Network Intrusion Identification Using Machine Learning

Mrs.S.Sasikala M. E (Ph. D)[1], Ms.A. priya[2], Ms.J. Sindhuja[3], Ms. R. Sujeetha[4]
[1]*Assistant Professor, Department of CSE, R P Sarathy Institute of Technology, Salem, India*
[2,3,4]*Department of CSE, R P Sarathy Institute of Technology, Salem, India*

***Abstract*—With the rapid growth of internet usage and digital communication, network security has become an essential concern for organizations and individuals. Cyber threats such as unauthorized access, denial-of-service attacks, and malicious activities can compromise sensitive data and disrupt network operations. Traditional security mechanisms are often unable to detect complex and evolving attacks effectively. This research proposes a Network Intrusion Detection System (NIDS) based on Machine Learning techniques to identify abnormal network behavior. The model utilizes the Random Forest algorithm to classify network traffic as either normal or malicious. The system is trained and evaluated using the NSL-KDD dataset, which contains labeled records representing both normal activities and different attack categories. The experimental results demonstrate that the proposed approach improves detection accuracy and enhances the reliability of intrusion detection systems. This work highlights the potential of machine learning methods in strengthening modern network security solutions.**

***Index Terms*—Machine Learning, Network Intrusion Detection System (IDS), Network Security, Random Forest Algorithm, NSL- KDD Dataset, Real-Time Traffic Monitoring, Scapy, Flask Web Application.**

I. INTRODUCTION

The widespread adoption of computer networks and online services has significantly increased the risk of cyber threats. Organizations rely heavily on digital infrastructures for communication, data storage, and business operations. As a result, protecting networks from malicious activities has become a major challenge.

Intrusion Detection Systems (IDS) are designed to monitor network traffic and identify suspicious behavior that may indicate a cyberattack. Conventional IDS techniques often depend on predefined rules or signatures, which makes them less effective when encountering new or unknown attacks. Machine Learning offers a more adaptive solution by enabling systems to learn patterns from historical data and detect anomalies automatically. By analyzing large volumes of network traffic data, machine learning models can distinguish between normal and abnormal behavior with improved accuracy.

In this study, a machine learning-based intrusion detection model is developed using the Random Forest algorithm. The system is trained with the NSL-KDD dataset, which is widely used in cybersecurity research. The objective of this work is to build an efficient model capable of identifying different categories of network attacks while maintaining high detection performance.

II. LITERATURE SURVEY

2.1. Ensemble-IDS: An Ensemble Learning Framework for Enhancing AI-Based Network Intrusion Detection Tasks

Osvaldo Arreche, Ismail Bibers2025, Modern cybersecurity threats continue to evolve in both complexity and prevalence, demanding advanced solutions for intrusion detection. Traditional AI-based detection systems face significant challenges in model selection, as performance varies considerably across different network environments and attack scenarios. To overcome these limitations, we propose a comprehensive ensemble learning approach that systematically integrates feature selection, model optimization, and rigorous evaluation components. Our framework evaluates fourteen distinct machine learning approaches, ranging from individual classifiers to sophisticated ensemble methods including bagging, boosting, and hybrid stacking/blending architectures. These techniques are applied to multiple base algorithms such as neural networks and tree- based models. Extensive testing was conducted on two complementary benchmark

datasets (RoEduNet-SIMARGL2021 and CICIDS-2017) to assess detection capabilities across varied threat landscapes. Our experimental results revealed several key findings. Ensemble techniques universally surpass standalone models in detection accuracy, with random forest achieving the best performance on RoEduNet-SIMARGL2021, while the blending and bagging methods approach yielded perfect scores (F1 > 0.996) on CICIDS-2017. Feature selection via information gain demonstrated particular value, reducing model training times by 94% while maintaining detection accuracy. Among ensemble methods, XGBoost showed exceptional computational efficiency, whereas stacking and blending architectures delivered maximum accuracy at the expense of greater resource requirements. This research provides practical guidance for security professionals in model selection based on specific operational constraints and threat profiles. To support community advancement, we have made our complete framework publicly available, facilitating reproducibility and future innovation in intrusion detection systems.

2.2. A novel ensemble learning-based model for network intrusion detection

Ngamba Thockchom, Moirangthem Marjit Singh 2023 The growth of Internet and the services provided by it has been growing exponentially in the past few decades. With such growth, there is also an ever-increasing threat to the security of networks. Several efficient countermeasures have been placed to deal with these threats in the network, such as the intrusion detection system (IDS).

This paper proposes an ensemble learning- based method for building an intrusion detection model. The model proposed in this paper has relatively better overall performance than its individual classifiers. This ensemble model is constructed using lightweight machine learning models, i.e., Gaussian naive Bayes, logistic regression and decision tree as the base classifier and stochastic gradient descent as the meta-classifier. The performance of this proposed model and the individual classifiers used to build the ensemble model is trained and evaluated using three datasets, namely, KDD Cup 1999, UNSW-NB15 and CIC-IDS2017.

The performance is evaluated for binary class as well as multiclass classifications. The proposed method also incorporates the usage of a feature selection method called Chi-square test to select only the most relevant features. The empirical results definitively prove that using an ensemble classifier can be immensely helpful in the field of intrusion detection system with unbalanced datasets where misclassifications can be costly.

2.3. Ensemble Learning-Based Intrusion Detection and Classification for Securing IoT Networks: An Optimized Strategy for Threat Detection and Prevention

Kumaresh Sheelavant1, Charan K. V. 2, B. Yamini Supriya3, Purshottam J. Assudani4, Chandra Bhushan Mahato5, Sanjay Kumar Suman6, *2023 The rapid proliferation of real- time Internet of Things (IoT) devices have increased the need for efficient and accurate security mechanisms. Real-time IoT devices are highly vulnerable to cyberattacks due to their continuous connectivity and limited security mechanisms. In this work, we propose Light Ensemble-Guard, an ensemble learning- based approach specifically designed for resource-constrained IoT environments. Our method achieves a high detection accuracy of 99.55%, with strong precision, recall, and F1- score, while maintaining a low computational cost of just 22.23 seconds. To ensure robustness, we conducted 5-fold cross-validation and ROC curve evaluations, confirming the model's reliability and generalizability. Light Ensemble-Guard integrates three lightweight classifiers LightGBM, XGBoost, and Extra Trees using a majority voting mechanism to improve detection performance on highly imbalanced datasets without burdening system resources. This ensemble strategy ensures an optimal balance between detection performance and computational resource utilization, making it well-suited for IoT networks, where processing power and memory are limited. The results highlight Light Ensemble-Guard as an effective, scalable and lightweight solution for real-time IoT security, significantly outperforming traditional models in both accuracy and computational efficiency.

III. PROPOSED METHODOLOGY

The proposed system is designed to detect malicious network activities using a machine learning-based intrusion detection approach. The methodology

consists of several stages including dataset collection, data preprocessing, feature extraction, model training using the Random Forest algorithm, real-time packet monitoring, and intrusion detection. Each stage contributes to improving the accuracy and efficiency of detecting network attacks.

1. Dataset Collection

The first step in the proposed methodology is obtaining a suitable dataset for training the machine learning model. In this project, the NSL-KDD dataset is used because it is a widely accepted benchmark dataset for intrusion detection research. The dataset contains different types of network traffic records labeled as either normal or malicious. These records include several features such as protocol type, service type, flag status, source bytes, and destination bytes, which help in identifying abnormal behavior in network traffic.

2. Data Preprocessing

Before training the machine learning model, the dataset must be cleaned and prepared. Data preprocessing involves removing redundant data, handling missing values, and converting

categorical features into numerical format. Encoding techniques are applied to convert attributes such as protocol type and service type into machine-readable values. Additionally, normalization may be applied to scale the numerical features to improve model performance.

3. Feature Extraction and Selection

Feature extraction is performed to identify the most relevant attributes from the dataset that contribute to intrusion detection. Important network features such as protocol type, service, flag status, source bytes, and destination bytes are selected. Feature selection helps reduce data dimensionality and improves the efficiency of the machine learning model while maintaining high detection accuracy.

4. Model Training Using Random Forest

After preprocessing and feature extraction, the processed dataset is used to train the machine learning model. The proposed system uses the Random Forest algorithm, which is an ensemble learning technique that builds multiple decision trees during the training phase. Each tree is trained on a randomly selected subset of the dataset using bootstrap sampling.

During training, each decision tree learns patterns that differentiate normal network traffic from attack traffic. Random feature selection during tree construction helps improve model diversity and reduces overfitting. The final classification result is obtained using majority voting among all decision trees in the forest.

5. Real-Time Packet Capture

Once the model is trained, the system captures real-time network packets using a packet sniffing tool such as Scapy. The captured packets contain information about network communication such as protocol type, source and destination addresses, and packet size. These packets are processed and converted into feature vectors similar to those used during the training phase.

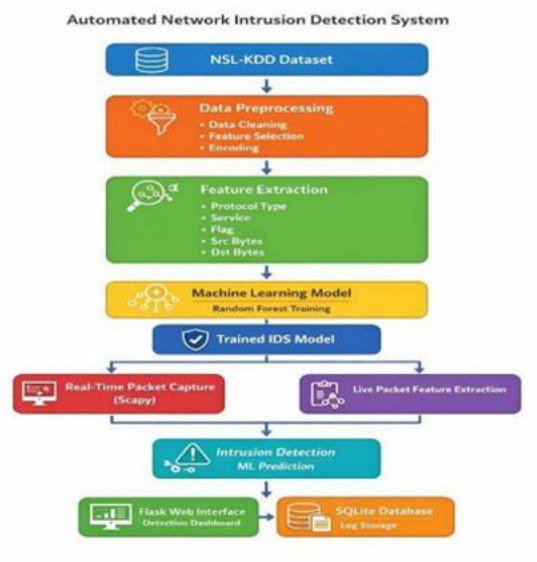
6. Intrusion Detection Using the Trained Model

The extracted features from real-time packets are provided as input to the trained Random Forest model. The model analyzes the features

and predicts whether the network traffic is normal or represents a potential attack. If the majority of decision trees classify the traffic as malicious, the system identifies it as an intrusion.

7. Result Visualization and Log Storage

The detection results are displayed through a web-based dashboard developed using Flask. This interface allows users to monitor network traffic and intrusion alerts in real time. Additionally, all detected events are stored in an SQLite database for logging and further analysis.



IV. DATASET DESCRIPTION

1. NSL-KDD

KDD'99 is outdated and contains redundant records, resulting in network intrusion detection inaccuracy. The problem is solved in NSL-KDD, which is a developed version of KDD'99. The training set of NSL-KDD has 125973 data points, whereas the testing set contains 22544 data points. It features 41 variables with numeric, binary, and nominal data types, as well as a label. Dos, probe, r2l, u2r, and regular class are the four major groups of attack types in the dataset. The distribution of each assault in training and testing sets is shown.

Dataset	Class	Train-set	Test-set
NSL-KDD	normal	67 343	9,711
	dos	45 927	7 458
	probe	11 656	2 421
	r2l	995	2 754
	u2r	52	200
	Total	125 973	22 544

2. Random Forest Algorithm

Random Forest is an ensemble machine learning algorithm that constructs multiple decision trees during training and outputs the final prediction using majority voting. It improves prediction accuracy and reduces overfitting compared to a single decision tree.

In this project, Random Forest is used to classify network traffic as normal traffic or intrusion attack based on network features.

Step 1: Bootstrap Sampling

Random Forest first generates multiple training datasets using bootstrap sampling from the original dataset.If the dataset contains N samples, multiple subsets are created by randomly selecting samples with replacement.

$D = \{x_1, x_2, x_3, \ldots, x_n\}$

Each subset is used to construct an independent decision tree.

Step 2: Feature Selection

At each node of the decision tree, a random subset of features is selected instead of using all features.If the total number of features is m, then a subset k feature is randomly selected:

$$k = \sqrt{m}$$

This randomness increases diversity among trees and improves model performance.

Step 3: Gini Impurity Calculation

To determine the best split at each node, the Gini Impurity measure is used.The Gini impurity is calculated as:

$$Gini = 1 - \sum_{i=1}^{c} (pi)^2$$

Where:

- p_i= probability of class i
- c= number of classes

Step 4: Tree Construction

Multiple decision trees are constructed independently using different bootstrap datasets.Each tree produces a classification output.

Step 5: Majority Voting

The final prediction is obtained using majority voting.

$Prediction = Mode\ (T_1, T_2, T_3, \ldots, T_n)$

Step 6: Accuracy Evaluation

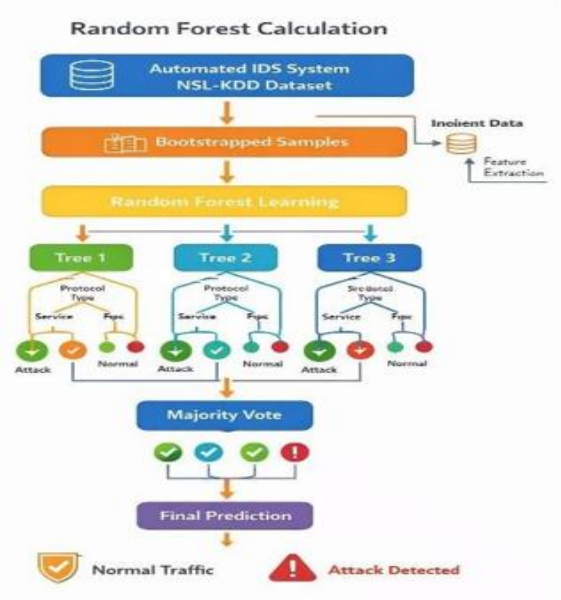
The performance of the Random Forest model is evaluated using accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

Example: $Accuracy = \frac{92}{100} \times 100 = 92\%$



V. RESULT AND ANALYSICS

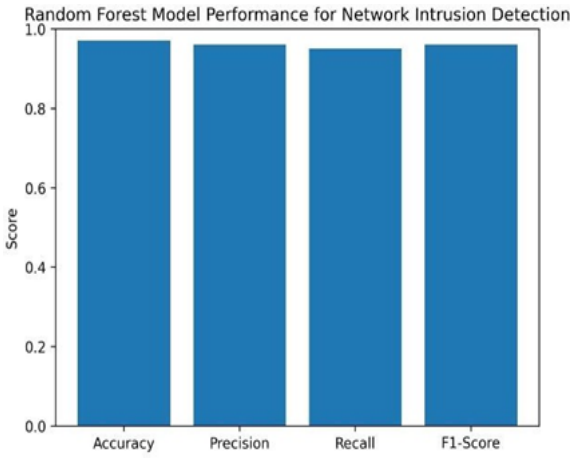
The performance of the proposed Network Intrusion Identification System was evaluated using the NSL-KDD dataset. The Random Forest machine learning algorithm was trained using selected network traffic features such as protocol type, service type, flag status, source bytes, and destination bytes.

After training the model, the system was tested using unseen network traffic data to measure its detection capability. The performance of the model was evaluated using standard machine learning evaluation metrics including accuracy, precision, recall, and F1-score.

The experimental results show that the Random Forest model provides high detection accuracy and effectively distinguishes between normal network traffic and malicious activities. The model achieved an overall accuracy of approximately 97%, demonstrating its capability to detect intrusion attempts with minimal error.

Precision indicates how many of the detected intrusions are actually correct, while recall measures the ability of the model to detect all possible attacks. The results show that the proposed system maintains a balanced performance between precision and recall, which is essential for reliable intrusion detection.

The graphical representation above illustrates the performance metrics of the trained model. The high values across all metrics indicate that the Random Forest algorithm performs efficiently for network intrusion detection tasks.



VI. DISCUSSION

The results obtained from the experiment demonstrate that machine learning techniques can significantly improve the effectiveness of intrusion detection systems. The Random Forest algorithm performs well because it combines multiple decision trees, which reduces overfitting and improves prediction accuracy. Compared to traditional rule-based intrusion detection systems, the proposed machine learning-based approach can automatically learn patterns from network traffic data and adapt to different types of attacks. The ensemble nature of Random Forest allows the model to capture complex relationships between network features.

Another advantage of the proposed system is its ability to analyze real-time network packets. By integrating packet capturing techniques with the trained model, the system can detect suspicious activities dynamically. This makes the approach suitable for modern network environments where cyber threats are constantly evolving.

However, the system may require additional computational resources when processing very large volumes of network traffic. Therefore, further optimization may be required to improve real-time performance in large-scale networks.

VII. FUTURE WORK

Although the proposed intrusion detection system achieves high accuracy, several improvements can be considered for future research.

First, the system can be extended by incorporating deep learning techniques such as neural networks or convolutional neural networks, which may further improve detection performance for complex attack patterns.

Second, the dataset used for training can be expanded by including more recent and diverse network traffic datasets. This will help the model learn new attack behaviors and improve its generalization capability.

Third, the system can be enhanced with real- time alert mechanisms, such as email notifications or security dashboards, to provide immediate alerts when an intrusion is detected.

Finally, integrating the system with cloud- based network monitoring platforms can improve scalability and allow the intrusion detection model to operate

efficiently in large distributed network environments.

VIII. CONCLUSION

NID is designed to provide basic acquisition strategies to protect existing systems on networks directly or indirectly online. But finally at the end of the day to the Network Administrator to make sure his network is out of danger. Many different methods have been used in the screening process. Among them machine learning plays a vital role.

This analysis works with machine learning algorithms such as KNN, DTC and Naïve Bayes. This does not completely protect the network from attackers, but IDS helps the Network Administrator track down the bad guys on the internet whose purpose is to bring your network into a hotspot and make it vulnerable to attack.

REFERENCE

[1] K. Wang, A. Zhang, H. Sun, and B. Wang, "Analysis of recent deep learning-based intrusion detection methods for in-vehicle network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1843–1854, Feb. 2023.

[2] Prasanth and S. Jayachitra, "A novel multi-objective optimization strategy for enhancing quality of service in IoT-enabled WSN applications," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 1905–1920, 2020.

[3] Smith, B. Johnson, and C. Lee, "A machine learning approach for cyber threat detection in IoT environments," *Journal of Network and Computer Applications*, vol. 175, p. 102900, Jan. 2023.

[4] Kalaiselvi and G. M. Nasira, "A new approach for diagnosis of diabetes and prediction of cancer using ANFIS," in *Proc. World Congress on Computing and Communication Technologies (WCCCT)*, 2014, pp. 188–190.

[5] M. Thiruvengadam *et al*., "Bioactive compounds in oxidative stress-mediated diseases: Targeting the Nrf2/ARE signaling pathway and epigenetic regulation," *Antioxidants*, vol. 10, no. 12, pp. 1–12, 2021.