International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 07 July-2026 | Impact Factor: 3.5

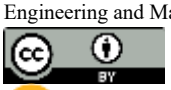
ML–Based Intrusion Detection System for Computer Networks

Pooja Verma[1]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
poojav074@gmail.com

Mr. Dhanna Ram[2]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
dr.mecs2009@gmail.com

Mr. Mohit kumar Saini[3]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
1saini1mohit1@gmail.com

How to Cite this Article:
Verma, P., Ram, D. & Saini, M. K. (2026). ML–Based Intrusion Detection System for Computer Networks. International Journal of Creative and Open Research in Engineering and Management, 02(7). https://doi.org/10.55041/ijcope.v2i7.028



https://doi.org/10.55041/ijcope.v2i7.028

Abstract**—***Modern computer networks face a persistent and growing range of cyber threats, including unauthorized access, malware infiltration, denial-of-service attacks, and data exfiltration. Conventional signature- and rule-based intrusion detection systems (IDS) struggle to identify zero-day attacks and evolving intrusion patterns as networks expand in scale and complexity, and they typically require frequent manual updates to remain effective. This paper presents a machine learning–based intrusion detection system capable of automatically analyzing network traffic and distinguishing malicious activity from legitimate behavior. By learning from historical traffic data, the proposed system adapts to emerging threats with greater accuracy than static, rule-driven approaches. The system is implemented in Python using established data-analytics and machine-learning libraries, and it evaluates several supervised classifiers—Random Forest, Support Vector Machine, Naïve Bayes, and Logistic Regression—to categorize network traffic as normal or intrusive. Experimental results show that the Random Forest classifier consistently achieves the strongest overall performance, and the integration of real-time packet capture with a visualization dashboard enables continuous, interpretable monitoring. The findings demonstrate that machine learning offers a scalable, adaptive, and largely automated foundation for securing contemporary network infrastructures.*

Index Terms—*Intrusion Detection System, Machine Learning, Network Security, Random Forest, Cybersecurity, Traffic Classification.*

I. INTRODUCTION

A. Background

Computer networks are the infrastructure of modern information technology, which helps to share resources and communicate in education, healthcare, banking, government and business. Cloud computing, mobile apps, and the Internet of Things (IoT) have made connectivity so prolific that business enterprises are running distributed systems and data is moving in and out of users, servers, and applications at all times. The expansion has thus introduced a lot of efficiency gains, but has also increased the range of attack for malicious actors, leading to cybersecurity as an organizational priority: the protection of networks, systems, and data from theft, disruption, and unauthorized access. With sensitive personal, financial and operational information being transmitted on networks, the risk of compromise can lead to severe financial damages, downtime and reputational loss.

B. Cyber Threat Landscape

There are many different threats to network environments. Malware (virus, worm, ransomware, spyware) spreads via infected downloads, email attachments or infected websites to steal information or disrupt systems. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks flood a system with so much traffic that legitimate users are unable to access it; distributed variants use many compromised hosts to coordinate the attacks making them more difficult to mitigate. Unauthorized access can be gained with weak credentials or improperly configured systems, allowing an attacker to view or alter protected systems, and data exfiltration transfers protected data without any knowledge or permission from the user. In addition to the technical attacks, phishing and spoofing attacks exploit human trust instead of software vulnerabilities.

C. Intrusion Detection Systems

Intrusion Detection System (IDS): These scan network or host activities for suspicious behavior, violation of the company's policy or unauthorized access, alerting the network administrator before significant damage is done. Network-based IDS (NIDS) look for suspicious activity or known attack signatures in network traffic while host-based IDS (HIDS) monitor activity on specific devices such as file-system changes, logs, and processes running on the host. Both possibilities are complementary and are often used simultaneously. An IDS (whether active or passive) has three main purposes: to continuously monitor traffic flow and system behavior, to detect unusual or known malicious patterns, and to alert to initiate a timely action.

D. Motivation and Problem Statement

While networks grow larger and faster, a number of limitations have made it difficult to effectively use the traditional IDS. Traffic from users, applications and cloud services is so great that it requires a lot of computational power, but many legacy systems are not yet capable of scaling to handle it and may not be able to perform full or complete analysis. Attacks can do damage in seconds, and IDS have to be real-time, which is challenging to reconcile with a high detection accuracy. In addition, cybercriminals continue to find fresh and new methods for evasion – zero-day exploits, polymorphic malware, and advanced persistent threats – which signature-based systems, based on the predetermined rules, cannot detect. Traditional IDS solutions also tend to produce a lot of false positives, making the administrator's job more difficult, and are not improved automatically based on historical experience [2, 10]. These restrictions have necessitated the

need for an intelligent adaptive detection mechanism that is able to learn from data instead of rule-based detection, which is the space where machine learning can fill the gap.

E. Machine Learning–Based IDS

A Machine Learning – Based Intrusion Detection System (ML – IDS) is a system that classifies network traffic based on learning algorithms and does not require a manually defined signature [10]. This usually involves a step-by-step process in data collection, data preprocessing, model training, and model detection. Traffic data is collected from traffic logs, packet captures or public benchmark data sets like NSL-KDD [11] or CICIDS2017 [12] and then preprocessed by removing the irrelevant features, normalizing the data and making it clean. Supervised algorithms (Decision Tree, Random Forest [13], Support Vector Machine [14] and K-Nearest Neighbors), used during training to learn to associate these features with normal or malicious labels, were able to generate a model that generalizes to unseen traffic, as demonstrated on the ensemble and hybrid approaches surveyed in [1]–[9]. During the detection stage, the trained model runs in real time to classify the incoming traffic and alerts or automatically responds to malicious traffic. Another drawback of static, rule-based IDS [2], [4], [5] is that the model can adapt itself to new attacks as new data comes in, so the model is data driven.

F. Objectives and Scope

The objectives of this study are to: (i) explore the main types of network intrusion and their typical behavior; (ii) design an intelligent, learning-based intrusion detection system; (iii) preprocess and analyze the network traffic data for proper model training; (iv) develop and compare classifiers for malicious activity detection and (v) perform the evaluation of the system with respect to standard intrusion-detection metrics aiming for better detection accuracy and fewer false alarms. The work is restricted to the design, training and evaluation of supervised machine-learning classifiers – Random Forest [13], Support Vector Machine [14], Naïve Bayes and Logistic Regression written in Python. In this study, the hardware-level implementation and integration with the commercial security platforms are excluded, and only a conceptual and practical framework is provided, as recommended in [2] and [7] that can be expanded to support enterprise, cloud, and IoT security scenarios.

II. RELATED WORK

There is a significant amount of recent research work on utilizing machine learning for ID in traditional, IoT, and cloud network contexts. An empirical study of six machine-learning models was performed by Alkadi et al. [1] on four benchmark IoT datasets (UNSW-NB15, BoT-IoT, ToN-IoT, and Edge-IIoT) and it is concluded that the detection accuracy can be improved significantly by paying attention to the preprocessing and hyperparameter tuning of the models, with MLP and clustering type models achieving up to 99.97% accuracy. Mohammad et al. [2] surveyed the general state of ML-based IDS research: they found that although systems are more accurate in their detection and more adaptable to other environments, there are still problems with interpretability, robustness to adversarial examples, and computational costs and they proposed explainable AI and privacy-preserving learning as potential future directions.

Specifically on IoT, Kikissagbe and Adda [3] reviewed the supervised, unsupervised, deep-learning and hybrid detection methods and found that machine learning can now be used to make meaningful contributions to the security of IoT, but more research is necessary to match the shifting threat landscape. In response to this transparency, Mohale and Obagbuwa [4] conducted a comparative study between several machine learning algorithms, Decision Trees, XGBoost, CatBoost and Random Forest, and explainable-AI algorithms, LIME, SHAP and ELI5, on the UNSW-NB15 dataset; they found that XGBoost and CatBoost

not only had the highest accuracy (87%) but also generated explainable feature attributions like time-to-live and destination service count.

Al Lail et al. [5] evaluated several ML algorithms on a modern attack set, and concluded that the best detection algorithm was the Random Forest, which detected 97% of modern attacks. On the ToN-IoT dataset, Hidayat et al. [6] trained classifiers, including Decision Tree, AdaBoost, K-Nearest Neighbor (KNN), MLP and LSTM deep-learning models, using the Pearson correlation coefficient in a hybrid feature-selection method, with the Decision Tree and MLP models achieving the best accuracy and the least number of false detections. The IDS proposed by Ajalkar et al. [7] was based on ML techniques and was implemented using 3 classifiers: Random Forest, Decision Tree, and SVM on the NSL-KDD dataset and CICIDS2017 dataset, with high accuracy, low false-positive rates, and good adaptability for use with enterprise, cloud, and IoT deployments.

To overcome this, Talukder et al. [8] presented a model using random oversampling, stacking-based feature embedding and PCA-based dimensionality reduction that successfully obtained up to 99.99% accuracy on UNSW-NB15, CIC-IDS-2017 and CIC-IDS-2018 datasets. Lastly, Kantharaju et al. [9] proposed a Self-Attention Progressive Generative Adversarial Network (SAPGAN) framework for IDS in IoT, which includes the missing-value recovery and War Strategy Optimization (WSO) for feature selection, and achieved significantly higher accuracy and lower computation time compared with the conventional deep-learning based IDS approaches. These studies are summarized in table I.

S. No	Author (Year)	Objective	Methodology	Key Findings
1	Alkadi et al. (2023)	Improve ML-based IDS for IoT networks	Six ML models evaluated across UNSW-NB15, BoT-IoT, ToN-IoT and Edge-IIoT with preprocessing and hyperparameter tuning	Accuracy up to 99.97%; MLP and clustering-based models performed best
2	Mohammad et al. (2024)	Assess ML-IDS effectiveness and open challenges	Conceptual review of ML-based detection approaches	High accuracy and adaptability; interpretability and data quality remain open issues
3	Kikissagbe&Adda (2024)	Survey ML techniques for IoT intrusion detection	Review of supervised, unsupervised, deep-learning and hybrid models	ML substantially strengthens IoT security but requires continued research
4	Mohale&Obagbuwa (2024)	Improve IDS interpretability	ML classifiers combined with XAI methods (LIME, SHAP, ELI5) on UNSW-NB15	XGBoost and CatBoost achieved 87% accuracy with improved transparency
5	Al Lail et al. (2023)	Develop an ML-based NIDS for modern attacks	Comparative evaluation of ML algorithms on a contemporary attack dataset	Random Forest achieved a 97% detection rate
6	Hidayat et al. (2022)	Detect intrusions using ML and DL	Hybrid feature selection with ML/DL models on the ToN-IoT dataset	Decision Tree and MLP delivered the best accuracy with low false rates

S. No	Author (Year)	Objective	Methodology	Key Findings
7	Ajalkar et al. (2025)	Design a scalable ML-based IDS	Random Forest, Decision Tree and SVM trained on NSL-KDD and CICIDS2017	High accuracy, low false positives, good adaptability across deployments
8	Talukder et al. (2024)	Handle large, imbalanced intrusion data	Oversampling, stacking feature embedding and PCA-based dimensionality reduction	Accuracy up to 99.99%, outperforming prior benchmarks
9	Kantharaju et al. (2024)	Improve efficiency of DL-based IDS	SAPGAN framework with optimized feature selection	Higher accuracy and markedly lower computation time than traditional models

TABLE I. Summary of Reviewed Literature

III. PROPOSED METHODOLOGY

A. System Design Approach

The proposed IDS is based on data-driven, learning-based rules instead of the rule-based paradigm used by existing IDS, and has been designed with the general principles of ML-IDS used in [10] and [2], [1]. The system gathers network traffic history, learns a pattern of normal and abnormal traffic and uses that pattern to classify new network traffic. The five principles that guide the design are: automation of the detection process, adaptability to previously unseen threats, real-time monitoring capability, minimization of false positives and scalability to large network environments. The entire system is developed in python with library functions for data analysis, machine learning, visualization and live packet capture.

B. Overall Methodological Workflow

Fig. 1 illustrates the end-to-end workflow of the proposed system, from raw traffic acquisition through model training and selection to real-time detection, attack classification, and administrator response. Each stage in the pipeline is described in the subsections that follow.

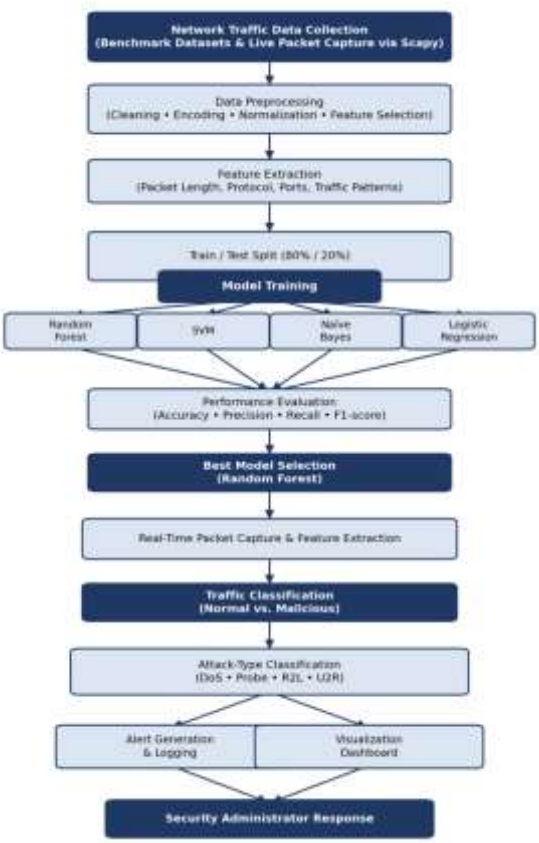


Fig. 1.Methodological workflow of the proposed machine learning–based intrusion detection system.

C. Data Collection

Network traffic data is obtained from publicly available intrusion-detection benchmark datasets—including NSL-KDD [11], a refined and widely adopted successor to the original KDD Cup 99 dataset, and CICIDS2017 [12], which offers more contemporary and diverse attack traffic—and is supplemented with real-time packet capture using Scapy. Collected data includes source and destination IP addresses, protocol type (TCP, UDP, ICMP), packet size and duration, and port and connection features, providing a representative mixture of normal and malicious traffic for model development, consistent with the dataset choices adopted in [7] and [8].

D. Preprocessing and Feature Extraction

As in pre-processing pipelines reported in [1] and [6] raw traffic data is generally noisy and is pre-processed before being used for training machine learning systems. Preprocessing involves handling missing, duplicate, or irrelevant records, transforming categorical data into numerical values (e.g., label encoding), normalizing numerical features (e.g., StandardScaler), and extracting relevant features for intrusion detection, such as packet length, protocol type, source ports, destination ports, and traffic frequency patterns. These steps directly enhance the accuracy of the models and decrease computational overhead when training the models.

E. Model Development and Training

Four supervised classification algorithms, namely Random Forest [13], Support Vector Machine [14], Naïve Bayes, and Logistic Regression are implemented and compared. The Random Forest is an ensemble of decision trees, proposed by Breiman [13] and it has been widely used to achieve better performance than one estimator baseline in ML-IDS research [1], [5], [8] and is therefore included. The data is split into training

data (80%) and test data (20%) and the models are trained on the labeled training data and then evaluated on the unlabeled test data to measure their generalization. The performance of the models are compared with regard to the standard classification metrics and the model that has the best performance is deployed in the real-time detection module.

F. Real-Time Detection and Attack Classification

After the training, the chosen model is used for live intrusion detection. The system captures network packets continuously with Scapy, extracts network features for every packet and passes the network features to the trained classifier for classification of network packets as normal or malicious. In case of detection of intrusion, further classification is performed in terms of four basic types: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R) attacks, similar to attack classifications presented in [7] and [11], to help the administrator decide on the attack type and respond accordingly.

G. Alerting, Visualization, and Evaluation

The system sends an alert with the type of the attack, its source ip address, packet details and the time of detection to be logged for later review, the alert is generated whenever detection of suspicious activity is detected. The graphical dashboard enhances this logging with real-time intrusion graphs, packet-rates, protocol distributions, attack-type counts and traffic heatmaps, facilitating quick analysis and decision making. The accuracy, precision, recall, and F1-score are used to evaluate the system, similar to those used in the literature reviewed [1], [4], [5] and [8] and the system supports periodic retraining for the system to adapt as new attack data is introduced. The overall implementation utilizes Python and the PANDAS, NUMPY, SCAPY, MATPLOTLIB, and SEABORN libraries for data handling and analysis, SCAPY for packet capture, TKINTER for the graphical interface, and JOBLIB for model persistence.

IV. RESULTS AND DISCUSSION

Random Forest, Support Vector Machine, Naïve Bayes, and Logistic Regression classifiers were used to train and test the proposed system, and evaluation metrics, such as accuracy, precision, recall, and F1-score, were used to compare the performance of the classifiers. The next figures provide a summary of the traffic characteristics observed during the evaluation process, and a comparison of the traffic models' performance.

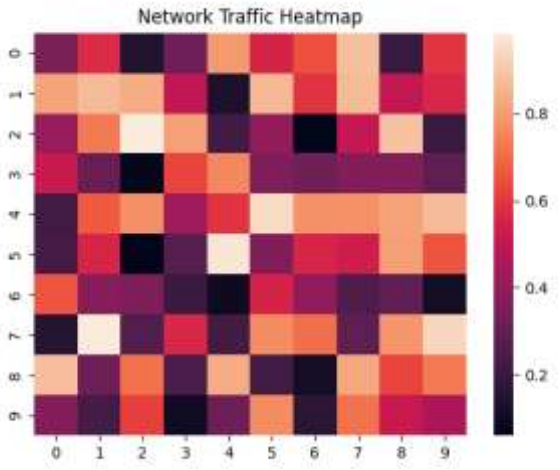


Fig. 2. Network traffic heatmap showing activity intensity across monitored connections; darker regions indicate clusters of unusual behavior and potential anomalies.

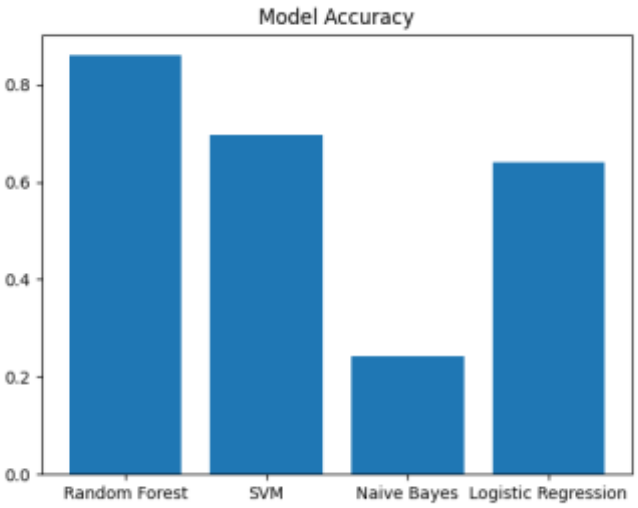


Fig. 3.Comparative accuracy of the Random Forest, SVM, Naïve Bayes, and Logistic Regression classifiers.

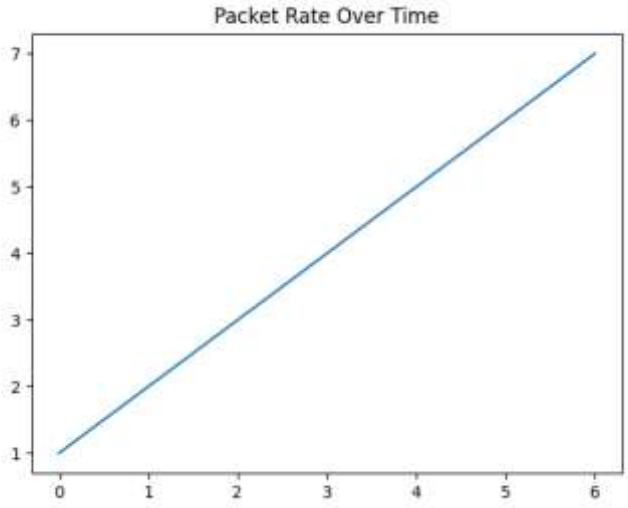


Fig. 4. Packet rate over time, highlighting traffic spikes indicative of congestion or possible denial-of-service activity.

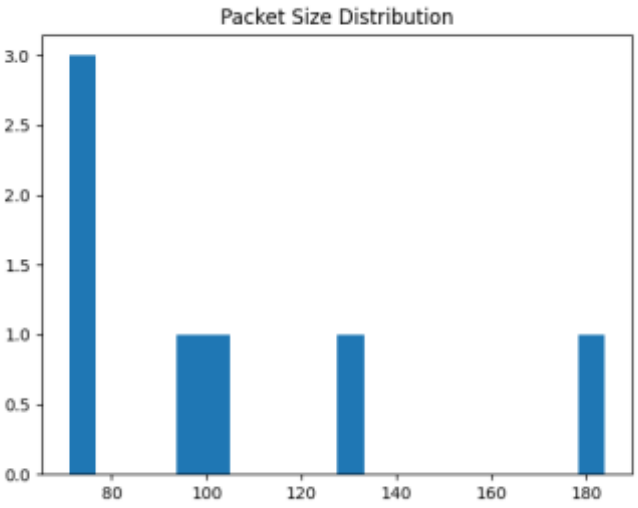


Fig. 5. Distribution of observed packet sizes; irregular distributions can indicate abnormal or malicious traffic.

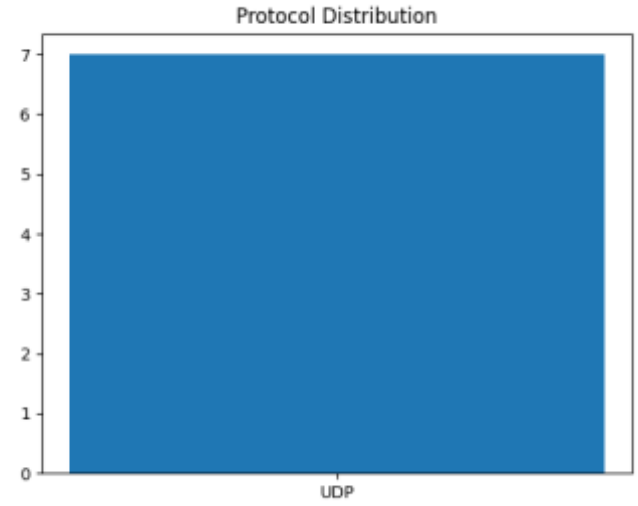


Fig. 6. Distribution of network protocols (TCP, UDP, ICMP) observed during monitoring.

The results obtained from the Random Forest classifier are always the highest and the model gives the best overall balance of precision and recall, which is consistent with results reported on similar intrusion detection tasks where Random Forest or its ensemble variants are concluded as the best models. The result is in line with the ensemble structure of Random Forest that combines predictions of multiple decision trees to boost accuracy and avoid overfitting, thus random forest can capture the complex and non-linear relationship of traffic data better than the single-estimator baselines.

The system has been found working effectively in real time apart from the offline model comparison. This was done continuously, as live packets were captured and features such as packet length, protocol type and port numbers extracted, classified as they were received, and the malicious packets flagged. This is also important because cyber threats can happen within seconds and this responsiveness is crucial to minimising potential harm in operational network environments.

The visualization dashboard added additional clarity by providing a packet rate trend, packet size histogram, protocol distribution and attack frequency summary in addition to the traffic heatmap. The usefulness of sudden packet rate increases to catch possible DoS activity and of abnormal protocol distributions to highlight possible suspicious communication patterns turned out to be helpful in making the system's output more accessible for security personnel with no particular data-science training, a goal that is strongly related to making AI systems explainable.

The classification of intrusions to one of four categories (DoS, Probe, R2L, U2R), along with the detailed logging of source IP addresses and when the intrusions were detected, aids incident response and in the future, analysis of repeat attackers. There are a few drawbacks, however. The accuracy of detection is also dependent on the quality and variety of the training set and can be affected by a lack of detection of attack types that are not represented or present in the training set, an obstacle to the wider adoption of ML-IDS. The feature set for the real-time detection is relatively straightforward and using deeper packet inspection or temporal traffic patterns can help make it more accurate. Although false positives are decreased compared to a rules-based IDS, they are not eliminated, and real-time processing may lead to computation overhead in high throughput networks, requiring additional optimization before enterprise scale deployment.

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This study presented the design and implementation of a machine learning–based intrusion detection system that classifies network traffic as normal or malicious with high accuracy. The results showed that the robustness of Random Forest was reflected in its superior performance, which is in agreement with previous studies on ensemble-based intrusion detection which mainly used complex, high-dimensional network data. Real-time packet monitoring enabled the system to continuously capture, process and classify the traffic; an additional visualization dashboard allowed the detection patterns and attack types to be interpreted. Overall, the above results suggest that the proposed system is an affordable, scalable and adaptable solution to the conventional rule-based IDS, which has a tendency to require pre-defined rules instead of learning from data, and can still be improved by adding more variety in data sets and reducing false-positives.

B. Future Scope

The present system can be expanded in various ways and directions in the future. Using deep-learning architectures like Artificial Neural Networks, Convolutional Neural Networks and Long Short-Term Memory networks could enhance the detection of sophisticated and zero-day attacks, which could capture more complex temporal and structural patterns in traffic data, as proposed by Hidayat et al. [6] and Kantharaju et al. [9] respectively, and hybrid and ensemble techniques could enhance accuracy and reduce false positive detections. Larger and more diverse real-world data sets, along with larger data sets and more sophisticated feature engineering, such as using flow-based, time-series, and deep-packet-inspection information, would likely help generalize to new types of attacks, and feature selection, dimensionality reduction, and data-balancing techniques would be helpful in making the algorithms more efficient at scale. Lastly, extending the system to cloud and IoT environments, extend the system to become an Intrusion Prevention System (IPS) with automated response actions as in Mohale and Obagbuwa [4], and add Explainable AI (XAI) to increase the trust of an analyst on the model decisions, are promising pathways for moving the research prototype to real practical, large-scale deployments.

REFERENCES

[1] S. Alkadi, S. Al-Ahmadi, and M. M. Ben Ismail, "Toward improved machine learning-based intrusion detection for internet of things traffic," Computers, vol. 12, no. 8, p. 148, 2023, doi: 10.3390/computers12080148.

[2] S. Mohammad, V. Vimal, and A. Sahu, "Enhancing network security through machine learning based intrusion detection systems," Int. J. Intell. Syst. Appl. Eng. (IJISAE), vol. 12, no. 21s, pp. 1117–1125, 2024.

[3] B. R. Kikissagbe and M. Adda, "Machine learning-based intrusion detection methods in IoT systems: A comprehensive review," Electronics, vol. 13, p. 3601, 2024, doi: 10.3390/electronics13183601.

[4] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability," 2024.

[5] M. Al Lail, A. Garcia, and S. Olivo, "Machine learning for network intrusion detection—A comparative study," Future Internet, vol. 15, p. 243, 2023, doi: 10.3390/fi15070243.

[6] I. Hidayat, M. Z. Ali, and Arshad, "Machine learning-based intrusion detection system: An experimental comparison," J. Comput. Cogn. Eng., vol. 00, no. 00, pp. 1–10, 2022, doi: 10.47852/bonviewJCCE2202270.

[7] D. Ajalkar, V. Chavan, and P. Bhosle, "Machine learning based intrusion detection system," Int. J. Adv. Res. Sci. Commun. Technol., vol. 5, no. 10, Apr. 2025, doi: 10.48175/IJARSCT-25679.

[8] M. A. Talukder, M. M. Islam, and M. A. Uddin, "Machine learning based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," J. Big Data, vol. 11, p. 33, 2024, doi: 10.1186/s40537-024-00886-w.

[9] V. Kantharaju, H. Suresh, and M. Niranjanamurthy, "Machine learning based intrusion detection framework for detecting security attacks in internet of things," Sci. Rep., vol. 14, p. 30275, 2024, doi: 10.1038/s41598-024-81535-3.

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv.Tutor., vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[11] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. 2nd IEEE Symp. Comput.Intell.Secur. Defense Appl. (CISDA), Ottawa, ON, Canada, 2009, pp. 1–6.

[12] I. Sharafaldin, A. HabibiLashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[13] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[14] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.