	My leader encourages his subordinates to use appropriate solutions to problems.

3.2. Data Preprocessing

1) Data Cleaning

The initial steps involved handling missing data and errors in data entry. Given the structured nature of survey data, missing values were imputed using the respective item's median score to maintain the data distribution's integrity.

2) Feature Engineering

Scores from related survey items were aggregated to create composite scores for each leadership dimension, enhancing the analytical robustness of the dataset.

3) Data Scaling

The features were standardized using the StandardScaler to ensure the model inputs had a mean zero and unit variance. This is particularly important for models sensitive to the scale of input features like SVM.

4) Categorization

The 'Total Transformational Leadership' score was calculated as the sum of all leadership dimension scores, and it was categorized into three groups (Low, Medium, and High) based on percentile thresholds to facilitate classification; the distribution is shown in Figure 1.

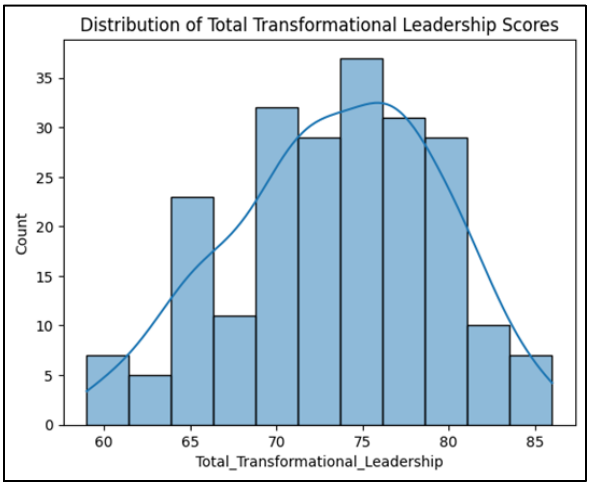


Figure 1. Distribution of Total Transformational Leadership Scores

3.3. Modelling Technique

1) Base Models

Random Forest is an ensemble learning method that constructs many decision trees at training time. It has high accuracy and robustness and works efficiently on large databases.

SVM is a powerful classifier that works well in high-dimensional spaces. It is useful when the number of dimensions surpasses the number of samples.

Gradient Boosting: It operates by constructing models in a phased sequence where each new model addresses the failures introduced by the already trained trees. It is widely used for its strong predictive power of any differential loss function.

2) Stacking Ensemble

A stacking approach was used, and the final estimator was a logistic regression model. Stacking allows combining the strengths of individual models to achieve better prediction accuracy and higher generalization to never-before-seen data. The selection of models was random. Random Forest, SVM, and Gradient Boosting were chosen because each had a particular strength that made them especially good at handling some complexity in data. Random Forest is not inclined to overfit and is good at dealing with non-linearity in data. SVM is effective in high-dimensional spaces, which is particularly important because leadership is multidimensional, and it is hard to compare the psychological traits of leadership by order of magnitude. Gradient Boosting is an excellent learner, and the need to minimize error sequentially is a good example of a problem where such a learner will be especially powerful. The diversity of these models, in the complementary nature of their strengths, is why they stick together.

Stacking allows the blending of these models efficiently to enhance the ensemble's overall well-performing capabilities. The final estimator was a logistic regression model because it provided clear probabilistic output. In addition, using logistic regression to stack predictions allows accounting for the logistic distribution of binary outcomes, which is the prettiest approach to dealing with a categorical response variable. This methodology allows for a sound framework for analyzing leadership more sophisticatedly using advanced data science tools and insights that can potentially revolutionize strategic human resource management.

3) Deep learning approach

To evaluate the effectiveness of deep learning in leadership classification, we implemented a Multi-Layer Perceptron (MLP) Neural Network. The model was designed with the following architecture:

- 4 hidden layers:
    - 128 neurons (first layer)

  - 64 neurons (second layer)
  - 32 neurons (third layer)
  - 16 neurons (fourth layer)
- Activation function: ReLU in all hidden layers
- Optimizer: Adam
- Loss function: Categorical Cross-Entropy
- Regularization: Alpha = 0.0001 to prevent overfitting
- Iterations: 1000 epochs for training stability

The dataset was normalized using MinMaxScaler to scale input features between 0 and 1. The target variable, representing leadership classification (Low, Medium, High), was one-hot encoded and categorized based on percentile thresholds. The dataset was split into 80% training and 20% testing sets. The model was trained using a batch size 16, optimizing through gradient descent-based Adam optimization.

4. Results

1) The ensemble machine learning model was validated using the data generated from the leadership behavior surveys. The classification effectively grouped the leadership styles as Low, Medium, and High depending on collecting the leadership dimensions' scores.

The Stacking Ensemble model, integrating predictions from Random Forest, SVM, and Gradient Boosting with a Logistic Regression meta-model, achieved the highest accuracy of 93.3%. This represents a significant improvement over the individual models, as the confusion matrix of each model is shown in Figures 2, 3, and 4, respectively.

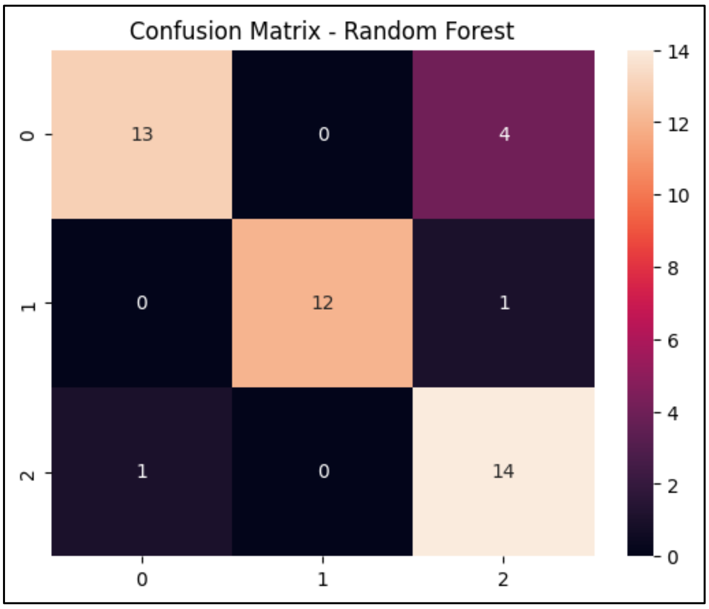


Figure 2. Random Forest Confusion Matrix

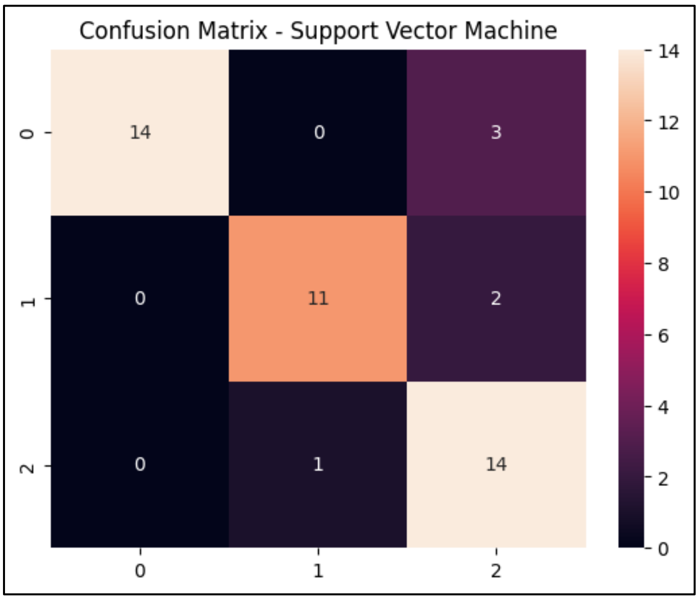


Figure 3. Support Vector Machine Confusion Matrix

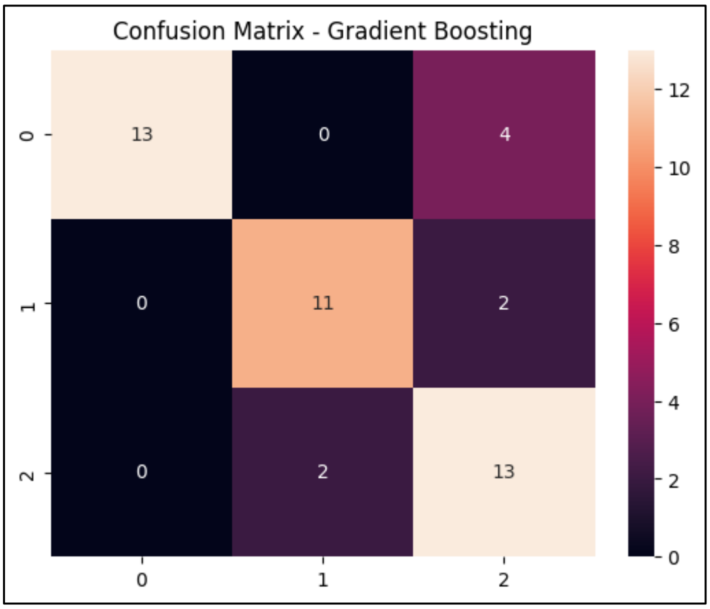


Figure 4. Gradient Boosting Confusion Matrix

The ensemble stacking model shows a better result, as shown in the confusion matrix of Figure 5.

These metrics indicate that the Stacking Ensemble model was particularly effective at identifying High and Medium leadership styles, with perfect precision in identifying Medium leadership styles. The 100% recall in High leadership style suggests that all instances of High leadership were correctly identified.

The superior performance of the Stacking Ensemble model can be attributed to its ability to harness the diverse strengths of the underlying base models, effectively mitigating their weaknesses. For instance, Random Forest provided a robust baseline with its decision tree-based approach, which is less prone to overfitting and good at handling the binary split of categorical data. At the same time, SVM contributed to the model's performance in high-dimensional spaces, which is crucial given the multi-dimensional nature of the data, and Gradient Boosting improved the model's ability to reduce bias and variance sequentially, addressing errors left by the previous models in the sequence.

The integration of these models through stacking, guided by a Logistic Regression meta-model, optimized the combination of their predictions. This likely led to improved accuracy and reliability in classifying leadership styles, as shown in the comparison histogram in Figure 6.

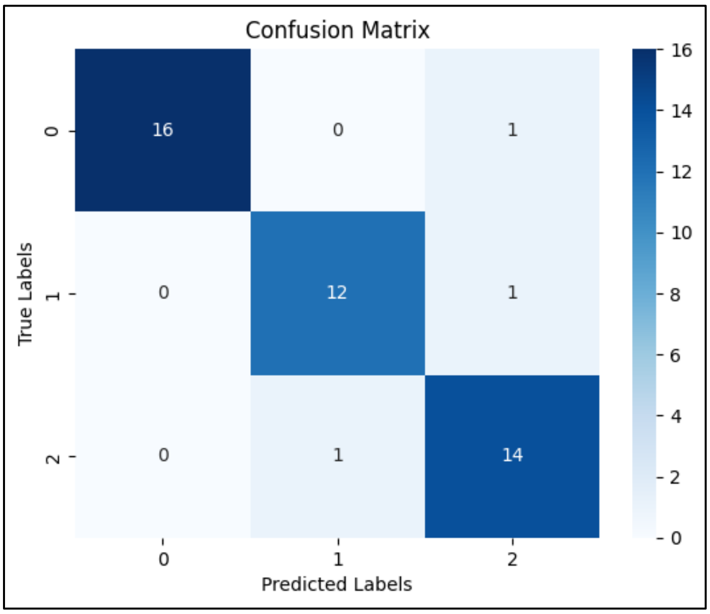


Figure 5. Confusion Matrix of the Ensemble model

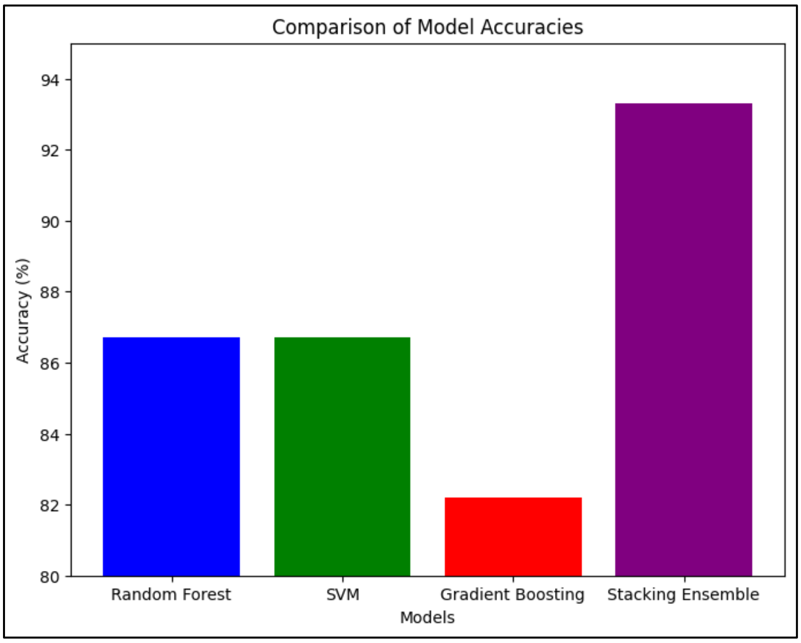


Figure 6. Comparison Histogram for the Models Accuracies

A comparison has been made to understand the results of each model, as shown in Table 2.

Table 2. Performance Metrics of the Models

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	86.6%	0.89	0.87	0.87
SVM	86.6%	0.89	0.87	0.87
GBM	82.2%	0.85	0.82	0.83
Stacking Ensemble	93.3%	0.94	0.93	0.94

Results showed that using advanced ensemble machine-learning techniques was highly effective for classifying leadership styles. Since the Stacking Ensemble model was created using different algorithms, this approach demonstrated high accuracy and described diverse leadership behaviors. The Stacking Ensemble model can be particularly useful for leader assessment.

2) MLP Results Show the distribution of correctly and incorrectly classified leadership levels. Diagonal values indicate correct classifications, while off-diagonal values indicate misclassifications. The "Medium" category had more misclassifications compared to "High" and "Low," as seen in Figure 7.

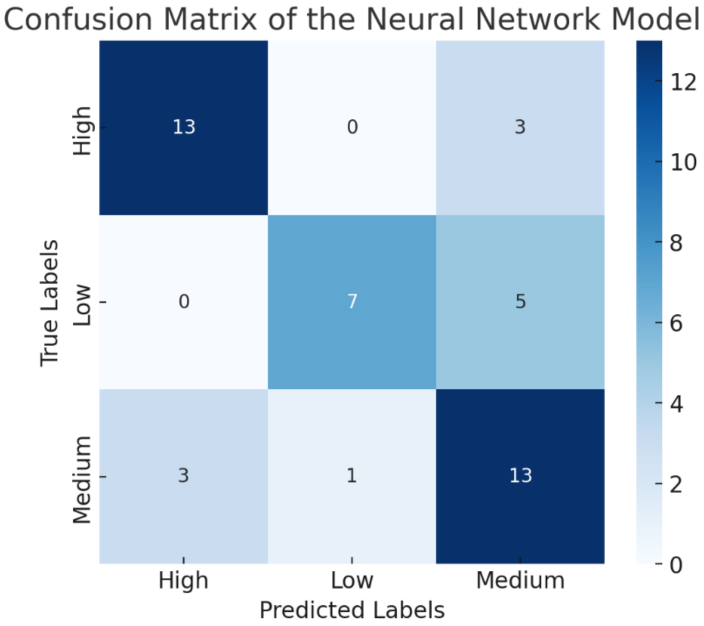


Figure 7. Confusion Matrix of the Neural Network Model

Classification Performance Metrics: A bar chart comparing Precision, Recall, and F1 scores across leadership levels. The "High" category had balanced precision and recall. The "Low" category had high precision but low recall, meaning it was often predicted correctly, but some "Low" cases were misclassified. The "Medium" category had lower precision but higher recall, meaning the model correctly identified most Medium cases but sometimes predicted them incorrectly, as seen in Figure 8.

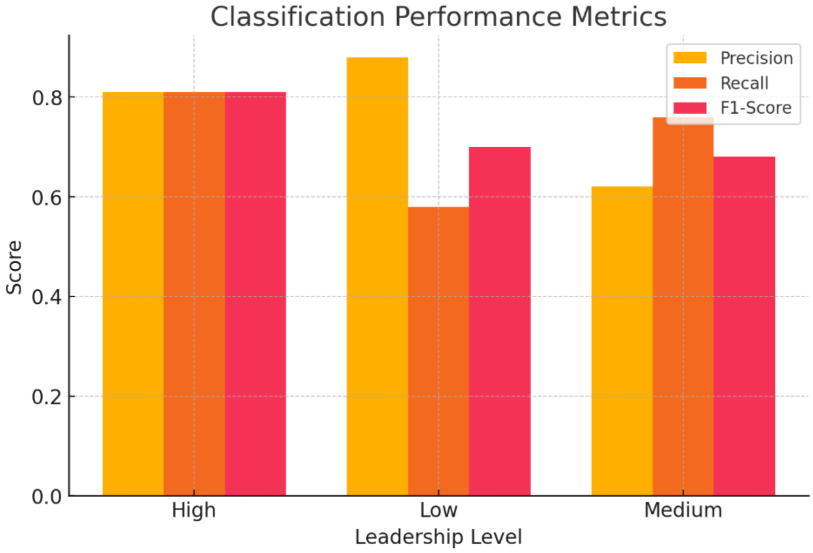


Figure 8. Classification Performance Metrics

3) Performance Comparison of Stacking Ensemble and Neural Network Models To evaluate the effectiveness of different machine learning approaches for leadership classification, we compared the Stacking Ensemble Model with a Neural Network Model (MLP). The results of both models are presented in Table 3.

Table 3. Model Performance Comparison

Metric	Stacking Ensemble Model	Neural Network (MLP)
Accuracy	93.3%	75.6%
Precision	0.94	0.78
Recall	0.93	0.75
F1-Score	0.94	0.77

The Stacking Ensemble Model outperformed the Neural Network Model in all key performance indicators. However, the improved Neural Network Model demonstrated a better classification balance, particularly for the Low and Medium leadership categories.

5. Discussion

The study's outcomes illustrate the effectiveness of ensemble machine-learning solutions for categorizing leadership styles based on behavioral data. The Stacking Ensemble model, which combined results of Random Forest, SVM, and Gradient Boosting, was characterized by 93.3% accuracy, which may be defined as significantly better compared to the results that can be achieved by traditional approaches using a single model. This example illustrates the solution's capacity to interpret multidimensional, complex data and deliver subtype classifications that less sophisticated methods may blur. The findings suggest that the Stacking Ensemble Model provides a more effective solution for leadership classification than the Neural Network Model. The superior performance of the ensemble method can be attributed to its ability to leverage multiple classifiers, effectively mitigating individual model weaknesses.

6. Conclusion

In conclusion, the study developed and validated a Stacking Ensemble machine learning model for classifying Leadership Styles as Low, Medium, and High, with an accuracy of 93.3%. The study further shows that advanced ensemble machine-learning approaches can reasonably interpret complex human behavioral data. The 'ensemble' of machine learners can interpret complex interdependencies between various aspects of leadership and present it understandably to all stakeholders.

The results of the study accordingly have significant practical implications. The study has developed a highly accurate tool that organizations can use to improve their assessments of their leaders. This could lead to a better understanding of leadership dynamics and better leadership development and performance. This tool could also spawn a series of other machine-learning developments that could similarly support HR departments and the organizations they serve. The applications could be far-reaching and transform the process companies assess, define, and develop their leaders. Moreover, the study avidly contributes to the academic discussion of machine learning applications in human resource management and partnership management strategies. The study also develops a practical tool for strategic human resource management. Further study may extend these benefits to practice and test future applications of the tool or tools as described above. Furthermore, his study demonstrates that ensemble learning approaches provide a more effective and interpretable leadership classification method than neural networks. However, deep learning remains a promising direction for further exploration, particularly with larger datasets and enhanced feature engineering.

References

[1] A. Siikaluoma, "LEADERSHIP PRACTICES SHAPED BY DIGITALIZATION". 2020, http://www.theseus.fi/handle/10024/343550

[2] A. Barthakur, V. Kovanovic, S. Joksimovic, Z. Zhang, M. Richey, and A. Pardo, "Measuring leadership development in workplace learning using automated assessments: Learning analytics and measurement theory approach," British Journal of Educational Technology, vol. 53, no. 6, pp. 1842–1863, Nov. 2022, doi: 10.1111/BJET.13218.

[3] L. Wei, "Genetic Algorithm Optimization of Concrete Frame Structure Based on Improved Random Forest," in 2023 International Conference on Electronics and Devices, Computational Science (ICEDCS), IEEE, Sep. 2023, pp. 249–253. doi: 10.1109/ICEDCS60513.2023.00051.

[4] K. Chen, H. Yao, and Z. Han, "Arithmetic optimization algorithm to optimize support vector machine for chip defect Identification," in 2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/M2VIP55626.2022.10041106.

[5] M. Kaur, C. Thacker, L. Goswami, T. TR, I. S. Abdulrahman, and A. S. Raj, “Alzheimer’s Disease Detection using Weighted KNN Classifier in Comparison with Medium KNN Classifier with Improved Accuracy,” in 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE, May 2023, pp. 715–718. doi: 10.1109/ICACITE57410.2023.10183208.

[6] Y. Fan and W. Lei, “Wind Speed Prediction Based on Gradient Boosting Decision Tree,” in 2022 International Conference on Big Data, Information and Computer Network (BDICN), IEEE, Jan. 2022, pp. 93–97. doi: 10.1109/BDICN55575.2022.00025.

[7] J. Faria, S. M. Azmat Ullah, and Md. R. Hasan, “Stroke Detection Through Ensemble Learning: A Stacking Approach,” in 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS), IEEE, Mar. 2024, pp. 01–06. doi: 10.1109/iCACCESS61735.2024.10499584.

[8] M. Yang, N. Slam, and Z. Zheng, “A Classification Model of Urban Fire Level with Stacking Ensemble Learning,” in 2023 IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE), IEEE, Dec. 2023, pp. 22–26. doi: 10.1109/ICEACE60673.2023.10442652.

[9] K. Kim and J. Jeong, “Multi-layer Stacking Ensemble for Fault Detection Classification in Hydraulic System,” in 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC), IEEE, Jul. 2022, pp. 341–346. doi: 10.1109/CSCC55931.2022.00066.

[10] V. N. Vasu, Surendran. R, Saravanan. M. S, and Madhusundar. N, “Prediction of Defective Products Using Logistic Regression Algorithm against Linear Regression Algorithm for Better Accuracy,” in 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), IEEE, Nov. 2022, pp. 161–166. doi: 10.1109/3ICT56508.2022.9990653.

[11] “How AI Is Transforming the Organization | MIT Press eBooks | IEEE Xplore.” Accessed: Apr. 09, 2023. [Online]. Available: https://c85689232ea394a8dc08a512c1f46793a2397178.vetisonline.com/book/9072232

[12] M. Sarstedt, “Revisiting Hair Et al.’s Multivariate Data Analysis: 40 Years Later,” The Great Facilitator, pp. 113–119, 2019, doi: 10.1007/978-3-030-06031-2_15.

[13] “REGRESSION ANALYSIS: EFFECTS OF LEADERSHIP STYLES ON ORGANIZATIONAL... | Download Table.” Accessed: Apr. 09, 2023. [Online]. Available: https://www.researchgate.net/figure/REGRESSION-ANALYSIS-EFFECTS-OF-LEADERSHIP-STYLES-ON-ORGANIZATIONAL-CULTURE_tbl4_272011654

[14] “Item Response Theory - Susan E. Embretson, Steven P. Reise - Google Books.” Accessed: Apr. 09, 2023. [Online]. Available: https://books.google.com.tr/books?hl=en&lr=&id=9Xm0AAAAQBAJ&oi=fnd&pg=PR1&dq=Embretson,+S.+E.,+%26+Reise,+S.+P.+(2013).+Item+response+theory+(2nd+ed.).+Psychology+Press&ots=Ec6UUtKXZi&sig=lX_g94yV0Phd1FMm1NO5mTusfok&redir_esc=y#v=onepage&q=Embretson%2C%20S.%20E.%2C%20%26%20Reise%2C%20S.%20P.%20(2013).%20Item%20response%20theory%20(2nd%20ed.).%20Psychology%20Press&f=false

[15] S. Oltedal and T. Rundmo, “Using cluster analysis to test the cultural theory of risk perception,” Transp Res Part F Traffic Psychol Behav, vol. 10, no. 3, pp. 254–262, May 2007, doi: 10.1016/J.TRF.2006.10.003.

[16] G. S. Insch, J. E. Moore, and L. D. Murphy, “Content analysis in leadership research: Examples, procedures, and suggestions for future use,” Leadersh Q, vol. 8, no. 1, pp. 1–25, Jan. 1997, doi: 10.1016/S1048-9843(97)90028-X.

[17] B. M. Doornenbal, B. R. Spisak, and P. A. van der Laken, “Opening the black box: Uncovering the leader trait paradigm through machine learning,” Leadership Quarterly, vol. 33, no. 5, Oct. 2022, doi: 10.1016/j.leaqua.2021.101515.

[18] E. Deopersaud and A. Capstone, “Natural Language Processing: Distinguishing Employee Views Toward Leadership,” 2022.

[19] Y. Zhang, S. Xu, L. Zhang, and M. Yang, “Big data and human resource management research: An integrative review and new directions for future research,” J Bus Res, vol. 133, pp. 34–50, Sep. 2021, doi: 10.1016/J.JBUSRES.2021.04.019.

[20] S. Bekesiene and S. Hoskova-Mayerova, “Decision tree-based classification model for identification of effective leadership indicators,” Journal of Mathematical and Fundamental Sciences, vol. 50, no. 2, pp. 121–141, 2018, doi: 10.5614/J.MATH.FUND.SCI.2018.50.2.2.

[21] S. Bekesiene, Š. Hošková-Mayerová, and P. Diliunas, “Identification of effective leadership indicators in the Lithuania army forces,” Studies in Systems, Decision and Control, vol. 104, pp. 107–122, 2019, doi: 10.1007/978-3-319-54819-7_9.

[22] C. Beyan, F. Capozzi, C. Becchio, and V. Murino, “Prediction of the leadership style of an emergent leader using audio and visual nonverbal features,” IEEE Trans Multimedia, vol. 20, no. 2, pp. 441–456, Feb. 2018, doi: 10.1109/TMM.2017.2740062.

[23] B. Chen, X. Chen, and H. Chen, “Construction of inclusive leadership knowledge graph based on Citespace and WOS core database,” Proceedings - 2022 International Conference on Machine Learning and Knowledge Engineering, MLKE 2022, pp. 337–340, 2022, doi: 10.1109/MLKE55170.2022.00071.

[24] S. Pongpaichet, K. Nirunwiroj, and S. Tuarob, “Automatic Assessment and Identification of Leadership in College Students,” IEEE Access, vol. 10, pp. 79041–79060, 2022, doi: 10.1109/ACCESS.2022.3193935.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 8, No. 1, 123-135, 2025
DOI: 10.35377/saucis...1610018

RESEARCH ARTICLE

Analysis of Queue Models in Simulation Applications

Abdullah Sevin[1*], Göktuğ Yaman[1], Durdali Atılgan[1]

[1]Department of Computer Engineering, Sakarya University, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Abdullah Sevin, Sakarya University,
Department of Computer Engineering,
Sakarya, Türkiye
asevin@sakarya.edu.tr

Article History:
Received: 07.01.2025
Revised: 13.03.2025
Accepted: 18.03.2025
Published Online:27.03.2025

ABSTRACT

With the advancement of technology, the speed and efficiency of information processing have become vital for meeting the growing demands of individuals and organizations. As time constraints increase, rapid and accurate access to information has gained critical importance. To address these challenges, organizations in the business and public sectors are increasingly relying on simulation methods, a core area of computer science, to optimize their responses to customer demands. Simulation provides a robust framework for analyzing and modeling complex systems. Within this framework, queue theory plays a central role by examining how systems handle incoming demands and offering insights into improving resource utilization, minimizing waiting times, and enhancing overall performance, particularly in service industries.

This study provides a detailed analysis of queue theory, exploring its fundamental principles, key features, and various models. Additionally, a comparative evaluation of different queueing models is conducted through simulation, assessing key performance metrics such as server utilization, maximum queue length, and average response time. The results indicate that model selection significantly impacts system efficiency, with certain models exhibiting superior performance under specific conditions. These insights equip organizations with the tools to develop more effective strategies, optimize their processes, and enhance responsiveness to evolving demands.

Keywords: Simulation, Queue theory, Performance metrics, Resource utilization, System modeling

1. Introduction

In today's rapidly evolving and highly connected world, waiting has become unavoidable, impacting individuals and organizations across various domains. Whether it is a customer waiting in line to order coffee, a patient waiting for their appointment at a hospital, or vehicles delayed at traffic lights, waiting is a universal phenomenon. As populations grow and societal demands become increasingly complex, businesses, public institutions, and other organizations face significant challenges in managing waiting times efficiently while maintaining high-quality service. The consequences of waiting extend beyond mere inconvenience. Excessive delays can lead to customer dissatisfaction, loss of business opportunities, and reduced operational efficiency. These inefficiencies in critical sectors such as healthcare, transportation, and logistics can also have broader societal and economic implications. Therefore, understanding and optimizing the dynamics of waiting systems is essential for enhancing service quality, resource utilization, and overall system performance.

Queue theory, a branch of operations research and applied mathematics, provides a robust framework for analyzing and managing waiting lines in various systems. By modeling the behavior of customers, service mechanisms, and system capacity, queue theory offers valuable insights into how organizations can minimize waiting times, improve resource allocation, and enhance customer satisfaction. From service industries like banking and retail to technical applications in computer networks and telecommunications, queue theory is critical in addressing the challenges posed by increasing demand and limited resources. Simulation, as a complementary tool, further enhances the practical application of queue theory. Using computational models, simulation allows organizations to mimic real-world processes, test different scenarios, and evaluate the effectiveness of strategies in a controlled environment. This combined approach enables data-driven decision-making, empowering organizations to develop efficient and adaptable solutions.

This study delves into the fundamental principles, characteristics, and applications of queue theory, particularly emphasizing its integration into simulation studies. By exploring performance metrics and real-world applications, the study aims to highlight the critical role of queue theory in improving organizational processes, meeting customer demands, and addressing

Cite as: A. Sevin, G. Yaman, and D. Atilgan, “Analysis of queue models in simulation applications,” *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 1, pp. 123-135, 2025, doi: 10.35377/saucis...1610018

the complexities of modern systems. As industries continue to evolve, the insights provided by queue theory and simulation remain indispensable for ensuring sustainability and competitiveness.

2. Literature Review

Queue theory, a fundamental branch of operations research, has been extensively studied across various disciplines due to its critical role in analyzing and optimizing waiting systems. From its theoretical foundations to practical applications, queue theory provides valuable insights into system performance, resource utilization, and customer satisfaction. Numerous studies have explored its use in diverse fields such as healthcare, transportation, telecommunications, and service industries, demonstrating its versatility and effectiveness. This section reviews the existing literature on queue theory, highlighting key contributions, practical implementations, and emerging challenges. Particular attention is given to integrating queue theory with simulation techniques, which offer enhanced capabilities for addressing dynamic and complex real-world systems.

Sztrik aimed to establish a technical foundation by addressing the subject of queue theory with its fundamental aspects [1]. Ulaş conducted a theoretical examination addressing a parallel-channel queue system by using the birth-death process to determine the parameters of a parallel queue system. Additionally, a two-heterogeneous-channel queue system was analyzed using a Poisson process [2]. Şimşek implemented a single-service channel model, examining the passages of tankers and other ships through the Istanbul Strait. The results indicated intense tanker traffic in the Istanbul Strait, with an increasing number of passing tankers [3]. Kiremitçi aimed to efficiently create transportation plans for ships and achieve cost savings by addressing the vehicle route planning problem in the maritime transportation sector [4]. Parlak focused on queue systems in healthcare institutions and the Central Physician Appointment System (MHRS). The study analyzes the effectiveness of appointment systems, information level, satisfaction level, and the benefits they provide patients accessing healthcare services [5].

Batur provided detailed information on the functioning of the evolving air transportation sector and aimed to offer solutions to the problems arising in this industry. The research included examples from Türkiye and worldwide air transportation [6]. Chaves conducted a study on two-stage queue models developed for the aviation sector. A single-channel multi-phase queue model was examined in the first stage, and in the second stage, a general distributed single-phase model was proposed [7]. Adan and Resing examined queue model examples in detail, providing technical information about mathematical modeling [8]. Ertuğrul and others studied queue theory and analyzed customer waiting queues at branches of two banks operating in the city of Denizli using queue theory [9]. Majid and Manoharan compared the M/M/c queue model from two different perspectives, comparing the model they developed [10]. Yıldız and Arslan examined students' waiting queues in the Central Cafeteria of Düzce University, calculating the average performance of the system [11]. Maragathasundari and others studied non-Markovian queue models for aircraft control systems [12].

Lan and Tang performed stability analysis by evaluating the probability of problem occurrence in the Geo/Geo/1 queue system [13]. Kim and others created a simulation-based queue model for unmanned and automatic systems by examining potential airport queue models [14]. Anosike and Nneka evaluated the adequacy of the Nigeria Nnamdi Azikiwe International Airport (NAIA) for current demand by mathematically modeling the passenger queue problem [15]. Smith analyzed M/M grouped queue models, considering rounding errors in numerical calculation situations [16]. Jawab and others aimed to optimize and enhance the passenger queue model at Fez-Sais Airport in Morocco [17]. Girginer and Şahin investigated the waiting queue problem during the use of sports equipment in sports facility operations through simulation methods. The study simulated the system in a sports facility using data collected over 45 days to identify factors affecting capacity issues [18]. Kumar and others studied the performance parameters of the M/M/1/N feedback customer queue [19].

Artalejo and Falin analyzed the renewable queue model, where a customer cannot receive service based on limited capacity, density, and other reasons. They compared M/G/1 and M/M/c queue models [20]. Smith and others examined a queue system in the M/M/N/N format where two types of users, prioritized and non-prioritized, attempt to reach N resources. This study aims to model future portable radio systems [21]. Ibe and Isijola examined a queue system with multiple vacations following the busy period in the M/M/1 queue model. They interpreted the differences between the model with zero duration, where no customer is served after the busy period, and the model with nonzero duration, where service can be provided immediately after the busy period [22]. Vandaele and others examined a traffic flow analysis traditionally based on experimental (empirical) methods and developed an analytical queue model to perform this analysis. They described the model established for traffic control and density analysis on a main road [23]. Using Bessel functions and probability techniques, Kumar examined the Markovian multi-phase queue system (M/M/c). According to the results, the established model could play a significant role in systems such as emergencies in hospitals and call centers [24].

Wang and Zhu proposed a dynamic queue model solved using the multiple-server model for excessive demand. They used an assumption involving customers joining the queue early or late, differentiating evaluation and cost criteria. They examined this model in systems such as shopping malls, restaurants, and highways [25]. Sharma and others explained the fundamentals and usage of queue theory, offering information on mathematical modeling [26]. Christien and others studied the sequencing of landing aircraft with different operational procedures in the three major ports of Europe. This study aimed to shed light on solving complex operations during peak hours [27]. Mehri and others worked on explaining basic queue models and focused on mathematical modeling [28]. Chew worked on a new modeling of the standard M/M/1 queue structure, proposing a queueing model in this new structure that includes single-phase and real/virtual customer types. They compared this model to the standard model with simulation support [29]. In his research book, Winston focused on the fundamentals of queue theory and mathematical modeling [30].

Awasthi studied the M/M/1/K finite capacity queue model on customer behaviors, including balking (customer leaving if the queue is too long) and reneging (customer leaving if the queue is moving too slowly) [31]. Som and Seth developed a Markov queue system for single-phase and finite arrivals. Addressing the encouraged arrival model, they created a model reflecting the effects of discounts and attractive offers implemented by companies. They examined the model numerically and through simulation [32]. Jhala and Bhatwala proposed a model to reduce airport queues and increase customer satisfaction [33]. Çevik and Yazgan developed a queueing model to determine customer waiting times in a bank and calculated the average efficiency of the system [34]. Poongodi and Muthulakshmi proposed a control chart method for the infinite capacity M/M/s queue model. This method aims to predict possible waiting times, maximum waiting times, and minimum waiting times in advance, considering customer satisfaction [35]. Using the operator analytics technique, Massey designed an M/M/1 Markov queue model for non-static situations. Using a common parameter to determine arrival and service rates, this technique reveals dynamic asymptotic behavior different from broad time interval analysis [36]. Idris and others analyzed data to determine flow constraints delaying departure operations at Boston Logan International Airport, comparing it with other airports [37].

Shone and others worked on the optimal control and modeling of aircraft queues at runway thresholds, providing a literature review of related studies [38]. Tiwari and others studied the M/M/1/N queue model with Poisson arrivals and exponential service times, researching the expected total minimum cost [39]. Thiagarag and Seshaiah worked on the landing aircraft queues, exploring the limits of analytical approaches and making inferences about how simulation methods should be used [40]. Bertsimas and Nakazato examined the MGEL/MGEM/1 queue model using the MGE method, a subclass of the Erlang distribution. They calculated the queue length distribution and waiting times using the first-come, first-served principle [41]. Karapetyan and others worked on the pre-departure sequencing of aircraft, conducting a study on the goals, requirements, and real-time decision mechanism of the system developed using an algorithm [42]. Aydın worked on determining the landing order and times of aircraft in the air. The study suggested that the aircraft scheduling problem could be solved using metaheuristic methods, achieving an optimal solution, and shedding light on future research related to ACP [43]. Arslan focused on efficiently using aircraft gates and optimizing the gate assignment process by considering factors that could affect passenger satisfaction [44].

Doğan worked on minimizing costs arising from unexpected flight routes by airline companies. For this purpose, they developed two different decision support systems by analyzing meteorological data retrospectively and making future predictions [45]. Fatima and others present the efficiency of patient management in healthcare institutions by comparing traditional queuing systems with modern technological advancements. The study highlights how integrating innovative technologies can improve operational efficiency, minimize delays, and enhance patient satisfaction [46]. Anita and others present a Markovian two-stage tandem queueing system with retrial policy and server vacation, where customers undergo service at both stations and those unable to be served immediately retry after a random time. The system's performance is analyzed through birth-death balance equations, and the effects of various parameters are illustrated graphically [47]. Dhibar and Jain analyzed a Markovian retrial queueing system with two types of customers, unreliable servers, and Bernoulli feedback, focusing on customer decisions to join or balk based on service profit and delay cost. The study employs Chapman-Kolmogorov equations and the probability-generating function method to derive performance metrics, and optimization techniques like PSO and GWO are used for cost optimization and QoS enhancement, with results validated through numerical simulations [48].

Amjath and others review past research on the performance evaluation and optimization of Material Handling Systems (MHSs) using queueing network models. It comprehensively analyzes relevant research questions and adopts systematic literature review, bibliometric, and content analysis techniques to offer insights for scholars and practitioners in material logistics [49]. Ambika and others examine a queueing model in production management with working vacations and Bernoulli vacations, where the manufacturing unit operates at a reduced rate during maintenance phases. Mathematical techniques are used to calculate transient state probabilities, and numerical examples illustrate the impact of these dynamics on production management [50]. Çakmak and Torun evaluate the performance of different queue management algorithms in LTE networks through simulations conducted in the NS-3 environment [51]. Çakmak and Albayrak comprehensively analyze various active queue management techniques used in mobile communication networks. It examines different algorithms, their working principles, and their impact on network performance [52]. In another work, they also analyze the performance of queue management algorithms between the Remote-Host and PG-W in LTE networks [53]. Gündoğar and others analyze a spring mattress manufacturing line to identify and eliminate bottlenecks using the Theory of Constraints (TOC). By applying simulation-based methods in Arena 13.5, they tested various scenarios to optimize production flow [54].

Overall, the reviewed literature demonstrates the broad applicability of queueing models across various domains, highlighting their potential to enhance operational efficiency, minimize waiting times, and improve customer satisfaction. Similarly, our study compared simulations of different queueing models to evaluate their performance under various conditions. By analyzing models such as M/M/1, M/D/1, and others, we aimed to provide deeper insights into their practical applications and identify which models most effectively optimize service systems. This comparative analysis offers valuable contributions to understanding queueing theory and its role in improving operational processes across diverse industries.

3. Queueing Theory

3.1. What is Queueing Theory?

Queueing theory is a branch of applied mathematics and operations research that analyzes waiting lines or queues. It studies the behavior of customers arriving for service, the processes they undergo, and the factors affecting system efficiency. By examining these dynamics, queueing theory provides valuable insights into optimizing service processes and minimizing delays. At its core (Figure 1), queueing theory evaluates key components such as arrival rates (the frequency of customer arrivals), service rates (the speed at which services are provided), and queue disciplines (rules governing the order of service). Ordinary queue disciplines include First-In-First-Out (FIFO), Last-In-First-Out (LIFO), and priority-based approaches, each suited to different operational contexts [55].

For instance, FIFO is often used in retail checkout lines, while priority-based systems are common in emergency healthcare services. This theory is not only concerned with the mathematical analysis of queues but also with their practical implications. By modeling waiting systems, organizations can improve resource utilization, reduce waiting times, and enhance customer satisfaction. The queueing theory finds applications in diverse fields, including banking, transportation, telecommunications, and healthcare, making it an essential tool for academic research and practical decision-making. As modern systems grow more complex, queueing theory is increasingly combined with simulation techniques to address dynamic and unpredictable scenarios. This integration enables organizations to test and refine strategies in virtual environments, ensuring optimal performance in real-world applications.

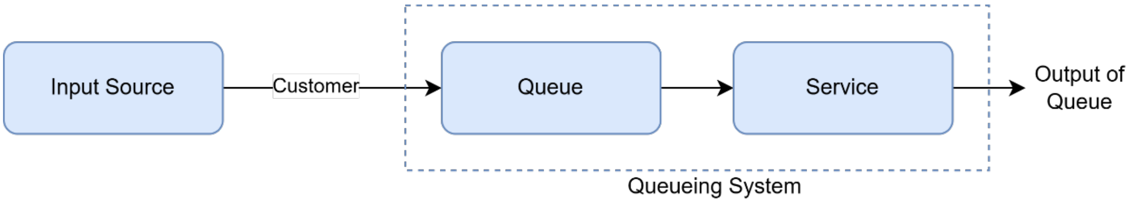


Figure 1. Basic Queueing System Schematic

3.2 Poisson Process

Let $\{ N(t): t \geq 0\}$ be a counting process and $\lambda > 0$. A counting process satisfying the following properties is called a Poisson process with rate λ [56]:

(Property 1) Independent Increments: The process has independent increments. The number of events appearing in non-overlapping time intervals is independent of each other. For any ordered time, indices $0 \leq t_1 < \cdots < t_n$, the random variables $N_{t_1}, N_{t_2} - N_{t_1}, \ldots, N_{t_n} - N_{t_{n-1}}$ are independent.

(Property 2) Poisson Distribution: The number of events occurring within a unit time interval follows a Poisson distribution with an average rate of λ. Additionally, the number of events appearing within a time interval of length t follows a Poisson distribution with a mean of λt:

$$\Pr\,(N_{t+s} - N_s = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}, k = 0,1,2, \ldots \tag{1}$$

(Property 3) Considering a small positive real number $h \geq 0$: the possibility of a single event appearing in the time period $[t, t+h)$ is:

$$\Pr\,(N_{t+h} - N_t = 1) = \lambda h + o(h) \tag{2}$$

On the other hand, the probability of at least two events occurring in $[t, t+h)$ is:

$$\Pr\,(N_{t+h} - N_t \geq 2) = \Pr\,(N_h \geq 2) = o(h) \tag{3}$$

The probability of no events occurring in $[t, t+h)$ is:

$$\Pr\,(N_h = 0) = 1 - \lambda h - o(h) \tag{4}$$

These probability equations imply that the likelihood of a substantial number of events happening in small time intervals tends to be small.

Theorem: If N_t has a Poisson distribution and X_n represents the time among the (n−1)th and nth events, then the inter-arrival times are independent and follow an exponential distribution with a mean of $1/\lambda$ [57].

3.3. Characteristics of Queueing Theory

Fundamentally, a queue consists of two main components:

- The side requests the service (also known as the arrival side or customers).
- The side provides or completes the service (also known as the service side or server).

The elements of the queue system are further detailed in the following subsections.

3.3.1. Called Population

The population of possible customers is known as the calling population. Although the number of potential customers is technically finite, it is often assumed to be infinite to simplify the model. This assumption is reasonable when the potential customer base is large, particularly if the number of customers currently receiving or waiting for service represents only a slight fraction of the total population. Assuming an infinite population implies that the arrival rate of customers is unaffected by the number of customers already in the system, allowing the arrival rate to remain constant over time [58].

3.3.2. System Capacity

Queue systems often limit the number of customers occupying the waiting line or the system. If the system reaches capacity, incoming customers cannot enter and are immediately redirected to the calling population. However, some systems are designed with infinite capacity, allowing unlimited customers to enter. In systems with restrained capability, a distinction is made between the arrival rate, the number of customers arriving per unit time, and the effective arrival rate, which represents the number of customers per unit time that enter the system [59].

3.3.3. Arrival Process

For infinite population models, the arrival process is characterized by the time intervals between consecutive customer arrivals. Arrivals can occur at planned or random times. In the case of random times, the intervals between arrivals are typically characterized by probability distribution. Customers can also arrive individually or in groups, with the party being of a fixed or random size. For random arrivals, the Poisson arrival process is the most significant model and the primary focus of our consideration. Let A_N, represent the inter-arrival time for the N, then between the $(N-1)_{th}$ and N_{th} customer, it follows an exponential distribution with an average of $(1/\lambda)$ per unit of time. The arrival rate is $\lambda, which$ denotes the average number of customers arriving per unit of time. Over a long time, interval T, the total number of arrivals follows a Poisson distribution with an average of λT customers [60].

3.3.4. Queue Behavior and Queue Discipline

Queue performance refers to customers' actions and decisions while waiting for service. During this waiting period, customers may sometimes decide not to join the queue, which is known as balking. Customers who have joined the queue might leave before receiving service, which is called reneging. Additionally, when multiple queues are available, customers might switch to a different queue if they perceive it to be moving faster, a behavior called jockeying. Queue discipline refers to the rules determining the order in which customers are selected for service. One of the most ordinary queue disciplines is FIFO, where the customer who has been waiting for the longest is served first. Another less common discipline is LIFO, where the most recent customer who joined the queue is served first. Service-In-Random-Order (SIRO) involves selecting customers for service randomly without regard to their arrival time or position in the queue. In some systems, customers may be served based on priority. In such cases, customers with higher priority (e.g., VIPs or urgent cases) are served before others, regardless of their position in the queue [61].

3.3.5. Service Times and Service Mechanism

The service times for consecutive arrivals are denoted as $S_1, S_2, \ldots$ can either be constant or random. In cases where they are random, $\{S_1, S_2, S_3, \ldots\}$ is typically modeled as a sequence of independent and identically distributed random variables. While service times for customers of the same type, class, or priority often share the same distribution, customers of several types may have distinct service time distributions. Furthermore, service times in some systems may vary based on factors such as the time of day or the length of the waiting line. A queue system comprises a network of service counters and interconnected queues. Each service center contains a certain number of servers, denoted b C, operating in parallel. When a customer reaches the front of the queue, they are assigned to the first available server. The parallel service mechanism can take different forms: a single server $(C = 1)$ or multiple servers $(1 < C < \infty)$[62].

4. Analysis of Queue Models in Simulation Applications

4.1. Representation of Queue Models

There are numerous queue representations, and a six-character notation is commonly used to represent these models. The first three characters of this notation were proposed by Kendall in 1953 and signify the arrival distribution, service time distribution, and the number of servers (channels). A. M. Lee later added the fourth and fifth characters in 1966. In 1968, Hamdy A. Taha defined the last character. This notation is often used in software to describe and define queue models. The Kendall notation summarizes three key factors: arrival distribution, service time distribution, and the number of servers, represented as A/B/C/D/E/F. These characters correspond to the following [63]:

- A: Distribution of arrivals
- B: Distribution of service times
- C: Total parallel servers (channels)
- D: Queue rules
- E: System capability
- F: Population size,

Standard notations that can replace A and B include M (exponential), D (constant or deterministic), E_k (Erlang), and G (general). These notations represent the characteristic distribution type for arrivals and service times.

4.2. The Importance of Queue Theory for Simulation

Queue theory is a crucial concept in simulation studies, especially when simulations assess and enhance the performance of businesses, organizations, or systems. The following aspects are of particular significance in highlighting the importance of queueing theory in the context of simulation [64]:

1. Analysis of Waiting Times: Queue theory is utilized to understand how waiting and queues form within a system. When used to model a specific process or point in a system during a simulation, it helps evaluate how waiting occurs and assesses its impact on process efficiency.
2. Optimization of Resource Utilization: Queue theory analyzes the effective utilization of resources such as personnel, machinery, service points, etc. It can be used in simulation models to develop strategies for enhancing the efficiency of specific resources or service points and optimizing overall capacity.
3. Improvement of Service Quality: Queue theory is valuable for evaluating and improving the service quality of a system. Since waiting times directly impact customer satisfaction, using queue theory through simulations allows for optimizing service processes, ultimately enhancing the overall customer experience.
4. Determination of Performance Metrics: In simulation models, queue theory can be used to identify specific performance metrics. These metrics, such as average waiting time and resource utilization efficiency, can be determined to assess and improve system efficiency.
5. Risk Analysis: Queue theory applies to understanding how a system behaves in certain scenarios and evaluating potential risks. Employing queue theory through simulations provides insights into how a system would respond in specific scenarios.

In conclusion, queue theory is a powerful tool for analyzing systems' effectiveness, performance, and resource utilization within simulation models. This theory can assist businesses in optimizing their processes, making more efficient use of resources, and ultimately improving customer satisfaction.

4.3. Applications of Queue Theory

Queue theory has diverse applications across multiple sectors, which is crucial in optimizing processes and improving efficiency. Some key areas where queue theory is commonly applied include [65]:

1. Service Sector: In service industries such as banks, hospitals, and restaurants, queue theory is employed to analyze and optimize customer service processes. It reduces waiting times, improves service quality, and manages staff capacity.
2. Transportation and Logistics: Queue theory is applied in transportation and logistics sectors, including traffic management, airport flight scheduling, and bus terminal operations. It is used to evaluate the effectiveness and efficiency of transportation systems, aiming to reduce waiting times and develop strategies for more effective resource utilization.

3. Telecommunications: Queue theory is utilized in telecommunication systems such as call centers, data transmission lines, and internet service providers. It helps evaluate network performance, optimize capacity, and enhance service quality.
4. Manufacturing and Industrial Processes: In industrial settings, including manufacturing lines, inventory management, and order processing, queue theory is applied to improve production efficiency and optimize material flow.
5. Computer Science: Queue theory is used in computer science fields such as computer networks, data transfer between processors, and database management. It helps evaluate system performance and response times.
6. Finance and Banking: Queue theory finds application in areas like bank queues, ATMs, and the processing of financial transactions. It aims to improve customer service, minimize waiting times, and increase transaction capacity.
7. Healthcare Services: In healthcare, queue theory is applied to emergency services, appointment scheduling, and treatment processes. Its goal is to reduce patient waiting times and assist in the more effective management of healthcare services.

These application areas highlight the versatility of queueing theory as a valuable tool for optimizing system performance across various industries. For instance, queuing theory can be employed in the banking sector to optimize customer service operations. Banks can adjust staffing levels by analyzing factors such as waiting times and transaction volumes to ensure customers are served more efficiently, particularly during peak hours. This reduces waiting times, improves customer satisfaction, and more effective resource utilization. In logistics, queuing models can enhance material handling and warehouse operations. Logistics companies can reduce bottlenecks, minimize delays, and improve throughput by optimizing the flow of goods and managing inventory more effectively. Queuing theory in this context contributes to better scheduling of tasks and resource allocation, leading to more streamlined operations and cost savings. By applying queuing theory in these real-world settings, organizations can achieve tangible benefits such as increased operational efficiency, reduced costs, and improved customer satisfaction, ultimately fostering a more competitive and sustainable business model.

4.4. Queue Theory Performance Metrics

Various sources may use various terms to describe the metrics used to evaluate the performance of queue models, but these metrics essentially measure the same core concepts. Waiting time refers to the duration customers spend waiting before receiving service. In contrast, the time spent on the system is the total time a customer spends, including waiting and service time. The distribution of the number of customers in the system reflects the number of customers present at any given time. Workload distribution is the total service time required for waiting customers and the remaining service time for the customer being served. The service station's busy time refers to the continuous duration during which the service station remains occupied and operational. These metrics provide valuable insights into a queueing system's overall performance and efficiency [66].

Key performance metrics include average waiting time and time spent in the system. These metrics are significant in studies aimed at understanding system performance. The operational characteristics of steady-state queue systems can be calculated using various formulas, which help assess how the system performs under different conditions.:

- λ (Average Arrival Rate): The mean number of customers that arrive on the system within a unit of time.
- μ (Average Service Rate): The mean number of customers that serve in the system within a unit of time.
- p (Average System Utilization Rate): ${}^{\lambda}/_{\mu}$, the mean system utilization rate.
- L (Average Number of Customers in the Queue System): $\lambda/(\mu - \lambda)$, The mean number of customers present in the queueing system at any given time.
- L_q (Average Number of Customers Waiting in the Queue): pL, The mean number of customers waiting in the queue for service at any given time.
- W (Average Time Spent in the System): $1/(\mu - \lambda)$, The total average time a customer spends in the system, encompassing both the waiting time and the service time.
- W_q (Average Time Spent Waiting in the Queue): pW, the average time spent on delay in the queue.
- P_n (Probability of having n Customers in the Queue): $(1 - p)p^n$, the possibility of having *n* customers in the queue at any given time.

These formulas and models should be used under the condition that the service rate is greater than the arrival rate ($\mu > \lambda$). Otherwise, the queue can excessively lengthen. Therefore, it is essential to ensure that this condition is met before using these formulas and models.

4.5. Performance Evaluation of Queueing Models

In this section, we compare several widely used queueing models based on three key performance indicators: Server Utilization, Maximum Queue Length (MQL), and Average Response Time (ART). These parameters are essential for evaluating the efficiency and effectiveness of different queuing systems under varying conditions. The queueing models considered in this comparison include the following:

- M/M/1: Single-server queue with exponential inter-arrival and service times.
- M/D/1: Single-server queue with exponential inter-arrival times and deterministic service times.
- M/N/1: Single-server queue with exponential inter-arrival times and Normal-distributed service times.
- M/U/1: Single-server queue with exponential inter-arrival times and uniform service time distribution.
- M/Weibull/1: Single-server queue with exponential inter-arrival times and Weibull-distributed service times.
- M/LogNormal/1: Single-server queue with exponential inter-arrival times and LogNormal-distributed service times.
- M/Erlang/1: Single-server queue with exponential inter-arrival times and Erlang-distributed service times.

In queueing theory, the calculation of key performance metrics such as server utilization, ART, and MQL generally follows specific formulas. Still, the exact calculation depends on the type of queueing model and its respective characteristics. Server Utilization is typically calculated using the formula $\rho = \frac{\lambda}{\mu}$, where λ is the arrival rate and μ is the service rate. This formula holds for most queueing models, although slight variations may exist depending on the system's characteristics. ART is calculated based on Little’s Law and specific model characteristics. For many models, the general formula is $W = \frac{1}{\mu - \lambda}$, with modifications made for different service distributions (e.g., Poisson, deterministic, Weibull). MQL is typically computed using $L_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$, but in models with capacity restrictions (such as M/N/1), this formula must be adjusted to account for the system's maximum capacity. Each queueing model (e.g., M/M/1, M/D/1, M/N/1) introduces specific variations in these formulas based on factors such as arrival rates, service rates, and the type of service distribution. Models with deterministic, uniform, or specialized distributions (e.g., Weibull or LogNormal) require tailored calculations that reflect these distribution characteristics. While general formulations exist, the exact metrics depend on the specific queueing model and its associated parameters.

The analysis uses a mean inter-arrival time of 4.5 minutes, representing the average time between successive arrivals. The mean service time, the average time a server takes to serve a customer, varies across three values: 2.5, 3.2, and 4. The Sigma value, representing the standard deviation of the service time distributions, is also set to 0.6. The distribution parameters are adjusted based on the mean service time for models involving non-exponential service times, such as Weibull, LogNormal, and Erlang. Specifically, for each distribution, the shape and scale parameters (for Weibull), the mean and sigma (for LogNormal), and the k and lambda values (for Erlang) are tailored to match the corresponding service time characteristics. The values for the mean service time (2.5, 3.2, and 4) and sigma (0.6) were chosen to represent different service scenarios with varying system utilization and congestion levels. The mean service time values allow for analyzing queueing performance under different operational conditions: 2.5 represents a relatively low service time, indicating a faster processing rate with lower congestion; 3.2 represents a moderate service time, balancing efficiency and queuing effects; and 4 represents a higher service time, simulating a more congested system with longer waiting times. These values ensure meaningful comparisons across different queueing models by examining their impact on key performance metrics such as server utilization, maximum queue length, and average response time under different conditions. The selected sigma value of 0.6 also introduces controlled variability in service times for models incorporating stochastic distributions, such as LogNormal and Weibull, ensuring realistic variations without extreme deviations. These parameters were carefully determined to comprehensively evaluate queueing behavior while maintaining system stability and producing interpretable results. To ensure high accuracy and minimize the impact of random fluctuations, the system is simulated with 1,000,000 arrivals. This large sample size guarantees that the results are statistically reliable and represent real-world conditions. The performance metrics, including server utilization, MQL, and ART, are computed for each model under these conditions. The results will provide valuable insights into the strengths and limitations of each queuing model, offering a basis for selecting the most appropriate model for different system requirements.

Figure 2 presents the ART for various queueing models at three mean service times: 2.5, 3.2, and 4 minutes. The comparison of ART across various queueing models reveals significant insights into the behavior of the systems under different service time distributions. The M/M/1 model consistently shows the highest ART, especially as the mean service time increases. This can be attributed to its exponential distribution, which leads to high arrival and service times variability, causing greater waiting times. In contrast, models with more predictable service time distributions, such as M/D/1, M/N/1, and M/U/1, exhibit lower response times. The M/D/1 model, with its deterministic service times, performs particularly well in maintaining stable response times. Models utilizing more flexible service time distributions, like M/Weibull/1 and M/LogNormal/1, show slightly higher response times than the deterministic models but still offer improvements over M/M/1. The Weibull and

LogNormal distributions capture more complex real-world service time behaviors, leading to more accurate performance predictions in certain systems. Lastly, the M/Erlang/1 model exhibits lower ART in most scenarios, demonstrating improved performance in handling service time variations. Its ability to achieve relatively lower response times suggests that it may offer advantages in systems where reducing waiting times is a priority. The results emphasize the importance of selecting an appropriate queueing model based on system characteristics. While M/M/1 provides a baseline for simple scenarios with exponentially distributed service times, models such as M/D/1, M/Weibull/1, and M/Erlang/1 show varying performance characteristics that may benefit systems with different service time distributions.

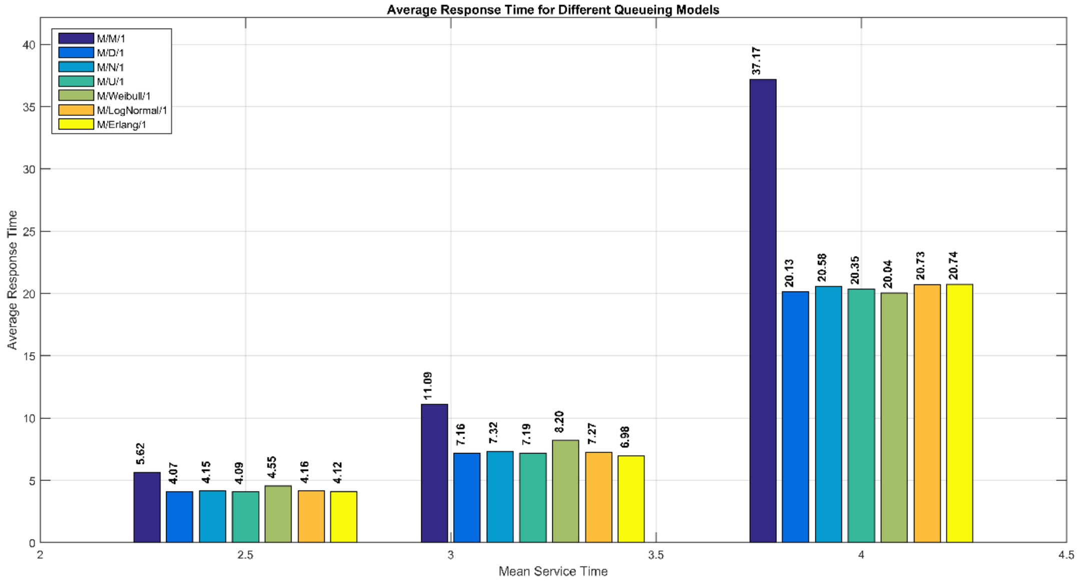


Figure 2. The Average Response Time of the Models

As the mean service time increases, the MQL in Figure 3 tends to grow for all models, reflecting the direct relationship between service time and queue length. However, the performance of different queue models varies. The M/M/1 model exhibits the highest MQL across all service times, with values of 21, 37, and 85, respectively. This is due to the variability introduced by the exponential distribution of service times, which leads to larger fluctuations and higher queue lengths. On the other hand, the M/D/1 model shows the best performance, with the lowest MQL at each mean service time value: 12, 20, and 42. This is expected due to the deterministic service times in the M/D/1 model, which result in more predictable and stable system behavior, reducing the chances of large queues building up. The M/N/1 and M/U/1 models display moderate increases in MQL as service time increases, but they still perform better than the M/M/1 model, with values of 13, 21, and 49 for M/N/1, and 15, 24, and 45 for M/U/1. These models exhibit better control over the queue than the M/M/1 model but are not as efficient as M/D/1. The M/Weibull/1 and M/LogNormal/1 models demonstrate moderate MQL values that are higher than M/D/1 but lower than M/M/1, showing that the flexibility of these distributions in capturing variability leads to slightly higher queue lengths (17, 26, 54 for M/Weibull/1 and 12, 21, 45 for M/LogNormal/1). Finally, the M/Erlang/1 model, while still relatively efficient in managing queue lengths, performs slightly worse than M/D/1, with MQL values of 13, 20, and 47. The results indicate that the Erlang distribution leads to shorter queue lengths than the M/M/1 and M/Weibull/1 models in the tested scenarios. This suggests that the choice of service time distribution plays a significant role in queue behavior, highlighting the need for careful model selection based on system requirements.

When analyzing the server utilization in Figure 4, values across different queue models, we observe that the values for all models are quite similar, particularly when the mean service time increases. This suggests that the models operate with relatively consistent utilization rates, with minimal differences between the deterministic and stochastic models. The M/M/1, M/D/1, M/N/1, M/U/1, M/LogNormal/1, and M/Erlang/1 models all show a steady increase in server utilization as the mean service time grows, with values progressing from 0.5552 to 0.7119, and then to 0.8871 as the service time reaches 4. The increased utilization across these models is expected because longer service times mean the server is busy for a greater portion of the time. However, the M/Weibull/1 model shows slightly lower utilization than the other models. At mean service times of 2.5, 3.2, and 4, the utilization values for M/Weibull/1 are 0.5234, 0.6694, and 0.8355, respectively. This is likely due to the shape of the Weibull distribution, which, depending on the shape parameter, can produce more variability in service times, leading to periods where the server is idle more often than in models with deterministic or less variable distributions.

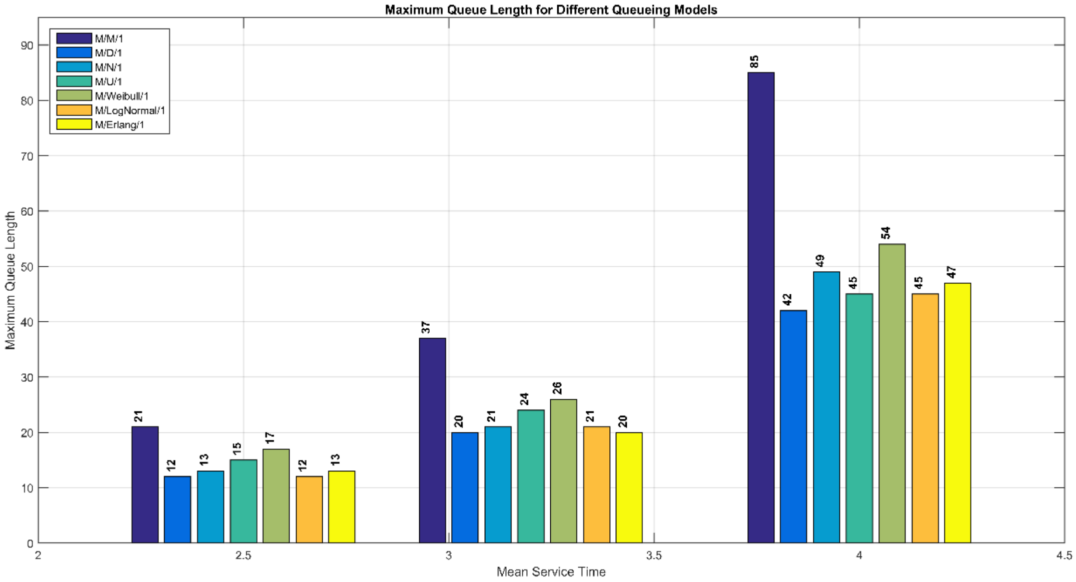


Figure 3. Maximum Queue Length of the Models

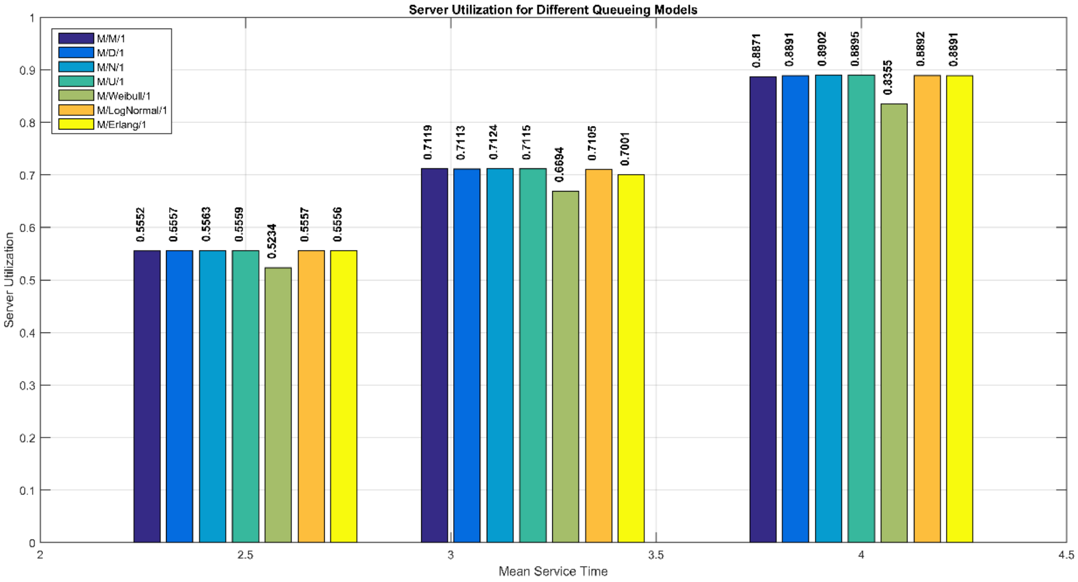


Figure 4. Server Utilization of the Models

5. Conclusions

This article comprehensively examines queueing theory and its role in simulation applications. It offers a detailed exploration of fundamental concepts, including input queues, output queues, service points, and waiting lines, laying a solid foundation for understanding the core elements of queueing theory. The study also delves into mathematical models for assessing processor speed, advanced queueing systems, and overall performance.

In addition to covering the basic principles of queueing theory, this article emphasizes its mathematical underpinnings, offering readers a well-rounded understanding of the subject. A simulation-based application compares the performance of different queueing models under varying service time conditions. The results reveal that models such as M/D/1 and M/Erlang/1 generally lead to shorter queue lengths and reduced average response times compared to M/M/1 in specific scenarios. This highlights the significance of choosing an appropriate model based on system requirements to optimize efficiency and performance.

In conclusion, this study demonstrates that queueing theory is a powerful tool for simulating and optimizing system performance when applied correctly. The findings underscore the importance of selecting suitable queueing models based on system characteristics, as different models exhibit distinct advantages in handling service time variations. By integrating theoretical insights with simulation-based analysis, this study provides a valuable resource for researchers and practitioners seeking to enhance system performance through queueing theory applications.

References

[1] Sztrik, J. (2016). *Basic queueing theory*. OmniScriptum GmbH, KG, Saarbrucken, Germany: GlobeEdit.

[2] Ulaş, M. (2007). İki hizmet kanalına sahip kuyruk sistemlerinin analizi. *Fen Bilimleri Enstitüsü-İstatistik Anabilim Dalı.*, Fırat Üniverstiesi, Yüksek Lisans Tezi.

[3] Şimşek, H. (2004). *Kuyruk teorisinin İstanbul Boğazı tanker ve gemi geçişleri ile Haydarpaşa Limanı konteyner terminaline uygulanması* İstanbul Teknik Üniversitesi, Doktora Tezi.

[4] Kiremitçi, S., (2011). Denizyolu Yük Taşımacılığında Rotalama ve Çizelgeleme, in Sosyal Bilimler Enstitüsü - İşletme Anabilim Dalı. İstanbul Üniversitesi, Doktora Tezi.

[5] Parlak, Ş., (2008). Hastane Randevu Sisteminin Hastalar Açısından Değerlendirilmesi, in Sağlık Bilimleri Enstitüsü, Sağlık Yönetimi Anabilim Dalı. Necmettin Erbakan Üniversitesi, Yüksek Lisans Tezi.

[6] Batur, B.S., (2008). Hava Yolcu ve Kargo Taşımacılığı: Dünya'da ve Türkiye'de Uygulamalar, in Sosyal Bilimler Enstitüsü - İşletme Anabilim Dalı. Dokuz Eylül Üniversitesi, Yüksek Lisans Tezi.

[7] Chaves, C.R., (2016). Approximation for Single-channel Multi-server Queues and Queuing Networks with Generally Distributed Inter-arrival and Service times in Engineering Management and Systems Engineering. 2016, Missouri University of Science and Technology, Doctoral Thesis.

[8] Adan, I., & Resing, J. (2015). Department of Mathematics and Computing Science Eindhoven University of Technology PO Box 513, 5600 MB Eindhoven, The Netherlands March 26, 2015.

[9] Ertuğrul, İ., Birsen, B., & Özçil, A. (2015). İki bankanın farklı şubelerindeki müşteri bekleme sürelerinin kuyruk modeliyle etkinlik analizi. *Çankırı Karatekin Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, *5*(1), 275-292.

[10] Majid, S., & Manoharan, P. (2017). Analysis of a M/M/c queue with single and multiple synchronous working vacations. *Applications and Applied Mathematics: An International Journal (AAM)*, *12*(2), 3.

[11] Yıldız, M. S., & Arslan, H. M. (2013). Bekleme Hatti Modeliyle Servis Sisteminin Analizi: Düzce Üniversitesi Merkez Yemekhanesi Örneği. *Journal of Management and Economics Research*, *11*(21), 169-184.

[12] Maragathasundari, S., Prabhu, C., & Palanivel, M. (2020). A study on stages of queuing system in aircraft control system. 3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Marzo 2020, 91-111. http://doi.org/10.17993/3ctecno.2020.specialissue4.91-111.

[13] Lan, S., & Tang, Y. (2017). Performance analysis of a discrete-time queue with working breakdowns and searching for the optimum service rate in working breakdown period. *Journal of Systems Science and Information*, *5*(2), 176-192.

[14] Kim, D. U., Jie, M. S., & Choi, W. H. (2018). Airport simulation based on queuing model using ARENA. *International Journal of Advanced Science and Technology*, *115*, 125-134.

[15] Ademoh, N. A., & Anosike, E. N. (2014). Queuing modelling of air transport passengers of Nnamdi Azikiwe international airport Abuja, Nigeria using multi server approach. *Middle East Journal of Scientific Research*, *21*(12), 2326-2338.

[16] Smith, D. K. (2002). Calculation of steady-state probabilities of M/M queues: further approaches. *Journal of Applied Mathematics and Decision Sciences*, *6*(1), 43-50.

[17] Jawab, F., Khachani, M., Akoudad, K., Moufad, I., Frichi, Y., Laaraj, N., & Zehmed, K. (2018, July). Queuing model for improving airport passengers treatment process. In *Proceedings of the ICIEOM: International Conference on Industrial Engineering and Operations Management (July 26-27, 2018, Paris, France)* (pp. 2095-2107).

[18] Girginer, N., & Şahin, B. (2007). Spor Tesislerinde Kuyruk Problemine Yönelik Bir Benzetim Uygulaması. *Spor Bilimleri Dergisi*, *18*(1), 13-30.

[19] Kumar, R., Som, B. K., & Jain, S. (2015). An M/M/1/N feedback queuing system with reverse balking. *Journal of Reliability and Statistical Studies*, 31-38.

[20] Artalejo, J., & Falin, G. (2002). Standard and retrial queueing systems: a comparative analysis. *Revista matemática complutense*, *15*(1), 101-129.

[21] Smith, P. J., Firag, A., Dmochowski, P. A., & Shafi, M. (2012). Analysis of the M/M/N/N queue with two types of arrival process: applications to future mobile radio systems. *Journal of Applied Mathematics*, *2012*(1), 123808.

[22] Ibe, O. C., & Isijola, O. A. (2014). M/M/1 multiple vacation queueing systems with differentiated vacations. *Modelling and Simulation in Engineering*, *2014*(1), 158247.

[23] Vandaele, N., Van Woensel, T., & Verbruggen, A. (2000). A queueing based traffic flow model. *Transportation Research Part D: Transport and Environment*, *5*(2), 121-135.

[24] Kumar, R. (2017). A transient solution to the M/M/c queuing model equation with balking and catastrophes. *Croatian Operational Research Review*, 577-591.

[25] Wang, S.; Zhu, L. (2004): "A Dynamic Queuing Model," Chinese Journal of Economic. Theory, 1, 14–35.

[26] Sharma, A. K., & Sharma, G. K. (2013). Queueing theory approach with queueing model: A study. *International Journal of Engineering Science Invention*, *2*(2), 1-11.

[27] Christien, R., Hoffman, E., Trzmiel, A., & Zeghal, K. (2018). An extended analysis of sequencing arrivals at three major

European airports. In *AIAA Aviaiton Technology, Integrations, and Operations Conference*. 2018: Atlanta - Georgia (USA).

[28] Mehri, H., & Djemel, T. (2009). Solving of waiting lines models in the airport using queuing theory model and linear programming. *ROADEF 2009*, 35.

[29] Chew, S. (2019). Continuous-Service M/M/1 Queuing Systems. *Applied System Innovation*, *2*(2), 16.

[30] Winston, W. L. (1991). *Operations research: applications and algorithms*. Boston, MA: PWS-Kent Publishing Company.

[31] Awasthi, B. (2018). Performance analysis of M/M/1/k finite capacity queuing model with reverse balking and reverse reneging. *J. Comp. Math. Sci*, *9*(7), 850-855.

[32] Seth, S. & Som, B. (2017). An M/M/1/N Queuing system with Encouraged Arrivals. Global Journal of Pure and Applied Mathematics, 13(7): p. 3443-3453.

[33] Jhala, N., & Bhathawala, P. (2017). Application of Queueing Theory to Airport Related Problems. Global Journal of Pure and Applied Mathematics, 13(7): p. 3863-3868.

[34] Çevik, O. Yazgan, A. E. (2008). Hizmet Üreten Bir Sistemin Bekleme Hattı (Kuyruk) Modeli İleEtkinliğinin Ölçülmesi. *Niğde Üniversitesi İktisadi ve idari bilimler fakültesi dergisi*, *1*(2), 117-124.119-128.

[35] Poongodi, T., & Muthulakshmi, S. (2013). Control chart for waiting time in system of (m/m/1):(?/fcfs) queuing model. *International Journal of Computer Applications*, *63*(3).

[36] Massey, W. A. (1985). Asymptotic analysis of the time dependent M/M/1 queue. *Mathematics of Operations Research*, *10*(2), 305-327.

[37] Idris, H. R., Anagnostakis, I., Delcaire, B., Hansman, R. J., Clarke, J. P., Feron, E., & Odoni, A. R. (1999). Observations of departure processes at logan airport to support the development of departure planning tools. *Air Traffic Control Quarterly*, *7*(4), 229-257.

[38] Shone, R., Glazebrook, K., & Zografos, K. (2018, September). Stochastic modelling of aircraft queues: A review. In *OR60: The OR Society Annual Conference* (pp. 61-83).

[39] Tiwari, S. K., Gupta, V. K., & Joshi, T. N. (2016). M/M/S queueing theory model to solve waiting lines and minimize estimated total cost. *International Journal of Science and Research*, *5*(5), 1901-1904.

[40] Thiagaraj, H. B., & Seshaiah, C. V. (2014). A queueing model for airport capacity and delay analysis. *Applied Mathematical Sciences*, *8*(72), 3561-3575.

[41] Bertsimas, D. J., & Nakazato, D. (1992). Transient and busy period analysis of the GI/G/1 queue: the method of stages. *Queueing Systems*, *10*, 153-184.

[42] Karapetyan, D., Atkin, J. A., Parkes, A. J., & Castro-Gutierrez, J. (2017). Lessons from building an automated pre-departure sequencer for airports. *Annals of Operations Research*, *252*, 435-453.

[43] Aydın, A. (2009). *Metasezgisel yöntemlerle uçak çizelgeleme problemi optimizasyonu* (Doctoral dissertation, Marmara Universitesi (Türkiye).

[44] Arslan, Ş., (2011) Uçakların Terminal Kapılarına Atanması Probleminin Farklı Yöntemlerle Çözümü ve Uygulaması, in Endüstri Mühendisliği Anabilim Dalı - Endüstri Mühendisliği Programı. Yıldız Teknik Üniversitesi, Yüksek Lisans Tezi

[45] Doğan, B. (2019). Beklenmedik uçak yönlendirmelerini azaltma: zaman serisi analizi ve yapay sinir ağları ile modelleme. Ankara: TOBB ETÜ Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.

[46] Fatima, A., Singh, V., Singh, S., & Khanna, P. (2025). Navigating queues in healthcare: A comparative analysis of queue management systems in healthcare. Recent Advances in Sciences, Engineering, Information Technology & Management, 172-178.

[47] Anitha, K., Poongothai, V., & Godhandaraman, P. (2025). Performance analysis of a queueing system with tandem nodes, retrial, and server vacations. Results in Control and Optimization, Volume 18, 100520.

[48] Dhibar, S., & Jain, M. (2025). Metaheuristics and strategic behavior of markovian retrial queue under breakdown, vacation, and Bernoulli feedback. Applied Intelligence, 55(4), 273.

[49] Amjath, M., Kerbache, L., Elomri, A., & Smith, J. M. (2024). Queueing network models for analyzing and optimizing material handling systems: a systematic literature review. Flexible Services and Manufacturing Journal, 36(2), 668-709.

[50] Ambika, K., Vijayashree, K. V., & Janani, B. (2024). Modelling and analysis of production management system using vacation queueing theoretic approach. Applied Mathematics and Computation, 479, 128856.

[51] Cakmak, M., & Torun, C. (2021). Performance comparison of queue management algorithms in LTE networks using NS-3 simulator. Tehnički vjesnik, 28(1), 135-142.

[52] Çakmak, M., & Albayrak, Z. (2018). A Review: Active queue management algorithms in mobile communication. In 2018 International Conference on Advanced Technologies, Computer Engineering and Science (ICONCS) (pp. 180-184).

[53] Çakmak, M., & Albayrak, Z. (2020). LTE Ağlarda Remote-Host ile PG-W Arasındaki Kuyruk Yönetim Algoritmalarının Performans Analizi. Academic Platform-Journal of Engineering and Science, 8(3), 456-463.

[54] Gundogar, E., Sari, M., & Kokcam, A. H. (2016). Dynamic bottleneck elimination in mattress manufacturing line using theory of constraints. SpringerPlus, 5, 1-15.

[55] Newell, C. (2013). *Applications of queueing theory* (Vol. 4). Springer Science & Business Media.

[56] Consul, P. C., & Jain, G. C. (1973). A generalization of the Poisson distribution. *Technometrics*, *15*(4), 791-799.

[57] Boxma, O. J., & Yechiali, U. (2007). Poisson processes, ordinary and compound. *Encyclopedia of statistics in quality and reliability*, 1-12.

[58] Steckley, S. G., Henderson, S. G., & Mehrotra, V. (2009). Forecast errors in service systems. Probability in the Engineering and Informational Sciences, 23(2), 305-332.

[59] Tan, X., Knessl, C., & Yang, Y. P. (2013). On finite capacity queues with time dependent arrival rates. Stochastic Processes and their Applications, 123(6), 2175-2227.

[60] Massey, W. A., & Whitt, W. (1993). Networks of infinite-server queues with nonstationary Poisson input. Queueing Systems, 13, 183-250.

[61] Tilt, B., & Balachandran, K. R. (1979). Stable and superstable customer policies in queues with balking and priority options. European Journal of Operational Research, 3(6), 485-498.

[62] Manitz, M. (2015). Analysis of assembly/disassembly queueing networks with blocking after service and general service times. Annals of Operations Research, 226, 417-441.

[63] Daskin, M. S. (2021). Fundamentals of Queueing Theory. In Bite-Sized Operations Management. Cham: Springer International Publishing.

[64] Dshalalow, J. H. (2023). Advances in Queueing Theory, Methods, and Open Problems. CRC Press.

[65] Gautam, N. (2012). Analysis of queues. CRC Press, LLC, Boca Raton, Florida, United States, 10, 2222496.

[66] Yang, K. K., Cayirli, T., & Low, J. M. (2016). Predicting the performance of queues–A data analytic approach. Computers & Operations Research, 76, 33-42.

Authors Contributions

Abdullah SEVİN contributed to the conception, design, writing, technical support, material support, and critical content review. Göktuğ YAMAN contributed to the conception, design, data collection, data analysis and interpretation, writing, technical support, material support, and literature review. Durdali ATILGAN contributed to the conception, design, writing, technical, and material support. Each author played a vital role in developing this work, ensuring its quality and accuracy.

Acknowledgments

Authors would like to acknowledge the support and guidance provided by colleagues, peers, and reviewers throughout the preparation of this article. No external funding was received for this work.

Conflict of Interest Notice

Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that scientific and ethical principles were followed during the preparation process of this study. All the studies referenced in this paper have been properly acknowledged in the bibliography. Ethical approval for this study was not required as it does not involve human participants or animals.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 1, 136-151, 2025
DOI: 10.35377/saucis...1638424

RESEARCH ARTICLE

A Comparative Analysis of Deep Learning Models for Prediction of Microsatellite Instability in Colorectal Cancer

Ziynet Pamuk[1*], Hüseyin Erikçi[1]

[1] Zonguldak Bulent Ecevit University, Department of Biomedical Engineering, Zonguldak, Türkiye, ror.org/01dvabv26

Corresponding author:
Ziynet Pamuk, Zonguldak Bulent Ecevit University, Department of Biomedical Engineering, Zonguldak, Türkiye
ziynet.pamuk@beun.edu.tr

Article History:
Received:12.02.2025
Revised: 06.03.2025
Accepted: 23.03.2025
Published Online: 27.03.2025

ABSTRACT
Colorectal cancer remains one of the most prevalent and fatal malignancies worldwide, underscoring the necessity for early and precise diagnostic approaches to enhance patient prognoses. This study proposes a deep learning-based model for predicting microsatellite instability (MSI) in colorectal cancer using hematoxylin and eosin (H&E)-stained histopathological tissue slides. A classification framework was constructed using convolutional neural networks (CNN) and optimized through transfer learning techniques. The dataset, comprising 150,000 unique H&E-stained histologic image patches, was sourced from an open-access Kaggle repository, with 80% allocated to training and 20% to testing. A comparative evaluation of nine pre-trained models demonstrated that the VGG19 architecture yielded the highest classification performance, achieving an accuracy of 90.60%, a precision of 88.60%, a sensitivity of 93.10%, and an AUC score of 90.60%. Considering its high performance, the proposed model is expected to assist pathologists in clinical decision-making, potentially enhancing diagnostic accuracy in real-world medical applications.

Keywords: Microsatellite instability, Deep learning, Colorectal cancer, Histopathologic image

1. Introduction

Colorectal cancer (CRC), the most common type of cancer worldwide, accounts for a significant proportion of cancer-related deaths [1]. According to the World Health Organization's International Agency for Research on Cancer (IARC), approximately 19.3 million individuals are diagnosed with cancer, while around ten million cancer-related deaths have been reported. With these figures, CRC represents 10% of cancer cases, and its death rate accounts for 9.4% of cancer-related mortality [2, 3]. Consequently, CRC is considered the third most diagnosed malignancy and is among the second leading causes of cancer-related deaths globally [4].

For the diagnosis of colorectal cancer, a tissue sample is typically obtained endoscopically, embedded in paraffin (FFPE), and fixed in formalin [5]. While FFPE is a standard method that aids in developing medical treatments and disease diagnosis, research on the preservation and preparation of biopsy samples has shown that FFPE tissue slides, when properly prepared, are highly durable and can be stored at ambient temperature for extended periods [6]. Subsequently, the tissue sample is stained with hematoxylin and eosin (H&E) and examined microscopically for diagnosis [5]. CRC patients are recommended to undergo advanced molecular testing, especially for microsatellite instability (MSI), which is caused by genetic alterations and affects 10-15% of these patients [7, 8]. MSI is a common tumor phenotype characterized by abnormal repetitions of short DNA motifs resulting from an insufficient mismatch repair (MMR) system, and it is associated with Lynch Syndrome (LS), an inherited cancer syndrome [9].

Systematic MSI screening is generally recommended for evaluating the response to colorectal immunotherapy [8] and has been controversially proposed for chemotherapy response. MSI diagnosis is traditionally performed using polymerase chain reaction (PCR) or next-generation sequencing (NGS) by analyzing H&E-stained CRC tissue and manually detecting abnormalities in microscopic images [10]. These methods are time-consuming and costly and require specialized expertise that may not be available at all centers, which can sometimes lead to errors [11]. Therefore, there is a growing need for tools that are widely accessible, cost-effective, and provide high-accuracy results in a shorter time.

Cite as: Z. Pamuk, H. Erikçi. "A comparative analysis of deep learning models for prediction of microsatellite instability in colorectal cancer". *Sakarya University Journal of Computer and Information Sciences*, vol. 8, no. 1, pp. 136-151, 2025, doi: 10.35377/saucis...1638424

Deep learning technologies, a subfield of artificial intelligence (AI), have become increasingly prevalent in healthcare [12]. Significant contributions to medicine have been made by developing segmentation and classification models that can diagnose and detect tumor mutations and molecular changes using convolutional neural networks (CNNs). These methods have been applied in detecting skin cancer using dermoscopic images [13] and in detecting lung [14], liver [15], and breast [16] cancers using histopathological images. Building upon these studies, this research aims to estimate the MSI status in colorectal cancer by conducting comparative analysis using deep learning models previously trained on H&E-stained histopathological slides.

The paper's organization is as follows: The studies on deep learning techniques applied to histopathological images for MSI prediction are presented in Section 2. The dataset is described in detail in Section 3.1. The deep transfer learning architecture, pre-trained models, and experimental setup parameters are explained in Sections 3.2 and 3.3, respectively. Performance metrics are provided in Section 3.4. The experimental results obtained from the proposed models and discussions are presented in Section 4. Finally, conclusions and future work are summarized in Section 5.

2. Related Works

Several studies have diagnosed MSI using H&E-stained histopathological images in cancer research, employing dual or multi-class classification based on deep learning methods. While some studies used raw data, others applied segmentation techniques through feature extraction. Additionally, the number of datasets used in these studies varies. Among these studies, convolutional neural networks and machine learning are the most commonly evaluated methods, particularly in studies incorporating transfer learning approaches. However, there is still no consensus on the ideal neural network architecture. Furthermore, variations are observed in the success rates of different models with varying parameters.

In contribution to the dataset used in our study, Kather et al. employed the ResNet18 model architecture over ImageNet to train and validate H&E-stained FFPE and frozen images from various independent cohorts, such as gastric and colorectal cancer from The Cancer Genome Atlas (TCGA). The German colorectal cancer cohort (DACHS) was used as an external confirmation set. It was observed that colorectal cancer exhibited better performance results than gastric cancer at the patient level. They demonstrated that MSI could be estimated with an area under the curve (AUC) value of 0.84 for the colorectal cancer set [17].

Cao et al. proposed a deep learning model based on multi-example learning to estimate MSI from histopathological images. They developed the Ensembled Patch Likelihood Aggregation (EPLA) model using two different colorectal cancer cohorts from TCGA-COAD and Asia and the ResNet18 architecture. Initially, it was trained and validated with TCGA-COAD, achieving an AUC of 0.8848. However, a lower AUC of 0.6497 was obtained using the EPLA model on the Asia-CRC external confirmation dataset. By utilizing previously trained networks within the model, the AUC was improved to 0.8504, and MSI estimation was successfully performed. They evaluated model performance by generating pathological signatures from the model [18].

Echle et al. developed a deep learning system based on MSI or dMMR estimation of colorectal cancer patients' H&E-stained slide images. This system was trained using the ESA-based ShuffleNet model architecture and validated externally with the Yorkshire Cancer Research Bowel Cancer Improvement Program (YCR-BCIP) cohort. The system demonstrated high performance in estimating MSI/dMMR tissue characteristics [19, 20].

Lee et al. introduced a two-phase classification method to estimate MSI in CRC. Initially, they segmented the tumor regions into MSI-L and MSI-H areas using a feature pyramid network. Then, tumor classification was performed using Inception-ResNetV2. The pathology images were magnified at 10x and 20x, and the suggested method was reported to outperform traditional methods in terms of performance [21].

Krause et al. developed synthetic images using conditional generative adversarial networks (CGAN) to estimate MSI status in CRC. Training with these synthetic images achieved an AUC of 0.742 on actual image data. When both actual and synthetic images were used for training, the AUC improved to 0.777, demonstrating an approach to augment small datasets with synthetic data [22].

Zhang et al. employed a different visualization method for MSI estimation in rectal cancer using high-resolution T2-weighted magnetic resonance imaging (MRI). Their model, developed with the MobileNetV2 neural network architecture, achieved AUC values of 0.573, 0.82, and 0.868 for the clinical, purely visual, and combined models. It was noted that the clinical model exhibited lower performance compared to the other two models [23].

Yamashita et al. developed the MSINet-100 deep learning model by scanning H&E-stained whole slide images magnified at 40x. The model was externally validated with images from TCGA and using 20x magnification. Additionally, they compared the performance of MSINet with five gastrointestinal pathologists and concluded that the deep learning model outperformed the pathologists in estimating MSI [24].

Bustos et al. developed a deep learning system that reduced multi-biases using adversarial networks from tissue microarrays to estimate MSI in colorectal cancer. This system was validated with data from 1,788 patients from EPICOLON and HGUA. Tissue types and model performance effects were evaluated using different magnifications at both tissue and patient levels. The study marked the first attempt to estimate MSI using a multi-bias ablation technique with tissue microarrays [25].

Further studies on MSI estimation in CRC include those by Lee et al., Qui et al., Su et al., Muti et al., and Saillard et al., who applied deep learning-based classification methods using neural networks for MSI estimation in multiple cancer types, including gastric cancer and Epstein-Barr virus-related cancers [26-30].

3. Material and Methods

3.1. Dataset

In this study, histopathological tissue images of colorectal cancer (COAD) from The Cancer Genome Atlas (TCGA), specifically focusing on MSI and MSS cases, were obtained from the open-access Kaggle website and provided by Joan Gibert. This dataset comprises 192,312 unique image patches extracted from the histopathological images of cancer patients. These images were reproduced from diagnostic slides that had been fixated with formalin and embedded in paraffin (FFPE) [31]. FFPE diagnostic slides play a crucial role in medical diagnoses, as they are produced by fixating a tissue sample in formaldehyde and embedding it in a paraffin wax block for sectioning. A clear, well-defined image is created to prepare the sample for computational analysis [32].

The dataset was categorized into two groups: MSS and MSI. It contains 117,276 images in the MSS category and 75,039 in the MSI category. During the creation of the dataset, Kather performed several preprocessing steps. These steps included automatic tumor detection, resizing the images to a 0.5 µm/px resolution with dimensions of 224x224 pixels, followed by color normalization using the Macenco method [33]. The images were then classified as either "MSS" (microsatellite stable) or "MSIMUT" (microsatellite unstable or highly mutated) based on the patients' conditions [34]. As a result of these preprocessing steps, the images were of high quality and were used in their final form as JPG files in the study. Sample images are shown in Figure 1.

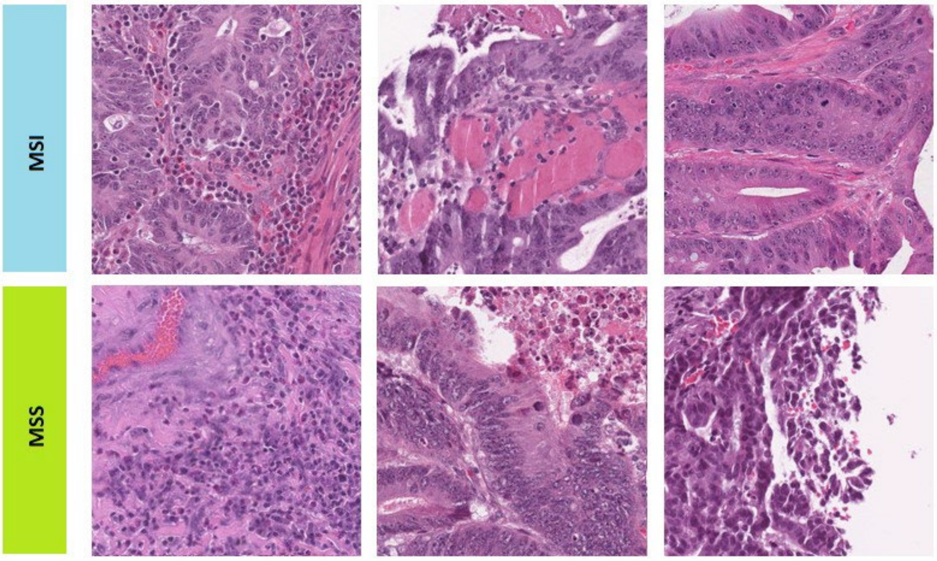


Figure 1. Histopathological Image Samples of MSI and MSS Classes

3.2. Architecture of Deep Transfer Learning

Deep learning is a subfield of machine learning that operates by mimicking the working style of the human brain, processing information through multiple layers. In recent years, as deep learning technology has been applied in areas such as autonomous systems, computer vision, and speech recognition, it has also made significant advancements in medical data analysis and medical image processing [35, 36]. In our study, we achieve substantial results by applying deep learning methods to medical imaging data.

Studies in classification, segmentation, and lesion detection are conducted using images obtained from medical imaging modalities, such as positron emission tomography-computed tomography (PET-CT), magnetic resonance imaging (MRI), computed tomography (CT), and x-rays. As a result of these analyses, the detection and classification of certain diseases have become more rapid and cost-effective [37-39].

Deep learning models rely on several algorithms, one of the most widely used being the Deep ESA algorithm in image classification. ESA is a multi-layer, feed-forward neural network that utilizes sensors to analyze data through controlled learning processes [40]. This algorithm captures the features of the image data used in the study and classifies them through various stages. As illustrated in Figure 2, feature extraction is initially performed in the convolutional layers. A filter is applied in the feature extraction process, and a comparison is made between the newly generated image and the image label by sliding the filter over the image. The network's inconsistencies are addressed by reducing parameters and performing calculations in the pooling layer. In short, the fully connected layer completes the classification process [41]. Therefore, data passes through three distinct layers—convolutional, pooling, and fully connected—during the application of ESA. These layers are discussed in further detail below.

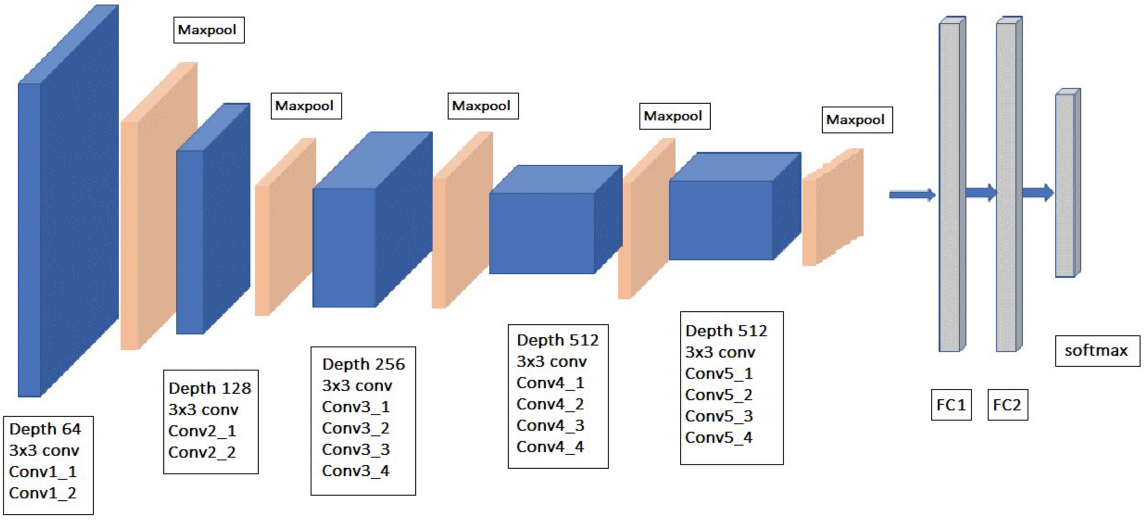


Figure 2. Architecture of the VGG19 Model

This layer is the first stage of the ESA algorithm. It is responsible for detecting the features of image data. The convolutional layer extracts both low-level and high-level features from the image by applying a filtering process. These filters assist in extracting the edge information and fine details of the image. As shown in Figure 3, the applied filters are 3x3x3 multidimensional matrices composed of pixel values. The first value of the filter represents the matrix's height, the second value represents the matrix's width, and the third value indicates the matrix's depth [42].

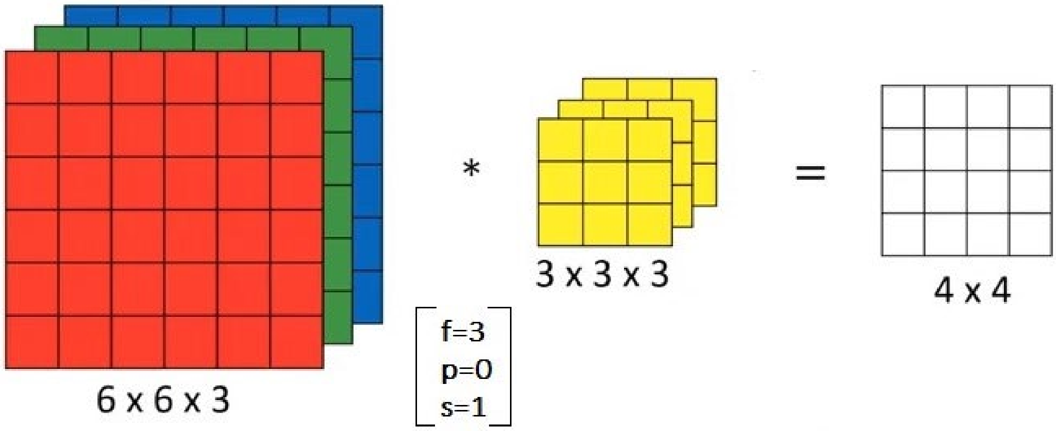


Figure 3. Calculation Process of Layers Using the Convolutional Formula (N+2p-f) +1

N: the input image size, p: padding (pixel adding), f: filter size, s: stride (scroll step).

The pooling layer is designed to reduce the number of computations and parameters within the network. Our study utilized both maximum pooling and global average pooling variants. Maximum pooling involves the application of a 2x2 filter, which selects the largest value within the area covered by the filter on the feature map, as illustrated in Figure 4. This process reduces the size of the image, thereby decreasing the computational load while preserving the essential features of the image data. On the other hand, global average pooling connects to the fully connected layer by aggregating all the neurons in the feature map into a single value, thereby reducing the dimensionality of the data [43].

This layer is the final layer of ESA and connects each neuron to every neuron in the subsequent layer. It is responsible for learning through artificial neural networks. In our study, the Rectified Linear Unit (ReLU) activation function was employed in this layer, while the sigmoid activation function was used to estimate the output data in the neurons. The mathematical formulations for these activation functions are provided in equations (1) and (2).

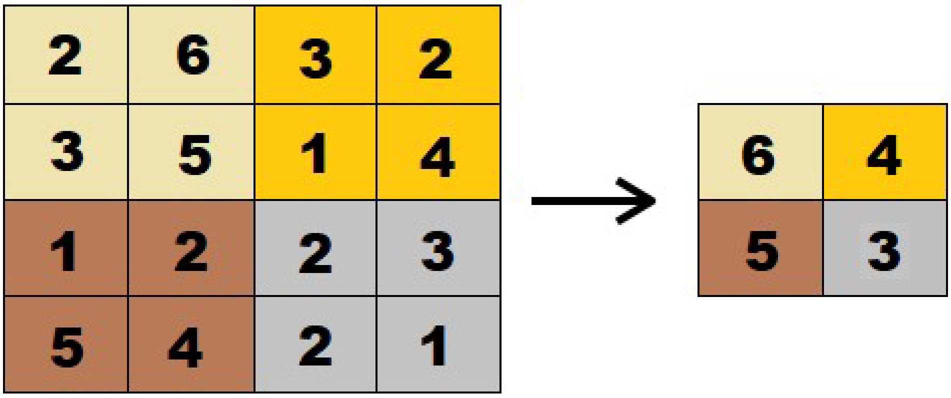


Figure 4. Illustration of the Pooling Process

$$ReLU(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \quad (1)$$

$$Sigmoid(x) = \sigma(x) = \frac{1}{1 + \exp(-x)} \quad (2)$$

In these equations, x represents the input data. Transfer learning is a machine learning technique in which a model developed for one study is reused as the starting point for a subsequent model in another study. In other words, instead of training convolutional neural networks (CNNs) from scratch, transfer learning accelerates the learning process by leveraging information from a previously trained model, thus facilitating faster transactions and improving training efficiency. Pre-trained networks are models that inherit weights from a model that has been previously applied to a similar or even a different problem. While these networks may not always offer the optimal solution for every new problem, they help avoid the need for redundant training, enabling faster results with reduced computational effort. Pre-trained networks typically leverage the ImageNet visual object recognition database [44, 45].

In this study, deep ESA-based pre-trained networks, including VGG19, VGG16, MobileNet, MobileNetV2, ResNet50, InceptionV3, ResNet18, GoogleNet, and AlexNet, were employed to classify colorectal cancer histopathological tissue images into two categories: MSI and MSS. The performance of these networks was compared. The applied processes are illustrated in Figure 5. The study's resources are also publicly available in the open-access repository: https://github.com/hsynrkc/MSI-Prediction.

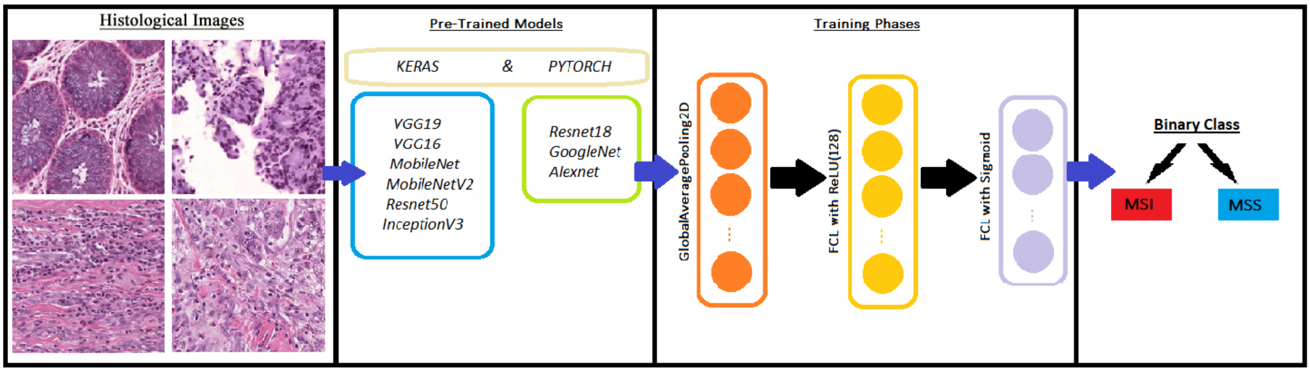


Figure 5. Schematic Flowchart Illustrating the Estimation Process of Pre-Trained Models for MSI and MSS Classification

Keras, an open-access deep learning library written in Python, is an application programming interface (API) for neural networks that operate with frameworks such as TensorFlow, Theano, and CNTK (Microsoft Cognitive Toolkit). Keras applications provide deep learning models with pre-trained weights, which can be utilized for tasks such as estimation, feature extraction, and fine-tuning. The models employed in this study are detailed below with their respective descriptions [46].

The VGG19 model is a convolutional neural network architecture with five pooling layers developed by the Visual Geometry Group (VGG) at the University of Oxford. It consists of 19 layers, 16 of which are convolutional, and three are fully connected. The network was trained using over one million images from the ImageNet database and performs classification tasks on various images with an input size of 224x224 pixels [47]. Among the models used in this study, VGG19 achieved the best performance regarding computational cost, accessibility, high accuracy, and low loss. After the convolutional layers in the VGG19 architecture, a fully connected layer with 128 neurons is added, utilizing the sigmoid activation function.

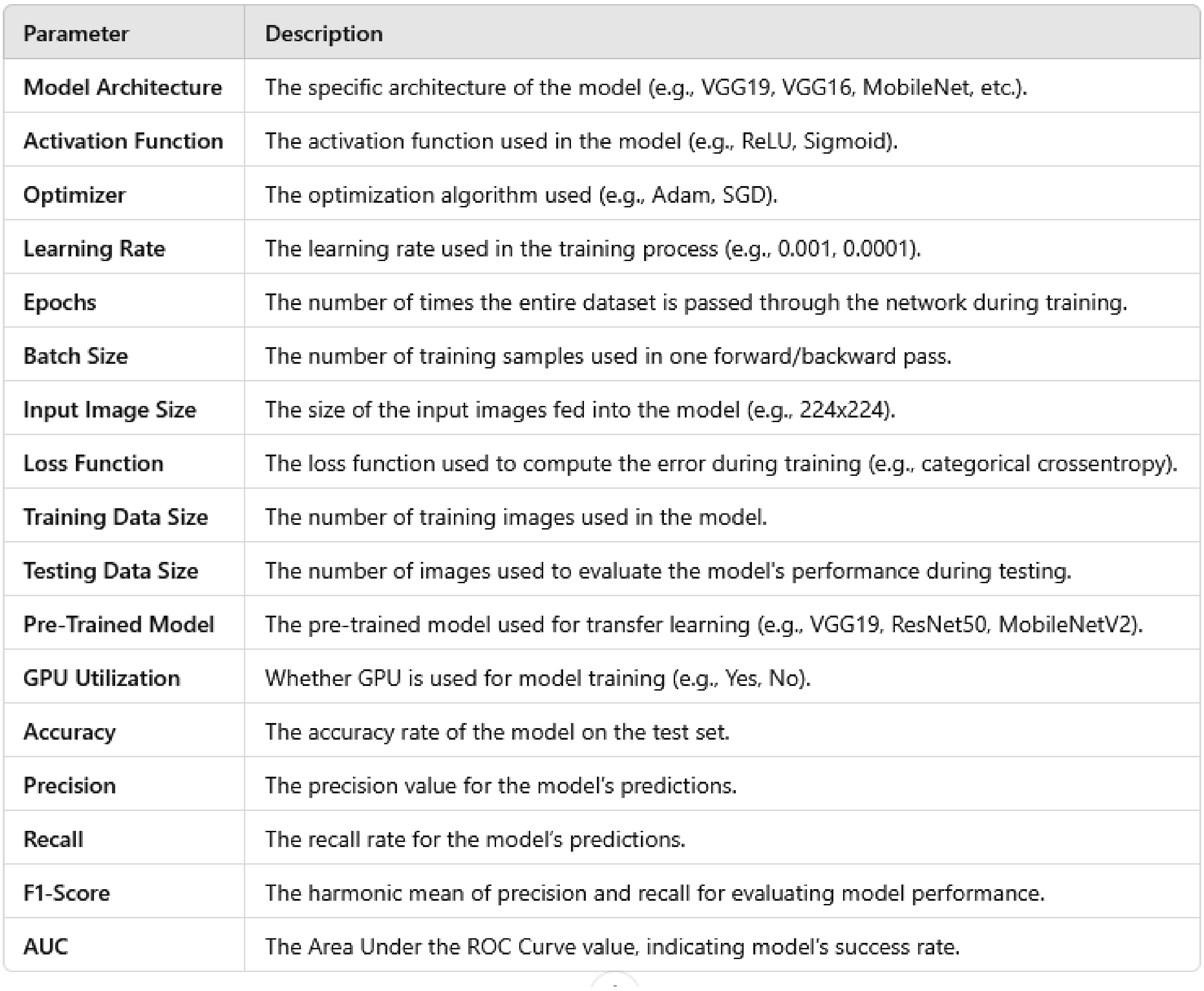
The VGG16 model is another convolutional neural network architecture consisting of 16 layers, including 13 convolutional layers and three fully connected layers. VGG16 and VGG19 are similar, with the key difference being the number of layers. A notable feature of this network is that it includes joint layers after 2x2 or 3x3 convolutional layers, distinguishing it from other models [47].

The MobileNet model, developed by Google researchers, is a low-dimensional convolutional neural network designed to reduce computational cost and the number of parameters by employing depthwise separable convolution layers [48, 49]. MobileNetV2, an evolution of MobileNet, incorporates bottleneck layers and inverted blocks after 32 filtered convolutional layers. This variant is more efficient in dimensionality, with fewer parameters and greater depth than the original MobileNet [50].

ResNet (Residual Network) is an artificial neural network that incorporates residual connections, which involve stacking bottleneck blocks. The ResNet50 model is a convolutional neural network with a depth of 50 layers, comprising 48 convolutional layers, one max pooling layer, and one average pooling layer. It is a widely used model in deep learning applications [51].

InceptionV3, developed by Google, is another convolutional neural network trained on the ImageNet dataset. It includes multiple convolutional layers, max-pooling layers, and a fully connected layer at the final stage [52].

Table 1. Parameters Used in Deep Learning Models for MSI and MSS Classification



Parameter	Description
Model Architecture	The specific architecture of the model (e.g., VGG19, VGG16, MobileNet, etc.).
Activation Function	The activation function used in the model (e.g., ReLU, Sigmoid).
Optimizer	The optimization algorithm used (e.g., Adam, SGD).
Learning Rate	The learning rate used in the training process (e.g., 0.001, 0.0001).
Epochs	The number of times the entire dataset is passed through the network during training.
Batch Size	The number of training samples used in one forward/backward pass.
Input Image Size	The size of the input images fed into the model (e.g., 224x224).
Loss Function	The loss function used to compute the error during training (e.g., categorical crossentropy).
Training Data Size	The number of training images used in the model.
Testing Data Size	The number of images used to evaluate the model's performance during testing.
Pre-Trained Model	The pre-trained model used for transfer learning (e.g., VGG19, ResNet50, MobileNetV2).
GPU Utilization	Whether GPU is used for model training (e.g., Yes, No).
Accuracy	The accuracy rate of the model on the test set.
Precision	The precision value for the model's predictions.
Recall	The recall rate for the model's predictions.
F1-Score	The harmonic mean of precision and recall for evaluating model performance.
AUC	The Area Under the ROC Curve value, indicating model's success rate.

PyTorch is an open-access Python library used to develop deep learning models. It provides a practical environment for creating neural networks with speed and flexibility, leveraging the power of graphics processing units (GPUs). PyTorch's popularity stems from its Pythonic nature and ease of use in constructing neural network models [53]. Below are descriptions of some models employed within the PyTorch environment:

The ResNet18 model is a convolutional neural network with 18 layers, trained using the ImageNet dataset. It is a residual network similar to ResNet50 and is implemented using PyTorch [51, 54].

GoogleNet, developed by Christian Szegedy and colleagues in 2014, is a convolutional neural network architecture with 22 layers. This PyTorch implementation model includes two auxiliary classifier layers, which combine multiple inception model layers [55].

AlexNet, developed by Alex Krizhevsky and colleagues in 2012, is a neural network architecture with a depth of 8 layers, consisting of five convolutional layers and three fully connected layers. It is implemented using PyTorch, and the ImageNet dataset was trained on it [56].

As shown in Table 1, the parameters used in the deep learning models for MSI and MSS classification are summarized, highlighting each model's key components and configurations.

3.3. Experimental Setup

The Python programming language was utilized to train the proposed deep transfer learning models. All colorectal cancer histopathological image experiments were conducted at no cost using Google Colaboratory (Colab) servers, leveraging online cloud services and Tesla T4 Graphics Processing Unit (GPU) hardware. In addition, personal computers equipped with an Intel Core i3-5005U processor, 2 GB RAM, an NVIDIA GeForce 920M graphics card, and 8 GB RAM were used for supplementary tasks. Training for ESA models (VGG19, VGG16, MobileNet, MobileNetV2, ResNet50, InceptionV3, ResNet18, GoogleNet, AlexNet) was performed using the root-mean-square propagation (RMSprop) optimization technique with the cross-entropy loss function.

In the proposed model, the batch size, learning rate, and number of epochs were set to 64, 1e-5, and 10, respectively. The datasets were randomly divided into two independent subsets for training and testing, with 80% of the data used for training and 20% for testing. During model training, 80% of the data was employed, while the validation process was conducted by randomly generating a 20% validation dataset from the training data. The model’s performance was evaluated using the remaining 20% of the dataset, which served as the test set.

3.4. Performance Metrics

Several performance metrics are utilized to evaluate the performance of deep learning transfer learning models. These metrics include:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3)$$

$$Precision = \frac{TP}{TP + FP} \qquad (4)$$

$$Recall = TPR = \frac{TP}{TP + FN} \qquad (5)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (6)$$

In these equations, TP, FP, TN, and FN represent the counts of true positive, false positive, true negative, and false negative cases, respectively. Specifically, for the test dataset and model, TP refers to the number of MSI cases correctly predicted as positive by the model, FP refers to MSS cases incorrectly predicted as positive (i.e., false positives), TN represents the number of MSS cases correctly predicted as negative, and FN refers to MSI cases incorrectly predicted as negative.

The proposed model for MSI diagnosis in colorectal cancer and other models used in this study are evaluated using classification performance metrics such as accuracy, precision, recall, and F1-score. Accuracy is the proportion of the total number of correct predictions made by the model to the total number of samples. Precision refers to the proportion of correctly predicted positive samples (MSI) out of all samples predicted as positive. Recall indicates the proportion of actual positive samples (MSI) the model correctly identified. The F1-score represents the harmonic meaning of precision and recall, balancing the two.

The ROC curve is used to visualize the model's ability to discriminate between the two classes (MSI and MSS) at various threshold values. It plots the false positive rate (FPR) on the x-axis, which is defined as 1−Specificity, and the true positive rate (TPR) on the y-axis. These are defined as:

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

4. Experimental Results and Discussion

In this study, we conducted dual-class classification between MSI (Microsatellite Instability) and MSS (Microsatellite Stable). To evaluate the performance of various models, we compared nine different pre-trained models: ResNet18, GoogleNet, and AlexNet (from the PyTorch library) and VGG19, VGG16, MobileNet, MobileNetV2, ResNet50, and InceptionV3 (from the Keras library).

For model training, 80% of the dataset was randomly selected for the training/validation set, while the remaining 20% was reserved for the test set. This process was repeated until each model's test set was evaluated.

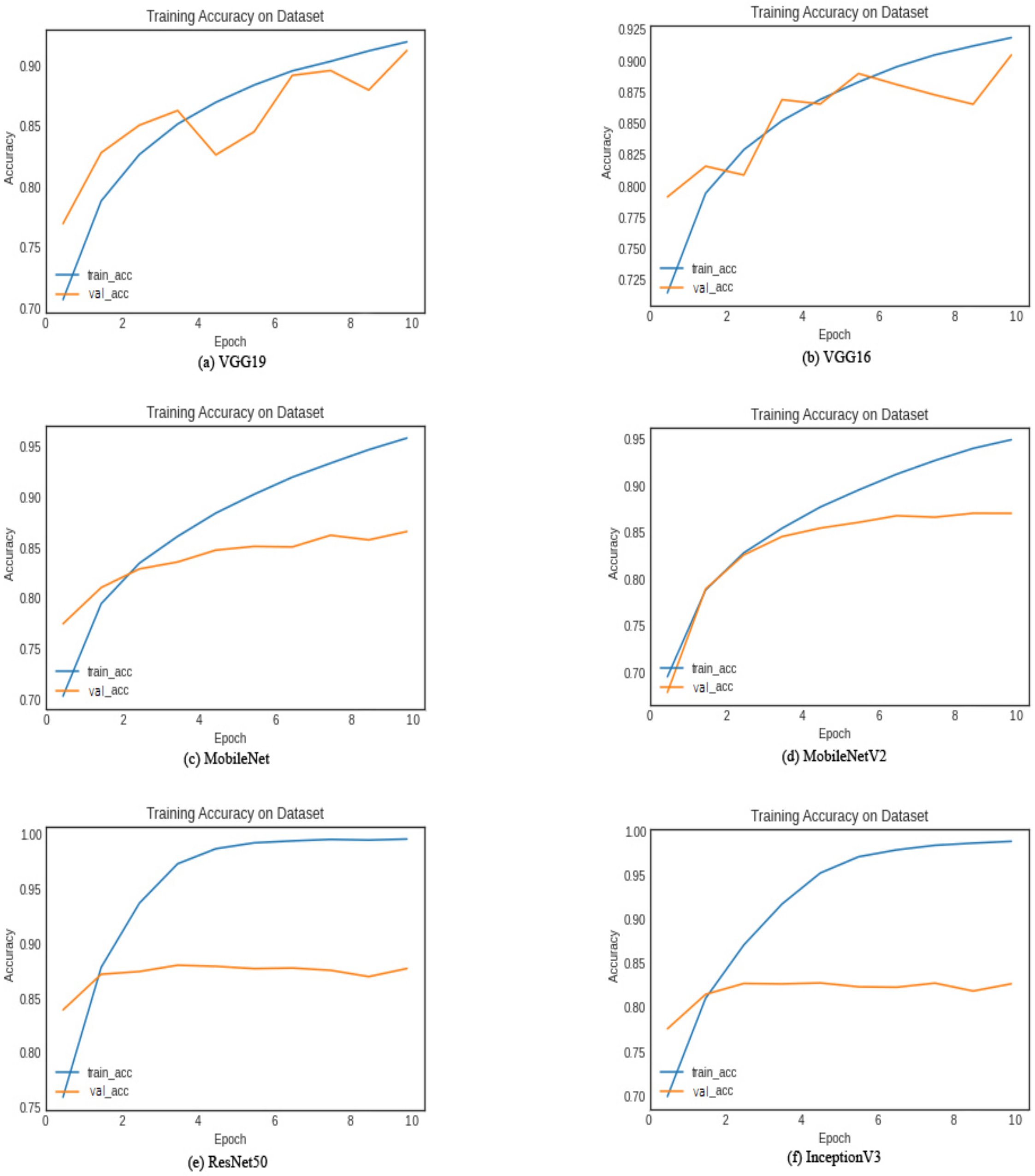
The detailed results of these experiments, including the performance metrics for each model, are summarized in Table 2.

Table 2. Training Accuracy and Loss Values of Pre-Trained Models Used in the Study

Models	Values						
Keras	Train			Validation		Test	
	Acc	Loss	Time	Acc	Loss	Acc	Loss
VGG19	**91**	**0.19**	**17666s**	**91**	**0.21**	**90**	**0.22**
VGG16	91	0.19	13989s	90	0.22	89	0.25
MobileNet	93	0.17	6040s	86	0.31	85	0.33
MobileNetV2	91	0.21	6653s	86	0.31	85	0.34
ResNet50	87	0.27	11355s	87	0.29	86	0.30
InceptionV3	81	0.40	7736s	81	0.39	80	0.41
Pytorch							
ResNet18	86	0.30	7166s	85	0.33	86	~
GoogleNet	82	0.38	4887s	83	0.37	83	~
AlexNet	75	0.53	3595s	77	0.52	76	~

According to the best loss value during the training period for the models applied to the dataset, the accuracy and loss values are presented in Table 1. In evaluating the model's performance, accuracy and loss rates are considered during training. The accuracy graphs for the Keras library models are shown in Figure 6.

The loss function is critical to evaluate the discrepancy between the true and predicted values. A decrease in the loss function signifies an increase in the model's robustness [58]. The VGG19 model outperforms the others, achieving lower loss values and higher detection rates for the MSI-MSS classes. While MobileNet models show better accuracy and loss values during the training phase, they perform worse than VGG19 and VGG16 in the accuracy and test phases. Additionally, although ResNet50 and InceptionV3 models yield good training results, their accuracy and test values are relatively poor. Therefore, VGG19 demonstrates the highest overall performance with an accuracy of 91%. The results summarized in Table 3 reflect the training outcomes on the histopathological images of all the pre-trained models used in this study. The accuracy values represent the general success rates of the models.



(a) VGG19

(b) VGG16

(c) MobileNet

(d) MobileNetV2

(e) ResNet50

(f) InceptionV3

Figure 6. Training and Validation Accuracy Curves of the Pre-trained Models

Table 3. Performance Metrics of Six Different Models

Models	SPE (%)	ACC (%)	PRE (%)	REC (%)	AUC (%)	F1-Score (%)
VGG19	**88.1**	**90.6**	**88.6**	**93.1**	**90.6**	**90.8**
VGG16	93.5	89.4	92.9	85.3	89.4	88.9
ResNet50	86.7	86.6	86.6	86.4	86.6	86.5
MobileNet	83.9	85.1	84.3	86.3	85.1	85.3
MobileNetV2	86.3	85.2	86.0	84.0	85.2	85.0
InceptionV3	81.9	80.9	81.5	79.8	80.9	80.6

The VGG19 model outperforms the other models primarily due to its deeper architecture, which enables superior feature extraction from complex medical images. Its 19 layers effectively capture low-level and high-level features, improving classification accuracy. Furthermore, VGG19 benefits from transfer learning, having been pre-trained on large datasets like ImageNet, which helps it generalize well to new tasks with limited data. The model's optimized activation functions, such as ReLU, enhance its learning efficiency and stability. These factors contribute to VGG19's superior performance metrics, including accuracy, precision, recall, AUC, and F1-score.

The VGG16 model, while similar to VGG19, has a slightly shallower architecture with 16 layers, which may limit its ability to capture more intricate features in complex data. As a result, it achieved lower performance in certain metrics, particularly recall, compared to VGG19. While it still benefits from transfer learning and optimized activation functions, the reduced depth of the network restricts its ability to exploit the available data fully.

ResNet50, which incorporates residual connections to overcome the vanishing gradient problem, performs well but not as highly as VGG models. Its architecture allows for deeper networks without the typical degradation of performance seen in traditional deep networks. Yet, its overall performance is limited by factors such as the model's relatively lower ability to extract fine-grained features and its complexity compared to VGG-based models. The accuracy, precision, and recall performance are slightly lower, which could be attributed to the model's difficulty in generalizing the medical image dataset used in this study.

MobileNet and MobileNetV2 are optimized for mobile and embedded devices and designed to be lightweight and efficient, but they sacrifice some accuracy in favor of computational efficiency. While good for applications requiring lower computational cost, these models show decreased performance compared to the VGG and ResNet models, particularly in accuracy and recall. The reduced model size and fewer parameters reduce the capacity to learn detailed features from the complex data.

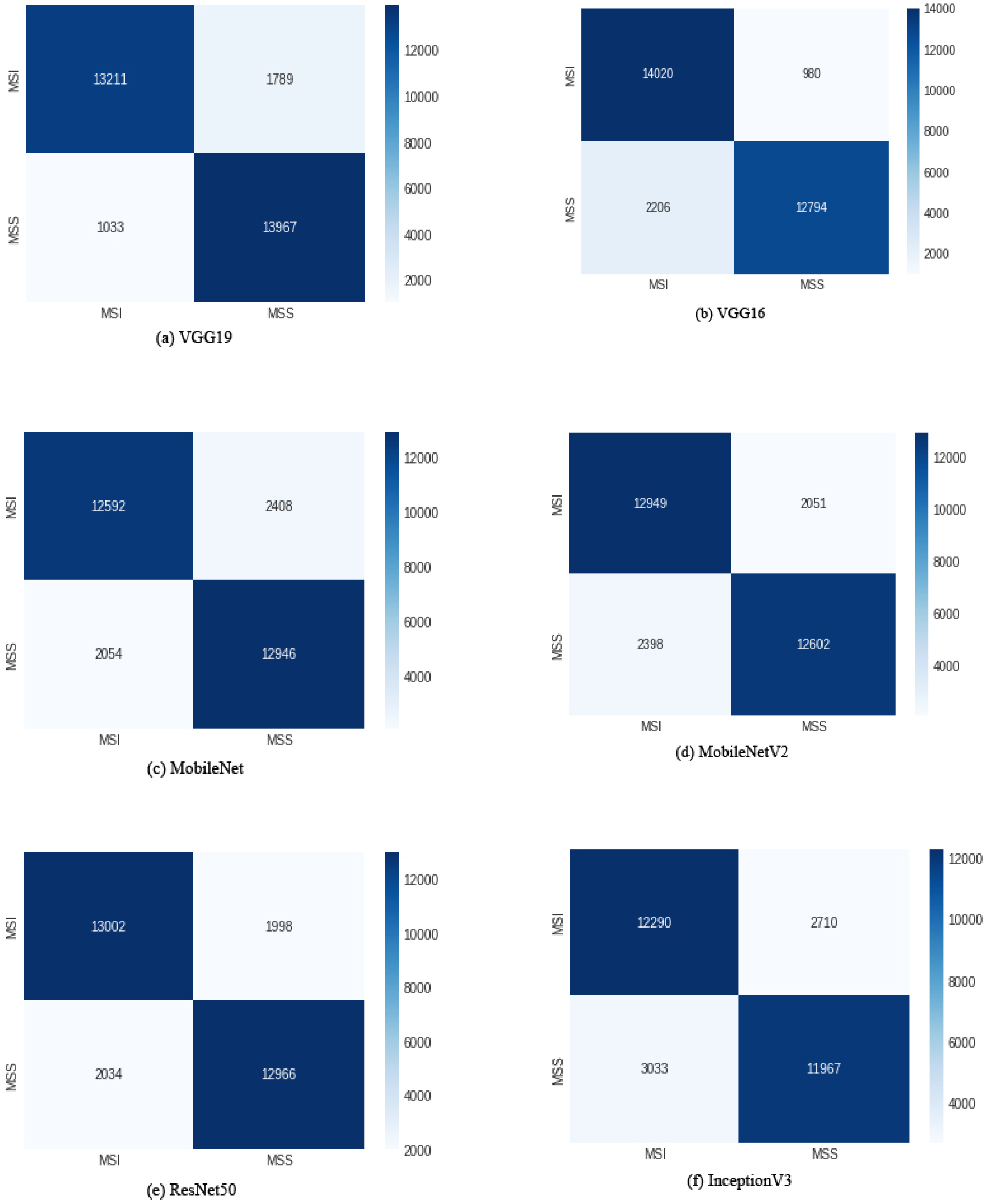
Finally, InceptionV3, despite its sophisticated architecture designed to capture a wide variety of features at multiple scales, demonstrates the lowest performance in this study. This could be due to its relatively high complexity, which may hinder its ability to effectively learn from smaller or specialized medical datasets, as well as potential issues with optimization for the specific task of heart disease prediction.

In conclusion, while deeper and more complex models such as VGG19 tend to perform better due to their ability to capture detailed features and benefit from transfer learning, lighter models like MobileNet and InceptionV3 show lower performance because of their design limitations and computational trade-offs.

The confusion matrix table, which is created separately for each model, is shown in detail in Figure 7. In these confusion matrices, the predictions made by the models based on the dataset split and the actual values of the images are presented. Based on the pre-trained VGG19, which provided the best performance among the models, the test dataset consists of 3000 images. Out of 15,000 MSI images, 1,321 were correctly predicted, while 1,789 were misclassified as MSS. For the MSS images, 13,967 out of 15,000 were correctly identified, while 1,033 MSI images were incorrectly classified. The confusion matrices of the other pre-trained models can be interpreted similarly.

The estimation rates presented in Table 3 are calculated separately for each model's performance metrics, with the corresponding calculation formulas provided in equations (3)-(8). The AUC value (Area Under the Curve) refers to the "area under the ROC curve," which indicates the model's success rate based on the area it covers. The ideal value for the area under the curve is 1 [59]. Figure 8 illustrates the pre-trained models' ROC curve and the area under the curve. The AUC, accuracy, F1 score, and other metrics presented in Table 3 are used as criteria for evaluating the performance of the models. According to studies by Ling et al., the AUC value is considered a more reliable metric than accuracy, as it provides a more accurate assessment of the model's performance [60].

A comparison of several studies on the estimation of MSI is presented in Table 4. Previous research has generally focused on a limited number of histopathological image datasets. In contrast, this study compares nine pre-trained models using a dataset comprising 150,000 image samples. Notably, the proposed method is fully automated, with no manual intervention, utilizing images resized to 224x224 pixels. Moreover, this study stands out by leveraging a larger dataset and incorporating more pre-trained models than many existing studies in the field. However, as the dataset size increases, the number of stages involved in the process decreases. This trade-off is mainly due to the limitations of the free cloud environment (Google Colab), which restricts the amount of processing time and capacity. Despite these constraints, the proposed system offers a cost-effective decision support tool for pathologists and doctors, aiding them in rapidly diagnosing and diagnosing MSI in colorectal cancer patients through histopathological images.



(a) VGG19

(b) VGG16

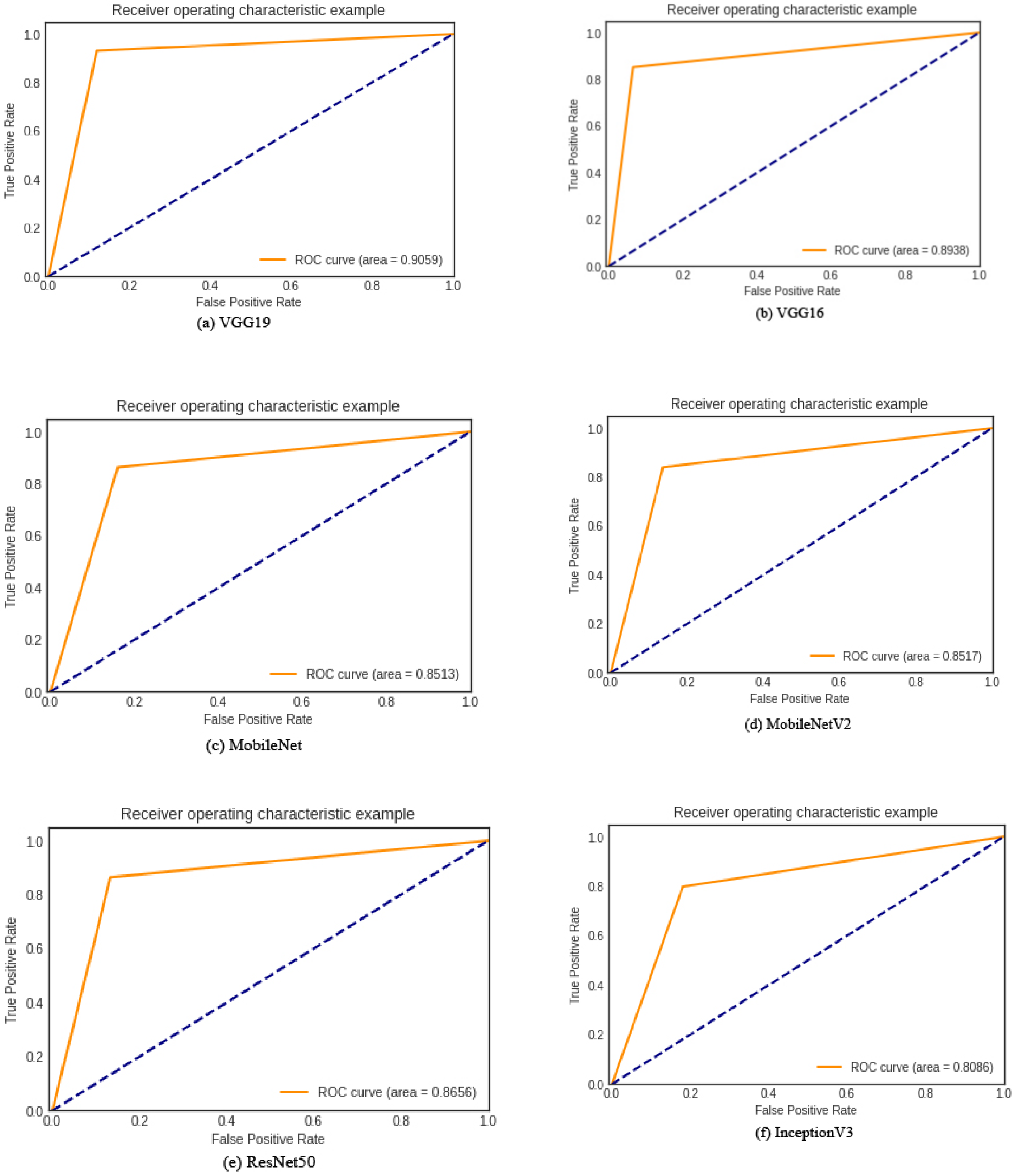
(c) MobileNet

(d) MobileNetV2

(e) ResNet50

(f) InceptionV3

Figure 7. Confusion Matrices for the Evaluated Models



(a) VGG19

(b) VGG16

(c) MobileNet

(d) MobileNetV2

(e) ResNet50

(f) InceptionV3

Figure 8. ROC Curves of the Pre-Trained Models

Table 4. Comparison of MSI Estimation Studies

Study	Cancer Type	Dataset	CNN Model	Performance
Kather et al. [17]	Colorectal Cancer	TCGA Cohort (738 patients)	ResNet18	AUC = 0.84
Cao et al. [18]	Colorectal Cancer	TCGA Cohort (429 WSI) Asian Cohort (785 WSI)	ResNet18	TCGA Cohort AUC = 0.885 Asian Cohort AUC = 0.850
Krause et al. [22]	Colorectal Cancer	TCGA Cohort (256 WSI) NLCS Cohort (1457 WSI) 10000 Synthetic Data	ShuffleNet	TCGA Cohort AUROC = 0.742 NLCS Cohort AUROC = 0.757 Synthetic images only AUROC = 0.743 Combination of both synthetic and real data AUROC = 0.777
Zhang et al. [23]	Rectal Cancer	West China Hospital (High-resolution T2-weighted MRI images - 491 patients)	MobileNetV2	AUC = 0.868
Bustos et al. [25]	Colorectal Cancer	Spain Hospitals EPICOLON (1705 patients) HGUA (283 patients)	ResNet34	AUC = 0.87
Qiu et al. [28]	Colorectal Cancer	TCGA Cohort (100000 H&E unique images)	ResNet34	AUC = 0.809
Su et al. [30]	Gastric cancer	Beijing Cancer Hospital (467 patients)	ResNet18	AUC = 0.785
Muti et al. [27]	Gastric cancer	Multicenter patient cohort (South Korea, Switzerland, Japan, Italy, Germany, UK, and the USA – 4128 patients)	ShuffleNet	Cross validation MSI AUROC = 0.836, EBV AUROC = 0.897. External validation MSI AUROC =0.863, EBV AUROC =0.859
This Study	Colorectal Cancer	TCGA Cohort (150000 H&E unique images)	VGG19	AUC = 0.906

5. Conclusion

The early detection of microsatellite instability (MSI) in colorectal cancer patients plays a crucial role in determining appropriate diagnostic and treatment strategies. This can help minimize side effects, reduce the time and costs associated with the disease, and ultimately improve patient outcomes. Colorectal cancer histopathological images are obtained through special staining techniques and scanning devices applied to tissue samples collected from patients, which are then digitized for analysis. This study proposes a deep transfer learning-based approach to estimate MSI from these images automatically.

Various activation functions, optimization methods, neuron counts, and epochs were explored to identify the optimal classification model for deep learning. Based on these parameters, the results of the models exhibit variations in performance. Classification accuracy is the primary metric used to evaluate the effectiveness of the models.

This study compared nine pre-trained models with identical activation and optimization methods to classify the MSI and MSS histopathological images derived from colorectal cancer patients. VGG19 demonstrated the best performance among the models evaluated, yielding accuracy, precision, recall, and F1-score values of 90.60%, 88.60%, 93.10%, and 90.80%, respectively. Models from the PyTorch library did not meet the expected performance standards. The findings suggest that the VGG19 model, with its superior performance, holds the potential to assist pathologists in making more informed decisions in clinical settings.

Further improvements can be made in future studies, such as exploring more advanced models, optimizing computational resources, and increasing the dataset size. These enhancements are expected to lead to even more accurate results. The high-performance demonstrated in this study contributes significantly to existing literature, offering a valuable tool for researchers and clinicians working on MSI estimation and colorectal cancer diagnosis. By advancing this methodology, we can provide more efficient decision support systems, ultimately aiding in better diagnosis and treatment planning.

References

[1] K. Bardhan and K. Liu, "Epigenetics and colorectal cancer pathogenesis," *Cancers (Basel)*, vol. 5, no. 2, pp. 676-713, 2013. doi: 10.3390/cancers5020676.

[2] R. L. Siegel, K. D. Miller, A. Goding Sauer, S. A. Fedewa, L. F. Butterly, J. C. Anderson, A. Cercek, R. A. Smith, and A. Jemal, "Colorectal cancer statistics, 2020," *CA Cancer J Clin*, vol. 70, no. 3, pp. 145-164, 2020. doi: 10.3322/caac.21601.

[3] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA Cancer J Clin*, vol. 71, no. 3, pp. 209-249, 2021. doi: 10.3322/caac.21660.

[4] American Cancer Society, *Colorectal Cancer Facts & Figures 2020-2022*, 2020.

[5] T. A. A. Tosta, P. R. de Faria, L. A. Neves, and M. Z. do Nascimento, "Computational normalization of H&E-stained histological images: Progress, challenges and future potential," *Artificial Intelligence in Medicine*, vol. 95, pp. 118-132, 2019. doi: 10.1016/j.artmed.2018.10.004.

[6] J. R. Wiśniewski, "Proteomic sample preparation from formalin-fixed and paraffin-embedded tissue," *JoVE (Journal of Visualized Experiments)*, no. 79, p. e50589, 2013. doi: 10.3791/50589.

[7] R. Bonneville et al., "Landscape of microsatellite instability across 39 cancer types," *JCO Precision Oncology*, vol. 1, pp. 1-15, 2017. doi: 10.1200/PO.17.00073.

[8] J. N. Kather, N. Halama, and D. Jaeger, "Genomics and emerging biomarkers for immunotherapy of colorectal cancer," *Seminars in Cancer Biology*, pp. 189-197, 2018. doi: 10.1016/j.semcancer.2018.02.010.

[9] J. N. Nojadeh, S. B. Sharif, and E. Sakhinia, "Microsatellite instability in colorectal cancer," *EXCLI Journal*, vol. 17, p. 159, 2018. doi: 10.17179/excli2017-948.

[10] A. J. Kacew et al., "Artificial intelligence can cut costs while maintaining accuracy in colorectal cancer genotyping," *Frontiers in Oncology*, vol. 11, p. 2066, 2021. doi: 10.3389/fonc.2021.630953.

[11] K. Li, H. Luo, L. Huang, H. Luo, and X. Zhu, “Microsatellite instability: A review of what the oncologist should know,” *Cancer Cell International*, vol. 20, no. 1, pp. 1-13, 2020, doi: 10.1186/s12935-019-1091-8.

[12] J. Xu, K. Xue, and K. Zhang, “Current status and future trends of clinical diagnoses via image-based deep learning,” *Theranostics*, vol. 9, no. 25, p. 7556, 2019, doi: 10.7150/thno.38065.

[13] S. Hosseinzadeh Kassani and P. Hosseinzadeh Kassani, “A comparative study of deep learning architectures on melanoma detection,” *Tissue and Cell*, vol. 58, pp. 76-83, 2019, doi: 10.1016/j.tice.2019.04.009.

[14] N. Coudray, P. S. Ocampo, T. Sakellaropoulos, N. Narula, M. Snuderl, D. Fenyö, A. L. Moreira, N. Razavian, and A. Tsirigos, “Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning,” *Nature Medicine*, vol. 24, no. 10, pp. 1559-1567, 2018, doi: 10.1038/s41591-018-0177-5.

[15] M. Chen, B. Zhang, W. Topatana, J. Cao, H. Zhu, S. Juengpanich, Q. Mao, H. Yu, and X. Cai, “Classification and mutation prediction based on histopathology H&E images in liver cancer using deep learning,” *npj Precision Oncology*, vol. 4, no. 1, p. 14, 2020, doi: 10.1038/s41698-020-0120-3.

[16] Ş. Öztürk and B. Akdemir, “HIC-net: A deep convolutional neural network model for classification of histopathological breast images,” *Computers & Electrical Engineering*, vol. 76, pp. 299-310, 2019, doi: 10.1016/j.compeleceng.2019.04.012.

[17] J. N. Kather, A. T. Pearson, N. Halama, D. Jäger, J. Krause, S. H. Loosen, A. Marx, P. Boor, F. Tacke, U. P. Neumann, et al., “Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer,” *Nature Medicine*, vol. 25, no. 7, pp. 1054-1056, 2019, doi: 10.1038/s41591-019-0462-y.

[18] R. Cao, F. Yang, S. C. Ma, L. Liu, Y. Zhao, Y. Li, D. H. Wu, T. Wang, W. J. Lu, W. J. Cai, et al., “Development and interpretation of a pathomics-based model for the prediction of microsatellite instability in colorectal cancer,” *Theranostics*, vol. 10, no. 24, pp. 11080-11091, 2020, doi: 10.7150/thno.49864.

[19] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848-6856, 2018, doi: 10.48550/arXiv.1707.01083.

[20] A. Echle, H. I. Grabsch, P. Quirke, P. A. van den Brandt, N. P. West, G. G. A. Hutchins, L. R. Heij, X. Tan, S. D. Richman, J. Krause, et al., “Clinical-grade detection of microsatellite instability in colorectal tumors by deep learning,” *Gastroenterology*, vol. 159, no. 4, pp. 1406-1416.e11, 2020, doi: 10.1053/j.gastro.2020.06.021.

[21] H. Lee, J. Seo, G. Lee, J. Park, D. Yeo, and A. Hong, "Two-stage classification method for MSI status prediction based on deep learning approach," *Applied Sciences*, 2021, doi: 10.3390/app11010254.

[22] J. Krause, H. I. Grabsch, M. Kloor, M. Jendrusch, A. Echle, R. D. Buelow, P. Boor, T. Luedde, T. J. Brinker, C. Trautwein, et al., "Deep learning detects genetic alterations in cancer histology generated by adversarial networks," *J. Pathol.*, vol. 254, no. 1, pp. 70-79, 2021, doi: 10.1002/path.5638.

[23] W. Zhang, H. Yin, Z. Huang, J. Zhao, H. Zheng, D. He, M. Li, W. Tan, S. Tian, and B. Song, "Development and validation of MRI-based deep learning models for prediction of microsatellite instability in rectal cancer," *Cancer Medicine*, vol. 10, no. 12, pp. 4164-4173, 2021, doi: 10.1002/cam4.3957.

[24] R. Yamashita, J. Long, T. Longacre, L. Peng, G. Berry, B. Martin, J. Higgins, D. L. Rubin, and J. Shen, "Deep learning model for the prediction of microsatellite instability in colorectal cancer: A diagnostic study," *The Lancet Oncology*, vol. 22, no. 1, pp. 132-141, 2021, doi: 10.1016/S1470-2045(20)30535-0.

[25] A. Bustos, A. Payá, A. Torrubia, R. Jover, X. Llor, X. Bessa, A. Castells, Á. Carracedo, and C. Alenda, "xDEEP-MSI: Explainable bias-rejecting microsatellite instability deep learning system in colorectal cancer," *Biomolecules*, vol. 11, no. 12, 2021, doi: 10.3390/biom11121786.

[26] S. H. Lee, I. H. Song, and H. J. Jang, "Feasibility of deep learning-based fully automated classification of microsatellite instability in tissue slides of colorectal cancer," *Int. J. Cancer*, vol. 149, no. 3, pp. 728-740, 2021, doi: 10.1002/ijc.33599.

[27] H. S. Muti et al., "Development and validation of deep learning classifiers to detect Epstein-Barr virus and microsatellite instability status in gastric cancer: A retrospective multicentre cohort study," *The Lancet Digital Health*, vol. 3, no. 10, pp. e654-e664, 2021, doi: 10.1016/S2589-7500(21)00133-3.

[28] W. Qiu, J. Yang, B. Wang, M. Yang, G. Tian, P. Wang, and J. Yang, "Evaluating microsatellite instability of colorectal cancer based on multimodal deep learning integrating histopathological and molecular data," *Frontiers in Oncology*, p. 3011, 2022, doi: 10.3389/fonc.2022.925079.

[29] C. Saillard, O. Dehaene, T. Marchand, O. Moindrot, A. Kamoun, B. Schmauch, and S. Jegou, "Self-supervised learning improves dMMR/MSI detection from histology slides across multiple cancers," *arXiv preprint arXiv:2109.05819*, 2021, doi: 10.48550/arXiv.2109.05819.

[30] F. Su, J. Li, X. Zhao, B. Wang, Y. Hu, Y. Sun, and J. Ji, "Interpretable tumor differentiation grade and microsatellite instability recognition in gastric cancer using deep learning," *Laboratory Investigation*, vol. 102, no. 6, pp. 641-649, 2022, doi: 10.1038/s41374-022-00742-6.

[31] J. Gibert, "TCGA COAD MSI vs MSS Prediction (JPG)," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/joangibert/tcga_coad_msi_mss_jpg. Accessed: Dec. 24, 2024.

[32] A. Janowczyk, "DOWNLOAD TCGA DIGITAL PATHOLOGY IMAGES (FFPE)," 2018. [Online]. Available: http://www.andrewjanowczyk.com/download-tcga-digital-pathology-images-ffpe. Accessed: Dec. 24, 2024.

[33] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. G., C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 1107-1110, doi: 10.1109/ISBI.2009.5193250.

[34] J. N. Kather, "Histological images for MSI vs. MSS classification in gastrointestinal cancer, FFPE samples," Zenodo, 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2530835. Accessed: Dec. 25, 2024.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.

[36] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4-21, 2016, doi: 10.1109/JBHI.2016.2636665.

[37] S. J. Choi, E. S. Kim, and K. Choi, "Prediction of the histology of colorectal neoplasm in white light colonoscopic images using deep learning algorithms," *Scientific Reports*, vol. 11, no. 1, p. 5311, 2021, doi: 10.1038/s41598-021-84299-2.

[38] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207-1220, 2021, doi: 10.1007/s10044-021-00984-y.

[39] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240-1251, 2016, doi: 10.1109/TMI.2016.2538465.

[40] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint* arXiv:1511.08458, 2015.

[41] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[42] İ. Özkan and E. Ülker, "Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri," *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, vol. 6, no. 3, pp. 85-104, 2017.

[43] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network-A deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679-688, 2018, doi: 10.1016/j.procs.2018.05.069.

[44] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14-23, 2015, doi: 10.1016/j.knosys.2015.01.010.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015, doi: 10.1007/s11263-015-0816-y.

[46] Keras. https://tr.wikipedia.org/wiki/Keras. Accessed 10.07.2024

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556.

[48] S. L. Rabano, M. K. Cabatuan, E. Sybingco, E. P. Dadios, and E. J. Calilung, "Common garbage classification using mobilenet," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2018, pp. 1-4. doi: 10.1109/HNICEM.2018.8666300.

[49] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, 2020. [Online]. Available: https://doi.org/10.1155/2020/7602384.

[50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510-4520. [Online]. Available: https://doi.org/10.48550/arXiv.1801.04381.

[51] K. Bardhan and K. Liu, "Epigenetics and colorectal cancer pathogenesis," *Cancers (Basel)*, vol. 5, no. 2, pp. 676-713, 2013. Doi: 10.3390/cancers5020676.

[52] R. L. Siegel, K. D. Miller, A. Goding Sauer, S. A. Fedewa, L. F. Butterly, J. C. Anderson, A. Cercek, R. A. Smith, and A. Jemal, "Colorectal cancer statistics, 2020," *CA Cancer J Clin*, vol. 70, no. 3, pp. 145-164, 2020. Doi: 10.3322/caac.21601.

[53] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA Cancer J Clin*, vol. 71, no. 3, pp. 209-249, 2021. Doi: 10.3322/caac.21660.

[54] ResNet-18-Pytorch. https://docs.openvino.ai/2021.1/omz_models_public_resnet_18_pytorch_resnet_1 8_pytorch.html. Accessed 25.08.2024

[55] T. A. A. Tosta, P. R. de Faria, L. A. Neves, and M. Z. do Nascimento, "Computational normalization of H&E-stained histological images: Progress, challenges and future potential," *Artificial Intelligence in Medicine*, vol. 95, pp. 118-132, 2019. Doi: 10.1016/j.artmed.2018.10.004.

[56] J. R. Wiśniewski, "Proteomic sample preparation from formalin fixed and paraffin embedded tissue," *JoVE (Journal of Visualized Experiments)*, no. 79, p. e50589, 2013. Doi: 10.3791/50589.

[57] R. Bonneville, M. A. Krook, E. A. Kautto, J. Miya, M. R. Wing, H-Z. Chen, J. W. Reeser, L. Yu, and S. Roychowdhury, "Landscape of microsatellite instability across 39 cancer types," *JCO Precision Oncology*, vol. 1, pp. 1-15, 2017. Doi: 10.1200/PO.17.00073.

[58] J. N. Kather, N. Halama, and D. Jaeger, "Genomics and emerging biomarkers for immunotherapy of colorectal cancer," *Seminars in Cancer Biology*, Elsevier, pp. 189-197, 2018. Doi: 10.1016/j.semcancer.2018.02.010.

[59] J. N. Nojadeh, S. B. Sharif, and E. Sakhinia, "Microsatellite instability in colorectal cancer," *EXCLI Journal*, vol. 17, p. 159, 2018. Doi: 10.17179/excli2017-948.

[60] A. J. Kacew, G. W. Strohbehn, L. Saulsberry, N. Laiteerapong, N. A. Cipriani, J. N. Kather, and A. T. Pearson, "Artificial intelligence can cut costs while maintaining accuracy in colorectal cancer genotyping," *Frontiers in Oncology*, p. 2066, 2021. Doi: 10.3389/fonc.2021.630953.

Authors Contributions

HE, obtained the study concept and dataset, conducted the experiments, and analysed the results. ZP wrote the draft and critically reviewed the draft. All authors reviewed the manuscript.

Conflict of Interest Notice

Authors declare that they have no conflict of interest.

Ethical Approval and Informed Consent

This article contains no data or other information from studies or experiments involving human or animal subjects.

Availability of data and material

The data used in the study were obtained from the Kaggle data repository.
https://www.kaggle.com/datasets/joangibert/tcga_coad_msi_mss_jpg

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAUCIS

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

http://saucis.sakarya.edu.tr/

e-ISSN: 2636-8129
Publisher: Sakarya University

Vol. 8, No. 1, 152-170, 2025
DOI: 10.35377/saucis...1635558

SAKARYA UNIVERSITY

REVIEW

Early Prediction of Students’ Performance Through Deep Learning: A Systematic and Bibliometric Literature Review

Ahmet Kala[1,2*], **Orhan Torkul**[1], **Tuğba Tunacan Yıldız**[4], **Ihsan Hakan Selvi**[5]

[1] Sakarya University, Faculty of Science, Industrial Engineering Department, Sakarya, Türkiye, ror.org/04ttnw109
[2] Sakarya University of Applied Sciences, Department of Information Technologies, Sakarya, Türkiye, ror.org/01shwhq58
[4] Abant Izzet Baysal University, Faculty of Science, Industrial Engineering Department, Bolu, Türkiye, ror.org/01x1kqx83
[5] Sakarya University, Faculty of Science, Information Systems Engineering, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Ahmet Kala, Sakarya University of Applied Sciences, Department of Information Technologies, Sakarya, Türkiye
ahmetkala@subu.edu.tr

ABSTRACT

Early prediction of student performance is a critical and challenging task in the field of Educational Data Mining (EDM), encompassing all levels of education. Although there is extensive literature on student performance within EDM, studies specifically focused on early prediction are limited and mostly rely on traditional machine learning methods. However, in recent years, the importance and use of deep learning (DL) methods have increased due to their ability to process large datasets. This systematic literature review focuses on the early prediction of student performance using DL techniques. A total of 39 articles selected from the Scopus and Web of Science databases were analyzed using systematic and bibliometric methods. The review addresses five key research questions, including the distribution of studies by publication year, type, and education level; the datasets and features used; DL models and techniques; the timing of early predictions; and the challenges, limitations, and opportunities encountered. The bibliometric analysis, conducted with the VOSviewer program, visualized relationships between keywords, authors, and articles. Overall, this review provides a comprehensive synthesis of existing research on the early prediction of student academic performance using DL, offering valuable insights into trends and opportunities for researchers, educators, and policymakers.

Keywords: Education, Educational data mining, Early prediction, Student performance, Deep learning, Bibliometric literature review, Systematic literature review

Article History:
Received: 07.02.2025
Revised: 04.03.2025
Accepted: 06.03.2025
Published Online: 27.03.2025

1. Introduction

Educational data mining (EDM) is an interdisciplinary field that focuses on extracting meaningful insights from educational data to enhance learning and teaching processes [1]. The International Educational Data Mining Society emphasizes that EDM aims to analyze educational data types, predict student performance, and develop innovative methods to improve learning outcomes. With the advent of deep learning (DL) techniques, EDM has gained significant momentum, enabling more accurate and early predictions of student performance compared to traditional machine learning (ML) approaches. EDM combines social science methods such as psychometry, psychology, and broad-based mathematical methods from statistics, artificial intelligence, and machine learning (ML) to deep learning (DL) [2].

Early prediction is defined as implementing predictive models utilizing key variables to accurately forecast student failure or dropout as early as possible [3], [4]. It involves leveraging technological information to detect potential or actual academic problems. Detecting at-risk students promptly allows for timely interventions, support, and preventative strategies, aiming to prevent academic setbacks. Student information sources for early predictions are diverse, encompassing questionnaires, activities, events, log files, demographic data, evaluation results, behavior data, grades, affective variables, and more. The challenge of early prediction is amplified in the EDM field due to numerous factors influencing a student’s final status. This challenge holds critical implications globally across all educational stages (primary, secondary, and tertiary education), necessitating early identification of at-risk students to implement adequate preventative measures and interventions [5].

Previous research in EDM has extensively explored various aspects of student performance prediction, including machine learning [6] – [9], student dropout [10], learning analytics [11] – [13], and data mining [12], [14], [15]. While traditional ML

Cite as: A. Kala, O. Torkl, T. Tunacan Yildiz, IH. Selvi, “Early prediction of students’ performance through deep learning: A systematic and bibliometric literature review,” *Sakarya University Journal of Computer and Information Sciences*, vol. 8, no. 1, pp. 152-170, 2025, doi: 10.35377/saucis...1635558

methods such as Decision Trees (DT), Random Forests (RF), and Support Vector Machines (SVM) have been widely used, recent studies highlight the superior performance of DL techniques in handling complex and large-scale educational data [29]. However, a comprehensive comparison of DL and ML models regarding computational cost, training time, and prediction accuracy remains underexplored. This gap necessitates a deeper analysis of the trade-offs between these approaches.

Researchers have written numerous articles in predicting student performance in EDM. These literature studies by researchers focus on machine learning [6], [7], [8], [9], student dropout [10], learning analytics [11], [12], [13], data mining [12], [14], [15], student performance predictions [16], [17], [18], e-learning [19], computer-supported collaborative learning [20], student retention [21], feature selection [22], affecting factors [23], classroom learning (Khan and Ghosh, 2021), predicting academic success [24], early prediction [25], [26], and big data [27], [28] topics. While traditional ML methods such as Decision Trees (DT), Random Forests (RF), and Support Vector Machines (SVM) have been widely used, recent studies highlight the superior performance of DL techniques in handling complex and large-scale educational data [29]. However, no one specifically focuses on early student performance prediction through DL techniques.

Previous literature studies emphasized the need for a literature review to examine the impact of DL methods on early prediction of student performance. In this study, we aimed to conduct a literature review that encompasses these two research areas. The contributions of this literature review article are as follows:

- Provides an overview of DL techniques and algorithms in early student performance prediction.
- Identifies existing uses of DL for early student performance prediction through a systematic literature review.
- Identifies gaps in the literature and highlights future research areas to enhance early prediction of student performance with DL.
- A bibliometric literature review explains relationships between keywords, authors, and articles and presents these relationships visually.

This review article is divided into six sections. Section 2 explains the steps of the review methodology used. Search results are presented in Section 3. Section 4 provides a systematic literature analysis of selected articles. Bibliometric analysis is introduced in Section 5. Section 6 presents the conclusion of the current literature.

2. Literature Review Methodology

This literature review is divided into two sections: Systematic and Bibliometric Review. The details of the review steps are presented in the following subsections.

2.1. Systematic Review

This study adopts a systematic literature review approach, adhering to the guidelines proposed by Kitchenham for software engineering researchers [30]. The primary objective is to analyze the current landscape of DL techniques and algorithms for predicting students' performance early, providing insights into existing studies and identifying gaps for future research. The systematic literature review procedure is outlined as follows.

1. Research Questions: The study addresses the following research questions (RQs):
    - RQ1: What is the distribution of studies by publication year, publication type, education type, and level?
    - RQ2: What datasets, attributes, and predicted attributes are used for early prediction?
    - RQ3: What DL models and techniques are employed for early prediction, and what are the performance evaluation methods?
    - RQ4: How early can student academic performance be predicted with an acceptable level of accuracy?
    - RQ5: What are the main challenges, limitations, and research opportunities identified in previous studies?
2. Search Process The systematic analysis encompasses studies from Scopus and Web of Science library databases until December 20, 2023. The search utilizes five key terms: "educational data mining," "data mining," "machine learning," "deep learning," and "early prediction of student performance." The search terms are structured for both Scopus and Web of Science databases. The search used the following search query:
    - Scopus: TITLE-ABS-KEY (("educational data mining" OR "data mining" OR "machine learning" OR "deep") AND ("deep learning" OR ("deep" AND "Neural Network")) AND student AND performance AND early)

- Web of Science: (ALL= (((“educational data mining” OR “data mining” OR “machine learning” OR “deep”)))) AND AB= (((“deep learning” OR (“deep” AND “Neural Network”)) AND student AND performance AND early))

3. Inclusion and Exclusion Criteria The review includes DL techniques and algorithms for early prediction of students’ performance, published until December 20, 2023. Excluded topics do not involve early prediction of student performance and DL, lack appropriate abstracts and keywords, must be in English, or have inaccessible full texts.
4. Quality Assessment Research papers from active scholarly journals in the specified databases are considered of sufficient quality, while those outside these databases are excluded from the Review.
5. Data Collection Relevant data was extracted for the selected articles and organized into an electronic spreadsheet. The information includes article details, type and level of education, datasets and attributes, DL/ML models, evaluation methods, early prediction status, limitations, contributions, and future research suggestions.
6. Data Analysis The collected data is analyzed by defined research questions. The analysis results are synthesized, and common themes are identified by comparing findings related to each research question.

2.2. Bibliometric Review

Bibliometric Review incorporates a research approach involving bibliometric analysis, which quantitatively assesses publications in scientific literature and their interconnections. This investigation aims to comprehend scientific production, publication trends, significant researchers, highly cited works, and institutional contributions within a specific subject, field, or discipline. Bibliometrics provides the methods and indicators commonly employed in these analyses. The outcomes of bibliometric reviews are typically represented through graphs, diagrams, and maps.

In this article, we utilized VOSviewer (Visualization of Similarity), a widely adopted tool for visualizing and conducting network literature analysis. This tool integrates text mining and network analysis techniques to identify crucial concepts and connections within literature. Such a tool proves beneficial for researchers and information professionals in pinpointing significant focal points within a field, focusing on specific topics, or monitoring developments in a particular subject.

Bibliometric data, obtained from the Scopus website, where all information of the 39 selected studies was found at the end of the systematic literature review process, were exported in CSV format. Using this dataset, an analysis was conducted in the VOSviewer program to comprehend and visually represent relationships among keywords, authors, and articles.

2. Search Results

Two hundred seventy-eight articles published in the Web of Science and Scopus databases up to November 2023 were obtained from the abovementioned search process. Of these articles, 69 were found to be duplicates present in both databases, and one was excluded. Consequently, a unique set of 209 articles was reached. Of these, 113 were journal articles, 61 were from international conferences, 21 were conference reviews, and 13 were of different types such as books, book chapters, meetings, proceedings papers, and Reviews. Each article’s abstract was meticulously examined, and 142 articles were excluded at this stage. Among the excluded articles, 116 were unrelated to student performance prediction and deep learning, 24 lacked free access and full-text availability, and two needed to be in English. A selection process involving reading the full texts was applied to the remaining articles, and 28 articles not related to student performance prediction and deep learning were also excluded. As a result, 39 articles were chosen.

The remaining 39 articles addressed five main research questions and conducted bibliometric analysis in the VOSviewer program. In the discussion sections (Sections 5 and 6), the obtained results were detailed and discussed, providing a comprehensive overview of the literature on the subject.

4. Systematic Analysis of Deep Learning in Early Prediction of Academic Performance Within EDM

This section of the systematic literature review discusses the findings obtained in response to the identified research questions.

4.1. RQ1. What is the Distribution of Studies by Publication Year, Publication Type, Education Type, and Level?

Table 1 presents the critical details of the selected 39 studies. These studies were published between 2017 and December 20, 2023. All the studies comprise journal and conference publications, with journal studies accounting for 65% of the total (24 studies).

Table 1. Distribution of Essential Information for Selected Studies

Ref.	Year	Type	Cited by	Scopus	WoS	Source Title	Publisher	Education Types			Education Levels		
								F	B	E	S	U	All
[31]	2022	J	8	✓	✓	Complex and Intelligent Systems	Springer	✓				✓	
[32]	2023	J	1	✓		Revue d'Intelligence Artificielle	IIETA	✓			✓		
[33]	2023	C	1	✓		ITIKD 2023	IEEE Inc.			✓		✓	
[34]	2023	J	0	✓	✓	Applied Sciences	MDPI			✓		✓	
[35]	2022	C	1	✓		IC3SIS 2022	IEEE Inc.			✓			✓
[36]	2023	J	5	✓		Expert Systems with Applications	Elsevier Ltd			✓		✓	
[37]	2023	J	1	✓		SN Computer Science	Springer	✓				✓	
[38]	2022	C	0	✓		TALE 2022	IEEE Inc.			✓		✓	
[39]	2023	J	2	✓		Heliyon	Elsevier Ltd			✓		✓	
[40]	2023	C	0	✓	✓	COMPSAC	IEEE Computer Society	✓				✓	
[41]	2023	J	1	✓		IEEE Access	IEEE Inc.		✓			✓	
[42]	2023	J	1	✓	✓	IEEE Access	IEEE Inc.		✓			✓	
[43]	2022	J	6	✓	✓	Applied Sciences	MDPI			✓		✓	
[44]	2021	J	20	✓		IEEE Access	IEEE Inc.		✓			✓	
[45]	2018	C	14	✓		INAPR 2018	IEEE Inc.	✓				✓	
[46]	2020	J	242	✓	✓	Computers in Human Behaviour	Elsevier Ltd			✓		✓	
[47]	2019	J	45	✓		Sustainability	MDPI			✓		✓	
[48]	2021	J	2	✓		Sustainability	MDPI			✓		✓	
[49]	2021	J	1	✓		JATIT	Little Lion Scientific	✓			✓		
[50]	2020	C	8	✓		EDM 2020	IEDMS			✓			✓
[51]	2022	J	6	✓	✓	iJET	IAOE			✓			✓
[52]	2021	J	16	✓	✓	iJET	IAOE	✓					✓
[53]	2018	C	29	✓		EDM 2018	IEDMS			✓			✓
[54]	2019	J	17	✓		Computing	Springer			✓			✓
[55]	2020	C	12	✓	✓	Lecture Notes in Computer Science	Springer			✓		✓	
[56]	2021	C	8	✓	✓	Lecture Notes in Computer Science	Springer	✓			✓		
[57]	2019	C	33	✓	✓	ISET 2019	IEEE Inc.		✓			✓	
[58]	2021	J	4	✓	✓	Optical Memory and Neural Networks	Pleiades journals			✓			✓
[59]	2020	C	25	✓		EDM 2020	IEDMS			✓		✓	
[60]	2020	J	51	✓	✓	Journal of Learning Analytics	UTS ePRESS			✓		✓	
[61]	2019	C	23	✓	✓	ICPS	ACM	✓				✓	
[62]	2019	J	53	✓		International Journal of Intelligent Systems	John Wiley and Sons Ltd			✓		✓	
[63]	2017	C	65	✓		EDM 2017	IEDMS			✓			✓
[64]	2020	J	18	✓	✓	IEEE Access	IEEE Inc.			✓	✓		
[65]	2021	J	9	✓	✓	IEEE Access	IEEE Inc.	✓				✓	
[66]	2021	J	10	✓		JOIV	Politeknik Negeri Padang			✓		✓	
[67]	2021	J	41	✓	✓	IEEE Access	IEEE Inc.	✓				✓	
[68]	2021	J	70	✓	✓	IEEE Access	IEEE Inc.			✓		✓	
[69]	2021	J	48	✓	✓	Sustainability	MDPI	✓			✓		

F: face-to-face education; B: hybrid (blended) education; E: e-learning; S: secondary school; U: university

The sources with the highest number of publications are listed in Figure 1. Twenty-three studies, approximately half of the total, were published by six different sources. Notable among these sources are IEEE Access (7 studies), International Conference on Educational Data Mining (ICEDM) (4 studies), Sustainability (3 studies), Applied Sciences (2 studies), The International Journal of Engineering Technologies (IJET) (2 studies), and Lecture Notes in Computer Science (LNCS) (2 studies).

The publishers with the highest number of publications are listed in Figure 2. Furthermore, 31 studies, constituting 79% of the total, were published by six different publishers. Prominent publishers include IEEE Inc. (12 studies), Multidisciplinary

Digital Publishing Institute (MDPI) (5 studies), Springer (5 studies), the International Educational Data Mining Society (IEDMS) (4 studies), Elsevier Ltd (3 studies), and the International Association of Online Engineering (IAOE) (2 studies).

The selected studies have been classified according to the type of education system and education level. As shown in Figure 3, the studies encompass e-learning (23 studies, 59%), traditional face-to-face education (12 studies, 31%), and hybrid (blended) education (4 studies, 10%). Upon evaluation of these studies, as seen in Figure 3, it was determined that 26 out of 39 studies (67%) were conducted with university students, 5 out of 39 studies (13%) focused on secondary school students, and the remaining 8 out of 39 studies (21%) were related to e-learning courses at all education levels. The predominant reasons for conducting studies primarily at the university level include data accessibility, ease of data collection, and the widespread use of computer-assisted education. Additionally, it was observed that studies at the higher education level were predominantly carried out at the undergraduate level. No studies were conducted at the graduate level or in primary schools.

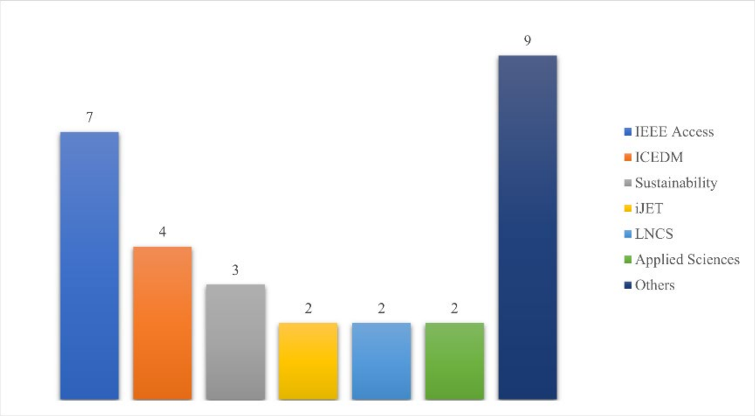


Figure 1. The Sources with the Highest Number of Publications

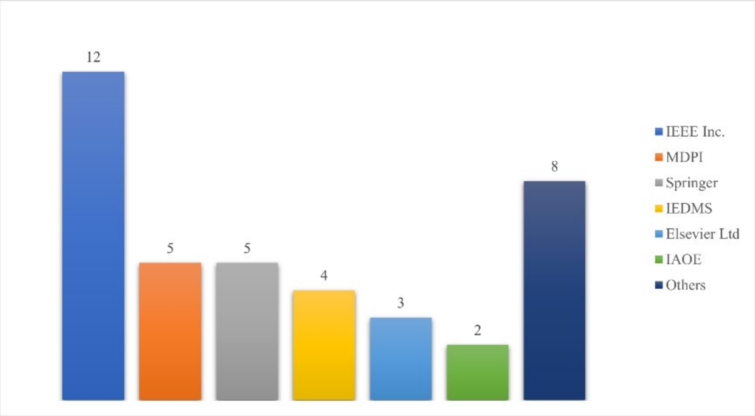


Figure 2. The Publishers with the Highest Number of Publications

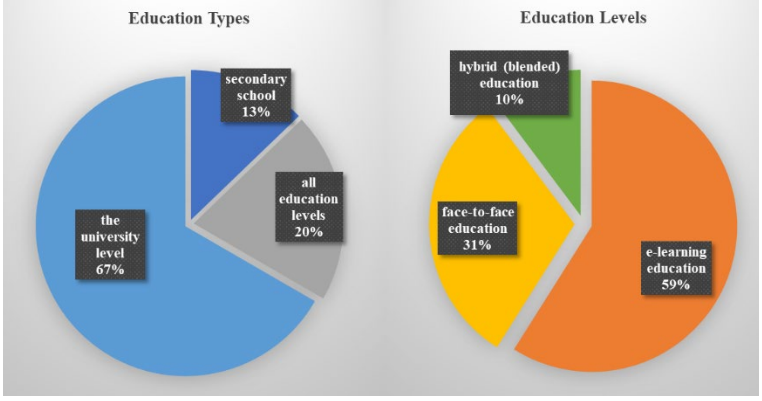


Figure 3. Education Types and Levels

Upon analyzing citation counts, it was determined that an average of 23 citations were obtained for each study. The citation counts, totals, and averages by year are presented in detail in Table 2.

Table 3 presents the researcher density in the selected articles. Authors are weighted based on the number of contributors to each paper. For example, in a paper with n authors, each author contributes to their country with a weight of 1/n.

The table indicates that the People's Republic of China is the most active country in this field, followed by the United States of America, Saudi Arabia, Pakistan, India, Indonesia, Taiwan, Canada, Egypt, Bahrain, Philippines, Japan, Yemen, and Malaysia. Among other countries, South Korea, Brazil, Oman, United Kingdom, Kerala, Australia, Spain, Tunisia, Nigeria, and Germany are noteworthy for their substantial contributions to this field.

Table 2. The Citation Counts, Totals, and Averages by Year

Year	Count of Cited by	The sum of Cited by	Average of Cited by
2017	1	65	65
2018	2	43	22
2019	5	171	34
2020	6	356	59
2021	11	229	21
2022	5	21	4
2023	9	12	1
Total	39	897	23

Table 3. The Authors' Countries Distribution

Country	Count	Score	References
China	10	8.9	[31], [35], [40], [41], [51], [54], [55], [59], [64], [69]
United States	7	5.3	[50], [53], [56], [61], [63], [64], [66]
Saudi Arabia	7	3.3	[36], [47], [52], [62], [65], [69]
Pakistan	6	3	[36], [47], [62], [68], [69]
India	4	2.8	[32], [35], [37], [59]
Indonesia	3	2.6	[42], [43], [46]
Taiwan	2	1.4	[43], [48]
Canada	2	1.1	[60], [69]
Egypt	2	1.1	[65], [67]
Bahrain	1	1	[33]
Yemen	1	1	[39]
Japan	1	1	[38]
Philippines	1	1	[57]
Malaysia	1	1	[49]
South Korea	1	1	[66]
Other	8	3.6	[35], [36], [37], [44], [46], [53], [56], [69]

4.2. RQ2. Datasets, Attributes, and Outcomes Used for Early Prediction of Student Performance

The datasets used for early student performance prediction vary widely regarding type, structure, and quality. The distribution of these datasets is detailed in Table 4.

Upon evaluating Table 4, as seen in Figure 4, it was observed that 63% of the studies (25 studies) were associated with Massive Open Online Courses (MOOCs), Virtual Learning Environments (VLEs), Learning Management Systems (LMSs), and Intelligent Teaching Systems (ITSs), with VLEs prominently featured among these datasets. Additionally, open datasets were generally employed in 25% of the studies (10 studies). Among these, the Open University Learning Analytics Dataset (OULAD) VLE general dataset, encompassing weekly VLE activity information for students, was utilized in ten studies. As seen in Figure 4, specifically 15 studies, one-third of the selected studies constitute general datasets. Among these general datasets are OULAD (10 studies), The edX open dataset (1 study), Udacity Data (1 study), xAPI-Edu-Data (1 study), and UCI Machine Learning Repository (4 study). These general datasets have unique advantages and limitations. The Open University Learning Analytics Dataset (OULAD) provides rich data on student interactions but can negatively impact model performance due to imbalanced class distributions. Similarly, datasets from MOOC platforms like edX are large-scale but often lack detailed behavioral information. These limitations can be addressed through data augmentation or advanced preprocessing techniques, which can enhance the reliability and accuracy of predictive models.

The categorization of attributes and predicted features for the early prediction of student performance is also detailed in Table 4. Accordingly, it was observed that the most frequently used attributes include student demographic information (age, gender, region, address, family size, mother's education, father's education, mother's job, father's job, current health status, etc.), evaluation results, activity data, LMS log data, and behavioral information. Other attributes such as student grades, grade points, test scores, learning outcomes, student details, and snapshot data were noteworthy. In evaluating the research, it was noted that 67% of the studies (26) aimed to predict final scores and grades. Other outcomes encompass grade point averages (GPAs), learning outcome scores, post-test scores, application scores, quiz performance scores, learning behavior, lecture grades, dropouts, and snapshot grades.

Table 4. Datasets, Features, and Estimated Attributes Distribution

References	General / Specific	Types	Datasets	Features							Estimated Attributes							
				Demographics	Academic	Events	Assessments	Behaviors	Log files	Others	Final Grades	Final Scores	GPAs	Dropouts	Graduation	The GPAs	Learning	Other
[31]	S	Other	The datasets of the university in Beijing	✓				✓								✓		
[32]	S	Other	The government and self-financed engineering colleges dataset	✓	✓		✓						✓					
[33], [34], [36], [39], [42], [46], [47], [59], [62], [68]	G	VLE	OULAD	✓	✓	✓	✓				✓							
[35]	S	MOOCs	MITx and Harvard X courses							✓				✓				
[37]	S	Other	The publicly accessible data source		✓	✓	✓											✓
[38]	S	Other	M2B Learning systems							✓	✓							
[40]	S	LMS	A sophomore course from the School of Computer Science and Engineering			✓	✓		✓			✓						
[41]	S	Other	The dataset of the university	✓	✓	✓	✓	✓				✓						
[43]	S	LMS	Moodle LMS			✓				✓		✓						✓
[44]	S	Other	The CS1 course is compulsory for 16 STEM degrees at the Federal University of the Amazonas.			✓					✓							
[45]	S	Other	The data used in this experiment are from computer science at Bina Nusantara University.				✓						✓					
[48]	S	LMS	A general education course at a university in northern Taiwan.			✓	✓		✓			✓						
[49], [56], [58], [69]	G	ITSs	UCI Machine Learning Repository	✓	✓	✓	✓	✓			✓							
[50]	S	ITSs and Other	Pyrenees and iSnap							✓								✓
[51]	G	MOOCs	The edX open dataset							✓							✓	
[52]	S	Other	Two data sets are mathematics and Portuguese language courses.	✓	✓	✓	✓				✓							
[53]	G	MOOCs	Udacity Data			✓									✓			
[54]	S	Other	The datasets are from two real e-learning system	✓	✓	✓	✓	✓	✓		✓							

Table 4. (Continued)

[55]	S	Other	Datasets from 505 university students			✓		✓				✓						
[57]	S	LMS	Moodle LMS			✓		✓			✓							
[60]	S	LMS	Moodle LMS			✓	✓		✓		✓							
[61]	S	Other	A dataset from University X				✓									✓		
[63]	S	MOOCs	Code.org							✓								✓
[64]	S	LMS	The Blackboard LMS			✓	✓	✓	✓		✓							
[65]	S	Other	The university dataset				✓						✓					
[66]	S	LMS	The Cyber University LMS system	✓	✓	✓			✓				✓					
[67]	S	Other	Students' grades				✓											✓
[69]	G	LMS	xAPI-Edu-Data	✓	✓	✓				✓	✓							

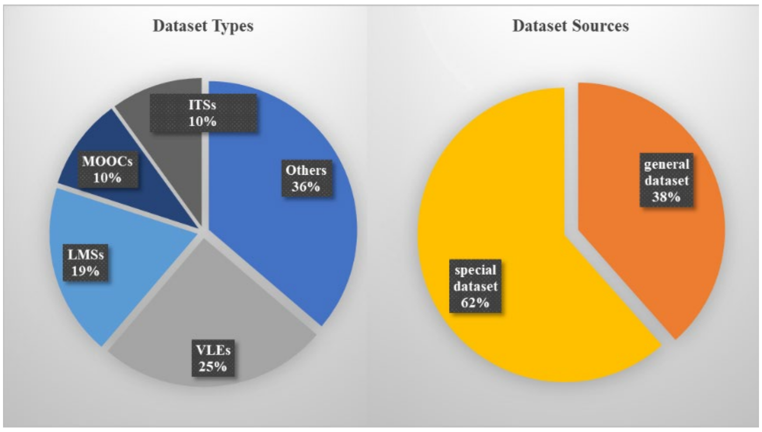


Figure 4. Dataset Types and Sources

4.3. RQ3. Proposed Models, Compared Models, and Performance Evaluation Methods for Early Prediction of Student Academic Performance

The distribution of the proposed models, compared models, classification, and evaluation methods are presented in Table 5. Figure 5 illustrates the proposed DL models in the selected articles. Long Short-Term Memory (LSTM), Deep Feed Forward Neural Networks (DFFNN), Bidirectional LSTM (BLSTM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Deep Belief Network (DBNN), Deep Neural Networks (DNN) DL, and Hybrid DL techniques were used. Among these techniques, it has been observed that many LSTM techniques were used.

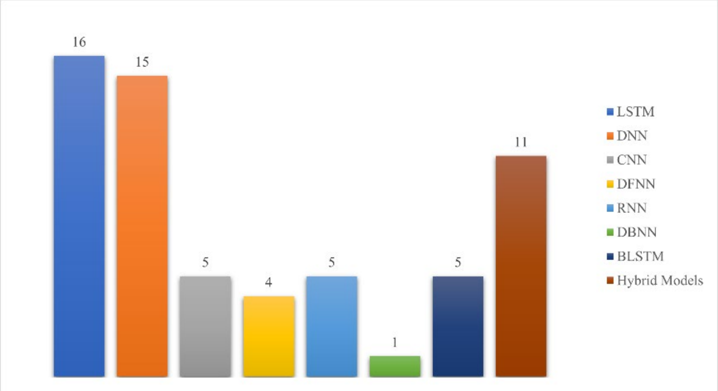


Figure 5. Proposed DL Models

It was observed that hybrid DL techniques were used in nine studies, and they were CNN-LSTM [31], [32], [43], [55], Levy Flight Rock Hyraxes Swarm Optimization (LFRHSO)-RNN [37], Atom Search Optimization (ASO)-DBN [58], BLSTM + the Condition Random Field (CRF) [65], DNN- Integrated Framework Based on Latent Variational Autoencoder (LVAEPre) [64], and LSTM-ANNs [39].

Among these studies, Chen et al. (2022) proposed a hybrid intelligent framework comprised of CNN and LSTM models to address the issue of unstable data distribution and predictability in VLE [43]. Li et al. (2020) Introduced the Sequential Prediction Based on Deep Network (SPDN) model, consisting of CNN and LSTM DL models, to predict students' 13-week course performance using online learning records and blog data from the university campus network [55]. In another study, Li et al. (2022) suggested an end-to-end hybrid DL model that combines CNN and LSTM models to automatically extract features from multi-source heterogeneous four-day behavioral data of students [31].

Table 5. The Distribution of The Proposed Models, Compared Models, Classification, and Evaluation Methods

References	Reg. / Class.	Classification	Proposed Model								Compared Models										Evaluation Methods						
			LSTM	Deep ANN	CNN	DFNN	RNN	DBN	BLSTM	Hybrid	SVM	RF	LR	DT	KNN	NB	ANN	GBM	AdaBoost	Others	Accuracy	F1 score	Precision	Recall	ROC-AUC	RMSE	Others
[31]	C	3								✓		✓	✓	✓					✓	✓	✓	✓	✓	✓			
[32]	R C	4	✓	✓						✓	✓	✓						✓		✓	✓	✓	✓	✓		✓	
[33]	C	2		✓							✓					✓	✓			✓	✓						
[34]	C	4				✓					✓	✓	✓			✓					✓	✓	✓	✓			
[35]	C	2		✓								✓	✓		✓	✓				✓	✓		✓				
[36]	C	2	✓								✓		✓	✓	✓		✓			✓	✓	✓	✓	✓	✓		✓
[37]	C	2		✓	✓		✓	✓		✓					✓						✓			✓			✓
[38]	C	2	✓									✓			✓	✓		✓		✓		✓	✓	✓			
[39]	C	4	✓			✓	✓			✓										✓	✓	✓	✓	✓			
[40]	C	2							✓												✓				✓		
[41]	C	3		✓	✓						✓			✓	✓					✓	✓	✓		✓			
[42]	C	2	✓																	✓	✓	✓					✓
[43]	C	2	✓							✓	✓		✓								✓	✓	✓	✓			
[44]	C	2		✓								✓								✓	✓	✓	✓	✓	✓		
[45]	R C	2		✓									✓	✓							✓					✓	
[46]	C	2		✓							✓		✓								✓		✓	✓			✓
[47]	C	2	✓								✓		✓				✓				✓		✓	✓			✓
[48]	C	2	✓		✓		✓													✓	✓	✓	✓	✓			
[49]	C	2		✓								✓						✓		✓	✓			✓			✓
[50]	C	2	✓										✓							✓	✓	✓	✓	✓	✓		
[51]	C	5		✓							✓		✓	✓		✓						✓	✓	✓			
[52]	C	2		✓							✓	✓									✓	✓	✓	✓			
[53]	C	2							✓											✓					✓		
[54]	C	2	✓	✓			✓				✓	✓								✓	✓	✓					
[55]	C	2							✓	✓		✓	✓	✓		✓					✓	✓			✓		
[56]	C	5			✓						✓	✓	✓	✓					✓	✓	✓						
[57]	C	2				✓															✓				✓		
[58]	C	3								✓	✓				✓		✓					✓	✓				✓
[59]	C	4	✓								✓		✓									✓		✓			
[60]	C	2	✓								✓	✓	✓	✓	✓	✓	✓	✓							✓		
[61]	R C	2	✓																	✓	✓	✓					✓
[62]	C	2	✓										✓				✓				✓		✓	✓			✓
[63]	C	2	✓										✓									✓	✓	✓			

Table 5. (Continued)

[64]	C	2		✓						✓				✓					✓		✓	✓	✓	✓			✓
[65]	C	2							✓	✓	✓	✓	✓	✓		✓	✓				✓	✓	✓	✓			
[66]	C	2		✓							✓	✓		✓							✓	✓	✓	✓	✓		
[67]	C	2		✓							✓	✓	✓	✓	✓			✓			✓	✓					✓
[68]	C	4				✓					✓	✓			✓		✓	✓	✓	✓	✓	✓	✓				
[69]	C	6	✓		✓		✓		✓		✓	✓	✓		✓	✓					✓	✓	✓	✓			

Venkatachalam and Sivanraju (2023) proposed a hybrid Student Achievement Prediction Model Using the Distinctive Deep Learning (SADDL) framework, which includes three modules: LSTM, CNN, and Multilayer Perceptron (MLP). The SADDL model has demonstrated superior performance to other machine learning models when utilizing the students' physiological, academic, and demographic data to achieve results [32]. Sayed et al. (2023) presented a method to predict student performance using the suggested hybrid model (LFRHSO-RNN) and a large student, administrator, and teacher data dataset. Additionally, they compared the proposed model with different hybrid models such as Grey Wolf Optimizer (GWO)-RNN, Fire- fly Algorithm (FFA)-RNN, Bat algorithm (BAT) -RNN, and Particle Swarm Optimization (PSO)-RNN [37]. Surenthiran et al. (2021) proposed a hybrid model based on DBNN, supported by ASO, which has been utilized to categorize students according to their historical performance [58]. Uliyan et al. (2021) reported high accuracy in examining students' retention status using a hybrid DL technique consisting of BLSTM and CRF [65].

Du et al. (2020) proposed an integrated framework, LVAEPre, based on latent variational autoencoder (LVAE) with DNN to alleviate the imbalanced distribution of the dataset and provide early warnings for students at further risk [64]. Al-azazi and Ghurab (2023) proposed a hybrid ANN-LSTM model consisting of Artificial Neural Network (ANN) and LSTM models to predict students' performance on a day-wise multi-class basis [39].

It has been observed that the proposed DL models were primarily compared with basic ML models such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (KNN), Artificial Neural Network (ANN), Gradient Boosting Machine (GBM), Adaptive Boosting (AdaBoost), and Naive Bayes (NB). Some studies compared the proposed models with the LSTM, CNN, RNN, and DNN DL models. Figure 6 illustrates the ML models in the selected articles that were compared.

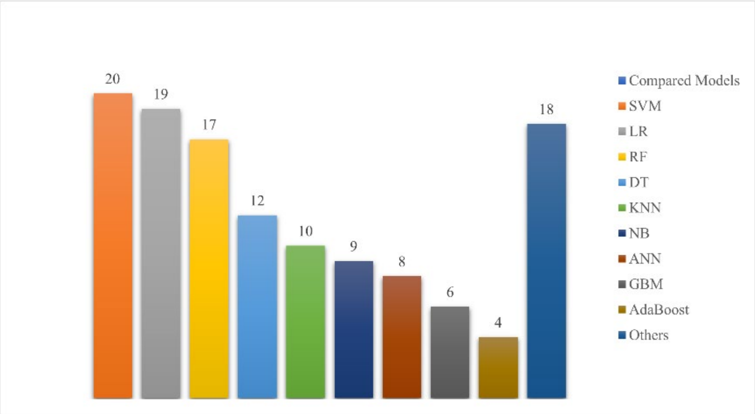


Figure 6. Compared ML Models

Consequently, the proposed DL models obtained the same or better results than the compared ML and DL models, demonstrating their effectiveness in early student performance prediction. In particular, hybrid DL models that integrated multiple architectures, such as CNN-LSTM and ASO-DBN, showed notable improvements in prediction accuracy, robustness, and generalizability across different datasets. These findings highlight the advantages of utilizing deep learning techniques over traditional machine learning methods, especially in handling complex, high-dimensional educational data and capturing intricate patterns in students' learning behaviors.

Figure 7 illustrates the classification types in the selected articles. It was observed that while regression models were performed in only two studies [45], [61], other studies were interested in classification. Considering all the studies that have been classified, Outcomes in 72% of them were divided into two classes, with the remaining 28% predicted by three classes, four classes, five classes, and six classes. In binary classification, pass-fail, true-false, passed-withdrawn, successful-unsuccessful, and at risk-not at risk can be given as examples. As several models are usually built, evaluating them and selecting the best-performing model is crucial. Figure 7 illustrates the evaluation methods in the selected articles. Root Mean Square Error (RMSE) was used in regression studies, while accuracy, precision, recall, F1 score, and Area Under the Curve (AUC) performance evaluation methods were used in classification studies.

4.4. RQ4. Early Prediction of Student Performance

Considering all the studies selected for the early prediction of student performance, it was observed that the early prediction times varied depending on the course length. In some studies, information regarding the length of the educational process was not provided [32], [37], [41], [44], [45], [49], [51], [52], [58], [61], [63], [64], [65], [66], [69]. The course length, prediction frequency, and early prediction time are given in Table 6.

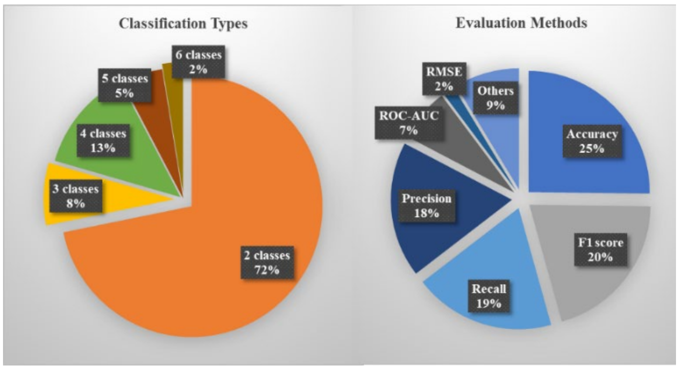


Figure 7. Classification Types and Evaluation Methods

Table 6. Distribution of Early Prediction Time

References	Course Length	Prediction Frequency	Early Prediction Time
[35]	5 Weeks	Weeks 1-5	Week 1
[38]	7 Weeks	Weeks 1-7	Week 7
[53]	8 Weeks	Weeks 1-8	Week 3
[55]	13 Weeks	Weeks 1-13	Week 7
[60]	16 Weeks	Weeks (6,8,10,12,16)	Week 6
[48]	17 Weeks	Weeks (3,6,9,12,18)	Week 9
[40]	21 Weeks	Weeks 1-12	Week 8
[34], [68]	40 Weeks	The sequence length (10,20,30,40,50,60,70,80,90,100)	20% sequence length
[47]	40 Weeks	Weeks (5,10,20,30,40)	Week 10
[62]	40 Weeks	Weeks (5,10,15,20,25)	Week 10
[46]	40 Weeks	Quarter 1-4	Quarter 1
[36]	40 Weeks	Weeks (5,10,20,30,38)	Week 20
[59]	40 Weeks	Weeks (5,10,15,20)	Week 20
[39]	40 Weeks	Days (0 and 270)	First 90 days
[33]	40 Weeks	Days (0,7,14,30,45,60)	First 0 days
[42]	40 Weeks	Days (0,20,40,60,80,100,120,140)	First 140 days
[31]	145 Days	The sequence length (5,10,15,20)	20% sequence length
[60]	70 Days	Days (28,42,56,70)	First 28 days
[51]	48000 Volumes	The volume of data (8000, 16000, 32000, 40000, 48000)	8000 Volumes
[67]	14 Academic years	Academic years	First two academic years
[50]	20 Minutes	Minutes (2,4,6,8,10,15,20)	First 10 minutes
[57]	3 Months	Midterm and final	The first month
[63]	12 Timesteps	Timesteps 1-12	Timestep 5

When Table 6 is assessed, it is observed that student performance is predicted earlier in the first quarter and midway through the prediction interval. This prediction interval varies from 20 minutes [50] to 14 academic years [67]. Additionally, an improvement in predicting student performance is observed as the predicted time interval increases. The best prediction results are generally obtained at the end of the prediction interval.

Figure 8 presents the highest accuracy values of studies conducting week-based early predictions using the 40-week OULAD general dataset [36], [46], [47], [62]. As the figure shows, prediction accuracy generally improves as the prediction interval progresses. For instance, models achieved accuracy rates ranging from 69% to 80% in the fifth week, which increased to 85%-97% by the 40th week. This trend provides a clear perspective on how prediction accuracy evolves, demonstrating a consistent improvement as more weeks of data become available.

4.5. RQ5 Limitations of Studies, Contributions to Literature, and Future Research Studies

4.5.1. Limitations of Studies

The limitations of studies were reported generally about datasets. These limitations have been listed as follows: the imbalanced distribution of the dataset [33], [34], [36], [39], [42], [46], [47], [48], [52], [59], [60], [62], [67], [68], the small sample size [50], [60], it does not structure [50], the dataset was limited [32], [37], [40], [64], short training period (Mao et al, 2020), the general dataset [33], [34], [36], [39], [42], [46], [47], [51], [53], [56], [59], [62], [68], considering only essential features (Yousafzai et al, 2021), insufficient enrolment [47], and same types of data [38], [45], [61], [65], [67].

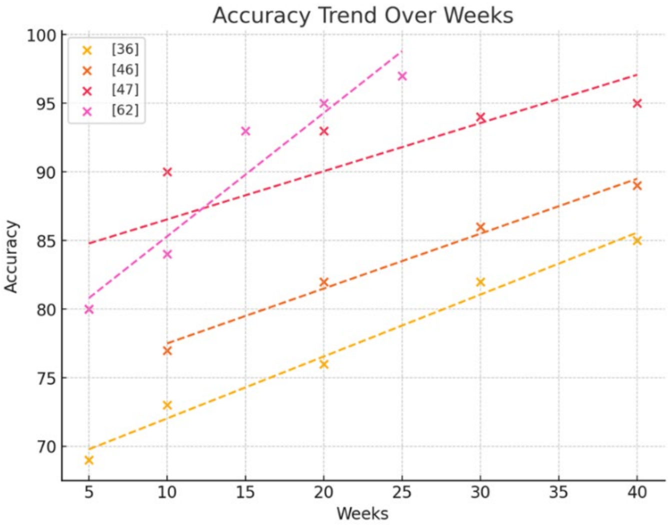


Figure 8. Accuracy Trend in 40-Week OULAD Studies

4.5.2. Contributions of Studies

When the selected studies are evaluated, they have contributed to the literature by developing new algorithms to address the following issues: examining behavioral data [44], [54], [55], [55], [56], [64], [68], predicting learning outcomes [50], [51], addressing the imbalanced distribution of the dataset [33], [34], [35], [43], [52], [60], enhancing the interpretability of prediction results [39], [43], integrating feature selection [51], [69], and tackling time series sequential classification problems [31], [36], [38], [40], [47], [53], [55], [59], [60], [62]. Additionally, the studies have addressed other significant problems, including focusing on multi-step exercises with unlimited solutions [50], [63], making predictions in a blended learning environment [57], forecasting dropouts [35], and predicting graduation [53].

4.5.3. Future Research Suggested by the Studies

The suggestions for future research from the reviewed studies were divided into two classes: data and models. Regarding the elimination of data limitations, solving the data imbalance problem [33], [36], [59], [60], [67], analyzing the data sparsity problems [59], [68], and expanding the data set to eliminate its limitations [31], [32], [36], [37], [38], [41], [43], [47], [48], [49], [52], [53], [55], [62], [64], [67], [69] have been left to future studies. Regarding the development of the models, it has been observed that the research of time-sensitive models [39], [50], the use of different models [32], [66], improving the proposed model [34], [40], [65], the use of natural language processing techniques [46], [47], [64], [68], and the dynamic estimation of the interpretability of DL models [31] have been left to future studies.

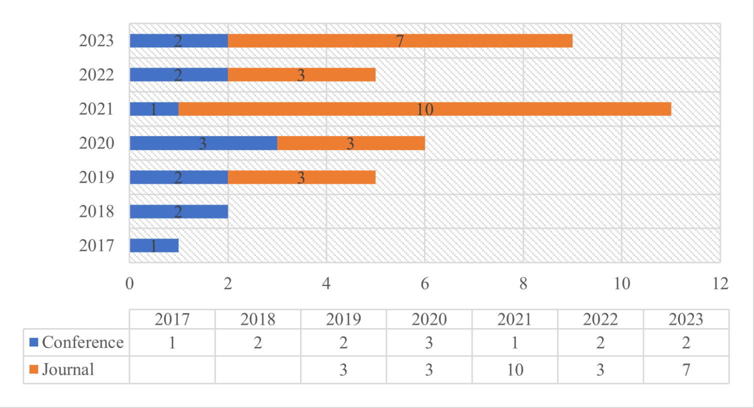
5. Comprehensive Science Mapping Analysis

This section presents bibliometric analysis results of the selected 39 articles downloaded from the Scopus database using VOSviewer. Figure 9 illustrates the periodic distribution of the total number of articles.

As shown in Figure 9, there is an increasing trend in the number of articles, with a year-over-year growth. The initial studies were only published as conference proceedings in 2017 and 2018. After 2019, the number of articles significantly increased. Moreover, there was a notable surge, particularly in 2021, where 11 works were published, including ten articles. Therefore, there is a growing interest in the subject. In this regard, more researchers are focusing on the topic.

5.1. Keyword Analysis

Table 7 displays the "Author Keywords Occurrence" and "Total Link Strength" for the top ten author keywords with a minimum keyword occurrence of 3 in the selected studies using the VOSviewer program. "Author Keywords Occurrence" indicates how often a specific keyword appears, while "Total Link Strength" represents the frequency and strength of co-occurrence between two keywords.



	2017	2018	2019	2020	2021	2022	2023
■ Conference	1	2	2	3	1	2	2
■ Journal			3	3	10	3	7

Figure 9. The Annual Number of Articles

Table 7. The Occurrence and the Total Link Strength of Author Keywords

Order	Keyword	Occurrences	Total link strength
1	deep learning	16	22
2	machine learning	12	18
3	educational data mining	8	11
4	early prediction	6	10
5	learning analytics	7	10
6	long short-term memory (LSTM)	3	8
7	virtual learning environment (VLE)	3	8
8	student performance prediction	5	6
9	deep neural networks	4	5

As shown in Table 7, deep learning is the most frequently occurring and highest total link strength keyword among the top ten keywords, based on a minimum keyword occurrence of 3. The other significant keywords are "machine learning," "educational data mining," "early prediction," and "learning analytics." The visual analysis presented by VOSviewer, shown in Figure 10, helps us understand the popularity of keywords and the connections between them. These visual analyses can assist in better understanding trends in literature and relationships between topics.

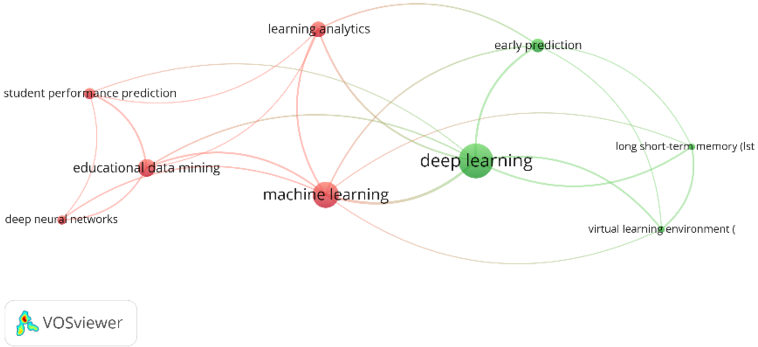


Figure 10. Network Visualization Maps of Co-Occurrence for Keywords

5.2. Co-Authorship for Country Analysis

VOSviewer's Co-authorship Analysis by Country is utilized to understand and visualize the collaboration frequencies and partnership relationships among authors operating in a specific country. Figure 11 presents the Country Collaboration Network Visualization Map provided by the VOSviewer program. As evident from Figure 11, two distinct groups are noticeable. The first group includes authors from China, the United States, and Egypt, while the second group encompasses authors from Saudi Arabia, the United Kingdom, and Pakistan. There are prominent collaboration relationships among authors from countries within both groups.

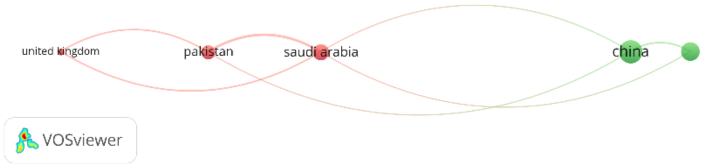


Figure 11. Network Visualization Maps of Co-Authorship for Country

5.3. Citation for Country Analysis

VOSviewer's Citation for Country Analysis features maps and visualizes the citation relationships of scientific publications produced in a specific country. This analysis helps understand the level of interaction of research outputs in the country, relationships with other countries, and international citation networks. These visualizations can give researchers important insights into understanding trends and networks in scientific knowledge production, identifying potential collaborations, and observing interdisciplinary interactions. Figure 12 presents the Network Visualization Map for country-based citation analysis provided by the VOSviewer program. As seen in Figure 12, two distinct groups are notable, similar to the analysis conducted by authors. The first group includes China, the United States, and India, while the second group encompasses Saudi Arabia, the United Kingdom, Indonesia, and Pakistan. There are evident citation network relationships among countries within both groups.

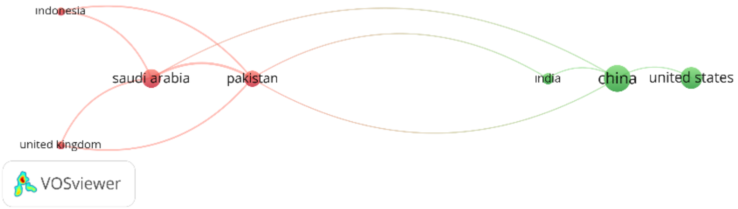


Figure 12. Network Visualization Maps of a Citation for The Country

5.4. Bibliographic Coupling for Sources Analysis

The Bibliographic Coupling for Sources Analysis in VOSviewer analyzes and visualizes connections between scientific sources. This analysis evaluates the similarity of sources in scientific articles and identifies strong relationships among these sources. Bibliographic coupling is based on two separate articles having the same reference. In other words, the connection between two articles is based on referencing the same sources. This analysis is used to identify scientific sources that work on similar topics or focus on similar research subjects and understand the connections between these sources. VOSviewer presents these bibliographic connections by creating network maps and visualizing relationships between sources. This visual analysis can help researchers understand important sources within a specific topic or discipline and the intense interactions between these sources. Figure 13 presents the network map of bibliographic connections for sources VOSviewer provides. As shown in Figure 13, four source groups are highlighted with red, blue, green, purple, and yellow lines. The purple lines, indicating IEEE Access and other blue sources, represent the most robust connections.

5.5. Sources Analysis

The source analysis was conducted based on the number of publications and the average citation count for each source, as illustrated in Figure 14. In terms of average citation count, "Computers in Human Behavior" has the highest average citation count. Regarding the number of publications, "IEEE Access" has the highest published articles.

6. Conclusion

This literature review offers an in-depth examination of the current advancements in deep learning (DL) methodologies applied to early predicting student performance in Educational Data Mining (EDM). The study systematically addressed five central research questions, delving into the distribution of existing research, the types of datasets and attributes utilized, the DL models proposed, the timing of early predictions, and the challenges and future directions highlighted in prior studies.

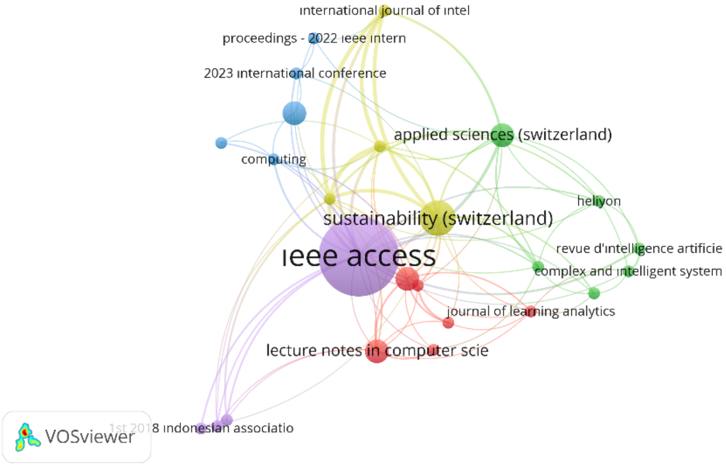

international journal of intel
proceedings - 2022 ieee intern
2023 international conference
applied sciences (switzerland)
computing
heliyon
sustainability (switzerland)
ieee access
revue d'intelligence artificie
complex and intelligent system
journal of learning analytics
lecture notes in computer scie
1st 2018 indonesian associatio
VOSviewer

Figure 13. Network Visualization Maps of Bibliographic Coupling for Sources

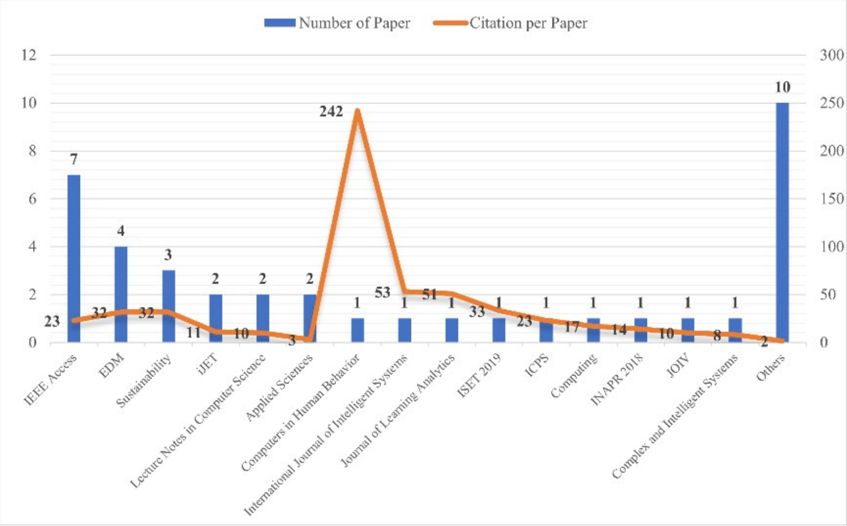


Figure 14. The Number of Publications and The Average Citation Count for Each Source

The results demonstrate that DL approaches, especially Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), outperform traditional machine learning (ML) techniques in managing intricate and voluminous educational datasets. Hybrid DL models have also gained traction as a viable solution, delivering enhanced accuracy and reliability in forecasting student outcomes. Nevertheless, persistent issues such as dataset imbalances, restricted sample sizes, and the demand for greater transparency continue to hinder the broader implementation of these methods.

The review emphasizes the critical role of early prediction in detecting at-risk students and enabling timely support mechanisms. The evaluation of prediction timelines reveals that while early forecasting is achievable, prediction accuracy increases as additional data is accumulated. This finding highlights the necessity for ongoing monitoring and adaptive predictive systems capable of adjusting to the evolving nature of student learning processes.

Future studies should prioritize overcoming the limitations outlined in this review, particularly concerning dataset quality and the interpretability of models. Broadening dataset diversity to include more representative samples and incorporating natural language processing (NLP) techniques could significantly improve the predictive power of DL models. Furthermore, creating time-sensitive models and investigating dynamic feature selection approaches present promising directions for further exploration.

In summary, this review underscores the transformative potential of DL in EDM for early student performance prediction. Researchers and educators can create more effective tools to improve student outcomes by addressing current challenges and leveraging emerging opportunities. The increasing research interest in this field indicates a promising future for DL in reshaping educational support and understanding.

References

[1] S. Keskin, F. Aydın, and H. Yurdugül, 'Eğitsel Veri Madenciliği ve Öğrenme Analitikleri Bağlamında E-Öğrenme Verilerinde Aykırı Gözlemlerin Belirlenmesi', Eğitim Teknolojisi Kuram ve Uygulama, vol. 9, no. 1, pp. 292–309, Jan. 2019, doi: 10.17943/etku.475149.

[2] K. Akgün and M. Bulut Özek, 'Eğitsel Veri Madenciliği Yöntemi İle İlgili Yapılmış Çalışmaların İncelenmesi: İçerik Analizi', Uluslararası Eğitim Bilim ve Teknoloji Dergisi, vol. 6, no. 3, pp. 197–213, Dec. 2020, doi: 10.47714/uebt.753526.

[3] J. Berens, K. Schneider, S. Görtz, S. Oster, and J. Burghoff, 'Early Detection of Students at Risk – Predicting Student Dropouts Using Administrative Student Data from German Universities and Machine Learning Methods,' Journal of Educational Data Mining, vol. 11, no. 3, p. 41, 2019.

[4] L. C. Yu et al., 'Improving early prediction of academic failure using sentiment analysis on self-evaluated comments', Journal of Computer Assisted Learning, vol. 34, no. 4, pp. 358–365, Aug. 2018, doi: 10.1111/jcal.12247.

[5] C. Romero and S. Ventura, 'Guest Editorial: Special Issue on Early Prediction and Supporting of Learning Performance', IEEE Transactions on Learning Technologies, vol. 12, no. 2, pp. 145–147, Apr. 2019, doi: 10.1109/TLT.2019.2908106.

[6] B. Albreiki, N. Zaki, and H. Alashwal, 'A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques', Education Sciences, vol. 11, no. 9, p. 552, Sep. 2021, doi: 10.3390/educsci11090552.

[7] P. Chakrapani and C. D, 'Academic Performance Prediction Using Machine Learning: A Comprehensive & Systematic Review', in 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India: IEEE, Apr. 2022, pp. 335–340. doi: 10.1109/ICESIC53714.2022.9783512.

[8] K. Alalawi, R. Athauda, and R. Chiong, 'Contextualizing the current state of research on the use of machine learning for student performance prediction: A systematic literature review', Engineering Reports, vol. 5, no. 12, Dec. 2023, doi: 10.1002/eng2.12699.

[9] A. Nabil, M. Seyam, and A. A. Elfetouh, 'Predicting students' academic performance using machine learning techniques: a literature review', International Journal of Business Intelligence and Data Mining, vol. 20, no. 4, p. 456, 2022, doi: 10.1504/IJBIDM.2022.123214.

[10] N. Iam-On and T. Boongoen, 'Generating descriptive model for student dropout: a review of clustering approach', Human-centric Computing and Information Sciences, vol. 7, no. 1, p. 1, Dec. 2017, doi: 10.1186/s13673-016-0083-0.

[11] Y. K. Hui and L. F. Kwok, 'A review on learning analytics', International Journal of Innovation and Learning, vol. 25, no. 2, p. 197, 2019, doi: 10.1504/IJIL.2019.097673.

[12] A. Namoun and A. Alshanqiti, 'Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review', Applied Sciences, vol. 11, no. 1, p. 237, Dec. 2020, doi: 10.3390/app11010237.

[13] M. Ingle, 'A Review On Research Areas In Educational Data Mining And Learning Analytics', INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 8, 2019, [Online]. Available: www.ijstr.org

[14] A. M. Shahiri, 'A Review on Predicting Student's Performance Using Data Mining Techniques', Procedia Computer Science, p. 9, 2015, doi: 10.1016/j.procs.2015.12.157.

[15] R. Ordoñez-Avila, N. S. Reyes, J. Meza, and S. Ventura, 'Data mining techniques for predicting teacher evaluation in higher education: A systematic literature review', Heliyon, vol. 9, no. 3, p. e13939, Mar. 2023, doi: 10.1016/j.heliyon.2023.e13939.

[16] P. S. Pawar and R. Jain, 'A review on Student Performance Prediction using Educational Data mining and Artificial Intelligence', in 2021 IEEE 2nd International Conference on Technology, Engineering, Management for Societal impact using Marketing, Entrepreneurship and Talent (TEMSMET), Pune, India: IEEE, Dec. 2021, pp. 1–7. doi: 10.1109/TEMSMET53515.2021.9768773.

[17] W. Xiao, P. Ji, and J. Hu, 'A survey on educational data mining methods for predicting students' performance', Engineering Reports, vol. 4, no. 5, May 2022, doi: 10.1002/eng2.12482.

[18] H. Nawang, M. Makhtar, and W. M. A. F. W. Hamza, 'A systematic literature review on student performance predictions', International Journal of Advanced Technology and Engineering Exploration, vol. 8, no. 84, Nov. 2021, doi: 10.19101/IJATEE.2021.874521.

[19] K. Aulakh, R. K. Roul, and M. Kaushal, 'E-learning enhancement through educational data mining with Covid-19 outbreak period in backdrop: A review', International Journal of Educational Development, vol. 101, p. 102814, Sep. 2023, doi: 10.1016/j.ijedudev.2023.102814.

[20] M. Saqr, R. Elmoazen, M. Tedre, S. López-Pernas, and L. Hirsto, 'How well centrality measures capture student

achievement in computer-supported collaborative learning? – A systematic review and meta-analysis', Educational Research Review, vol. 35, p. 100437, Feb. 2022, doi: 10.1016/j.edurev.2022.100437.

[21] D. A. Shafiq, M. Marjani, R. A. A. Habeeb, and D. Asirvatham, 'Student Retention Using Educational Data Mining and Predictive Analytics: A Systematic Literature Review', IEEE Access, vol. 10, pp. 72480–72503, 2022, doi: 10.1109/ACCESS.2022.3188767.

[22] M. Zaffar, M. A. Hashmani, K. S. Savita, and S. A. Khan, 'A review on feature selection methods for improving classification performance in educational data mining', International Journal of Information Technology and Management, vol. 20, no. 1/2, p. 110, 2021, doi: 10.1504/IJITM.2021.114161.

[23] A. Abu Saa, M. Al-Emran, and K. Shaalan, 'Factors Affecting Students' Performance in Higher Education: A Systematic Review of Predictive Data Mining Techniques', Tech Know Learn, vol. 24, no. 4, pp. 567–598, Dec. 2019, doi: 10.1007/s10758-019-09408-7.

[24] E. Alyahyan and D. Düştegör, 'Predicting academic success in higher education: literature review and best practices', Int J Educ Technol High Educ, vol. 17, no. 1, p. 3, Dec. 2020, doi: 10.1186/s41239-020-0177-7.

[25] J. López-Zambrano, J. A. Lara Torralbo, and C. Romero, 'Early Prediction of Student Learning Performance Through Data Mining: A Systematic Review', Psicothema, no. 33.3, pp. 456–465, Ağustos 2021, doi: 10.7334/psicothema2021.62.

[26] H. Q. Alatawi and S. Hechmi, 'A Survey of Data Mining Methods for Early Prediction of Students' Performance', in 2022 2nd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia: IEEE, Jan. 2022, pp. 171–174. doi: 10.1109/ICCIT52419.2022.9711642.

[27] S. M. Muthukrishnan, M. K. Govindasamy, and M. N. Mustapha, 'Systematic mapping review on student's performance analysis using big data predictive model', Journal of Fundamental and Applied Sciences, vol. 9, no. 4S, p. 730, Jan. 2018, doi: 10.4314/jfas.v9i4S.41.

[28] X. Bai et al., 'Educational Big Data: Predictions, Applications and Challenges', Big Data Research, vol. 26, p. 100270, Nov. 2021, doi: 10.1016/j.bdr.2021.100270.

[29] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, 'A Systematic Review of Deep Learning Approaches to Educational Data Mining', Complexity, vol. 2019, pp. 1–22, May 2019, doi: 10.1155/2019/1306039.

[30] Kitchenham Barbara and Charters Stuart M., 'Guidelines for performing systematic literature reviews in software engineering', School of Computer Science and Mathematics, Keele University., Durham, UK, EBSE Technical Report EBSE-2007-01, 2007.

[31] X. Li, Y. Zhang, H. Cheng, M. Li, and B. Yin, 'Student achievement prediction using deep neural network from multi-source campus data', Complex Intell. Syst., vol. 8, no. 6, pp. 5143–5156, Dec. 2022, doi: 10.1007/s40747-022-00731-8.

[32] B. Venkatachalam and K. Sivanraju, 'Predicting Student Performance Using Mental Health and Linguistic Attributes with Deep Learning', Revue d'Intelligence Artificielle, vol. 37, no. 4, pp. 889–899, Aug. 2023, doi: 10.18280/ria.370408.

[33] A. A. Almahdi and B. T. Sharef, 'Deep Learning Based An Optimized Predictive Academic Performance Approach', in 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD), Manama, Bahrain: IEEE, Mar. 2023, pp. 1–6. doi: 10.1109/ITIKD56332.2023.10099652.

[34] X. Wen and H. Juan, 'Early Prediction of Students' Performance Using a Deep Neural Network Based on Online Learning Activity Sequence', Applied Sciences, vol. 13, no. 15, p. 8933, Aug. 2023, doi: 10.3390/app13158933.

[35] C. A. Anjali and V. R. Bai, 'An Early Prediction of Dropouts for At-risk Scholars in MOOCs using Deep Learning', in 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), IEEE, Jun. 2022, pp. 1–6. doi: 10.1109/IC3SIS54991.2022.9885328.

[36] H. Waheed, S.-U. Hassan, R. Nawaz, N. R. Aljohani, G. Chen, and D. Gasevic, 'Early prediction of learners at risk in self-paced education: A neural network approach', Expert Systems with Applications, vol. 213, p. 118868, Mar. 2023, doi: 10.1016/j.eswa.2022.118868.

[37] B. T. Sayed, M. Madanan, and N. Biju, 'An Efficient Artificial Intelligence-Based Educational Data Mining Approach for Higher Education and Early Recognition System', SN Computer Science, vol. 4, no. 2, p. 130, Dec. 2022, doi: 10.1007/s42979-022-01562-7.

[38] S. Leelaluk, T. Minematsu, Y. Taniguchi, F. Okubo, T. Yamashita, and A. Shimada, 'Scaled-Dot Product Attention for Early Detection of At-risk Students', in 2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Hung Hom, Hong Kong: IEEE, Dec. 2022, pp. 316–322. doi: 10.1109/TALE54877.2022.00059.

[39] F. A. Al-azazi and M. Ghurab, 'ANN-LSTM: A deep learning model for early student performance prediction in MOOC', Heliyon, vol. 9, no. 4, p. e15382, Apr. 2023, doi: 10.1016/j.heliyon.2023.e15382.

[40] H. Wan, M. Li, Z. Zhong, and X. Luo, 'Early Prediction of Student Performance with LSTM-Based Deep Neural Network', in 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, Jun. 2023, pp. 132–141. doi: 10.1109/COMPSAC57700.2023.00026.

[41] K. Qin, X. Xie, Q. He, and G. Deng, 'Early Warning of Student Performance With Integration of Subjective and Objective Elements', IEEE Access, vol. 11, pp. 72601–72617, 2023, doi: 10.1109/ACCESS.2023.3295580.

[42] S. S. Kusumawardani and S. A. I. Alfarozi, 'Transformer Encoder Model for Sequential Prediction of Student

Performance Based on Their Log Activities', IEEE Access, vol. 11, pp. 18960–18971, 2023, doi: 10.1109/ACCESS.2023.3246122.

[43] H.-C. Chen et al., 'Week-Wise Student Performance Early Prediction in Virtual Learning Environment Using a Deep Explainable Artificial Intelligence', Applied Sciences, vol. 12, no. 4, p. 1885, Feb. 2022, doi: 10.3390/app12041885.

[44] F. D. Pereira et al., 'Explaining Individual and Collective Programming Students' Behavior by Interpreting a Black-Box Predictive Model', IEEE Access, vol. 9, pp. 117097–117119, 2021, doi: 10.1109/ACCESS.2021.3105956.

[45] E. Tanuar, Y. Heryadi, Lukas, B. S. Abbas, and F. L. Gaol, 'Using Machine Learning Techniques to Earlier Predict Student's Performance', in 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), Jakarta, Indonesia: IEEE, Sep. 2018, pp. 85–89. doi: 10.1109/INAPR.2018.8626856.

[46] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, 'Predicting academic performance of students from VLE big data using deep learning models', Computers in Human Behavior, vol. 104, p. 106189, Mar. 2020, doi: 10.1016/j.chb.2019.106189.

[47] N. R. Aljohani, A. Fayoumi, and S.-U. Hassan, 'Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment', Sustainability, vol. 11, no. 24, p. 7238, Dec. 2019, doi: 10.3390/su11247238.

[48] C.-C. Yu and Y. (Leon) Wu, 'Early Warning System for Online STEM Learning—A Slimmer Approach Using Recurrent Neural Networks', Sustainability, vol. 13, no. 22, p. 12461, Nov. 2021, doi: 10.3390/su132212461.

[49] M. Y. S. L. Z. Aljuid, 'Deep Learning Based Method For Prediction of Software Engineering Project Teamwork Assessment in Higher Education', Journal of Theoretical and Aplied Information Technology, vol. 99, no. 9, 2021, [Online]. Available: www.jatit.org

[50] Y. Mao, S. Marwan, and T. W. Price, 'What Time is It? Student Modeling Needs to Know', p. 12, 2020.

[51] H. Mi, Z. Gao, Q. Zhang, and Y. Zheng, 'Research on Constructing Online Learning Performance Prediction Model Combining Feature Selection and Neural Network', Int. J. Emerg. Technol. Learn., vol. 17, no. 07, pp. 94–111, Apr. 2022, doi: 10.3991/ijet.v17i07.25587.

[52] N. M. Aslam, I. U. Khan, L. H. Alamri, and R. S. Almuslim, 'An Improved Early Student's Academic Performance Prediction Using Deep Learning', International Journal of Emerging Technologies in Learning (iJET), vol. 16, no. 12, p. 108, Jun. 2021, doi: 10.3991/ijet.v16i12.20699.

[53] B.-H. Kim, E. Vizitei, and V. Ganapathi, 'GritNet: Student Performance Prediction with Deep Learning'. arXiv, Apr. 19, 2018. Accessed: Sep. 08, 2022. [Online]. Available: http://arxiv.org/abs/1804.07405

[54] X. Wang, P. Wu, G. Liu, Q. Huang, X. Hu, and H. Xu, 'Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments', Computing, vol. 101, no. 6, pp. 587–604, Jun. 2019, doi: 10.1007/s00607-018-00699-9.

[55] X. Li, X. Zhu, X. Zhu, Y. Ji, and X. Tang, 'Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network', in Advances in Knowledge Discovery and Data Mining, vol. 12084, H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, Eds., in Lecture Notes in Computer Science, vol. 12084. , Cham: Springer International Publishing, 2020, pp. 567–579. doi: 10.1007/978-3-030-47426-3_44.

[56] O. Ojajuni et al., 'Predicting Student Academic Performance Using Machine Learning', in Computational Science and Its Applications – ICCSA 2021, vol. 12957, O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B. O. Apduhan, A. M. A. C. Rocha, E. Tarantino, and C. M. Torre, Eds., in Lecture Notes in Computer Science, vol. 12957. , Cham: Springer International Publishing, 2021, pp. 481–491. doi: 10.1007/978-3-030-87013-3_36.

[57] R. C. Raga and J. D. Raga, 'Early Prediction of Student Performance in Blended Learning Courses Using Deep Neural Networks', in 2019 International Symposium on Educational Technology (ISET), Hradec Kralove, Czech Republic: IEEE, Jul. 2019, pp. 39–43. doi: 10.1109/ISET.2019.00018.

[58] S. Surenthiran, R. Rajalakshmi, and S. S. Sujatha, 'Student Performance Prediction Using Atom Search Optimization Based Deep Belief Neural Network', Optical Memory and Neural Networks, vol. 30, no. 2, pp. 157–171, Apr. 2021, doi: 10.3103/S1060992X21020119.

[59] H. Karimi, T. Derr, J. Huang, and J. Tang, 'Online Academic Course Performance Prediction using Relational Graph Convolutional Neural Network', in The International Conference on Educational Data Mining (EDM), 2020, pp. 7–7. [Online]. Available: https://github.com/hamidkarimi/dope.

[60] F. Chen and Y. Cui, 'Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance', JLA, vol. 7, no. 2, pp. 1–17, Sep. 2020, doi: 10.18608/jla.2020.72.1.

[61] Q. Hu and H. Rangwala, 'Reliable Deep Grade Prediction with Uncertainty Estimation', in Proceedings of the 9th International Conference on Learning Analytics & Knowledge, ACM, Mar. 2019, pp. 76–85. doi: 10.1145/3303772.3303802.

[62] S. Hassan, H. Waheed, N. R. Aljohani, M. Ali, S. Ventura, and F. Herrera, 'Virtual learning environment to predict withdrawal by leveraging deep learning', Int J Intell Syst, vol. 34, no. 8, pp. 1935–1952, Aug. 2019, doi: 10.1002/int.22129.

[63] L. Wang, A. Sy, L. Liu, and C. Piech, 'Learning to Represent Student Knowledge on Programming Exercises Using Deep Learning', in The International Conference on Educational Data Mining (EDM), 2017, pp. 6–6.

[64] X. Du, J. Yang, J.-L. Hung, and B. Shelton, 'Educational data mining: a systematic review of research and emerging trends', Information Discovery and Delivery, vol. 48, no. 4, pp. 225–236, May 2020, doi: 10.1108/IDD-09-2019-0070.

[65] D. Uliyan, A. S. Aljaloud, A. Alkhalil, H. S. A. Amer, M. A. E. A. Mohamed, and A. F. M. Alogali, 'Deep Learning

Model to Predict Students Retention Using BLSTM and CRF', IEEE Access, vol. 9, pp. 135550–135558, 2021, doi: 10.1109/ACCESS.2021.3117117.

[66] H. S. Park and S. J. Yoo, 'Early Dropout Prediction in Online Learning of University using Machine Learning', JOIV : International Journal on Informatics Visualization, vol. 5, no. 4, p. 347, Dec. 2021, doi: 10.30630/joiv.5.4.732.

[67] A. Nabil, M. Seyam, and A. Abou-Elfetouh, 'Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks', IEEE Access, vol. 9, pp. 140731–140746, 2021, doi: 10.1109/ACCESS.2021.3119596.

[68] M. Adnan et al., 'Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models', IEEE Access, vol. 9, pp. 7519–7539, 2021, doi: 10.1109/ACCESS.2021.3049446.

[69] B. K. Yousafzai et al., 'Student-Performulator: Student Academic Performance Using Hybrid Deep Neural Network', Sustainability, vol. 13, no. 17, p. 9775, Aug. 2021, doi: 10.3390/su13179775

Authors Contributions
Authors contributed equally to the study.

Conflict of Interest Notice
There is no conflict of interest to declare

Ethical Approval and Informed Consent
The ethics committee approval was not required as the study.

Plagiarism Statement
This article has been scanned by Ithenticate™.