ISSN 2723-6471

Predicting Network Performance Degradation in Wireless and Ethernet Connections Using Gradient Boosting, Logistic Regression, and Multi-Layer Perceptron Models

Chyntia Raras Ajeng Widiawati[1], Sarmini[2,*], Dwi Yuliana[3]

[1]*Information Technology, Computer Science Faculty, Universitas Amikom Purwokerto, Indonesia*

[2,3]*Information System, Computer Science Faculty, Universitas Amikom Purwokerto, Indonesia*

(Received: August 13, 2024; Revised: October 6, 2024; Accepted: November 20, 2024; Available online: December 29, 2024)

Abstract

This study explores predicting network performance degradation in wireless and Ethernet connections using three machine learning algorithms: XGBoost, Logistic Regression, and Multi-Layer Perceptron (MLP). Key metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, were employed to evaluate model performance. The MLP classifier achieved the highest accuracy (98.7%) and AUC-ROC (0.9998), with a precision of 1.0000 and recall of 0.8622, resulting in an F1-score of 0.9260. Logistic Regression provided reasonable baseline performance, with an accuracy of 93.67%, AUC-ROC of 0.9565, and an F1-score of 0.5992, but struggled with non-linear dependencies. XGBoost showed limited utility in detecting degradation events, achieving an F1-score of 0 despite a perfect AUC-ROC (1.0), indicating sensitivity to imbalanced data. Through hyperparameter tuning, MLP demonstrated robustness in capturing complex patterns in network latency metrics (local_avg and remote_avg), with remote_avg emerging as the most predictive feature for identifying degradation across both network types. Visualizations of latency dynamics demonstrate the higher predictive relevance of remote latency (remote_avg) in both network types, where spikes in this metric are closely associated with degradation. The findings underscore the effectiveness of using latency metrics and machine learning to anticipate network issues, suggesting that MLP is particularly well-suited for real-time, predictive network monitoring. Integrating such models could enhance network reliability by enabling proactive intervention, crucial for sectors reliant on continuous connectivity. Future work could expand on feature sets, explore adaptive thresholding, and implement these predictive models in live network environments for real-time monitoring and automated response.

Keywords: Network Performance Degradation Prediction, Machine Learning In Network Monitoring, Wireless and Ethernet Latency Analysis, Predictive Network Maintenance, Multi-Layer Perceptron

1. Introduction

Modern organizations rely on network infrastructure for daily operations, communication, and seamless data exchange, making reliability a core focus. Disruptions can cause service outages and affect both internal and customer-facing processes. Research [1] highlights that network reliability is crucial to prevent cascading failures that disrupt business continuity and cause financial losses. The rise of digital platforms has increased the need for robust infrastructure. Companies like Google have invested in fiber optics to improve connectivity in underserved areas, boosting operational efficiency and engagement [2]. Network management is vital for secure communication, sustaining relationships, and ensuring organizational integrity [3]. The shift to cloud computing emphasizes reliable networks to maintain service levels and customer satisfaction [4]. Network performance degradation remains a challenge, with common factors like high latency, packet loss, and bandwidth issues.

High latency impacts application responsiveness and is often caused by congestion, routing inefficiencies, or long distances, especially in WANs. Studies [5], [6] note that traditional synchronization algorithms can worsen latency issues, particularly in high-latency environments, while [7] emphasizes that heavy IoT traffic leads to delays, complicating real-time data processing. Server downtime poses a significant challenge to maintaining network

***Corresponding author: Sarmini (sarmini@amikompurwokerto.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i1.519

performance, often stemming from hardware failures, software bugs, or planned maintenance. In cloud environments, downtime can disrupt virtual services and impact SLAs, with research [8] noting that server reboots to fix anomalies can affect service accessibility. Inefficient VM migration, intended to balance load, may also introduce downtime if not managed properly [9], [10].

Effective planning is critical to minimize such interruptions. Addressing latency, packet loss, bandwidth issues, and downtime requires strategic network management and a deep understanding of network dynamics across diverse scenarios and infrastructure setups. Managing wireless and Ethernet connections adds complexity due to their distinct characteristics. Ethernet generally offers stable, low-latency performance due to its wired nature, minimizing susceptibility to interference and maintaining consistent performance [11]. Conversely, wireless connections are flexible but prone to environmental factors, such as signal attenuation and interference from other devices, which can lead to higher latency and performance variability [12]. Physical barriers and external interference exacerbate these fluctuations, making wireless networks inherently less stable than Ethernet [13]. Complex wireless routing paths further contribute to higher latency due to signal propagation delays. Unpredictable network issues can disrupt operations, causing financial losses, reduced productivity, and reputational damage. With mobile devices and cloud services deeply integrated, quickly identifying and resolving network anomalies is crucial [14]. Issues like high latency, packet loss, and security breaches can lead to service outages, impacting business continuity and customer satisfaction [15]. Failing to address latency spikes or packet loss promptly can result in delays and user frustration [16]. The increasing sophistication of cyber threats, such as DDoS attacks, underscores the need for proactive detection [16], [17]. Machine learning (ML) and deep learning (DL) enable real-time anomaly detection, allowing operators to take preventative action [18]. Combined with software-defined networking (SDN), real-time monitoring dynamically adjusts network settings, enhancing resilience to network issues [17], [19].

The primary goal of this research is to develop predictive models that accurately forecast network performance degradation in wireless and Ethernet connections. This study employs three ML algorithms—XGBoost, Logistic Regression, and Multi-Layer Perceptron (MLP)—to predict degradation events based on historical network data. Each algorithm has unique strengths in handling classification tasks, and this research seeks to evaluate their performance in predicting network issues, such as latency spikes and connectivity drops. Through a comparative analysis, the study aims to determine which of these algorithms provides the most reliable and accurate predictions, thereby offering valuable insights into their applicability in network management contexts. This research utilizes two distinct datasets, one for wireless network data and another for Ethernet network data, to capture each connection type's unique behaviors and performance characteristics. Key variables include timestamp, location, source, local_avg, and remote_avg, representing various network activity aspects. The target variable, network performance degradation, is defined based on these indicators, allowing the models to identify patterns associated with network issues. To assess the effectiveness of each predictive model, the study involves training and testing on both datasets, followed by a comparative analysis of accuracy, precision, recall, and other relevant metrics. The findings contribute to the field by exploring how different supervised learning models predict degradation events across wireless and Ethernet environments, providing insights that can enhance proactive network management strategies.

2. Literature Review

2.1. Network Performance Degradation

Network performance degradation can be attributed to several common causes, including network congestion, hardware limitations, interference (particularly in wireless networks), and cable damage in Ethernet connections. These factors are critical in affecting network efficiency and reliability, leading to disruptions in service quality and user experience. Network Congestion is one of the most frequent causes, occurring when the demand for bandwidth surpasses the network’s capacity. Congestion leads to packet loss, increased latency, and, consequently, a decline in network throughput. For example, Golgiri and Javidan explain that congestion can result in wasted energy through packet retransmissions, which impacts energy efficiency and service quality [20].

Network degradation often manifests through various indicators, including high latency, increased response times, frequent disconnections, and packet loss. These indicators provide valuable insights into network health and are

essential for diagnosing performance issues. High Latency is a common symptom of network degradation, where delays in data transmission result from congestion, inefficient routing, or long physical distances between nodes. Yang et al. describe how high latency can impede synchronization algorithms in wireless sensor networks, leading to errors and inefficiencies [6]. In the context of high-performance applications, Geng et al. note that even slight increases in latency can severely impact operations, particularly in latency-sensitive environments such as deep learning networks [21]. Increased Response Times are closely related to latency and indicate underlying network issues. Raju and Manjunath propose a method to measure response times, suggesting that delayed responses often correlate with packet loss and can reflect broader network performance challenges [22].

To effectively explain network performance degradation and its effects, mathematical expressions can provide clarity on key metrics such as packet loss, which are critical indicators of network health. Packet loss is a critical indicator of network congestion and reliability. It is calculated as the ratio of lost packets to the total transmitted packets:

$$P_{loss} = \frac{P_{sent} - P_{received}}{P_{sent}} \times 100\% \quad (1)$$

High packet loss can signal serious network issues and typically results in increased latency and decreased throughput. Together, these equations encapsulate the key aspects of network performance, helping quantify the degradation indicators discussed in this section.

2.2. Machine Learning in Network Monitoring

ML applications in network monitoring and analysis have significantly advanced areas such as anomaly detection, traffic classification, and fault prediction, providing new avenues for managing and optimizing network performance. In the field of anomaly detection, ML techniques have proven effective in identifying unusual patterns in network traffic that could indicate security threats or performance issues. Jadidi et al. explored using ML algorithms to analyze logs from Industrial Control Systems (ICS), which can reflect system behaviors and help detect anomalies indicative of potential threats [23]. In another study, Sokolov et al. applied classical ML algorithms, including K-means and Naive Bayes, to detect anomalies in industrial settings, demonstrating that these techniques are both scalable and efficient in real-world applications [24]. Moreover, Nusrat emphasized the need for ML-based anomaly detection in Internet of Things (IoT) networks, where the vast number of connected devices generates complex and voluminous data, making traditional monitoring approaches less effective [25].

Kernel-based methods are another ML approach that has proven effective in modeling non-linear patterns. Montesinos-López et al. discussed how kernel methods, integrated with ML algorithms like Support Vector Machines (SVM), can accurately model complex, non-linear data distributions, making them suitable for large-scale network data analysis [26]. Kernel methods enable the transformation of data into higher-dimensional spaces, capturing intricate relationships that traditional methods may overlook. López et al. also demonstrated that sparse kernel methods enhance computational efficiency while maintaining the ability to detect non-linear relationships, which is particularly useful in high-dimensional network data [27]. DL models have further enhanced ML's capability to manage non-linear data patterns in network monitoring. Konanur et al. explored the use of CNNs and RNNs for processing non-linear data in network analysis, noting that these architectures excel in handling the hierarchical and temporal aspects of complex network traffic [28]. These deep learning architectures can learn hierarchical data representations, which is particularly advantageous for analyzing the multi-dimensional nature of network traffic data. The use of LSTM-based deep learning techniques in this intrusion detection system (IDS) proves highly effective in handling dynamic, non-stationary network data, enabling the system to accurately detect evolving patterns of network behavior and malicious activities in real-time [29]. This adaptability ensures that ML techniques can efficiently address the unique challenges presented by non-linear network data.

2.3. Related Work on Predictive Models

Gradient Boosting algorithms, especially eXtreme Gradient Boosting (XGBoost), have become popular for predictive modeling in various domains thanks to their high accuracy and ability to handle complex datasets. XGBoost is a scalable ML system that has been optimized for speed and performance, making it particularly well-suited for large-scale data applications. Numerous studies have demonstrated the superior predictive capabilities of XGBoost, positioning it as a leading choice for tasks requiring both precision and computational efficiency. One of the primary

advantages of XGBoost is its accuracy. For example, Jiang et al. reported that XGBoost achieved an impressive area under the curve (AUC) of 84.8% in predicting overall survival for patients with renal cell carcinoma, outperforming other ML models in their study [30].

Logistic regression has become a valuable tool for binary classification tasks in network monitoring, particularly in applications like anomaly detection and equipment failure prediction. Its simplicity and interpretability make it a preferred choice in scenarios requiring clear, actionable insights from model outputs. Numerous studies have applied logistic regression to network-related challenges, highlighting its effectiveness in identifying system vulnerabilities. For example, Muideen et al. developed a logistic regression classifier for predicting failures in air pressure systems, demonstrating its utility in real-time applications [31]. Similarly, Huang et al. used logistic regression within wireless sensor networks to assess reliability, showcasing its capability to model relationships between predictors and the likelihood of system failures [32].

MLP have become essential tools in predictive modeling for network performance, owing to their capability to capture non-linear dependencies and complex feature interactions. MLP, as a form of feedforward neural network, contain multiple layers that allow for deep representation learning. This layered structure is particularly effective in network environments where relationships between performance metrics and other variables are often non-linear. For instance, [33] emphasizes that even a single hidden layer in an MLP can approximate complex non-linear functions, highlighting the model's flexibility and its adaptability to diverse data structures. This attribute makes MLP well-suited for analyzing complex network behaviors, which are often characterized by intricate interactions that linear models cannot adequately capture.

3. Methodology

The flowchart in figure 1 outlines the research methodology, beginning with Data Collection and Preparation to gather and clean datasets, followed by Data Preprocessing and Labeling to remove outliers, engineer features, and label degradation events. Next, Model Selection and Training involves tuning and training XGBoost, Logistic Regression, and MLP models. In Model Evaluation, these models are assessed using metrics like accuracy, precision, recall, F1-score, and AUC-ROC. Finally, Visualization and Comparison provides insights through ROC curves, feature importance plots, and confusion matrices, enabling a clear comparison of model performance.

Figure 1. Research Method Flowchart

3.1. Data Description

This study utilizes two distinct datasets representing wireless and Ethernet network performance, sourced from Kaggle, providing key features for analyzing network behavior and identifying potential degradation. The use of Kaggle datasets ensures a diverse and well-documented collection of network data, enhancing the study's applicability to real-world network scenarios. Both datasets include essential columns, such as timestamp, location, source, local_avg, and remote_avg. These features enable a comprehensive examination of network performance over time and across different geographical or network segments, facilitating a robust analysis of potential degradation patterns. The timestamp feature marks the exact moment each measurement is recorded, aiding in the identification of temporal trends and potential cyclical patterns in network behavior. The location and source columns specify the geographical and network origins of the data, offering insight into how performance variations may be influenced by physical or network-related factors. The local_avg and remote_avg columns, representing average latency values recorded at both local and remote points within the network, serve as primary indicators of network performance by capturing round-trip times within different segments of the network infrastructure. Significant deviations in these latency metrics often indicate potential performance bottlenecks or issues, reflecting how efficiently the network handles data transfer. Latency in both wireless and Ethernet networks can be affected by factors such as interference, environmental conditions, and congestion, which vary between the two network types. The decision to focus on latency-related

features stems from their direct impact on user experience and network reliability, making them critical for degradation analysis. However, it is acknowledged that other potentially impactful features, such as bandwidth utilization or protocol types, were not explored. Future evaluations may consider these additional factors to provide a broader context and enhance predictive performance. The study's target variable is a performance degradation label, which classifies network behavior based on local_avg and remote_avg values, distinguishing between normal operation and varying levels of degradation. This label, whether binary or multi-class, identifies instances of normal and degraded network conditions. In binary classification, the label indicates whether an issue is present (yes/no), while the multi-class approach differentiates levels of degradation, such as normal operation, high latency, and downtime. Thresholds calculated from the 95th percentile of `local_avg` and `remote_avg` values provide a systematic basis for assigning these labels, aiding in the categorization and analysis of network performance issues.

3.2. Data Preprocessing

The initial preprocessing step for the wireless and Ethernet datasets involved removing extreme outliers to maintain a focus on realistic latency values. Records where `local_avg` and `remote_avg` exceeded a threshold of 10 milliseconds were filtered out, as this threshold aligns with commonly observed values in network latency studies and industry standards. This value was chosen based on typical network latency conditions, ensuring that the analysis concentrated on meaningful performance patterns rather than rare anomalies.

Next, the 95th percentile for both `local_avg` and `remote_avg` was calculated on the trimmed datasets, establishing a threshold for identifying significant latency spikes indicative of network performance degradation. The choice of the 95th percentile provided a balanced approach, capturing major latency issues while filtering out minor fluctuations. This threshold formed the basis for labeling degradation events, distinguishing normal operation from periods of high latency or downtime. Alternative thresholds, such as the 90th or 99th percentiles, were considered but found to either include too many minor fluctuations or miss critical spikes, further justifying the selected percentile.

To ensure data completeness, missing values were handled carefully: imputation techniques, such as replacing missing entries with mean or median values, were applied where appropriate. This approach maintained data consistency but could introduce bias by assuming uniformity in missing values. In cases where critical fields like local_avg and remote_avg were missing, rows were removed to prevent incomplete records from skewing the analysis. While these basic imputation strategies ensured a usable dataset, more sophisticated methods, such as KNN (K-Nearest Neighbors) imputation, could further enhance data quality by considering patterns and similarities in neighboring data points. Noise was addressed through smoothing techniques and filtering to reduce the impact of anomalous spikes that do not reflect typical network behavior. Data normalization was also performed to bring latency values within a standard scale, facilitating model convergence during training. Additionally, feature scaling methods, such as Min-Max scaling, were applied to ensure consistency across input features, reducing potential biases caused by differing feature magnitudes. These preprocessing steps collectively strengthened the dataset's reliability, ensuring that the predictive models were built on accurate and comprehensive data.

3.3. Labeling Performance Degradation

The performance degradation labeling in the dataset relied on thresholds derived from the 95th percentile of `local_avg` and `remote_avg` latency values, calculated after removing extreme outliers. Instead of using a standardized dataset, the function `label_degradation_non_standardized` applied these thresholds directly to the original latency data, preserving the natural variance and distribution in latency measurements. This approach provided a more accurate depiction of real-world network behavior, as the thresholds reflected latency spikes that significantly deviated from typical network performance, identifying instances of notable degradation. For both wireless and Ethernet datasets, the labeling function applied the respective thresholds to classify entries with `local_avg` or `remote_avg` values exceeding these limits as degradation events. This classification effectively separated periods of normal operation from high-latency conditions, accommodating each dataset's unique latency characteristics and allowing for a tailored threshold-based approach. By avoiding standardization, this method retained critical fluctuations in latency values that are pertinent to assessing network health and identifying potential performance issues. Once labeling was completed, a summary of degradation and non-degradation counts was generated for both datasets to validate the accuracy of the threshold-based labeling. This summary acted as a quality check, confirming that the labeled instances reflected

realistic network conditions and degradation patterns, consistent with expected operational challenges in both network types. This validation step ensured that the labeled data served as a reliable foundation for the study's predictive analysis, accurately distinguishing between normal and degraded performance conditions.

3.4. Model Selection

This study employed XGBoost, Logistic Regression, and MLP models to predict network performance degradation, each selected for specific strengths. XGBoost was chosen for its effectiveness in modeling non-linear relationships and handling complex feature interactions, essential in capturing the multi-dimensional data patterns typical of network degradation. Its iterative boosting approach reduces bias over successive iterations, enhancing prediction accuracy and generalizability in multi-variable analyses. Logistic Regression served as a baseline model, offering a simpler, interpretable benchmark ideal for binary classification tasks, providing insight into individual feature contributions to network performance states. The use of Logistic Regression was further justified due to its strong foundation in statistical modeling and ease of implementation, making it a reliable comparator to assess the added value of more complex models. Meanwhile, MLP was selected for its ability to capture non-linear patterns and feature interdependencies via a neural network structure, well-suited for tasks involving intricate data relationships. Alternative models, such as SVM and Random Forests, were considered but not prioritized due to their relative limitations in capturing highly non-linear feature interactions without extensive tuning or increased computational costs. Together, these models provided a robust framework for analyzing network degradation, each contributing unique capabilities to the predictive process.

3.5. Model Training and Evaluation

The dataset was split into 70% for training and 30% for testing to ensure that models learned from the data while assessing generalizability on new instances. During training, models identified patterns in network behavior, while testing provided an unbiased measure of predictive accuracy. To optimize model performance, hyperparameter tuning was conducted for XGBoost and MLP. Grid search optimized parameters such as learning rate and depth for XGBoost, while random search refined the MLP's hidden layers, neurons, and learning rate, enhancing the models' ability to capture complex network patterns. To address potential overfitting, the MLP model employed regularization techniques, including L2 regularization, and incorporated dropout layers to randomly deactivate neurons during training. Early stopping was also utilized to halt training once validation performance stopped improving, ensuring the model did not learn noise in the data. Model evaluation employed accuracy, precision, recall, F1-score, and AUC-ROC to comprehensively assess predictive performance. Accuracy measured overall prediction success, providing a general sense of how often the models correctly classified network states. Precision and recall focused on identifying degradation events, with precision indicating the proportion of true positive predictions among all positive predictions, and recall capturing the ability of the models to detect all actual degradation events. The F1-score balanced precision and recall, offering a singular metric to assess the trade-off between these two measures, which is particularly critical in imbalanced datasets. AUC-ROC assessed each model's ability to distinguish between degraded and non-degraded states, indicating overall classification performance. While metrics like Matthews Correlation Coefficient (MCC) can be informative for imbalanced data, the selected metrics provided a comprehensive and interpretable measure of the models' real-world applicability and sensitivity to network degradation, which aligned with the study's objectives.

4. Results and Discussion

4.1. Model Performance

The performance of each model—XGBoost, Logistic Regression, and MLP—was evaluated on the test set using accuracy, precision, recall, F1-score, and AUC-ROC as key metrics, shown in figure 2.

 ISSN 2723-6471

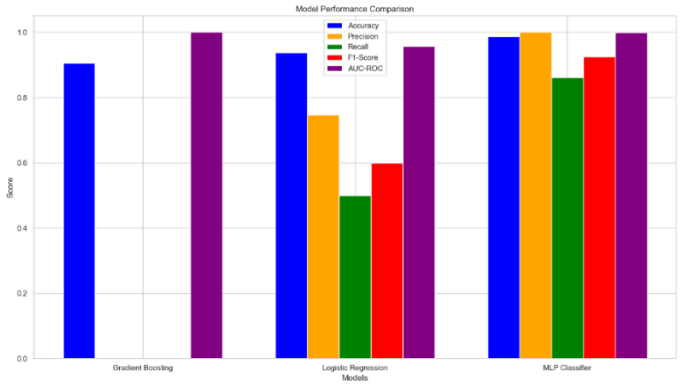

Figure 2. Model Performance Comparison

XGBoost was optimized using grid search over a range of hyperparameters, including learning_rate values from 0.01 to 0.1, max_depth ranging from 3 to 6, and n_estimators between 50 and 200. The optimal values were found to be {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}. For Logistic Regression, a simpler search was conducted to fine-tune the regularization parameter C, with the optimal value determined as {'C': 0.1}. The MLP Classifier underwent tuning of multiple hyperparameters, such as activation functions (relu, tanh), alpha values ranging from 0.0001 to 0.01 for regularization, and varying hidden_layer_sizes configurations (e.g., single and multi-layer structures). The best configuration was {'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes': (100,)}.

The MLP classifier achieved the highest overall performance, with an accuracy of 98.7% and an AUC-ROC of 0.9998, indicating strong predictive capabilities and the ability to distinguish effectively between degraded and non-degraded network states. Its precision for the positive class (indicating degradation) reached 1.0000, with a recall of 0.8622, leading to an F1-score of 0.9260. This reflects MLP's robustness in capturing complex patterns within the dataset, making it well-suited for network degradation prediction. Logistic Regression, used as a baseline model, performed reasonably well, achieving an accuracy of 93.67% and an AUC-ROC of 0.9565. Its precision and recall for the degradation class were 0.7465 and 0.5004, respectively, resulting in an F1-score of 0.5992. Although its overall accuracy was high, Logistic Regression struggled to predict the degradation class with the same precision as MLP. This outcome underscores the limitation of simpler linear models in scenarios where data exhibits non-linear relationships, which more complex models like MLP better capture. Table 1 compares the predictive performance of the MLP, Logistic Regression, and XGBoost models on network degradation detection, highlighting accuracy, AUC-ROC, and key metrics for the degradation class (precision, recall, and F1-score).

Table 1. Model Evaluation Results

Model	Accuracy (%)	AUC-ROC	Precision	Recall	F1-Score
MLP	98.70	0.9998	1.0000	0.8622	0.9260
Logistic Regression	93.67	0.9565	0.7465	0.5004	0.5992
XGBoost	90.54	1.0000	0	0	0

While achieving perfect AUC-ROC (1.0), the XGBoost model showed poor recall and precision for the degradation class, resulting in an F1-score of 0.0000. Despite tuning attempts, XGBoost failed to classify any instances of the degradation class, suggesting that this model was biased towards the majority class (non-degradation). Although its accuracy was 90.54%, the model's inability to detect degradation events limits its practical utility for network monitoring tasks, where identifying performance degradation is critical. This discrepancy highlights the limitations of using AUC-ROC in imbalanced datasets, as the high score can be misleading for evaluating minority class predictions, particularly in contexts where detecting degradation events is essential. In comparing the three models, MLP demonstrated the highest predictive capability across all metrics. Its ability to maintain high precision and recall for the degradation class indicates that it effectively captures the complex and non-linear patterns inherent in network performance data. Logistic Regression, though simpler and more interpretable, was less effective in predicting degradation accurately, highlighting the trade-off between model complexity and predictive accuracy in this context.

 ISSN 2723-6471

XGBoost's perfect AUC-ROC score reflects a good fit for the majority class but reveals its limitations in handling imbalanced data effectively. AUC-ROC primarily measures the ability of a model to distinguish between classes; however, in imbalanced datasets, a high AUC-ROC can be misleading as it may indicate strong performance overall while the model fails to accurately predict the minority class, such as degradation events in this study.

The differences in performance among the models illustrate the impact of class imbalance on model outcomes. While MLP provided the most balanced and accurate results, Logistic Regression can still be useful as a preliminary tool due to its simplicity and interpretability. However, XGBoost's failure to classify the degradation class suggests that further adjustments, such as incorporating oversampling techniques or recalibrating the model's loss function, may be necessary to enhance its sensitivity to minority classes in imbalanced datasets. In summary, MLP emerged as the superior model for predicting network performance degradation in this study, effectively balancing accuracy, recall, and precision. Logistic Regression, despite its limitations, provided a reasonable baseline, while XGBoost's limitations in handling class imbalance indicate that it may require additional modifications for optimal performance in this application. These results suggest that deep learning approaches like MLP, which can handle non-linearities and complex feature interactions, are particularly well-suited for network performance degradation prediction.

4.2. Feature Importance

The feature importance analysis conducted on the XGBoost model provided valuable insights into which variables contributed most significantly to predicting network performance degradation. In this analysis, the `local_avg` and `remote_avg` latency values emerged as the most influential predictors, indicating that fluctuations in these metrics are strongly associated with network degradation. This finding aligns with the initial hypothesis that latency is a critical indicator of network health, especially for performance-sensitive applications. Additional features, such as `timestamp` and `location`, also showed importance, albeit to a lesser degree, suggesting that temporal and spatial factors might influence network stability but are secondary to latency metrics. Confusion matrices were generated for each model (XGBoost, Logistic Regression, and MLP) to provide a clearer understanding of their classification performance in terms of true positives, false positives, false negatives, and true negatives. The XGBoost model demonstrated strong classification capability in identifying non-degradation events (true negatives) but showed limitations in detecting degradation events (true positives), reflecting a possible imbalance in the dataset or threshold sensitivity. Logistic Regression, used as a baseline, also captured non-degradation cases effectively, though its performance in recognizing degradation cases was less precise compared to MLP and XGBoost. The MLP achieved the highest accuracy in distinguishing between degradation and non-degradation cases, as evidenced by its balanced distribution of true positives and true negatives in the confusion matrix. This suggests that MLP's neural network structure allowed it to capture complex relationships within the data that Logistic Regression and XGBoost might have overlooked. The confusion matrix thus provided crucial insights into each model's strengths and limitations, especially in the context of degradation prediction where accurate identification of both positive and negative cases is essential.

4.3. Model Comparison

The comparative analysis of the three models—XGBoost, Logistic Regression, and MLP—demonstrated notable differences in their ability to predict network performance degradation. Among these, MLP consistently outperformed the other models across multiple evaluation metrics, including accuracy, precision, recall, and F1-score. The MLP's performance superiority can be attributed to its neural network architecture, which effectively captures non-linear relationships and complex interactions within the dataset. This capability proved advantageous in analyzing network performance data, where dependencies between features like `local_avg` and `remote_avg` latency values are inherently non-linear. Logistic Regression, used as a baseline model, achieved a reasonable level of accuracy and demonstrated high interpretability. However, its linear nature limited its ability to capture the complex feature interactions present in the dataset. As a result, while Logistic Regression performed well in classifying non-degradation cases, it underperformed in identifying degradation events compared to MLP and XGBoost. This limitation is consistent with the general understanding that linear models may struggle with complex datasets, especially when subtle feature interactions significantly impact the prediction outcome.

XGBoost, while expected to perform robustly due to its ensemble-based structure, displayed mixed results. Although it achieved a perfect AUC-ROC score of 1.0, this high score reflected its proficiency in distinguishing between classes

 ISSN 2723-6471

on a probability basis rather than absolute classification accuracy for degradation cases. In practice, XGBoost's performance was limited by its sensitivity to imbalanced classes and threshold selection, as it often failed to classify degradation events accurately despite capturing high-level patterns in the data. This outcome underscores the importance of not only achieving a high AUC but also ensuring practical applicability in identifying true degradation events. The dataset characteristics played a substantial role in influencing each model's performance. The high dimensionality and non-linear interactions among features, such as `local_avg` and `remote_avg`, were better suited to MLP's architecture, allowing it to learn complex patterns through multiple layers of processing. In contrast, Logistic Regression, which lacks the capacity to model non-linear dependencies effectively, fell short in handling this complexity. XGBoost, though generally robust in handling complex data, was affected by the dataset's imbalances, where degradation events were comparatively rarer, resulting in lower recall for this class.

4.4. Effectiveness of Features

The analysis of feature importance revealed that certain variables played a crucial role in predicting network performance degradation. Among these, `local_avg` and `remote_avg` emerged as the most influential features across all models, particularly in XGBoost and MLP. These features, representing the average latency values measured locally and remotely, provide direct insights into network performance. High values in these latency metrics are indicative of potential bottlenecks or delays, making them strong predictors of degradation events. This finding aligns with prior research emphasizing latency as a key factor in assessing network performance. `Location` was another significant feature in the models' predictions, especially in scenarios where network performance varied based on physical or logical network segments. This feature becomes particularly relevant in identifying specific locations prone to higher traffic loads, interference, or hardware constraints, which can lead to performance issues. In this study, `location` helped the models differentiate areas with stable performance from those more susceptible to degradation, thereby enhancing predictive accuracy. XGBoost, in particular, leveraged this feature effectively, capturing its interactions with latency metrics and highlighting the importance of context-based data in network monitoring.

Interestingly, the effectiveness of each feature varied depending on the model. The MLP model, for example, was able to capture non-linear interactions between `local_avg`, `remote_avg`, and `location` due to its neural network architecture, which allows it to learn complex relationships within the data. This capability enabled MLP to identify subtle patterns that simpler models might overlook. In contrast, Logistic Regression, constrained by its linear structure, relied heavily on individual feature significance, limiting its ability to capture complex dependencies but still benefiting from the latency features as straightforward indicators of performance issues. The scatter plots in figure 3 illustrate the relationship between `local_avg` and `remote_avg` latency values for wireless and Ethernet networks, with degradation events highlighted.

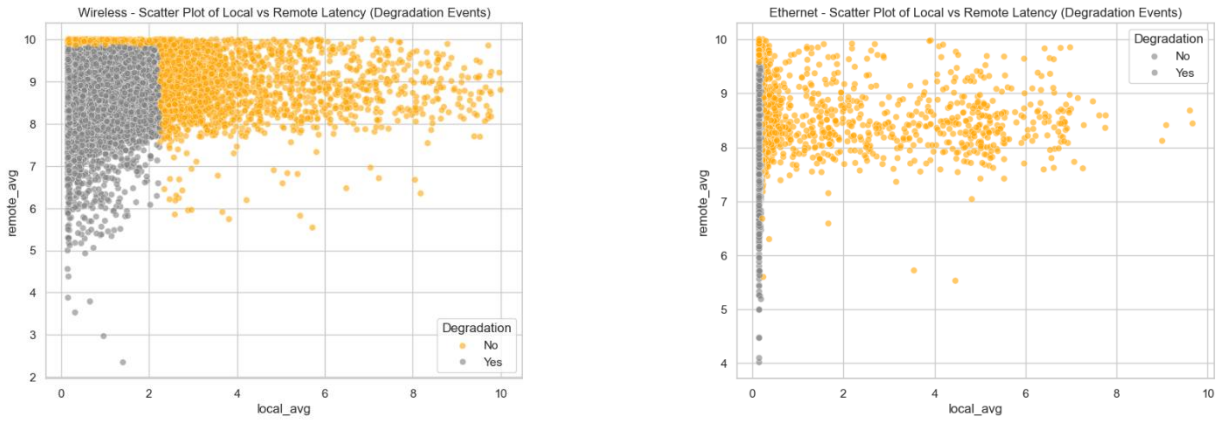

Figure 3. Scatter Plots of Wireless and Ethernet Degradation Events

In the wireless network plot, degradation events (orange points) cluster around higher latency values, particularly when `local_avg` exceeds 4 ms and `remote_avg` is also elevated. This pattern suggests that high latency in either metric significantly contributes to degradation in wireless networks. In contrast, the Ethernet network plot shows a distinct clustering of low `local_avg` values near zero, with degradation events occurring even when `local_avg` is minimal

 ISSN 2723-6471

but `remote_avg` remains high. This behavior indicates that Ethernet degradation is influenced more by remote latency spikes than by local latency, aligning with the typically stable nature of Ethernet connections where local latency is minimal.

4.5. Impact on Network Monitoring

Accurate prediction of network performance degradation transforms monitoring practices, especially in environments using both wireless and Ethernet connections. Unlike traditional reactive methods, predictive models like XGBoost, Logistic Regression, and MLP allow for proactive monitoring by identifying patterns that suggest imminent degradation. This approach enables network administrators to anticipate and address issues, improving reliability and stability in complex networks. In wireless networks, which are sensitive to interference and dynamic loads, predictive models support preventive actions such as load balancing or frequency adjustments. For Ethernet, where degradation often stems from congestion or hardware issues, predictions allow for targeted maintenance on vulnerable connections, reducing unexpected downtime. Figure 4 illustrates time-series latency measurements, highlighting degradation events across wireless and Ethernet networks.

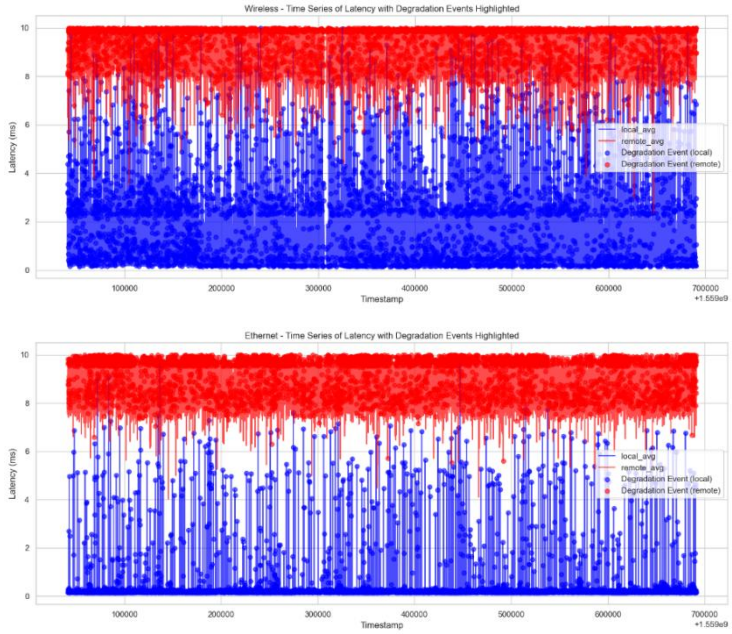

Figure 4. Time Series of Latency Over Time for Wireless and Ethernet Networks

In wireless networks, latency values for local_avg typically cluster between 0 and 2 ms, while remote_avg ranges from 8 to 10 ms, with degradation events concentrated in the higher remote_avg levels. This pattern suggests that remote latency spikes are a key factor in wireless performance degradation. Similarly, in Ethernet networks, local_avg remains low, generally below 2 ms, whereas remote_avg clusters around 8 to 10 ms, with degradation events also linked to elevated remote_avg values. These findings indicate that remote latency is a critical factor in degradation across both network types. Integrating predictive models with real-time monitoring systems could automate responses to early degradation warnings, minimizing manual intervention and speeding up reaction times. Embedded in network management software, these models could trigger alerts when degradation is anticipated, allowing for rapid resource allocation adjustments or reconfigurations. Such proactive alerts are essential in high-stakes sectors like healthcare, finance, and industrial automation, where even brief network disruptions can have serious consequences. Furthermore, predictive insights aid in capacity planning and resource optimization. Network administrators can use degradation predictions to guide decisions on infrastructure upgrades and align resources with demand, especially during high-traffic periods. By anticipating bandwidth constraints, administrators can allocate resources more efficiently, maintaining performance without over-provisioning and reducing network management costs. Together, time-series latency visualizations and predictive models highlight the importance of monitoring local_avg and remote_avg, with a particular focus on remote latency spikes as indicators of potential degradation.

4.6. Limitations

This study faced limitations that could impact the robustness and generalizability of its findings. The dataset, while suitable for initial analysis, may not capture the full range of network performance degradation across varied conditions, and a larger, more diverse dataset could enhance model generalization. To address this limitation, potential solutions include data augmentation or synthetic data generation. Data augmentation can involve creating new samples by introducing controlled variations or perturbations in existing data, thereby increasing data diversity without additional data collection efforts. Alternatively, synthetic data generation methods, such as using generative adversarial networks (GANs) to produce realistic but artificial data points, can provide additional training examples reflective of diverse network scenarios. These approaches could mitigate the dataset size constraint, improving the robustness and applicability of the predictive models to real-world network environments.

Feature selection was another constraint; focusing on `timestamp`, `location`, `local_avg`, and `remote_avg` might have overlooked other important factors, such as environmental influences in wireless networks or hardware variations in Ethernet setups. Additionally, the use of a static threshold based on the 95th percentile for labeling degradation may not adapt well to all network conditions, suggesting that dynamic or adaptive thresholding could improve model flexibility in real-time applications. Future research could integrate advanced models, such as deep learning, to capture complex patterns and conduct real-time monitoring tests to evaluate these models in practical scenarios. Expanding to diverse network types, including 5G and IoT, would also help confirm the generalizability of the findings, ultimately contributing to more versatile and resilient network monitoring models.

5. Conclusion

This study analyzed the predictive capabilities of three supervised learning models—XGBoost, Logistic Regression, and MLP—for identifying network performance degradation in both wireless and Ethernet environments. Among these models, MLP demonstrated the highest predictive accuracy and robustness, particularly in handling complex feature interactions and non-linear patterns present in the dataset. Key features such as `local_avg` and `remote_avg` latency values were found to be the most significant contributors to the prediction of degradation events, as they directly captured fluctuations in network performance. These findings highlight the model's effectiveness in discerning subtle changes in network conditions that might indicate imminent degradation. This research contributes to the growing body of literature on predictive modeling for network management by applying supervised learning methods to the challenge of performance degradation prediction. The study’s integration of XGBoost, Logistic Regression, and MLP provides a comparative analysis, offering insights into the strengths and limitations of each model within different network contexts. By demonstrating that MLP can effectively predict degradation events with high accuracy, this research underscores the potential of neural networks in network performance monitoring. These findings can serve as a foundation for implementing ML-based monitoring tools, enabling proactive management and rapid response to network issues.

Future research could enhance these models by incorporating additional network-specific features, such as bandwidth usage and protocol type, which may offer deeper insights into factors affecting network degradation. Additionally, adapting thresholds dynamically based on evolving network conditions could significantly improve the models’ ability to identify degradation accurately. Dynamic thresholding would allow models to respond to fluctuations in network load, congestion, and changing operational conditions, improving their sensitivity and adaptability. Exploring unsupervised learning techniques could further provide a new dimension to network monitoring by identifying novel patterns and anomalies without the need for labeled data. This approach would complement supervised models, broadening their applicability in dynamic network environments with limited labeled data. Testing predictive models in real-world network settings would validate their robustness and applicability under diverse operational conditions, offering valuable feedback on their practicality and reliability. Real-time deployment within network monitoring systems would facilitate proactive measures by detecting early signs of network degradation, aligning ML capabilities with the demands of modern, high-speed networks and enhancing their overall performance and responsiveness.

6. Declarations

6.1. Author Contributions

Conceptualization: C.R.A.W., S., and D.Y.; Methodology: S.; Software: C.R.A.W.; Validation: C.R.A.W., S., and D.Y.; Formal Analysis: C.R.A.W., S., and D.Y.; Investigation: C.R.A.W.; Resources: S.; Data Curation: S.; Writing Original Draft Preparation: C.R.A.W., S., and D.Y.; Writing Review and Editing: S., C.R.A.W., and D.Y.; Visualization: C.R.A.W. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received financial support for the research, authorship, and/or publication of this article through the Research Grant from Universitas Amikom Purwokerto, Year 2024.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Banerjee, M. Warnier, and F. M. T. Brazier, "Self-Organizing Topology for Energy-Efficient Ad-Hoc Communication Networks of Mobile Devices," *Complex Adapt. Syst. Model.*, vol. 8, no. 1, pp. 1-21, 2020, doi: 10.1186/s40294-020-00073-7.

[2] R. Mukherjee, "Jio Sparks Disruption 2.0: Infrastructural Imaginaries and Platform Ecosystems in 'Digital India,'" *Media Cult. Soc.*, vol. 41, no. 2, pp. 175–195, 2018, doi: 10.1177/0163443718818383.

[3] M. A. Hayudini, "Network Infrastructure Management: Its Importance to the Organization," *Nat. Sci. Eng. Technol. J.*, vol. 2, no. 1, pp. 80–86, 2021, doi: 10.37275/nasetjournal.v2i1.15.

[4] R. Qi, W. Liu, J. Gutierrez, and M. Narang, "Sustainable and Resilient Network Infrastructure Design for Cloud Data Centers," *Service science: research and innovations in the service economy*, vol. 2017, no. 1, pp. 227–259, 2017, doi: 10.1007/978-3-319-65082-1_11

[5] J. S. Lee, J. Lee, and M. Stacey, "Attributions for Underachievement Among Students Experiencing Disadvantage and Support for Public Assistance to Them," *Aust. J. Soc. Issues*, vol. 58, no. 3, pp. 1-19, 2023, doi: 10.1002/ajs4.266.

[6] T. Yang, Y. Dong, and X. Zhang, "Frequency Tracking Synchronization Algorithm for High Latency Wireless Sensor Networks," *2013 47th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2013*, vol. 2013, no. 10, pp. 1-5, 2013, doi: 10.1109/ciss.2013.6624255.

[7] S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, "An Analytical Model to Minimize the Latency in Healthcare Internet-of-Things in Fog Computing Environment," *Plos One*, vol. 14, no. 11, pp. 1-31, 2019, doi: 10.1371/journal.pone.0224934.

[8] C. Cunha and L. A. Silva, "Reboot-Based Recovery of Performance Anomalies in Adaptive Bitrate Video-Streaming Services," *Int. J. High Perform. Comput. Netw.*, vol. 10, no. 4/5, pp. 403-414, 2017, doi: 10.1504/ijhpcn.2017.10007211.

[9] K. O. Park, "A Study on Sustainable Usage Intention of Blockchain in the Big Data Era: Logistics and Supply Chain Management Companies," *Sustainability*, vol. 12, no. 24, pp. 1-15, 2020, doi: 10.3390/su122410670.

[10] T. Alyas, I. Javed, A. Namoun, A. Tufail, S. Alshmrany, and N. Tabassum, "Live Migration of Virtual Machines Using a Mamdani Fuzzy Inference System," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 3019–3033, 2022, doi: 10.32604/cmc.2022.019836.

[11] H. Emesowum, A. Paraskelidis, and M. Adda, "Fault Tolerance and Graceful Performance Degradation in Cloud Data Center," *J. Comput.*, vol. 13, no. 8, pp. 889–896, 2018, doi: 10.17706/jcp.13.8.889-896.

[12] T. Isotalo, J. Palttala, and J. Lempiainen, "Impact of Indoor Network on the Macrocell HSPA Performance," *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, Beijing, China, vol. 2011, no. 1, pp. 294-298, 2010, doi: 10.1109/icbnmt.2010.5705098.

[13] K. Ayub and V. Zagurskis, "Adoption Features and Approach for UWB Wireless Sensor Network Based on Pilot Signal Assisted MAC," *Int. J. Commun. Netw. Inf. Secur. Ijcnis*, vol. 8, no. 1, pp. 40-46, 2022, doi: 10.17762/ijcnis.v8i1.1574.

[14] G. Cantali, E. Deniz, O. Ozay, O. Yıldırım, G. Gûr, and F. Alagöz, "PIM Detection in Wireless Networks as an Anomaly Detection Problem," *2023 International Balkan Conference on Communications and Networking (BalkanCom)*, İstanbul, Turkiye, vol. 2023, no. 7, pp. 1-6, 2023, doi: 10.1109/balkancom58402.2023.10167980.

[15] Y. Ukon, S. Yoshida, S. Ohteru, and N. Ikeda, "Real-Time Virtual-Network-Traffic-Monitoring System With FPGA Accelerator," *NTT Tech. Rev.*, vol. 19, no. 10, pp. 51–60, 2021, doi: 10.53829/ntr202110ra1.

[16] V. Ali, A. A. Norman, and S. R. Azzuhri, "Characteristics of Blockchain and Its Relationship With Trust," *Ieee Access*, vol. 11, no. 2, pp. 15364–15374, 2023, doi: 10.1109/access.2023.3243700.

[17] J. Ramprasath and V. Seethalakshmi, "Improved Network Monitoring Using Software-Defined Networking for DDoS Detection and Mitigation Evaluation," *Wirel. Pers. Commun.*, vol. 116, no. 3, pp. 2743–2757, 2021, doi: 10.1007/s11277-020-08042-2.

[18] R. Liu and E. Wang, "Blockchain and mobile client privacy protection in e-commerce consumer shopping tendency identification application," *Soft Comput. - Fusion Found. Methodol. Appl.*, vol. 27, no. 9, pp. 6019–6031, Apr. 2023, doi: 10.1007/s00500-023-08099-8.

[19] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 236–262, 2016, doi: 10.1109/comst.2015.2477041.

[20] R. Golgiri and R. Javidan, "TMCC: An Optimal Mechanism for Congestion Control in Wireless Sensor Networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 454-459, 2016, doi: 10.14569/ijacsa.2016.070561.

[21] J. Geng, J. Yan, and Y. Zhang, "P4QCN: Congestion Control Using P4-Capable Device in Data Center Networks," *Electronics*, vol. 8, no. 3, pp. 1-17, 2019, doi: 10.3390/electronics8030280.

[22] Surya. S. Raju and S. Manjunath.S., "An Efficient Prelude to Measure Packet Loss and Delay Estimate With Elevated Security Feature," *Int. J. Comput. Appl.*, vol. 26, no. 3, pp. 23–27, 2011, doi: 10.5120/3083-4221.

[23] Z. Jadidi, A. Dorri, R. Jurdak, and C. Fidge, "Securing Manufacturing Using Blockchain," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, vol. 2021, no. 2, pp. 1920-1925, 2020, doi: 10.1109/trustcom50675.2020.00262.

[24] N. Sokolov, A. I. Pyatnitsky, and K. S. Alabugin, "Applying Methods of Machine Learning in the Task of Intrusion Detection Based on the Analysis of Industrial Process State and ICS Networking," *Fme Trans.*, vol. 47, no. 4, pp. 782–789, 2019, doi: 10.5937/fmet1904782s.

[25] A. Nusrat, "Machine Learning Techniques for Detecting Anomalies in IoT Networks," *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 10, pp. 16–23, 2023, doi: 10.22362/ijcert/2023/v10/i10/v10i103.

[26] A. Montesinos-López, O. A. Montesinos-López, J. C. Montesinos-López, C. Flores-Cortés, R. D. Rosa, and J. Crossa, "A Guide for Kernel Generalized Regression Methods for Genomic-Enabled Prediction," *Heredity*, vol. 126, no. 4, pp. 577–596, 2021, doi: 10.1038/s41437-021-00412-1.

[27] O. A. M. López, B. A. Mosqueda-González, A. P. González, A. M. López, and J. Crossa, "A General-Purpose Machine Learning R Library for Sparse Kernels Methods With an Application for Genome-Based Prediction," *Front. Genet.*, vol. 13, pp. 1-12, 2022, doi: 10.3389/fgene.2022.887643.

[28] S. Konanur V. R., W. L. Woo, and E. S. L. Ho, "Predicting Sleeping Quality Using Convolutional Neural Networks," *Advances in Cybersecurity, Cybercrimes, and Smart Emerging Technologies*, vol. 4, no. 3, pp. 175–184, 2023, doi: 10.1007/978-3-031-21101-0_14.

[29] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data," *Applied Sciences*, vol. 14, no. 2, Art. no. 2, pp. 1-9, Jan. 2024, doi: 10.3390/app14020479.

[30] X. Jiang, "A Review of Financial Services Research Based on Blockchain Technology," *Adv. Econ. Manag. Polit. Sci.*, vol. 92, no. 1, pp. 124–130, 2024, doi: 10.54254/2754-1169/92/20231231.

 ISSN 2723-6471

[31] A. A. Muideen, C. K. M. Lee, J. Chan, B. Pang, and H. Alaka, "Broad Embedded Logistic Regression Classifier for Prediction of Air Pressure Systems Failure," *Mathematics*, vol. 11, no. 4, pp. 1014-1025, 2023, doi: 10.3390/math11041014.

[32] F. Huang, Z. Jiang, S. Zhang, and S. Gao, "Reliability Evaluation of Wireless Sensor Networks Using Logistic Regression," *2010 International Conference on Communications and Mobile Computing, Shenzhen*, vol. 2010, no. 5, pp. 334-338, 2010. doi: 10.1109/cmc.2010.49.

[33] J. Naskath, G. Sivakamasundari, and A. A. S. Begum, "A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN," *Wireless Pers Commun*, vol. 128, no. 4, pp. 2913–2936, Feb. 2023, doi: 10.1007/s11277-022-10079-4.