International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025
© 2025 IJERST. All Rights Reserved

Research Paper

INTELLIGENT THREAT DETECTION IN MOBILE EDGE NETWORKS USING MACHINE LEARNING

Dr. G.N.V. Vibhav Reddy[1], B. Shiva[2], B. Kiran Venkat[2], M. Shivakumar[2], B. Praveen[2]
[1]Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (AI&ML)
[1,2]Sree Dattha Institute of Engineering and Science, Ibrahimpatnam, 501510, Telangana

ABSTRACT

Mobile Edge Computing (MEC) is an evolving paradigm that brings computation and data storage closer to end users, aiming to enhance the performance of mobile applications. However, the rise of MEC technologies has introduced significant security concerns, as malicious activities increasingly target mobile edge systems. These threats include data breaches, denial-of-service (DoS) attacks, and intrusion attempts that exploit infrastructure vulnerabilities, undermining the integrity, availability, and confidentiality of the system. Traditionally, mobile edge security relied on signature-based and rule-based detection methods. These systems depended on predefined patterns of known attacks, limiting their ability to detect novel or evolving threats. As cyber-attacks have grown more sophisticated, the need for advanced detection techniques has become critical. The integration of machine learning (ML) algorithms has transformed the landscape of mobile edge security by enabling real-time detection and prevention of both known and unknown attacks. Given the growing reliance on MEC for critical applications, it is essential to develop robust, adaptive security solutions that ensure data protection and system reliability. Earlier approaches faced challenges such as high false-positive rates, limited detection capabilities, and the need for manual updates to threat databases. To overcome these limitations, ML-based systems are proposed to automate the detection of malicious activities in mobile edge environments. Using classifiers such as Decision Trees, Random Forests, and Deep Neural Networks, these systems can process large volumes of data, identify complex patterns, and accurately classify malicious behaviors. This approach enhances the accuracy, scalability, and efficiency of security mechanisms, aiming to build self-improving systems that continuously learn from new data and adapt to emerging threats.

Keywords: Mobile Edge Computing (MEC), Cybersecurity, Machine Learning, Intrusion Detection, Adaptive Threat Detection.

Received: 14-7-2025 Accepted: 21-8-2025 Published: 28-8-2025

1.INTRODUCTION

Mobile Edge Networks (MENs) are revolutionizing modern computing by enabling data processing closer to the data source, significantly enhancing performance for latency-sensitive applications like autonomous vehicles, augmented reality, and remote surgeries. The global edge computing market is projected to reach USD 111.3 billion by 2028, driven by the proliferation of IoT devices, 5G infrastructure, and real-time analytics. However, this rapid growth also expands the attack surface, making MENs increasingly vulnerable to cyber threats due to their distributed architecture and limited resource nodes. Common attacks include man-in-the-middle, DDoS, data poisoning, spoofing, and privilege escalation, with research indicating that over 57% of threats go undetected due to limited visibility. Edge nodes often lack robust security infrastructure, relying instead on lightweight protection mechanisms that attackers exploit to implant malware, exfiltrate data, or disrupt operations. These risks are particularly pronounced in sectors like healthcare, finance, and industrial IoT, where the impact of a breach can be severe.

The dynamic nature of mobile edge environments—characterized by shifting traffic patterns, device mobility, and behavioral variability—renders traditional, static security solutions ineffective. As 75% of enterprise data is expected to be processed at the edge by 2025, there is a critical need for intelligent, scalable, and context-aware security systems capable of real-time anomaly detection. This has led to the integration of data-driven machine learning models for proactive threat identification and automated response. In real-world corporate settings such as AWS Greengrass, Azure IoT Edge, and Google Anthos, edge computing is extensively used for localized processing, control systems, and content caching. These environments handle sensitive data, making them prime targets for cybercriminals, while centralized security models prove inadequate. Similarly, in industrial environments like those operated by Siemens and Schneider Electric, edge computing enables predictive maintenance and automation, where any intrusion can halt production or compromise control systems. In healthcare, companies like GE Healthcare and Philips use edge devices for real-time patient monitoring, where a malicious attack could result in life-threatening misdiagnoses.

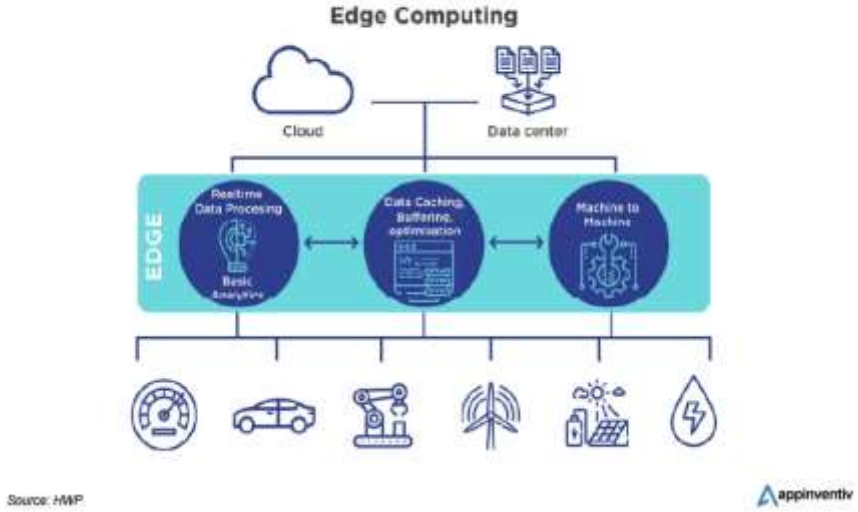

Fig.1: Edge computing architecture

Despite the benefits, MENs face major security challenges. Their decentralized nature, limited resources, and heterogeneous devices reduce the effectiveness of traditional, cloud-oriented security mechanisms. Frequent changes in network topology and device behavior further limit static rule-based systems, often resulting in high false positive rates or undetected anomalies. The absence of a unified monitoring framework across edge nodes allows attackers to exploit inconsistencies and remain undetected. Moreover, even when threats are detected, a lack of interpretability and delayed response compromises the effectiveness of mitigation efforts. These challenges necessitate the development of a real-time, adaptive, and intelligent detection system tailored to mobile edge environments.

Detecting malicious activity in MENs is vital for safeguarding sensitive data, maintaining uptime, and ensuring secure communication in real-time applications. Successful intrusion can lead to financial losses, operational disruptions, reputational damage, or even loss of life in critical sectors. Implementing advanced detection mechanisms not only improves threat mitigation and reduces false alarms but also supports regulatory compliance and strengthens system resilience. Ultimately, this contributes to building a secure, self-defending edge ecosystem.

The primary research objective is to assess the effectiveness of current machine learning models—such as Decision Trees, Random Forests, Support Vector Machines, and clustering-based approaches—in detecting malicious behavior in MENs. The study evaluates these models in terms of accuracy, latency, and computational efficiency within real-time environments. Additionally, it

proposes a deep learning-based detection framework using a Deep Neural Network (DNN) architecture that learns from raw edge traffic data to identify complex threat patterns, reduce false positives, and generalize across different edge scenarios, ensuring a scalable and robust solution for practical deployment. The advantages of this approach include enhanced real-time detection, higher accuracy through intelligent data learning, scalability across heterogeneous edge devices, reduced false positives, and adaptability to new threats. It is also computationally lightweight, supports contextual analytics for better incident response, integrates easily with existing platforms, automates threat prioritization, and enables continuous learning for ongoing improvement. Applications of such a system span a wide range of domains. In healthcare, it secures real-time patient data processing; in industrial IoT, it protects infrastructure and operational commands; and in intelligent transport, it maintains secure vehicle communication. It also enhances security in smart homes, mobile banking, video surveillance, agricultural monitoring, retail analytics, public safety systems, and smart city infrastructure, where secure and reliable data processing is essential for operational efficiency and user safety.

2. LITERATURE SURVEY

R. Braden et. al. [1] discusses the fundamental requirements for internet host communication layers, providing an early framework for secure data transmission. The report outlines critical security considerations necessary for maintaining the integrity and confidentiality of communications, which serve as the foundation for modern mobile edge security mechanisms. The study highlights vulnerabilities in host-based communication that can be exploited by malicious entities, necessitating advanced security measures. M. Korczy´nski and A. Duda [2] introduce Markov chain fingerprinting to classify encrypted traffic, an approach that enhances threat detection without decrypting data. This method is particularly relevant for mobile edge security as it enables anomaly detection while preserving user privacy. Their research demonstrates the effectiveness of probabilistic models in distinguishing between normal and malicious activities within encrypted network streams, a crucial aspect in preventing security breaches.

B. Anderson and D. McGrew [3] propose a machine learning-based approach to identifying encrypted malware traffic using contextual flow data. The study emphasizes the limitations of traditional signature-based detection and highlights the need for adaptive techniques that can analyze traffic behavior in real time. Their work underscores the importance of leveraging AI-driven models to enhance security at the edge of the network, where conventional methods struggle to provide timely threat identification. E. Rescorla [4] presents an update on the Transport Layer Security (TLS) protocol, detailing improvements in encryption standards that enhance security for mobile and edge computing environments. The paper addresses vulnerabilities in earlier versions of TLS and introduces mechanisms that mitigate risks associated with data interception and manipulation. The research is instrumental in securing communication channels against sophisticated cyber threats in edge networks.

C. Xu et al. [5] conduct a comprehensive survey on regular expression matching techniques for deep packet inspection, highlighting their application in network security. Their findings reveal the efficiency and accuracy of different matching algorithms in detecting malicious activities. The study also explores hardware-accelerated solutions that improve the performance of real-time security systems, making them more suitable for mobile edge environments where low latency is critical. [6] C. Meyer and J. Schwenk provide a chronological analysis of SSL/TLS attacks, examining the weaknesses exploited in previous breaches. Their work is crucial in understanding the evolution of security threats and the necessity for continuous improvement in cryptographic protocols. The study emphasizes the role of proactive security measures in preventing similar attacks in mobile edge networks.

Y. Zhao et al. [7] investigate exception-triggered DoS attacks on wireless networks, revealing vulnerabilities that can disrupt edge computing operations. Their research discusses the impact of such attacks on network availability and proposes countermeasures to mitigate the risk. The findings highlight the importance of robust security frameworks capable of identifying and preventing denial-of-service threats in real-time. [8] J. Salowey et al. explore session resumption mechanisms in TLS to enhance security without maintaining server-side state. The study demonstrates how these mechanisms improve authentication efficiency while minimizing the risks associated with session hijacking. Their work contributes to the development of lightweight security solutions for mobile edge computing, where resource constraints are a major concern.

R. Hummen et al. [9] tailor end-to-end IP security protocols for the Internet of Things (IoT), addressing the challenges posed by limited processing capabilities in edge devices. The research introduces optimized security mechanisms that balance performance with protection, ensuring secure communication in IoT-driven edge networks. Their findings support the need for scalable security solutions that can adapt to diverse mobile edge computing environments. B. Anderson, S. Paul, and D. McGrew [10] analyze how malware utilizes TLS for encrypted communication without requiring decryption. Their research presents a method to identify malicious patterns in encrypted traffic, demonstrating the potential of machine learning in security applications. The study reinforces the importance of behavioral analysis techniques in detecting cyber threats in mobile edge environments.

3. PROPOSED SYSTEM

The proposed system aims to strengthen the detection of malicious activities in mobile edge computing environments by utilizing advanced machine learning techniques. It begins with the collection of a comprehensive dataset containing various types of malicious behavior relevant to mobile edge networks. This dataset forms the foundation for training and evaluating machine learning models and includes diverse attack vectors, vulnerabilities, and anomalies to ensure a robust detection system. Following this, the dataset undergoes preprocessing, where null values are removed and label encoding is applied to convert categorical variables into numerical form, ensuring the data is clean and structured for model training.

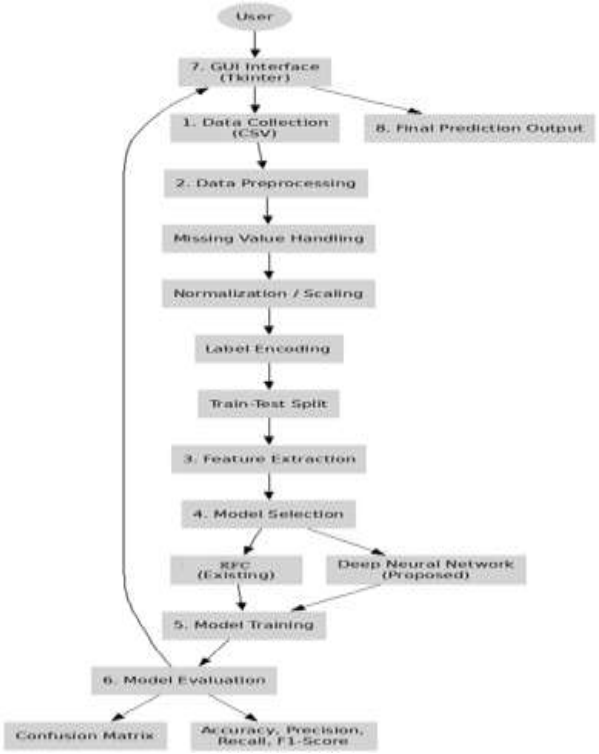

Fig.2: Architectural Block Diagram of the Project.

As a baseline model, the Random Forest Classifier is implemented due to its robustness and ability to handle high-dimensional data. This ensemble learning method constructs multiple decision trees using bootstrap sampling and random feature selection, aggregating their outputs to enhance prediction accuracy and reduce overfitting. Despite its strengths, Random Forest has some limitations, including computational intensity, high memory consumption, slower prediction times, and reduced interpretability compared to single decision trees.

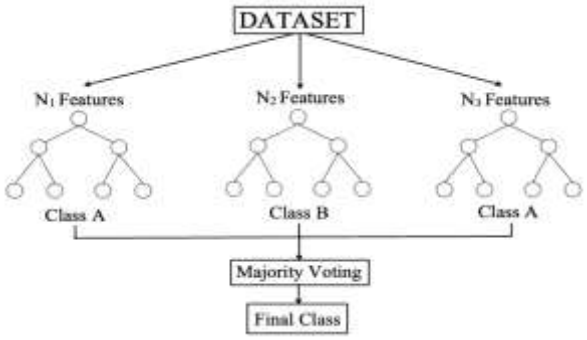

Fig.3: Architecture of Random Forest Classifier with Majority Voting Mechanism

To overcome these challenges and improve detection performance, a Deep Neural Network (DNN) is proposed as an advanced alternative. The DNN consists of an input layer, multiple hidden layers, and an output layer, with neurons in each layer performing weighted computations followed by non-linear activation functions such as ReLU. During training, the network adjusts weights using backpropagation and optimizes performance through techniques like Adam optimization and cross-entropy loss calculation.

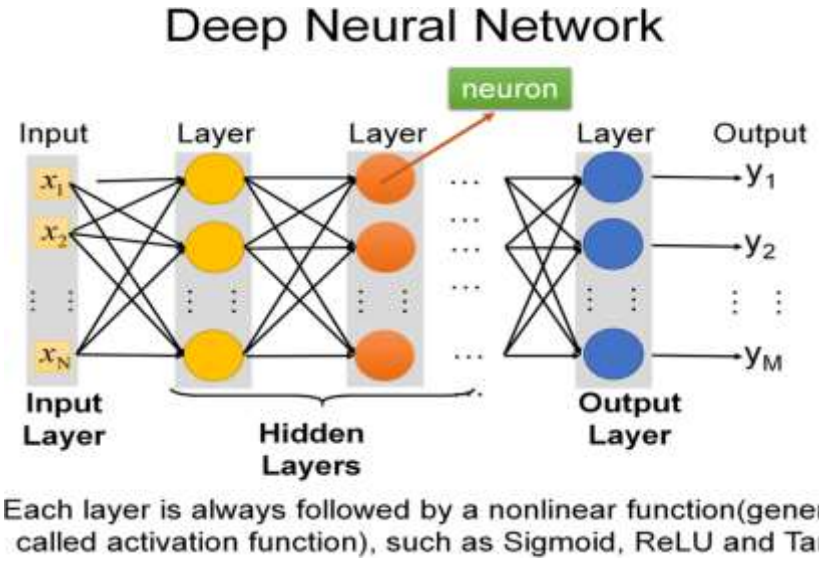

Fig4: Deep Neural Network Architecture

The system splits the dataset into training and testing sets, typically with an 80:20 ratio, and applies normalization to ensure consistent feature scaling. In model development, hyperparameters are tuned for both the Random Forest (e.g., number of trees, max depth) and the DNN (e.g., number of layers, neurons, learning rate). Throughout training, the models are monitored on a validation set to prevent overfitting. The performance of both models is compared using key evaluation metrics such as accuracy, precision, recall, and F1-score. This comparison provides insights into the superior capability of deep learning over traditional models in detecting malicious activity within mobile edge environments.

The DNN demonstrates significant advantages: it achieves higher classification accuracy, learns complex feature representations automatically, and scales well to large, high-dimensional datasets. It

also generalizes better across varying data distributions, supports flexible design through various activation and optimization functions, and maintains performance consistency. Its robustness to noise and adaptability to both binary and multi-class classification further enhances its utility. By capturing intricate patterns and minimizing false positives, the proposed DNN model offers a scalable, efficient, and intelligent solution for real-time malicious activity detection in mobile edge computing frameworks.

4. RESULTS AND DISCUSSION

9.1 Implementation and Description

The developed code provides a comprehensive GUI-based system for detecting malicious activities using machine learning and deep learning models, including Decision Tree, Random Forest, and Deep Neural Network (DNN). Built with Tkinter, the interface enables users to upload a dataset, preprocess it by handling missing values and encoding the target variable, and split it into training and testing sets. Core libraries such as Pandas, NumPy, Scikit-learn, TensorFlow/Keras, Seaborn, and Matplotlib support data handling, model training, and performance visualization. The system allows users to train and evaluate models individually, visualize results using ROC curves and bar charts, and perform predictions on new data using the trained Random Forest model. Preprocessing steps and model outputs are displayed in a text widget, and global variables manage data flow across components. The DNN model is optimized with dropout layers and activation functions, with support for loading saved weights to avoid retraining. The system supports real-time evaluation, visual analytics, and comparative performance analysis, offering an efficient, interactive solution for security analytics in mobile edge computing environments.

The dataset comprises a comprehensive list of Android API methods, permissions, actions, and classes, each associated with binary labels across multiple instances. These labels indicate the presence (1) or absence (0) of specific features within each instance. The dataset is structured to facilitate the analysis of these features in relation to Android applications, potentially serving as a foundation for machine learning models aimed at malware detection or permission analysis.

The dataset is composed of features that are specific to Android applications, including API methods (such as transact and onServiceConnected), permissions (like SEND_SMS and READ_PHONE_STATE), system actions (e.g., android. intent. action. BOOT COMPLETED), and Android classes (such as android.os. Binder and android. telephony. SmsManager). These features are essential for analyzing the behavior and functional capabilities of Android apps. Each instance in the dataset is encoded as a sequence of binary values, where a value of 1 indicates the presence and 0 indicates the absence of a particular feature. This binary representation allows for efficient processing and classification of instances based on the combination of features present in each application.

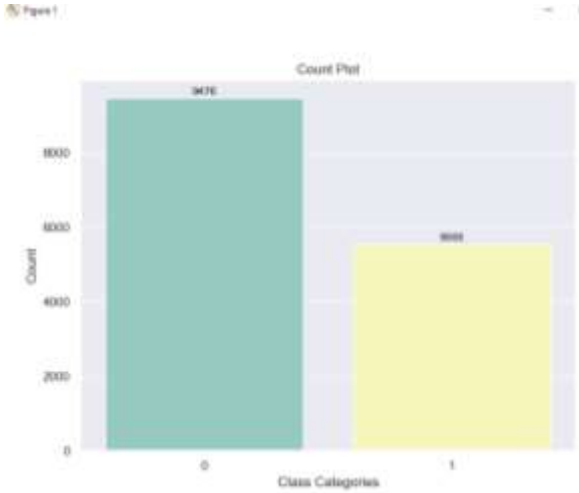

Fig.5: Count Plot of The Target Column.

Figure 5 shows count plot visualizes the distribution of two classes, revealing a significant class imbalance. This information is crucial for understanding the data and making informed decisions about further analysis or modelling. The class imbalance might reflect a real-world phenomenon. For example, if this data represents credit card transactions, it's natural to have many more legitimate transactions (Class 0) than fraudulent ones (Class 1).

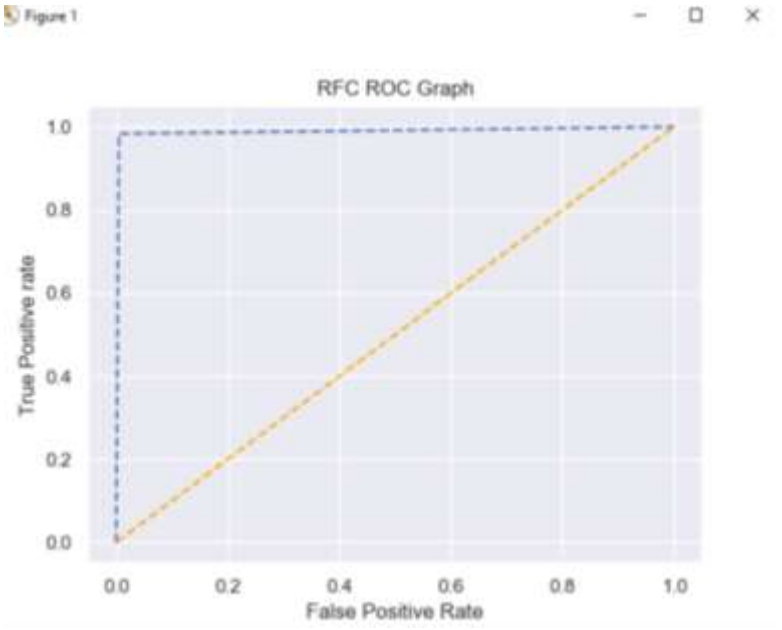

Fig.6: ROC curve of the Random Forest

Figure 6 Shows the ROC curve demonstrates the excellent performance of the Random Forest Classifier. The high AUC and the shape of the curve indicate a model with strong predictive power. However, it's crucial to consider potential overfitting and evaluate the model's performance on diverse datasets to ensure its real-world applicability.

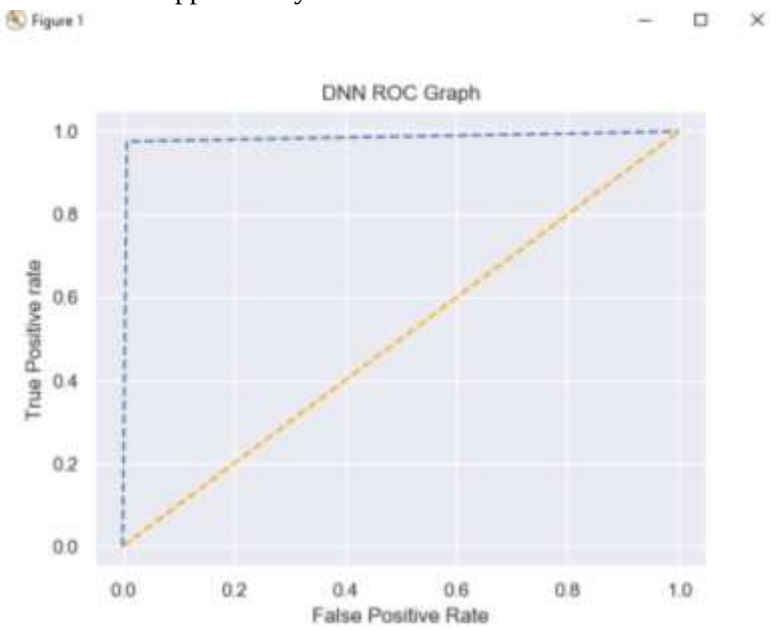

Fig.7: ROC curve of the Deep Neural Network.

Figure 7 shows the ROC curve illustrates the very strong performance of the Deep Neural Network model. The high AUC and the curve's shape indicate a model with excellent predictive capabilities. Further evaluation and validation are essential to confirm its robustness and generalizability.

Fig.8: Predicted output on test input.

Figure 8 shows the output of a security system classifying data as either "Secured" or "Attack Detected" based on binary data representing network traffic or system events. The system appears to be using pattern recognition to identify attacks, highlighting its potential usefulness in real-time security monitoring.

Metric	Random Forest (RF)	Decision Tree (DT)	Deep Neural Network (DNN)
Accuracy	99.17%	97.87%	99.67%
Precision	99.19%	97.53%	99.67%
Recall	99.00%	98.10%	99.43%
F1-Score	99.09%	97.68%	99.55%

Table 1: Performance metrics of all models.

The performance comparison table presents metrics for three classification models: Random Forest, Decision Tree, and Deep Neural Network. Among the three, the Deep Neural Network model achieved the highest overall accuracy (99.67%) with excellent precision, recall, and F1-score, indicating strong performance in both identifying and correctly classifying positive and negative cases. The Decision Tree model, while still accurate (97.87%), showed slightly lower precision and recall, particularly for the minority class. The Random Forest performed closely to DNN with balanced metrics and fewer misclassifications. Confusion matrices further illustrate that Random Forest made the least errors, reinforcing its superiority for this task.

5. CONCLUSION

The integration of deep learning (DL) into mobile edge computing (MEC) has significantly enhanced the security and efficiency of mobile networks. By leveraging the computational capabilities of edge devices, DL models can process data locally, thereby reducing latency and conserving bandwidth.

This localized processing is especially advantageous for real-time applications such as intrusion detection and threat analysis, where rapid responses are critical. Furthermore, the adaptability of DL algorithms allows them to learn from evolving attack patterns, offering robust defense mechanisms against sophisticated cyber threats. However, the deployment of DL in MEC environments presents several challenges. Edge devices often have limited computational resources, which can restrict the complexity of DL models that can be effectively implemented. Additionally, ensuring the privacy and security of data processed at the edge is vital, as these devices are vulnerable to various threats. Addressing these challenges requires ongoing research and the development of optimized DL models specifically designed for resource-constrained environments. Despite these obstacles, the potential benefits of integrating DL into MEC for enhanced security are substantial, paving the way for more resilient, intelligent, and efficient mobile networks.

REFERENCES

[1]. R. Braden, "Requirements for Internet hosts-communication layers," RFC 1122, Internet Engineering Task Force, Tech. Rep., Oct. 1989.

[2]. M. Korczy´nski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," In Proc of IEEE INFOCOM, Toronto, ON, Canada, pp. 781-789, 2014.

[3]. B. Anderson and D. Mcgrew, "Identifying encrypted malware traffic with contextual flow data," In Proc of the 2016 ACM Workshop on Artificial Intelligence and Security, New York, NY, USA, pp. 35-46, 2016.

[4]. E. Rescorla, "The transport layer security (TLS) protocol version 1.3," RFC 8446, Internet Engineering Task Force, Tech. Rep., Aug. 2018.

[5]. C. Xu, S. Chen, J. Su, S. M. Yiu, and L. C. K. Hui, "A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms," IEEE Communication Surveys and Tutorials, vol. 18, no. 4, 4th Quart., pp. 2991-3029, 201.

[6]. C. Meyer and J. Schwenk, "Lessons learned from previous SSL/TLS attacks: A brief chronology of attacks and weaknesses." IACR Cryptology ePrint Archive, vol. 2013, p. 49, 2013.

[7]. Y. Zhao, S. Vemuri, J. Chen, Y. Chen, H. Zhou, and Z. Fu, "Exception trig-gered DoS attacks on wireless networks," In Proc. of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 13–22, 2009.

[8]. J. Salowey, H. Zhou, P. Eronen, and H. Tschofenig, "Transport layer security (TLS) session resumption without server-side state," RFC 5077, Internet Engineering Task Force, Tech. Rep.

[9]. R. Hummen, H. Wirtz, J. H. Ziegeldorf, J. Hiller, and K. Wehrle. "Tailoring end-to-end IP security protocols to the internet of things," In Proc. Of 21st IEEE International Conference on Network Protocols (ICNP), pp. 1-10, 2013.

[10]. B. Anderson, S. Paul, and D. McGrew, "Deciphering malwares use of tls (without decryption)," J. Comput. Virol. Hacking Techn., vol. 14, no. 3, pp. 1–17, 2016.