International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

An Adaptive Hybrid Forest Framework for Real-Time Intrusion Detection

R.Dhivya[#1], R.B.Rajashre Rajesware[#2], R.Premalatha[#3] , J.Prasanth[#4] , S.Ragulgandhi[#5]
[#]*Computer Science and Engineering, Muthayammal Engineering College*
[1]dhivyamec24@gmail.com
[2]rajashreravikumar@gmail.com
[3]divyathilaga1603@gmail.com
[4]*prasanthjagathesh@gmail.com*
[5]ragulsragul2@gmail.com

***Abstract*— In the face of increasingly complex cyber threats, the necessity for robust Network Intrusion Detection Systems (NIDS) has never been greater. Conventional rule based systems often struggle to keep pace with evolving attack methodologies , necessitating the integration of machine learning techniques to bolster detection capabilities. Network attacks like probing, Denial of Service, R2L, and U2R affect countless systems daily. Implementing an NIDS that uses machine learning algorithms enhances the ability to detect and respond to these threats effectively. Machine Learning algorithms to prevent this attack. In the Preprocessing techniques include data cleaning and normalization which help in managing imbalanced datasets and improving the accuracy of detection algorithms. The feature selection is based on CFS-BA is used to determine a subset of the original features to eliminate irrelevant features and dimensionality reduction, which selects the optimal subset based on the correlation between features. In order to increase the detection ability of IDS and prevent the service providers from attack, propose an efficient ML based IDS using Light gradient boosting method and Random Forest algorithms. Which is used to classify anomalies, and identify patterns in complex network traffic data.**

***Keywords*— *Network Intrusion Detection System ,(NIDS), Machine Learning, Random Forest, Cybersecurity, Real-Time Detection, Data* Preprocessing, Anomaly Detection, Threat Detection**

I. INTRODUCTION

Intrusion detection in cybersecurity refers to the process of identifying unauthorized access, misuse, or malicious activity in a computer system, network, or application. The goal of intrusion detection is to detect potential threats or attacks as early as possible to minimize damage, data loss, or security breaches. The Intrusion detection System can accelerate and automate network threat detection by alerting security administrators to known or potential threats , or by sending alerts to a centralized security tool. A centralized security tool contains security information and event management system that can helps to combine data from other sources to help security teams to identify and respond to cyberattacks that might slip by other security measures However, traditional IDS methods often suffer from high false alarm rates and limited detection accuracy. An IDS uses two primary threat detection methods : signature-based or anomaly-based detection. A signature-based IDS maintains a database of attack signatures against which it compares network packets. If a packet triggers a match to one of the signatures, the IDS flags it. To be effective, signature databases must be regularly updated with new threat intelligence as new cyberattacks emerge and existing attacks evolve. Brand new attacks that are not yet analysed for signatures can evade signature-based IDS. Anomaly-based detection methods use machine learning to create and continually refine a baseline model of normal network activity. Then it compares network activity to the model and flags deviations such as process that uses more bandwidth than normal, or a device opening a port. Network Intrusion Detection systems provide continuous network monitoring across on-premise and cloud infrastructure to detect malicious activity like policy violations, lateral movement or data exfiltration.

Wireless networks[1] are inherently more vulnerable to cyber threats due to their open nature and dynamic behaviour. The deep learning techniques to improve the detection accuracy and minimize false positive rates in real- time environments, ensuring enhanced network protection. The framework leverages filter-based [2] feature selection methods to identify and retain the most relevant features from network traffic data while eliminating redundant and irrelevant information. A method to transform NIDS datasets into two-dimensional (2D) images[3] using various image transformation techniques, subsequently integrating these into three-channel RGB color images. This transformation enables the application of sophisticated image classification models, such as Convolutional Neural Networks (CNNs), to the intrusion detection domain. The unique challenges of CNN-based[4] communication systems, where real-time detection and low-latency response are critical for vehicle safety and performance. The proposed model leverages deep learning architectures, including Convolutional Neural Networks (CNN) and Attention-based Gated Recurrent Units (GRU), to enhance detection accuracy and reduce false positives. The study explored that how attackers exploit cognitive biases[5] emotional responses, and social norms to manipulate human behaviour and identified key strategies to enhance resistance against such attacks. Social engineering cyberattacks exploit human psychology to manipulate individuals into divulging confidential information or performing actions that compromise security.

EXISTING SYSTEM

Network Intrusion Detection system is a mechanism that is used within the network to identify the malicious event. It uses K-Nearest Neighbour algorithm for intrusion Detection. Early intrusion detection systems relied heavily on signatures [11]. One of the major drawbacks of using the K-Nearest Neighbour (K-NN) algorithm in intrusion detection is the issue of class imbalance, where normal network traffic vastly outnumbers actual attack instances. This imbalance can cause the model to become biased towards the majority class, leading to poor detection performance for minority classes, which are typically the real threats It requires large volumes of labelled data for effective training something that is often difficult to acquire in the cybersecurity domain. Moreover, many K-NN-based intrusion detection systems suffer from high false positive rates, which can inundate security teams with excessive alerts and ultimately cause alert fatigue. K-Nearest Neighbour is a data mining classifier. KNN is a supervised classifier[12]. The output of the target variable is predicted by finding the k closest neighbour, by calculating the Euclidean Distance. It is a non-parametric classification technique which does not make any assumptions about underlying data [11].

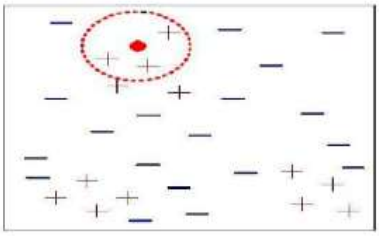

Figure 1 . KNN Classification of Data Instances

III.PROPOSED SYSTEM

To enhance the detection capability of Intrusion Detection Systems (IDS) and protect service providers from potential attacks, we propose an efficient machine learning-based IDS framework. This system leverages Light Gradient Boosting detecting anomalies in network traffic or system logs. (LightGBM) Random Forest algorithms and Correlation-based Feature Selection with Bat Algorithm (CFS-BA). At each node, a subset of features is chosen randomly to split the data, preventing overfitting. During classification, each tree votes on the output, and the majority vote determines the final classification result. The combined results from multiple trees improve accuracy, robustness, and generalization ability. Random Forest assesses patterns in network traffic, identifying abnormal activities indicative of cyber threats. The model continuously refines itself with new data, adapting to evolving attack techniques.

Correlation-based Feature Selection with Bat Algorithm (CFS-BA) is a hybrid feature selection technique used to improve model performance by selecting relevant features. Identifies a subset of features that have high correlation with the target class but low correlation with each other. Inspired by the echolocation behaviour of bats, BA optimizes feature selection by finding the best subset of features. It improves accuracy by ensuring only the most relevant features are used in training. Helps eliminate redundant and irrelevant features, improving model performance. CFS evaluates feature subsets based on their correlation with the target class and low inter-correlation among features. The Bat Algorithm, inspired by the echolocation behavior of bats, is used to optimize the selection process. This combination helps in identifying the most informative features while reducing redundancy. In this project, CFS-BA enhances intrusion detection accuracy by focusing on the most significant network attributes. Combining LightGBM and Random Forest improves detection accuracy by leveraging the strengths of both classifiers. Integrating CFS-BA ensures only the most relevant features are used, reducing noise and computational cost

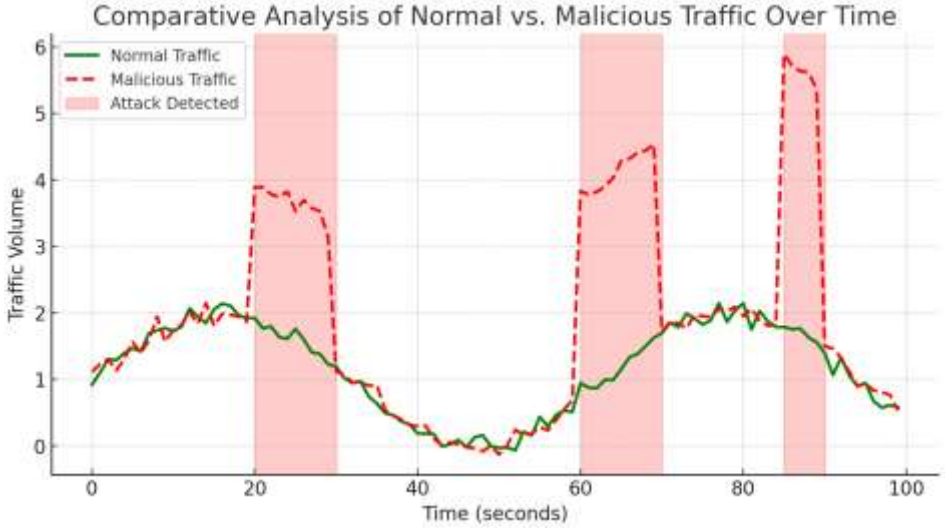

Figure 1. Comparative analysis of traffic

IV .METHODOLOGY

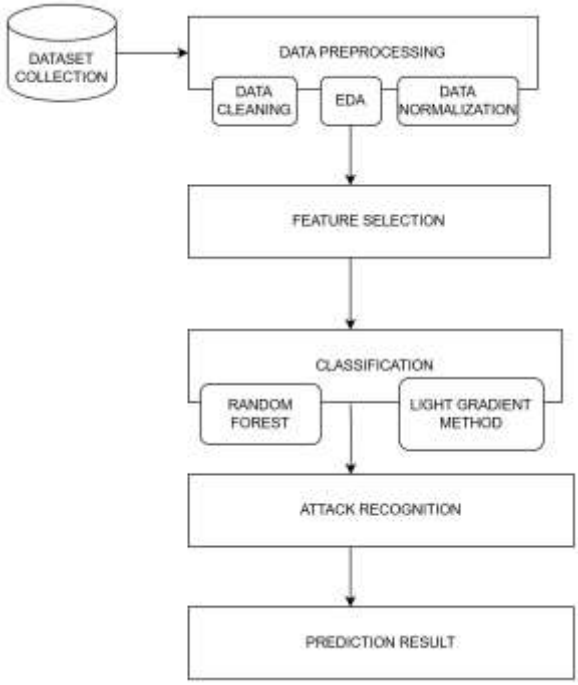

Figure 2. System Architecture

A. DATASET COLLECTION

The process of gathering, organizing, and preparing data from various sources to be used for training and testing machine learning models. In the context of intrusion detection systems (IDS), dataset collection involves obtaining network traffic logs, cyberattack records, and security event data to develop models that can detect malicious activities.

B.DATA PREPROCESSING

In this process it enhances data quality and ensures optimal performance of machine learning models. It involves cleaning the data by handling missing values, removing duplicates, and filtering out noise and irrelevant information. The data is split into training, validation, and testing subsets, often using stratified splitting to maintain class distributions. These preprocessing steps collectively improve model accuracy, reduce computational overhead, and ensure a focus on meaningful data for effective network intrusion detection. Data cleaning, exploratory data analysis, and data normalization employing two techniques, min–max normalization and Z-score normalization, are all part of the data preprocessing process.

Missing Values: Network datasets often contain missing values due to incomplete packet captures or logging errors. For that remove rows with excessive missing data. Impute missing values using techniques like mean, median, mode, or predictive imputation. Removing Duplicate Record: Duplicate entries in network logs may skew analysis and detection results. Identify and drop duplicate rows to maintain data integrity. Noise Removal: Raw network traffic may include noise, such as irrelevant packets or corrupted data. Filter out noise using domain knowledge or predefined thresholds (e.g., ignoring packets below a certain size). Normalization and Scaling: Network traffic features, such as packet size or duration, often have varying scales, which can bias models. Normalize data to a standard range (e.g., [0,1]) or scale features using methods like z-score standardization

C.FEATURE EXTRACTION

This process focuses on identifying key network traffic attributes and optimizing data representations for precise anomaly detection. This enhances accuracy and efficiency in real-time intrusion detection. First, the dataset for the Adaptive Hybrid Forest Framework for Real-Time Intrusion Detection is sourced from publicly available repositories such as Kaggle, containing network traffic logs and intrusion-related events. The obtained dataset is preprocessed and annotated by labeling various network activities, including normal traffic, malicious attacks, and anomalies. The dataset is then divided into two subsets: one for training and the other for testing purposes. The training set is used to train the ensemble deep learning models, including Decision Trees, Random Forests, and Gradient Boosting, while the testing set is utilized for performance evaluation. The trained models analyze various network features and classify traffic patterns to detect potential intrusions accurately. Additionally, performance evaluation helps understand the decision-making process of the model, providing insights into the accuracy and reliability of intrusion detection. This systematic approach—from data collection and preprocessing to model training and evaluation—ensures the efficiency and adaptability of the Adaptive Hybrid Forest Framework in real-time cybersecurity applications.

$$Mean = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where:

x_i = Size of each packet in the network traffic.

N = Total number of packets.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\, log_2 P(x_i)$$

where:
H(X) = Entropy of the system (higher values indicate anomalies).
P(xi) = Probability of occurrence of a particular event xi
n = Number of unique events in the dataset.

D.CLASSIFICATION

Classification methods in NIDS are used to distinguish between normal and malicious network traffic. Machine learning models classify network activities into predefined categories, such as normal traffic, Denial-of-Service (DoS) attacks, malware, or botnet activity. Two widely used classification methods in NIDS are Light Gradient Boosting Machine (LightGBM) and Random Forest.

1.*CORRELATION-BASED FEATURE SELECTION AND BAT ALGORITHM (CFS-BA)*

The CFS-BA-based Network Intrusion Detection System combines Correlation-based Feature Selection (CFS) and the Bat Algorithm (BA) to improve the accuracy and efficiency of detecting malicious network activities. The process begins with the collection of raw network traffic data from sources such as the NSL-KDD or CICIDS2017 datasets. This data is then preprocessed to handle missing values, normalize numerical features, and encode categorical variables. Once the dataset is prepared, the feature selection phase begins. Here, the CFS method evaluates subsets of features based on their correlation with the target class (i.e., normal or attack) and their lack of inter-correlation. The Bat Algorithm is an optimization technique inspired by the echolocation behavior of bats—explores different combinations of features, guided by a fitness function derived from the CFS evaluation. Each "bat" in the algorithm represents a possible feature subset and navigates the search space to maximize detection accuracy and minimize redundancy. Over successive iterations, the algorithm converges on the most relevant subset of features. This optimal feature set is then passed to a machine learning classifier Random Forest is trained to distinguish between normal and malicious network traffic. By focusing only on the most relevant features, the CFS-BA approach reduces computational cost, enhances classification performance, and improves the system's ability to detect intrusions accurately and efficiently.

2. *LIGHT GRADIENT BOOSTING MACHINE(LightGBM)*

LightGBM is a high-performance boosting algorithm designed for speed and efficiency, making it ideal for large-scale network traffic data.

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x)$$

where:
$F_m(X)$ = Model after the mth iteration (current updated prediction).
$F_{m-1}(X)$ =Model from the (m−1)th iteration (previous predictions).
$h_m(X)$ = The mth weak learner (typically a decision tree).
γ = The learning rate — a small value (e.g., 0.1) that scales the update.

3.RANDOM FOREST

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for improved classification accuracy.

$$P(X) = \frac{1}{N}\sum_{i=1}^{N} T_i(X)$$

Where,
P(X) = Final prediction for input X
N = Total number of decision trees in the forest.
Ti(X)= Prediction made by the ith decision tree for input X.

E.ATTACK RECOGNITION

Attack recognition refers to the process of identifying and classifying malicious activities in network traffic. It helps in detecting cyber threats such as Denial-of-Service (DoS) attacks, malware, phishing, botnets, and unauthorized access.

F.PREDICTION RESULT

The prediction result will be generated by a machine learning or rule-based model when analyzing network traffic. It determines whether a particular data instance (network packet, connection, or flow) is classified as normal traffic or an attack.

Algorithm	Detection Rate
K-NN	85%
Light Gradient Boosting Method	92%
Random Forest	90%
CFS-BA	95%

Table 1. Detection rate testing

Figure 2. Prediction - Anomaly detected

Figure 3. Prediction - Normal

IV. CONCLUSIONS

Data pre-processing and feature selection play a pivotal role in enhancing the performance of machine learning models for Network Intrusion Detection Systems (NIDS). Effective pre-processing, which includes data cleaning, handling missing values, removing duplicates, noise filtering, and scaling, ensures that the dataset is clean and consistent, reducing computational overhead and improving model accuracy. Exploratory Data Analysis (EDA) helps uncover insights, detect patterns, and validate assumptions, laying a solid foundation for model development. Feature selection further refines the dataset by identifying the most relevant attributes, thereby improving classification accuracy and reducing dimensionality. The hybrid approach combining Correlation-Based Feature Selection with the Bat Algorithm provides an optimized feature subset that balances relevance and redundancy. CFS ensures that features are informative and non-redundant, while BA's adaptive search capability effectively explores the feature space for optimal solutions. The implementation of an Intrusion Detection System (IDS) using the Random Forest algorithm within a Flask-based web application demonstrates a robust, efficient, and scalable solution for detecting malicious activities in network traffic. Random Forest, known for its ensemble learning approach and high accuracy, effectively handles complex and high-dimensional datasets, making it ideal for intrusion detection

ACKNOWLEDGMENT

The authors sincerely thank the Department of Computer Science and Engineering at Muthayammal Engineering College for their constant support and guidance during this project. We are also thankful to the researchers and experts whose helpful ideas and work in the field of intrusion detection inspired and improved the design of the proposed framework.

REFERENCES

[1] A.Psathas, L.Iliadis, A.Papaleonidas , D.Bountas,"A Hybrid Deep Learning Ensemble for Cyber Intrusion Detection", in Proceedings of the 22nd Engineering Applications of Neural Networks Conference, 2021.
[2] A.Javed, S.U.Rehman, M.U.Khan, M.Alazab, T.G.Reddy,''CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU,'' IEEE Trans. Netw. Sci. Eng., vol. 8, no. 2, pp. 1456–1466, Apr. 2021.
[3] Liqun Yang, Jianqiang Li, Liang Yin, Zhonghao Sun, Yufei Zhao, Zhoujun Li., "Real-time Intrusion Detection In Wireless Network: A Deep Learning based Intelligent Mechanism". IEEE Access 8: 170128-170139 (2020).
[4] M.A.Siddiqi, W.Pak, "A study on the psychology of social engineering based cyberattacks and existing countermeasures", Appl. Sci., vol. 12, no. 12, p. 6042, Jun. 2022.
[5] M.A.Siddiqi , W.Pak, "Optimizing filter-based feature selection method flow for intrusion detection system", Electronics, vol. 9, no. 12, p. 2114, Dec. 2020.
[6] M.A.Siddiqi, W.Pak, "An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection", IEEE Access, vol. 9, pp. 137494137513, 2021.
[7] M.Ganaiea, M.Hu, A.Malika, M.Tanveera , P.Suganthan, "Ensemble deep learning: A review," Arxiv preprints, p. arXiv:2104.02395v2, 2022.
[8] N.Sainis, D.Srivastava, R.Singh, "Feature classification and outlier detection to increased accuracy in intrusion detection system International Journal of Applied Engineering Research", 13(10):7249–7255, 2018. 26.
[9] S.Xu, X.Han, T.Tian, B.Jiang, Z.Lu, C.Zhang, "Few-shot network traffic anomaly detection based on Siamese neural network,'' in Proc. IEEE Int. Conf. Commun., May 2023, pp. 3012–3017.
[10] J. Padhye, V. Firoiu, and Z.Li, J.Wu, S.Mumtaz, A.M.Taha, S.Al-Rubaye, A.Tsourdos, ''Machine learning and multi-dimension features based adaptive intrusion detection in ICN,'' in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2020.
[11] https://medium.com/datadriveninvestor/knn-algorithm-and-implementation-from-scratch- b9f9b739c28f
[12] http://www.scholarpedia.org/article/K-nearest_neighbor
[13] M.A.Jabbar, B.A.Deekshatulu, P.chandra, "Heart Disease classification using nearest neighbor classifier using Feature subset selection", Anale. Seria Informatică. Vol. XI fasc. 1 – 2013.