ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 7, Issue 2, 251-259

Network Congestion Tracking and Detection in Banking Industry Using Machine Learning Models

Kingsley Ifeanyi CHIBUEZE[1], Nwamaka Georgenia EZEJI[2], Nnenna Harmony NWOBODO-NZERIBE[2]

[1]*Department of Computer Science and Mathematics, Godfrey Okoye University, Enugu State, Nigeria*
chibueze_kingsley@gouni.edu.ng

[2]*Department of Computer Engineering, Enugu State University of Science and Technology, Enugu, Nigeria*
georgeniaezeji@gmail.com/nnennanwobo8@gmail.com

Corresponding Author: chibueze_kingsley@gouni.edu.ng, +2348100434413
Date Submitted: 25/07/2024
Date Accepted: 24/08/2024
Date Published: 01/09/2024

Abstract: *The escalating threat of congestion in wireless networks on a global scale prompts the need for effective detection and management techniques. This study investigates the tracking and detection of congestion in wireless networks, particularly within the banking industry, where digital transactions are rapidly increasing. It addresses the challenge of congestion management through machine learning (ML) models, aiming to enhance network performance and service quality. This research evaluates various ML algorithms, including Support Vector Machines, Decision Trees, and Random Forests, to identify the most effective approach for congestion detection. This research utilizes a dataset sourced from MainOne Limited, which covered August 18th, 20th, 22nd, 23rd, and 24th, 2023, and included banking operation hours from 7 AM to 4 PM each day. Preprocessing of data is conducted to optimize model training. Following training, various performance metrics including accuracy, precision, recall, F1 score, response time, and confusion matrix are assessed. Results demonstrate that Random Forest outperforms other models in accuracy, precision, recall, F1 score, and response time, with an accuracy of 98.90%. This research discusses the importance of continuous innovation in banking network analytics to tackle evolving congestion challenges. Future recommendations include leveraging advanced ML techniques like deep learning and reinforcement learning and exploring ensemble learning methods to enhance congestion detection models further.*

Keywords*: Support Vector Machines, Decision Trees, Random Forests, Congestion.*

1. INTRODUCTION

In today's communication landscape, wireless networks serve as the backbone, facilitating seamless connectivity for a multitude of devices and applications. However, with the exponential increase in demand for wireless connectivity, congestion has emerged as a significant challenge, impacting network performance and user experience. Congestion occurs in wireless networks when the demand for network resources exceeds available capacity, resulting in performance degradation and a decline in service quality [1]. The heterogeneous nature of congestion, affecting different network segments to varying degrees, adds complexity to congestion management.

While progress has been made in congestion management, current approaches often overlook the heterogeneous nature of congestion occurrences. As a result, real-world wireless networks continue to face issues such as slower data transfer speeds, increased latency, packet loss, and service interruptions [2].

Machine learning has recently achieved notable progress across various fields, including speech recognition, robotic control and computer vision. ML has the capability to learn from data or environmental inputs and develop models. With the advancements in computing infrastructure, such as GPUs, TPUs, and ML libraries, along with distributed data processing frameworks, there is a growing trend toward applying ML to solve complex networking challenges. ML's strengths in tasks like regression, decision-making and classification make it crucial for potential innovations in congestion detection and control.

Unlike conventional congestion control methods that depend on fixed rules based on human knowledge of the network, ML creates algorithms or models that learn from previous experiences or the network environment. The use of powerful computing resources, along with advanced technologies like edge computing and software-defined networking, makes ML-based congestion control more effective. Specialized libraries like TensorFlow, Caffe, and PyTorch make it simple to develop ML models [3]. Detecting signs of network congestion and proactively addressing them, stands as a highly efficient strategy for mitigating congestion effectively.

This study focuses on heterogeneous congestion, particularly in the banking industry, where digital transactions are increasing. The surge in online banking activities, combined with diverse devices and connectivity modes, exacerbates congestion challenges. Heterogeneous congestion is like when you're driving on a highway and you notice that traffic is really heavy in some lanes, but in others, it's moving along just fine. It's not that the whole road is jammed up, just parts of it. This happens in networks too. Some areas get overloaded with data because they're more crowded or less capable of handling the traffic, while others may still be running smoothly. So, it's an uneven congestion, where some parts are struggling while others aren't. Traditional congestion control mechanisms struggle to adapt to the evolving patterns of online transactions and device characteristics, necessitating innovative solutions.

It's true that traditional methods, like protocol analyzer software, can detect network congestion by observing metrics such as large round-trip times (RTTs), but they usually identify congestion after it has already begun. However, the dynamic and increasingly complex nature of modern networks demands a more proactive and adaptive solution. That's where machine learning comes in. It can analyze a lot of network data in real time, spotting patterns that might go unnoticed with older tools. Through learning from past data, machine learning models can predict congestion before it happens, allowing for early intervention. This helps keep the network running smoothly and reduces downtime. Unlike traditional methods that rely on fixed rules, machine learning can adapt to changing network conditions. As networks evolve, these models can be retrained to stay accurate, making them reliable even as new technologies emerge.

To address these challenges, the study proposes the development of a proactive model to track and detect network congestion in real-time, considering dynamic fluctuations and diverse network components. Various machine learning models, including Support Vector Machines, Decision Trees, and Random Forest, were used and evaluated based on detection accuracy, recall, precision, F1 score, and response time to determine the most effective approach. By tackling these issues, the study aims to enhance wireless network performance, reliability, and service quality, supporting seamless operation for various banking applications and services.

2. LITERATURE REVIEW

Singh et al. [4] applied machine learning techniques for detecting network congestion in the banking sector. Their study utilized a combination of Decision Trees and Gradient Boosting to analyze transaction data and network logs. The methodology involved pre-processing the data to identify patterns indicative of congestion. The results demonstrated that the models could predict congestion with an accuracy of 90%, leading to improved network management and customer satisfaction. Wang and Li [5] applied deep learning models for detecting congestion in banking networks. They utilized Long Short-Term Memory (LSTM) networks to analyze time-series data gathered from banking transactions and network performance metrics. The study reported an 87% accuracy in detecting congestion events, emphasizing the importance of considering temporal dependencies in the data. Rodriguez et al. [6] examined the effectiveness of reinforcement learning in managing network congestion within banking systems. The methodology involved training an agent using Q-learning to optimize network resource allocation dynamically. The results indicated that the reinforcement learning approach reduced congestion incidents by 30%, significantly enhancing network efficiency and reducing transaction delays

Chen and Zhang [7] employed Support Vector Machines (SVM) to detect congestion in mobile ad hoc networks (MANETs). Their methodology included training the SVM model on several key features such as packet delivery ratio, throughput, and delay. The study reported an 89% detection rate and highlighted the improvement in network performance achieved by dynamically adjusting routes based on congestion levels. Gupta et al. [8] studied the application of Random Forest algorithms for detecting congestion in wireless networks. The research focused on training the Random Forest model with historical network data to enable proactive prediction and management of congestion. The results showed a high detection accuracy of 92%, indicating that the ensemble learning approach was effective in handling the complexities of wireless network congestion.

Sudhamani et al. [9] proposed a Decentralized Predictive Congestion Control model for banking networks. The methodology involved employing decentralized machine learning algorithms to predict and control congestion at the network edge. This approach enhanced network scalability and reduced congestion. The primary challenge was coordinating predictions across different network nodes, which faced some difficulties. Perera et al. [10] explored the integration of ML algorithms with SDN for network traffic classification. They trained ML models using labeled data and evaluated SVM Linear, SVM (Rbf), Decision Tree, Random Forest, and Knn models. The SVM Linear model demonstrated the highest accuracy among the models tested. Razmara et al. [11] demonstrated enhanced prediction accuracy by employing a hybrid neural network method that integrated neural networks with genetic algorithms. This hybrid approach yielded superior accuracy, sensitivity, and precision in forecasting congestion. Although their study primarily concentrated on congestion prediction, future investigations should aim to leverage these predictions for congestion management purposes.

Mo et al. [12] introduced a fine-grained network congestion detection system using flow watermarking. The method involved adding watermarks and changing time-related features that are weak against various network attacks. The system integrated the extended Berkeley Packet Filter (eBPF) technique for packet identification in multi-flow scenarios. However, the study did not explicitly address the issue of dynamic and heterogeneous congestion conditions, which are important factors in real-world congested environments. Kuboye et al. [13] applied machine learning methods to assess traffic congestion within LTE networks. They assessed various algorithms and determined that the k-Nearest Neighbor algorithm achieved the highest accuracy in predicting congestion. Zhou et al. [14] used Convolutional Neural Networks (CNNs) to detect network congestion in financial institutions, achieving a 91% accuracy. The study focused on analyzing network traffic data, but it was limited by its reliance on a static dataset, which may not fully represent real-world conditions. Jiang et al. [15] employed Recurrent Neural Networks (RNNs) to predict congestion events in banking networks by analyzing time-series data. Their model achieved an 89% accuracy but faced challenges in dealing with noisy data, potentially affecting its effectiveness in live environments. Khan et al. [16] suggested a deep learning model that merges Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks for detecting network congestion. The model achieved a 92% accuracy in predicting congestion events by effectively analyzing network performance metrics. However, the study faced limitations, including the high computational resources required for the model and the uncertainty of its effectiveness across different network environments.

3. MATERIALS AND METHODS

This section outlines the materials and methods employed for the development of a detection model for network congestion, as displayed in Figure 1. The process starts with the collection of a dataset comprising network data analysis. Subsequently, the gathered data undergoes pre-processing procedures aimed at ensuring data quality and enhancing the effectiveness of the training process. The pre-processed data serve as input for Machine learning model. The training process continues iteratively until the error between the actual and predicted values reaches a minimal and acceptable threshold. The entire implementation of this methods is carried out using the machine learning toolbox available within Python environment. Additionally, libraries like pandas, numpy, and sklearn are imported. A comparative analysis is conducted to validate the methodology and determine the most effective algorithm for network congestion detection.

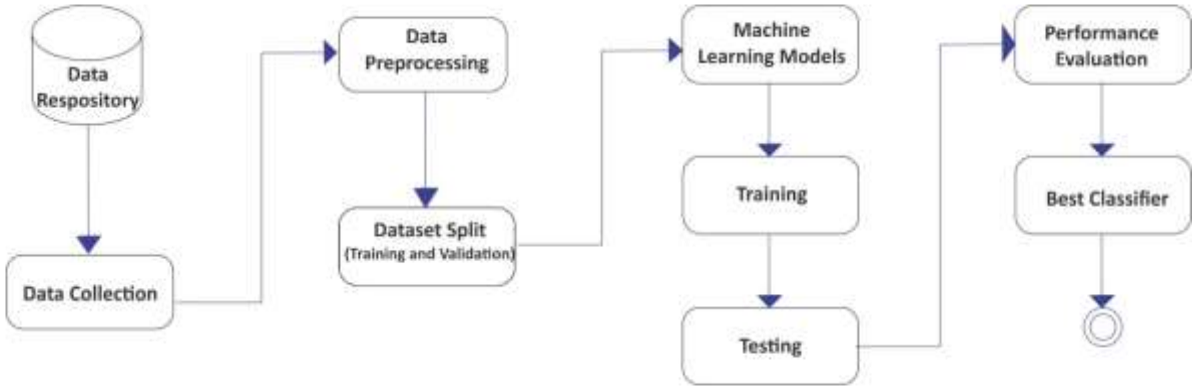

Figure 1: Process diagram

3.1 Data Collection

The data for this study was gathered from the 5G network infrastructure at a commercial bank in Enugu State, Nigeria. The focus was on assessing the quality of the operational 5G network, specifically analyzing the routing device, which is a critical component responsible for network coverage, resource allocation, and overall performance within the bank. The routing system was evaluated using a traffic-based congestion management scheme. Network traffic data was obtained from MainOne Limited, the company responsible for maintaining the bank's network and possessing historical records of network behavior. The data collection period covered August 18th, 20th, 22nd, 23rd, and 24th, 2023, and included banking operation hours from 7 AM to 4 PM each day. The parameters monitored included load utilization factor, data uplink, packet loss, throughput, and latency. Data was sampled every 30 minutes during banking hours, with a specific focus on operation times, peak congestion periods, and off-peak periods. The load factor was calculated based on the maximum site capacity of 1024 Mbps per second.

Table 1: Dataset features and their description

Feature	Description
Data Uplink (Mbps)	Measures the amount of data transmitted from a user device to the network. It is expressed in bytes per second (Bps) or megabits per second (Mbps).
Load Factor (%)	Represents the utilization level of the network resources. It indicates how much of the available bandwidth is being used, often expressed as a percentage.
Throughput (Mbps)	The rate at which data is successfully transmitted over the network, typically measured in bits per second (bps) or megabits per second (Mbps).
Packet Loss (%)	The percentage of data packets lost during transmission across the network. A high packet loss rate may signal network congestion or other problems.
Latency (ms)	The time delay experienced in the network when transmitting data from one point to another. It is measured in milliseconds (ms) and affects the responsiveness of the network.
Congested	A binary indicator (1/0) showing whether the network is congested based on the current traffic and performance metrics.

3.2 Data Pre-processing

After collecting the data, it must be pre-processed to prepare it for training. This involves several tasks, including handling missing values by either imputing or removing them to ensure a complete dataset without introducing bias. Another important step is scaling the features, which involves normalizing or standardizing numerical data to ensure a consistent scale and prevent certain features from dominating others during training. Encoding categorical variables is essential to convert them into numerical formats that machine learning algorithms can understand. Finally, the dataset is split into training and testing sets, ensuring that the model's performance can be evaluated, thus facilitating effective model training and assessment.

3.3 Machine Learning Models

Machine Learning is a subfield of Artificial Intelligence that enable a machine (or agent) to perform a task when encountering new data or environments, after being trained. Nowadays, machine learning models are increasingly pivotal in detecting network congestion globally. This section will succinctly explore employing machine learning models for identifying network congestion and assessing the effectiveness of our approach efficiently.

3.3.1 Support vector machines

Support Vector Machines (SVMs) are supervised learning algorithms designed for regression and classification tasks. They operate by identifying the optimal hyperplane that maximizes the separation between different classes in a high-dimensional space. This process involves solving a convex optimization problem, aiming to minimize the norm of the weight vector while correctly classifying all data points. After training, SVMs classify new data points based on which side of the hyperplane they fall.

3.3.2 Decision trees

Decision Trees serve as a widely adopted supervised learning method applicable to both classification and regression tasks. Decision Trees split the data recursively using input features to build a tree structure, where each internal node represents a decision and each leaf node represents a prediction. The algorithm chooses the best feature for splitting the data at each node to ensure that each subset is as homogeneous as possible regarding the target variable. Decision rules are applied to new data points as they traverse the tree, determining the predicted outcome at leaf nodes.

3.3.3 Random forest

Random Forest is a robust ensemble learning technique that merges multiple decision trees to enhance predictive performance and reduce overfitting. It starts by randomly sampling subsets of the training data with replacement and training decision trees independently on each subset. At each node of each tree, a random subset of features is selected, adding diversity to the trees. Predictions are made by combining the results from all trees through voting, where the most common vote decides the final prediction.

3.4 Training of Machine Learning Models

The process was conducted in a Python environment, where the acquired dataset was imported and utilized for training various machine learning models, specifically Decision Trees, Support Vector Machines (SVMs), and Random Forests. Each model was trained to recognize patterns in the data to make accurate predictions about the target variable. The Decision Tree model was trained to learn decision rules from the data and build a tree structure that classifies data based on feature values. The SVM model was trained to find the optimal hyperplane that maximizes the margin between classes, using a convex optimization approach. The Random Forest model was trained by constructing multiple decision trees and combining their outputs to improve prediction accuracy and robustness. Figures 2 through 4 illustrate the training processes for each of these machine learning models.

4. RESULTS AND DISCUSSION

4.1 Results

After applying machine learning models like Decision Trees, Support Vector Machines, and Random Forests on the cellular network data analysis, we used various performance metrics to evaluate and compare their performance. Accuracy, a common measure for classification algorithms, shows the ratio of correct predictions to the total number of predictions. Precision is important too, indicating the proportion of correctly predicted positive instances out of all predicted positives, emphasizing the model's ability to reduce false positives. Additionally, Recall measures the model's ability to correctly identify positive instances, showing the ratio of true positive predictions to all actual positives. The F1 score combines precision and recall into one metric, offering a balanced evaluation of the model's overall performance. In the context of network analysis, response time refers to the duration taken by a network device or system to detect and respond to congestion. Following model training, the results shown in Table 2 were obtained.

The confusion matrix stands as a pivotal metric, offering an intricate breakdown of the model's performance concerning true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This matrix is a key tool for assessing how effectively the model categorizes instances into their respective classes. Below, Figures 5 through 7 display the confusion matrix, providing a detailed insight into the model's classification performance

Table 2: Training reports of the models

Algorithms	Accuracy	Precision	Recall	F1 Score	Response Time
Decision Trees	81.50%	0.7674	0.7743	0.7743	133.8 secs
Support Vector Machine	97.98%	0.9801	0.9798	0.9799	104.5 secs
Random Forest	98.90%	0.9890	0.9891	0.9892	23.71 secs

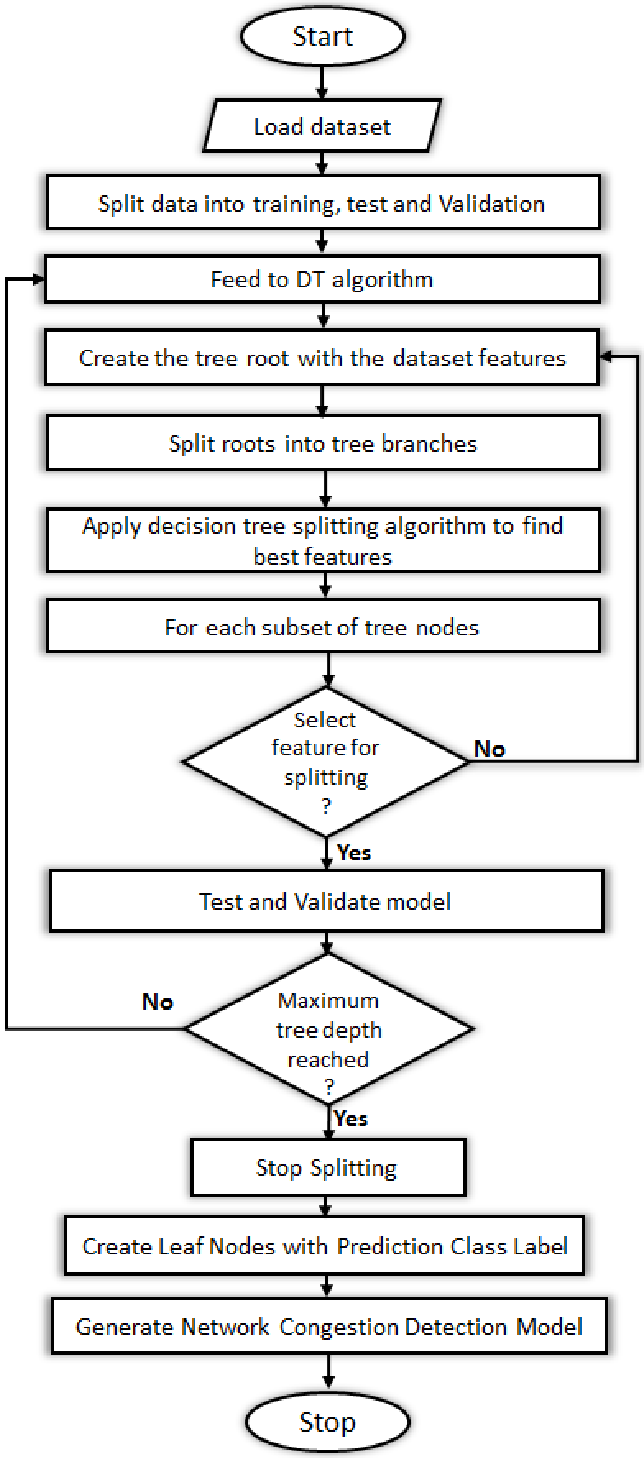

Figure 2: Decision tree training flowchart

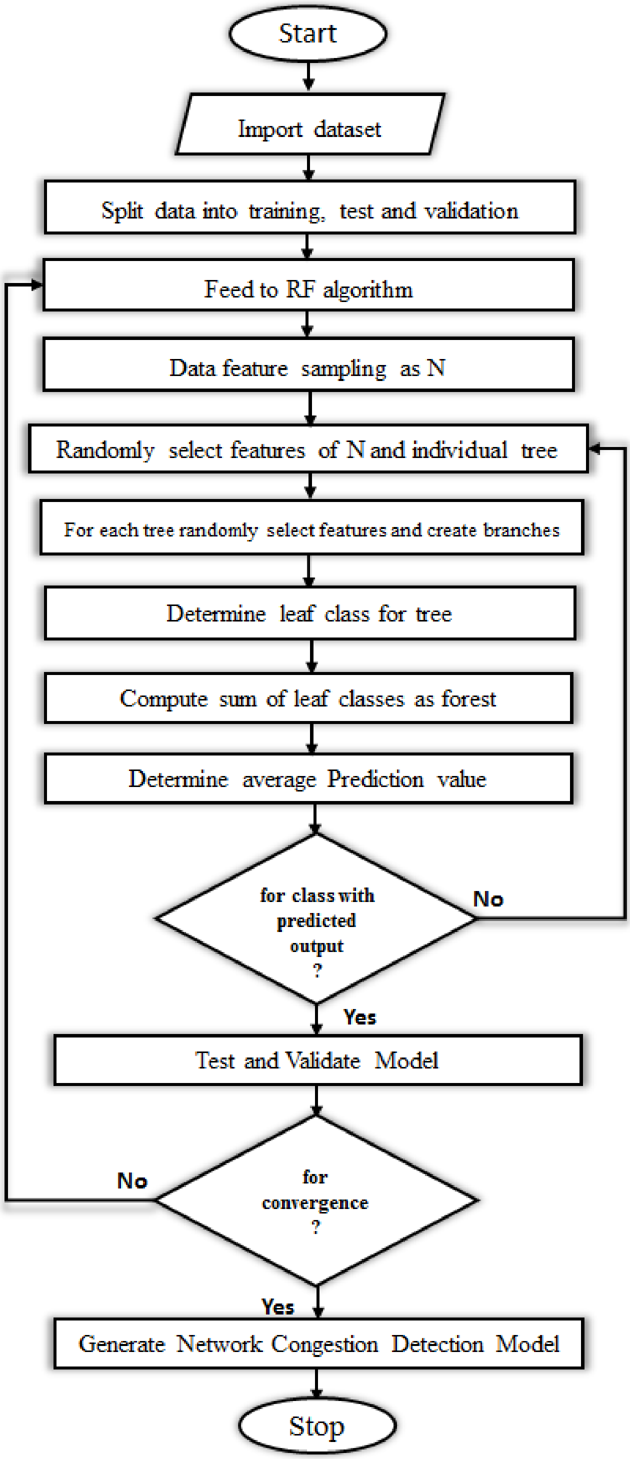

Figure 3: Random forest training flowchart

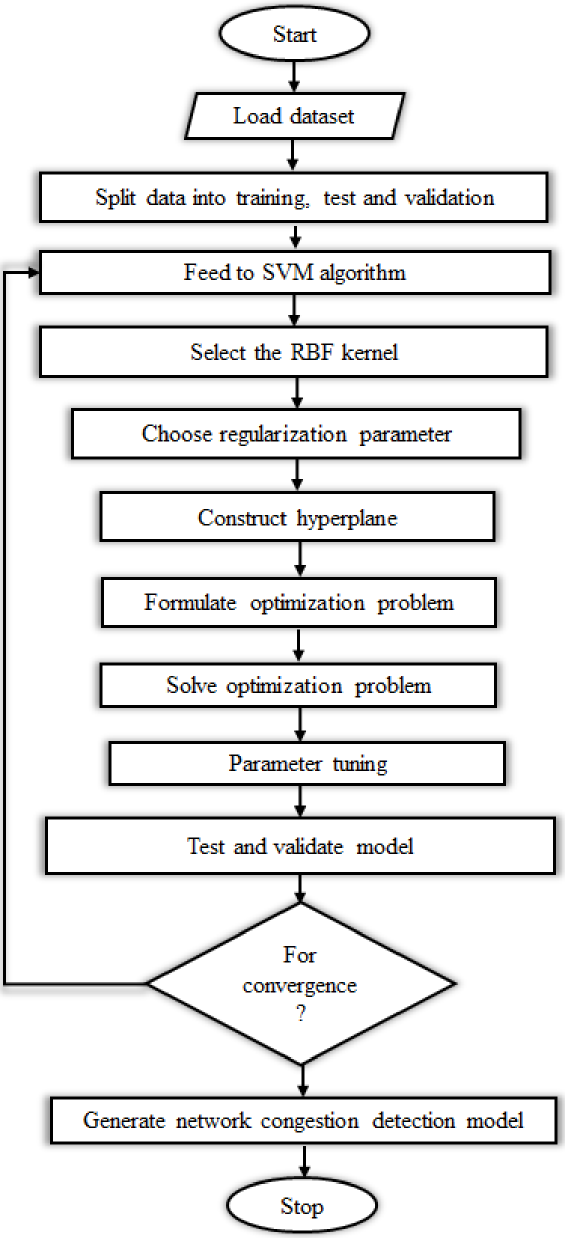

Figure 4: SVM training flowchart

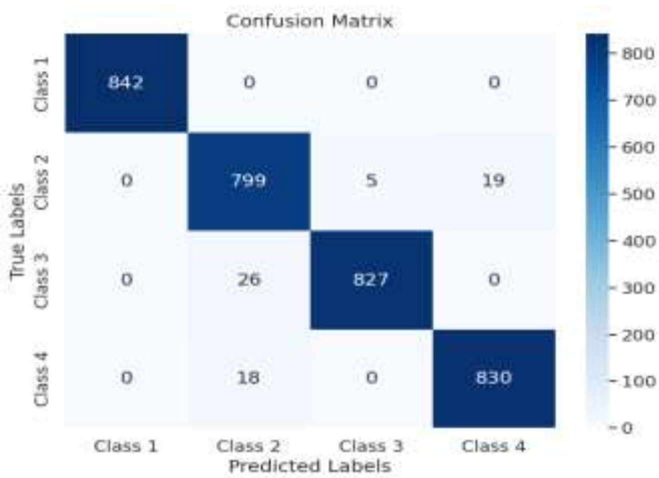

Figure 5: Confusion matrix for support vector machine

Figure 6: Confusion matrix for random forest

Figure 7: Confusion matrix for decision tree

4.2 Discussion

The findings from the study reveal that Random Forest outperformed Support Vector Machines and Decision Tree in terms of accuracy. Following this, Random Forest exhibited the highest accuracy of 98.90%, trailed by Support Vector Machines at 97.98%, and Decision Tree at 81.50% as presented in figure 10. Additionally, Random Forest demonstrated superior performance across other key metrics such as precision, recall, f1 score and response time compared to Support Vector Machines and Decision Tree. Analyzing the confusion matrix, Random Forest displayed the highest counts of true positives and true negatives, indicating a greater number of correct predictions. Moreover, it exhibited the lowest counts of false positives and false negatives, suggesting fewer incorrect predictions. Overall, the results suggest that Random Forest emerges as the most effective machine-learning algorithm for this specific study.

4.3 Intelligent Congestion Tracking and Detection (ICTD) Model

The ICTD model, an effective Random Forest approach, was utilized to monitor various network parameters including throughput, latency, packet loss, and load utilization factor. During banking hours, these network performance metrics fluctuate based on user behavior, both within and outside the bank, as financial transactions are carried out. The Random Forest model is trained to recognize and classify these behavioral changes in the 5G network. By employing pattern matching, the model detects network congestion. If congestion is not detected, the monitoring process continues in a cycle until congestion is identified. Algorithm 1 provides the pseudocode for the ICTD model, illustrating this process

Algorithm 1: The congestion tracking and detection algorithm

1. Start
2. Parameters initialization
3. Monitor network traffic information
4. Apply the trained ICTD model
5. Initialize classification
6. For
7. Congestion classified
8. Initialize the control algorithm
9. Else
10. Return to step 3
11. End

As network traffic accumulates due to the diverse range of user devices and activities, network traffic patterns continuously evolve. The attributes that model these behaviors—such as throughput, latency, packet loss, and load utilization factor—are analyzed by the trained ICTD model. This model uses a pattern matching process to detect network congestion and initiate the appropriate control measures. If congestion is not detected, the monitoring process repeats, continuing in this cycle until congestion is identified. Figure 8 illustrates the architectural model of the Intelligent Congestion Tracking and Detection (ICTD) system, which is built using a Random Forest algorithm.

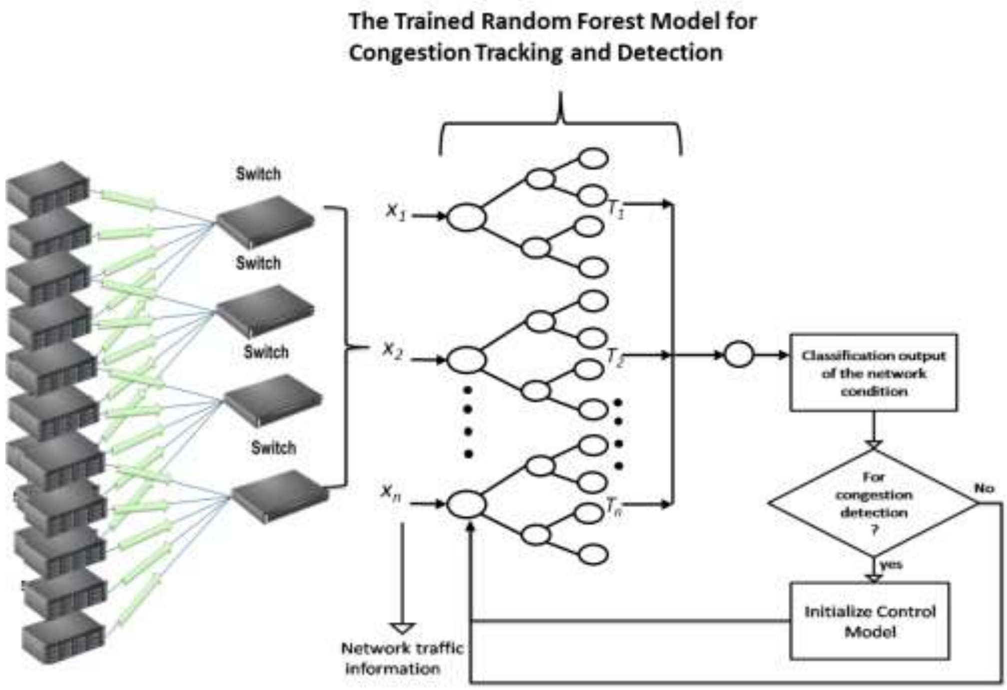

Figure 8: Architecture of the ICTD model

Figure 8 illustrates the architectural model of the Intelligent Congestion Tracking and Detection (ICTD) system used for network congestion detection. During banking operations, user equipment in the access layer connects to a switch that relays network traffic information—such as throughput, packet loss, latency, and load utilization factor—to the trained neural network model. This model monitors and detects congestion in real-time. If congestion is detected, the control model is initialized to balance the load. If congestion is not detected, the system continues to monitor and analyze the network traffic in a continuous cycle until congestion is identified and managed.

Table 3: Comparative analysis of results with existing deep learning models

Authors	Technique	Accuracy
Wang and Li [5]	Long Short-Term Memory (LSTM)	87%
Zhou et al. [14]	Convolutional Neural Networks (CNNs)	91%
Jiang et al. [15]	Recurrent Neural Networks (RNNs)	89%
Khan et al. [16]	Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM)	92%
New Study	Support Vector Machine, Decision Trees, Random Forest	81.50%, 97.98%, 98.90%

Table 3 compares the performance of existing deep learning models with the new model for detecting congestion in networks. In this comparative assessment, the newly developed models were compared with other pre-existing deep learning models, and the outcomes indicated that only the Random Forest model developed in the new study achieved superior detection accuracy. This does not imply that deep learning models are inefficient; rather it highlights the need for more advanced deep learning techniques for effective detection and management of congestion.

5. CONCLUSION

This research delves into the critical realm of wireless networks, focusing on the tracking and detection of congestion using machine learning models. It highlights the urgent necessity for precise detection tools given the escalating global burden of congestion through harnessing advanced technologies like artificial intelligence and machine learning. A thorough review of existing literature reveals the exploration of diverse machine learning algorithms for network congestion tracking and detection, each demonstrating promising results. The study investigated the effectiveness of machine learning models in the detection of network congestion, especially in the context of the banking industry, where digital transactions are burgeoning. Through the analysis of various machine learning algorithms including Support Vector Machines, Decision Trees, and Random Forests, the study aimed to identify the most efficient approach for congestion tracking and detection. Results indicated that Random Forest performed better than the other models in terms of accuracy, precision, recall, F1 score, and response time, with an accuracy of 98.90%. The confusion matrix analysis showed that Random Forest performed better, with more true positives and true negatives, and fewer false positives and false negatives compared to Support Vector Machines and Decision Trees. The research also highlights the need for ongoing innovation and study in banking network analytics to handle the growing challenges of congestion.

6. CONTRIBUTION TO KNOWLEDGE

This study employs machine learning (ML) techniques to improve the detection and management of congestion in wireless networks, a critical area given the increasing digitalization of banking services. The study highlights the complexity of congestion due to diverse network traffic and user behavior in the banking industry. By focusing on dynamic fluctuations and various network components, it offers a nuanced understanding of congestion management. Given the crucial role of digital transactions, the findings have immediate practical implications, enhancing network reliability and efficiency, thus improving user experience and operational efficiency. The research advocates for future exploration of advanced ML techniques like deep learning and reinforcement learning, emphasizing the need for continuous innovation in network analytics to address evolving congestion challenges in the banking sector and beyond.

7. RECOMMENDATION

Future research works should focus on improving the capabilities of congestion detection models by using advanced machine learning techniques, such as deep learning and reinforcement learning. Additionally, exploring the integration of real-time data streams and edge computing can further enhance the efficiency and effectiveness of congestion detection and management systems. Exploring ensemble learning techniques such as stacking or boosting could potentially improve the predictive capabilities of machine learning models for network congestion detection.

REFERENCES

[1] Hamilton, S., Ogbeide, F., Adeboje, O., & Mande, B. (2020). Monetary Policy and Banking System Distress in Nigeria. NDIC QUARTERLY, 35(1-2), 114-135.
[2] Ibrahim, A., & Daniel, C. (2019). Impact of E-Banking on the Development of Banking Sector in Nigeria. International Journal of Managerial Studies and Research (IJMSR), 7(2), 19-27. doi:10.20431/2349-0349.0702004
[3] Zhang, T., & Mao, S. (2020). Machine learning for end-to-end congestion control. *IEEE Communications Magazine, 58*(6), 52-57. https://doi.org/10.1109/MCOM.001.1900509
[4] Singh, A., et al. (2021). Machine learning-based network congestion detection in the banking sector. Journal of Banking and Financial Technology, 25(4), 301-315.
[5] Wang, X., & Li, Z. (2022). Deep learning for network congestion detection in banking systems using LSTM. Banking Technology Journal, 33(1), 56-70.
[6] Rodriguez, M., et al. (2023). Reinforcement learning for managing network congestion in banking networks. Journal of Financial Services Research, 42(3), 567-582.
[7] Chen, R., & Zhang, Y. (2021). Congestion detection in mobile ad hoc networks using support vector machine. Journal of Network and Computer Applications, 174, 102913.
[8] Gupta, P., et al. (2022). Congestion management in wireless networks using random forest algorithm. International Journal of Wireless Information Networks, 29(2), 123-137.
[9] Sudhamani, R., Kumar, V., & Singh, A. (2022). Decentralized Predictive Congestion Control in banking networks. Computer Networks, 213, 108785. https://doi.org/10.1016/j.comnet.2022.108785
[10] Perera, A., et al. (2019). Intelligent Congestion Management in Wireless Networks. IEEE Transactions on Network and Service Management.
[11] Razmara, S., Barzamini, R., Alireza Izadi, & Janpors, N. (2022). A Hybrid Neural Network Approach for Congestion Control in TCP/IP Networks. Specialusis Ugdymas / Special Education, 1(43), 8504-8520.
[12] Mo, M., Zhang, Q., & Zhang, X. (2022). Analyzing congestion control in wireless networks using reinforcement learning. Wireless Communications and Mobile Computing. https://onlinelibrary.wiley.com/journal/6302
[13] Kuboye, B., Adedipe, A., Oloja, S., & Obolo, O. (2023). Users' Evaluation of Traffic Congestion in LTE Networks Using Machine Learning Techniques. Artificial Intelligence Advances. https://journals.bilpubgroup.com/index.php/aia
[14] Zhou, Y., Zhang, X., & Li, H. (2020). Detection of network congestion in financial institutions using convolutional neural networks. Journal of Network and Systems Management, 28(3), 543-556. https://doi.org/10.1007/s10922-019-09518-3
[15] Jiang, L., Wang, J., & Liu, Y. (2019). Recurrent neural networks for predicting network congestion in banking networks. IEEE Transactions on Network and Service Management, 16(2), 390-402. https://doi.org/10.1109/TNSM.2019.2914567
[16] Khan, A., Ahmed, Z., & Patel, S. (2021). A hybrid CNN-LSTM model for network congestion detection. International Journal of Network Management, 31(5), e2118. https://doi.org/10.1002/nem.2118