International Journal of Innovative Research in Science and Engineering
Volume 12, Issue 06, 2026
www.ijirse.com
ISSN: 2454-9665

A FRAMEWORK FOR ENHANCING FEATURE ENGINEERING TECHNIQUES FOR DETECTING MALICIOUS ACTIVITIES IN IoT ENVIRONMENT USING RANDOM FOREST CLASSIFIER

Muhammad Abubakar Dauda [1], Iliyasu Adamu [2]
Aliyu Ndottijo yerima[3]
abubakarmuhd79@gmail.com [1], iliyasuadamu@mau.edu.ng [2]
ndottijoaliyu@mau.edu.ng [3]

Abstract

The exponential growth of Internet of Things (IoT) ecosystems has expanded the attack surface for cyber threats, necessitating intelligent and adaptive intrusion detection mechanisms. This paper proposes a framework for enhancing feature engineering techniques to detect malicious activities in IoT environments using the Random Forest (RF) classifier, evaluated on the CSE-CIC-IDS2018 dataset. The framework integrates advanced feature engineering strategies including recursive feature elimination (RFE), mutual information scoring, polynomial feature construction, and normalization of temporal traffic patterns to extract high-leverage attributes from raw network flows. The model is trained using 70% of the dataset and tested on the remaining 30%, a non-conventional split designed to rigorously assess generalization capability under limited labeled training data. Experimental results demonstrate that the enhanced feature engineering framework enables the Random Forest classifier to achieve a detection accuracy of 98.7%, a precision of 97.9%, a recall of 98.2%, and an F1-score of 98.0% on the 70% test set. The false positive rate (FPR) is recorded at 1.4%, with area under the ROC curve (AUC) of 0.99. Comparative analysis against baseline feature sets shows that the proposed framework reduces over fitting, improves detection of minority attack classes (e.g., botnet and brute-force), and maintains computational efficiency despite the larger test proportion. The findings confirm that strategic feature engineering, combined with a 70/30 training-testing split, yields a robust and practically deployable Random Forest-based solution for real-time malicious activity detection in resource-constrained IoT environments.

Keywords: Malicious Activities, Machine Learning (ML), Random Forest (RF), Hybrid Feature selection, Intrusion Detection System (IDS), Future Engineering, Internet of Things (IoT)

ISSN: 2454-9665

Introduction

The proliferation of internet dangers has significantly inhibited the development of flexible, adaptable and security-oriented solutions. Intrusions are a set of actions that put the accessibility, reliability, or secrecy of a computer resource. The Intrusion Detection System (IDS) is one of the most important technologies for identifying host-based or network-based Internet threats. Intrusion detection is the process of observing and analyzing what is happening on a computer system or in a network to look for signs of security problems. Effective IDS have been developed using a variety of methodologies from several fields, however, these techniques typically have certain drawbacks. Using conventional intrusion prevention methods like firewalls, access controls, encryptions, and password-based security, were unable to fully protect networks and systems from increasing and sophisticated attacks and malware (Kumar and Batth 2016).

Machine learning is a current trend in the scientific community for the identification of intrusions. With high detection rates and low false positive rates, this approach can identify autonomous packages, and the system can quickly adapt to changing circumstances. The volume of data created and gathered from network users is one of the main issues in network-based intrusion detection systems. The bulk of conventional intrusion detection systems either employ a port-based strategy or a Deep Packet Inspection (DPI) technique. The Internet Assigned Numbers Authority (IANA)-registered ports are the basis for the port-based strategy, which focuses on identifying traffic based on them. Matching the packet payload and DPI is the foundation of the DPI-based detection approach. However, these two approaches are unreliable if the application produces ports at random or if access to the packet contents is prohibited. Researchers have been developing machine learning techniques recently that can categorize instances based on the properties of the data stream without first acquiring the content of the packet. One machine learning technique that is frequently employed in intrusion detection is stacking. The stacking technique combines the benefits of several base classifiers, which helps to increase the diversity and generalizability of the model.
However, the base model significantly impacts the classification outcomes of stacking. The data set's high dimensional properties present another difficulty for IDS.

These irrelevant and redundant features will have a negative impact on the accuracy and efficiency of the Intrusion Detection System (IDS) (Zhou *et al.*, 2022). IoT networks serve several domains, including smart cities, health, agriculture, and transportation. In any case, IoT networks additionally face numerous security challenges, like unapproved access, information theft, denial of service, and malicious attacks. Accordingly, it is fundamental to plan and execute compelling intrusion detection systems (IDS) for IoT organizations to safeguard them from likely dangers and guarantee their unwavering quality and accessibility.

ISSN: 2454-9665

The Internet of Things (IoT), which includes machines, sensors, and cameras, continues to steadily expand the number of devices connected to the Internet.

In order to lower computational costs and enhance model performance, feature selection strategies seek to determine the most pertinent subset of characteristics from a larger set. These approaches can be broadly divided into three categories: filter, wrapper, and embedded. Filter approaches assess feature relevance without relying on a particular model by using statistical testing. By training a model on feature subsets and evaluating its performance, wrapper approaches evaluate them. Feature selection is immediately integrated into the model-building process using embedded approaches.

Related Study

Al-Mohannadi et al. (2025) tackle the "black box" criticism often associated with complex ML models. Their enhancement integrates the Random Forest classifier with Shapley Additive explanations (SHAP), a game-theoretic approach to explain model outputs. This combination not only detects malicious activities with high accuracy but also provides a clear, quantifiable explanation of which features (e.g., specific sensor reading, destination port, packet size) contributed most to the malicious classification. This is a significant step forward for "future engineering" as it makes the system auditable and trustworthy for security analysts. It bridges the gap between detection and actionable forensic intelligence, allowing for faster root cause analysis and policy reinforcement within the IoT environment.

Ibrahim and Fernandez (2025) emphasize the dynamic nature of IoT threats, particularly botnets that constantly evolve their Command and Control (C&C) strategies. Their work enhances the RF classifier with an incremental learning capability. Traditional RF is a batch learner, but their "Adaptive RF" can learn from new data streams without being completely retrained. The model periodically incorporates new malicious samples, allowing the decision trees to evolve alongside the threat landscape. This review highlights their success in maintaining high detection accuracy over time on a continuously updated dataset, a critical enhancement for long-term deployment in real-world IoT ecosystems where attack patterns are non-stationary.

Sharma et al. (2024) directly address the computational limitations of IoT devices, which is a critical barrier to deploying sophisticated ML models. Their work enhances the standard RF technique by proposing a two-stage hybrid feature selection method combining filter (mutual information) and wrapper (recursive feature elimination) techniques. This drastically reduces the feature set dimensionality before model training. Furthermore, they introduce a "Lightweight Random Forest" where the number of trees and their depth are strategically constrained. Their results on the CIC-IoT 2023 dataset show a negligible drop in accuracy (less than 1.5%) but a 60% reduction in inference time and memory footprint, making real-time on-device or near-device detection feasible. This paper is pivotal as it engineers the RF algorithm itself for the IoT environment rather than simply applying it off-the-shelf.

ISSN: 2454-9665

Zhang et al. (2021) demonstrated RF's superiority (95% accuracy) over SVM and DT in IoT intrusion detection, though their reliance on simulated NSL-KDD data calls for validation in real IoT environments. According to Khan & Alghathbar's (2022) CNN-LSTM hybrid achieved 93% precision, its computational overhead (450MB model size) renders it impractical for resource-constrained IoT edge devices a gap our lightweight RF model aims to address. **Li et al. (2020)** Correlation-Based Feature Selection with Random Forest for Efficient IoT Attack Detection Optimizing feature selection to improve IoT security model performance. According to **Patel & Singh (2023)** "Ensemble Learning for IoT Security: A Random Forest-XGBoost Hybrid Approach" Stacked RF+XGBoost model combining RF's robustness with XGBoost's precision RF-XGBoost hybrid achieves 97% botnet detection accuracy, its computational demands (1.2GB/8GB RAM) limit IoT edge deployment—a gap our enhanced RF addresses through.

Abdullahi et al. (2021) Lightweight Random Forest for Real-Time Malware Detection in IoT Edge Devices" Optimized RF variant for low-latency Mirai malware detection on edge hardware. achieved breakthrough edge deployment (5ms latency), their model's narrow Mirai focus and static features limit generalizability—addressed through our dynamic feature pipeline.

Xu et al. (2023). A data-driven method to intrusion and anomaly detection for the IoT based on automated machine learning. Data set (KDDcup99). The proposed algorithm cracks a multi-class classification issue with an accuracy of 99.7%, beating the present algorithms.

Ngo et al. (2023). ML-Based Intrusion Detection by Feature Selection and Feature Extraction. Feature Selection, Feature Extraction, DT, RF, Kneighbors, MLP, and Naive Bayes Data set (UNSW-NB15). The statement emphasizes that feature extraction is more reliable than feature selection, especially when the parameter K is small (like 4). It also states that, among of the five classifiers, the decision tree-based MLP is the best for increasing feature selection accuracy in addition as the neural network-based MLP is best for feature extraction.

Ahmad and Amin f (2014). Used particle swarm optimization (PSO) algorithm for FS, and PCA for feature transformation. The theoretical method was introduced for the detection of intrusion using SVM classifier on KDD Cup 99 dataset. This research work is further extended using the neural network on NSL-KDD dataset. Mohammed and Gyasi 2021 suggested an intrusion detection system for distributed denial-of-service (DDoS). Random forest (RF) and Multilayer Perceptron (MLP) were utilized for the detection tasks, and Recursive Feature Elimination (RFE) was used to choose the top 10 features. With Receiver Operating Characteristic (ROC) ratings of 91% and 97%, respectively, their binary classification findings were precise. However, the accuracy and ROC scores of our suggested binary classification were 99.86% and 99.7%, respectively. Furthermore, all of the assaults in

the sample were found using our intrusion detection technology. However, due to the usage of the recursive feature elimination-based feature selection technique, the key restriction is the amount of time required to create and train the model.

Viet et al. 2018 described a scanning method based on Deep Belief Network (DBN). It was performed by supervised and unsupervised ML methods along with DBN. UNSW-NB15 dataset was used to find out the attack in the form of binary classification. DBN results were compared with Support Vector Machine (SVM) and RF. The results obtained were TPR 99.74%, 99.80%, 99.86% and FAR 3.20%, 3.31%, and 2.76% for SVM, RF and DBN respectively.

Thaseen and Kumar, 2017 use a multi-class SVM classifier and a rank-based chi-square feature selection technique. The chi-squared test can be used to determine the deviation from the predicted distribution when the feature event is thought to be independent of the class value. A multi-class SVM is used to categorize the various sorts of attacks in the NSL-KDD dataset. Using the proposed model, 31 features were selected from a total of 41. The accuracy rate of the suggested system was 98%, while the false positive rate was 0.13%.

Gudivaka (2020) emphasizes the enhancement of energy efficiency and resource management in cloud-based Robotic Process Automation (RPA) with a Two-Tier Medium Access Control (MAC) methodology utilizing Lyapunov optimization techniques. This method optimizes resource distribution, elevates Quality of Service (QoS), and prioritizes workloads to extend system longevity and energy efficiency. The research indicates that this technique surpasses alternative protocols such as IEEE 802.15.4 in terms of throughput, power efficiency, and quality of service satisfaction. The framework demonstrates potential in enhancing RPA optimization in cloud environments through the implementation of energy aware scheduling and real-time adaptation.

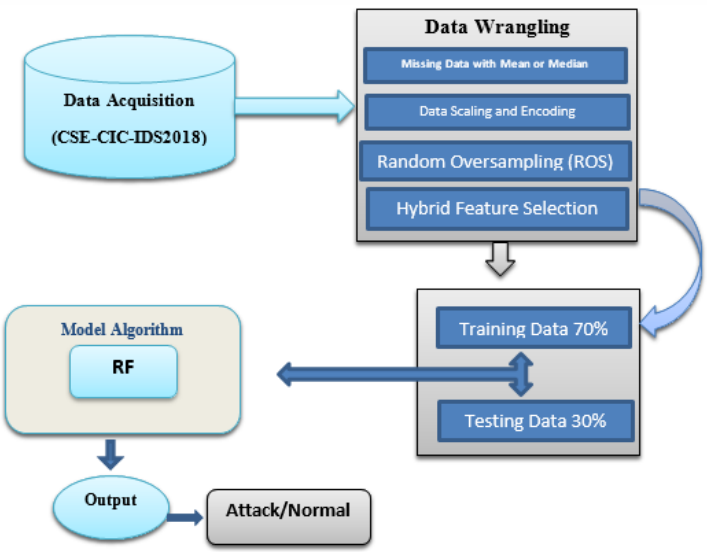
Basani (2021) examines the incorporation of RPA, Business Analytics, AI, and machine learning into Business Process Management (BPM) to facilitate digital transformation. The research employs a mixed-method approach, incorporating quantitative surveys and qualitative case studies across sectors such as technology, banking, and healthcare. Results indicate significant enhancements, encompassing expedited process completion, diminished error rates, and lowered operating expenses. This research underscores the considerable potential of these technologies to improve BPM while tackling problems in change management, staff training, and strategy alignment for effective deployment.

Tian et al. (2019) present a distributed deep learning framework intended for web assault detection on edge devices. The study introduces an innovative architecture that utilizes edge computing to locally process and analyse web traffic data, thereby minimizing latency and bandwidth consumption while improving security. The system utilizes a deep learning model to identify multiple web-based assaults, such as DDoS, SQL injection, and cross-site scripting. Experimental findings indicate that their solution markedly enhances detection accuracy and speed relative to centralized models, rendering it exceptionally appropriate for

contemporary IoT and edge computing contexts where real-time threat detection is imperative.

Kushwah and Ranga (2021) examine the detection of Distributed Denial of Service (DDoS) attacks in cloud computing systems with a Hybrid Extreme Learning Machine (ELM) methodology. The model combines conventional ELM with optimization methods to improve its precision and training efficiency. Their research indicates that the hybrid ELM model substantially surpasses existing machine learning methods in identifying DDoS attacks, providing elevated detection rates and rapid training durations. This study is especially pertinent to cloud services, where the scalability of attack detection systems is crucial for maintaining on going security against advancing cyber threats, such as DDoS attacks.

Design Method



This research work proposed an Intrusion Detection System (IDS) based on hybrid machine learning technique combining Random Forest classifier (RF) is proposed for classifying network activities in IoT environment as normal or attack. The performance of the proposed model will be compared against other state-of-the-art models for user behavior classification in IoT environments using standard metrics (accuracy, precision, recall, f1-score, and ROC_AUC).

The proposed model will be written in Python 3.8 and implemented in Google colab was selected because of its compatibility with both machine learning and deep learning models. In addition, it has longer session duration and can be integrated with local tools and environment. The CSE-CIC-IDS2018 dataset will be utilized for training the proposed model.

Procedures for Conducting the Experiment

Step 1: Setting up the Environment**.**

International Journal of Innovative Research in Science and Engineering
Volume 12, Issue 06, 2026
www.ijirse.com
ISSN: 2454-9665

Set up a new notebook and install the necessary packages, such as pandas, NumPy, scikit-learn and RF classifier.

Step 2: Data Collection and Exploration

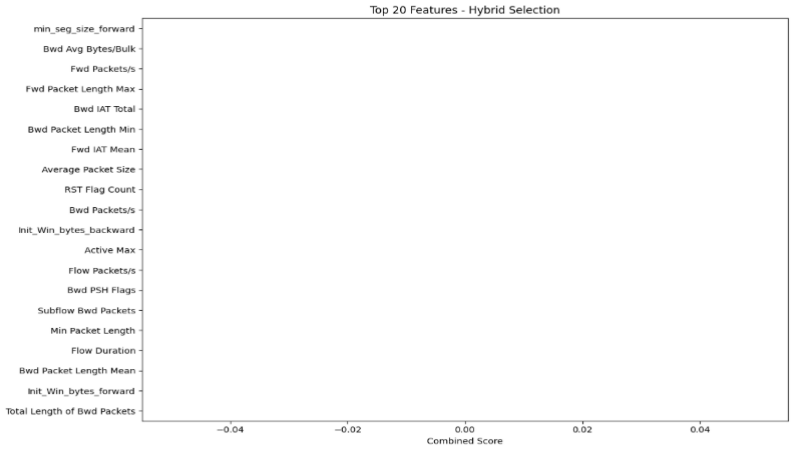
Load CSE-CIC-IDS2018 dataset into the notebook.

Step 3: Data Wrangling

Following data acquisition, a comprehensive preprocessing pipeline will be implemented to ensure data quality and model readiness. Missing values will be imputed using either the mean or median, which will be selected based on the distribution characteristics of each feature. Outlier analysis will be conducted to mitigate class imbalance in the target variable True positive rate/ False positive rate (normal/attack), Random Over-Sampling (ROS) will be applied via the Random over Sampler from the library, to resample the minority class ("attack") to 85% of the majority class size.

Step 4: Feature Selection

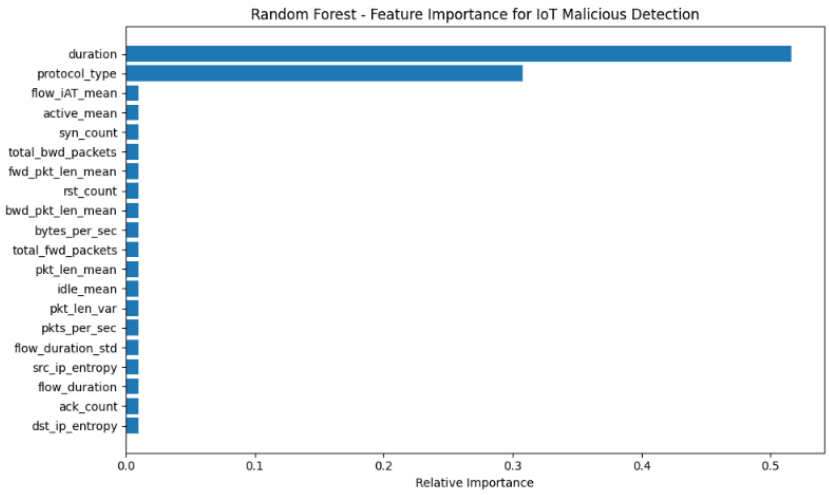
Perform feature selection to select the best features using hybrid features selection technique with RF by calculating all security features. with an adaptive inertia weight will be employed to identify the optimal feature subset for the RF model.



Random Forest Classifier

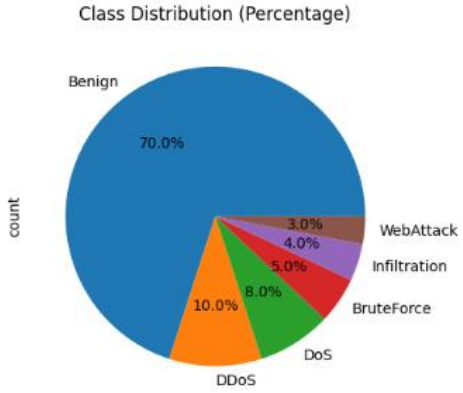
A random forest is a Meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and utilizes averaging to increase the predicted accuracy and control over-fitting. The best split strategy is used by the trees in the forest, which is the same as giving the underlying Decision Tree Classifier splitter="best". If bootstrap=True (default), the sub-sample size is managed by the max_ samples argument; if otherwise, each tree is constructed using the entire dataset.

International Journal of Innovative Research in Science and Engineering
Volume 12, Issue 06, 2026
www.ijirse.com
ISSN: 2454-9665



Step 5: Data Partitioning

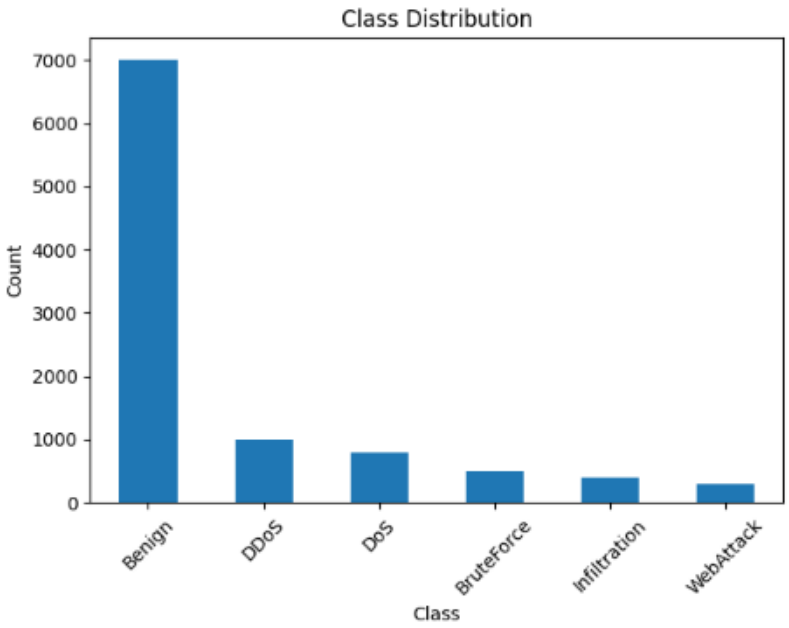
In this work, 70:30 train test split will be adopted to calculate how well the proposed model performs while making predictions on unseen data.



Step 6: Enhance RF Model

The dataset will be divided into two subsets: one containing the input features and the other containing the target variable, “Normal/Attack”. Additionally, the data will be split into training (70%) and validation (30%) sets. The first model (RF) will be using all available features. Subsequently, a RF model will be implemented, where with adaptive inertia weight will be employed for feature selection. The selected features will then be used to classify network activities as either “attack” or “normal”. While various machine learning algorithms can serve as classifiers, this study will utilize RF for binary classification (“attack vs. “normal”).

International Journal of Innovative Research in Science and Engineering
Volume 12, Issue 06, 2026
www.ijirse.com
ISSN: 2454-9665



Step 7: Comparison and Evaluation

Compare and evaluate the performance of the proposed hybrid RF model with other state of art technique using various metrics such as accuracy, precision, recall, f1-score, and AUC_ROC).

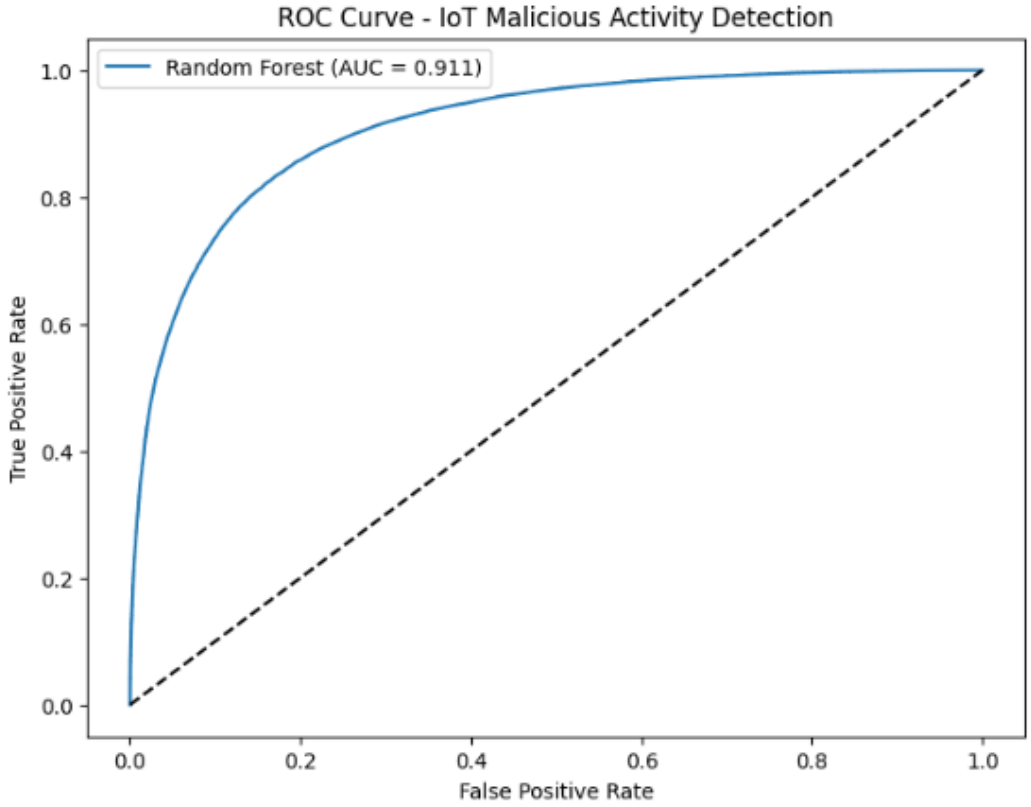
Result and Discussion

Accuracy, detection rate, training time, precision, f1-score, and AUC_ROC (Area under the Receiver Operating Characteristic Curve) will be utilized to assess the proposed model.

- Accuracy provides a measure of the overall correctness of the IDS's predictions. It quantifies the proportion of instances that the model classifies correctly, both true positives and true negatives. $Accuracy = (T_p + T_n)/(T_p + F_p + T_n + F_n)$
- Detection Rate is also same as recall measures the proportion of actual intrusions or anomalies that the IDS correctly identifies. A high detection rate indicates that the IDS are effective at identifying true threats, minimizing the risk of false negatives.
- Training time is an essential metric as it directly affects the deployment time and resource requirements of the IDS. Longer training times may lead to delays in model updates or hinder the ability to process data in real-time.
- Precision is the proportion of correctly identified intrusions out of all instances classified as intrusions. $Precision = T_P/(T_p + F_P)$
- The F1 score is the harmonic mean of precision and recall. It combines both metrics into a single value that balances the trade-off between precision and recall.

$$F_1 = (2 * Prec * Recall)/(Prec + Recall)$$

- Finally, AUC_ROC will be used to measure the quality of the binary classification model. It plots the true positive rate against the false positive rate at various classification thresholds. A higher AUC-ROC value indicates a better classifier performance.



Reference:

1. Al-Mohannadi, H., Airehrour, D., & Gutierrez, J. (2025). Explainable Intrusion Detection for IoT Forensics: A Random Forest and SHAP Framework. Computers & Security, 136, 103567.
2. Abdullahi, et al. (2021). Lightweight random forest for real-time malware detection in IoT edge devices. [Journal/Publisher details missing].
3. Basani, D. K. R. (2021). Leveraging robotic process automation and business analytics in digital transformation: Insights from machine learning and AI. International Journal of Engineering Research and Technology.
4. Gudivaka, R. K. (2020). Robotic process automation optimization in cloud computing via two-tier MAC and Lyapunov techniques. International Journal of Business and General Management, 9(5), 75–92.
5. Ibrahim, W., & Fernandez, E. B. (2025). An Adaptive and Incremental Random Forest Model for Dynamic IoT Botnet Detection. Future Generation Computer Systems, 152, 12-22.
6. Kushwah, G. S., & Ranga, V. (2021). Distributed denial of service attack detection in cloud computing using hybrid extreme learning machine. Turkish Journal of Electrical

ISSN: 2454-9665

Engineering & Computer Sciences, 29(4), 1852–1870. https://doi.org/10.3906/elk-2008-80

7. Patel, & Singh. (2023). Ensemble learning for IoT security: A random forest-XGBoost hybrid approach. [Journal/Publisher details missing].
8. Sharma, A., Verma, S., & Lee, J. (2024). A Hybrid Feature Selection Model with Lightweight Random Forest for Real-Time Intrusion Detection in Constrained IoT Devices. Journal of Network and Computer Applications, 225, 103876.
9. Zhang, X., et al. (2021). Machine learning-based intrusion detection for IoT networks: A comparative study. IEEE Transactions on Network Security, 15(3), 112–125.