LAUTECH Journal of Computing and Informatics (LAUJCI) – ISBN : 2714-4194
Volume 5 No.1, April 2026 – www.laujci.lautech.edu.ng

Machine Learning Driven Intrusion Detection for Internet of Things Networks: A Comparative Study of Ensemble and Traditional Models

Adetoye A. Adeyemo
Department of Computer Eng., Abiola Ajimobi Technical University, Ibadan, Oyo State, Nigeria
adetoye.adeyemo@tech-u.edu.ng

Ozichi N. Emuoyibofarhe
Computer Science Programme, Bowen University, Osun state, Nigeria
ozichi.emuoyibofarhe@bowen.edu.ng

Adeyinka O. Abiodun
National Open University of Nigeri Abuja,
aabiodun@noun.edu.ng

James O. Adegboye
Department of Computer Science
Federal University of Technology, Ilaro
olujoba.adegboye@federalpolyilaro.edu.ng

Sunday A. Ajagbe
Department of Computer Engineerin
Ladoke Akintola University of Technology, Ogbomosho, Oyo State Nigeria
saajagbe@pgschool.lautech.edu.ng

ABSTRACT
This study investigates the effectiveness of an intrusion detection system (IDS) powered by machine learning (ML) for securing Internet of Things environments. The growth of IoT devices has increased exposure to complex cyber threats that traditional security mechanisms struggle to detect. In this work, a supervised learning based intrusion detection framework was developed and evaluated using the publicly available UNSW-NB15 dataset, which represents real IoT network traffic scenarios. The methodology involved data preprocessing, feature engineering, model training, and performance evaluation using multiple metrics including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC). Five ML models were examined, namely Random Forest, XGBoost, Decision Tree, Logistic Regression and Naive Bayes. Experimental results show that ensemble models achieved superior performance. Random Forest recorded the highest performance with an accuracy of 99.78%, precision of 9.59%, recall of 98.27%, F1-score of 97.93% and AUC of 99.99%, followed by XGBoost with accuracy of 99.76%, precision of 97.72%, recall of 97.84%, F1-score of 97.78% and AUC of 99.99%. Decision Tree achieved an accuracy of 99.68%, precision of 96.76%, recall of 97.28%, F1-score of 97.02% and AUC of 98.55%, while Logistic Regression recorded an accuracy of 99.28%, precision of 91.07%, recall of 95.86%, F1-score of 93.40% and AUC of 99.92%. Naive Bayes produced lower performance with accuracy of 92.91%, precision of 40.16%, recall of 66.30%, F1-score of 50.02% and AUC of 93.56%, indicating reduced capability in modeling complex IoT traffic patterns. Further analysis using confusion matrices, ROC curves, and calibration plots confirmed the robustness and reliability of ensemble approaches. The findings demonstrate that ML driven intrusion detection is effective for IoT security, with XGBoost and Random Forest offering the best balance between detection performance and false alarm reduction.

Keywords: Intrusion Detection, Machine Learning (ML), Internet of Things (IoT), XGBoost, Random Forest, Network Traffic Classification

1. Introduction

The digital environment is experiencing a major change as the Internet of Things (IoT) technology moves closer to becoming part of our everyday lives. Smart sensors, for instance, are used in industries to monitor pipelines; on the other hand, there are wearable devices for monitoring our health, and the connection scale is unprecedented. Recent projections suggest that the number of connected devices will surpass

25.44 billion (Sharma and Bhushan, 2026). This hyper-connectivity promises effectiveness and new ideas, it also creates a huge and very easy to penetrate attack surface that traditional security measures are increasingly ill-equipped to defend (Ogenyi et al., 2025). The great diversity of IoT ecosystems, which consist of devices with different communication standards, short battery life, and low processing power, renders them vulnerable to sophisticated cyber-attacks.

With the devices turning out to be an essential component of critical infrastructure, the consequences of a security breach have moved from just losing data to the possibility of physical and systemic interruption. Traditional security frameworks, which often rely on static, signature-based detection, are struggling to keep pace with the dynamic nature of modern threats (Rahman et al., 2025). These older systems of detection are good for malware "fingerprints" that are already known, but they are seldom effective against zero-day attacks or subtle, evolving malicious patterns. Therefore, the researchers have turned to the anomaly-based IDS that are equipped with ML. These smart systems, apart from detecting known threats, also get trained on the "heartbeat" of a network and thus identify deviations from normal behavior that might signal an intrusion (Kalpani and Rodrigo, 2026).

However, implementing machine learning in the IoT context is not without its hurdles. The "resource-constrained" feature associated with many IoT edge devices means that complex, power-hungry algorithms are often impractical in real-time deployment (Mallick et al., 2026). Furthermore, as network traffic becomes more high-dimensional and non-linear, simpler models like Naive Bayes or Logistic Regression frequently degrade in detection accuracy (Bibers et al., 2025). This has led to the rise of ensemble learning, which is a strategy that incorporates the strengths of several models to achieve a more reliable and robust outcome. Some of these models are: Extreme Gradient Boosting (XGBoost) and Random Forest (RF). They have emerged as frontrunners, consistently demonstrating superior performance in handling the intricate dependencies of IoT traffic (Kouassi et al., 2025; Ali et al., 2025).

Despite these advancements, a significant gap exists in balancing high detection rates with model interpretability and computational efficiency. A lot of top-notch models operate as "black boxes," and do not give much explanation as to why a particular activity was marked as threatening, which is something that security experts who make crucial decisions need to know (Hosain and Çakmak, 2025). Furthermore, the shift from theoretical precision to practical, real-time use on edge hardware still calls for active research (Alturki and Alsulami, 2025).

Regardless of the remarkable progress in machine learning based intrusion detection systems, there remains a limited gap in the literature regarding the empirical comparison of traditional and ensemble classifiers specifically in the IoT network environments. Many existing studies evaluate models using different datasets, feature selection methods and performance metrics, which makes it difficult to come to a reliable conclusions about their relative effectiveness. Most especially, there is limited work that systematically compares both categories of models using the same IoT dataset, uniform preprocessing techniques, and standardized evaluation metrics. This lack of uniformity impedes reproducibility and

complicates the true performance tradeoffs between simpler models and complex ensemble approaches. Therefore, there is a need for a detailed and controlled comparative study which evaluates these models under identical experimental conditions using a representative IoT dataset such as UNSW NB15, in order to provide clear insights into their detection capability, reliability and suitability for real world deployment

This research paper seeks to address these challenges by providing the most exhaustive comparison of traditional and ensemble ML models suited for IoT intrusion detection. Our target is to evaluate the performance of XGBoost, Logistic Regression, Decision Trees, Random Forest, and Naive Bayes. The purpose is to identify the most reliable and effective balance between accuracy, false alarm reduction, and reliability.

2. Literature Review

The growth rate of Internet of Things (IoT) devices has basically changed modern connectivity, yet it has simultaneously expanded the attack surface for cyber threats. As IoT ecosystems are integrated into critical infrastructures like wearable devices, smart homes, and industrial sectors, the demand for robust Intrusion Detection Systems (IDS) has never been more pressing. Traditional security systems often struggle with the heterogeneous nature and resource constraints of IoT devices, making researchers to increasingly turn toward machine learning (ML) as a scalable and adaptive solution.

2.1 The Evolution of IDS in IoT Networks

Before now, research into IoT security centered on signature-based detection, which, while effective against known threats, has failed to identify zero-day attacks or dynamic malicious patterns. Recent surveys, such as those carried out by (Tekin et al., 2023), highlight a significant transition to anomaly-based detection powered by machine learning. These systems learn the "normal" behavior of network traffic and signal any anomaly, which makes them a better option for the dynamic and unpredictable nature of IoT systems. Kikissagbe et al. (2024) stressed that supervised learning remains the cornerstone of this field, which provides a structured way to sort traffic into benign or malicious categories based on historical data.

2.2 Comparative Performance of Machine Learning Models

The strength of an IDS is mostly determined by the underlying algorithm's ability to handle high-dimensional and non-linear data. Standard models like Logistic Regression and Naive Bayes were initially explored due to their computational efficiency. However, as noted by Vitorino et al. (2022) these models often fall short in complex IoT scenarios because they struggle to capture the intricate dependencies between networks features.

On the other hand, one of the issues that is attributed to tree-based models like Decision Trees, which offer better interpretability and performance, is overfitting. This disadvantage has led to the general adoption of ensemble methods. Alharthi et al. (2025) conducted a detailed study proving that while individual classifiers provide a baseline, ensemble techniques significantly improve detection accuracy and reliability.

2.3 The Rise of Ensemble Learning: Random Forest and XGBoost

Recent studies have consistently identified Extreme Gradient Boosting (XGBoost) and Random Forest (RF) as the algorithms that outperformed others when it comes to IoT intrusion detection.

i. Random Forest: RF reduces variance and improves generalization by aggregating the decisions of multiple trees. Hamidou and Mehdi (2025) found that RF achieved an exceptional F1-score across various IoT datasets, which makes it a reliable option for reducing false positives, which is a critical requirement for resource-constrained IoT systems.
ii. XGBoost: It is known for its performance and speed. XGBoost has emerged as a powerhouse in the field. According to Kaddour et al. (2024), one of the most important features is its capacity to learn from errors made by previous trees and constantly re-evaluate and correct them. This specifically helps to detect certain kinds of attacks that simpler models would not be able to identify. Studies by Alharthi et al. (2025) reported an exceptional performance of XGBoost in some particular IoT traffic scenarios.

2.4 Literature Comparison for IoT Intrusion Detection

Table 1 below provides a detailed comparison of various machine learning studies focused on intrusion detection in Internet of Things (IoT) environments.

Table 1: Comparison with existing approaches

Author(s) & Year	Dataset Used	Models Applied	Key Metrics	Key Findings	Limitations
Sunday et al. (2024)	UNSW-NB15	XGBoost, Random Forest, Linear SVC, Decision Tree	Accuracy, Precision, Recall, AUC	XGBoost (90% Accuracy) and Random Forest (0.91 AUC) showed strong performance. RF and Linear SVC achieved the highest AUC.	Study used an unbalanced dataset, which can pose challenges for model generalization across different attack types.
Ghadami et al. (2025)	NSL-KDD, UNSW-NB15, BoT-IoT	ASPCNNLSTM (Hybrid CNN-LSTM), GAN, AOA-SCA	Accuracy, Precision, Sensitivity	ASPCNNLSTM achieved 98.67% accuracy on UNSW-NB15. GAN-based balancing and hybrid feature selection	Increased computational complexity due to hybrid optimization; longer processing time for feature selection.

				significantly improved results.	
Mahfouz et al. (2020)	CIC-IDS 2017	Multiple Linear Regression	Accuracy	Achieved 73.79% accuracy in detecting DDoS attacks effectively.	Moderate accuracy compared to ensemble methods; requires further optimization in feature selection.
Kikissagbe et al. (2024)	Various (Review)	SVM, KNN, Random Forest, Decision Tree	Accuracy, FPR	Random Forest consistently provides high F1-scores and low false positives across multiple IoT datasets.	Computational costs of ensemble methods must be balanced against limited IoT edge device processing power.
Alharthi et al. (2025)	Various	ML and Deep Learning (DL) models	Accuracy, Reliability	Ensemble techniques significantly improve detection accuracy and reliability compared to individual classifiers.	The "black-box" nature of advanced models remains a barrier for applications requiring high explainability

To provide a robust context for this work, an expanded comparison that is beyond the UNSW-NB15 dataset to include studies utilizing other benchmark datasets such as NSL-KDD, BoT-IoT, and CICIDS2017 has been done. Table 1 compares prominent papers that provides information about the Author(s), Dataset Used, Models Applied, Key Metrics, Key Findings and Limitations. This comparison highlights that this research proposed ensemble-based approach (XGBoost and Random Forest) remains highly competitive and state-of-the-art across different network traffic scenarios, while also acknowledging the common challenges in the field, such as computational constraints and model interpretability.

3. Methodology

This section gave an elaborate description of the approach used for designing and evaluating an ML-based IDS for IoT environments. The methodology covers data collection, data preparation, model development, evaluation strategy, and the whole system architecture used to attain reliable attack detection.

3.1 System Overview

The planned IDS is designed to monitor IoT network traffic and automatically categorize activities as normal or malicious. The system is developed with a layered processing pipeline that starts with data acquisition from IoT devices and ends with intelligent decision-making using trained machine learning models. This structured architecture ensures scalability, accuracy, and suitability for real-world IoT usage.

3.2 Dataset Description and Preprocessing

Network traffic data emanating from IoT devices is first collected and saved in a centralized repository. Since raw IoT traffic often contains noise, missing values, and redundant features, preprocessing is applied before model training. This includes data cleaning, feature selection, normalization, and label encoding. These steps help improve learning efficiency and ensure a fair comparison across all ML models.

The dataset used in this study is the publicly available UNSW-NB15 dataset, which was developed at the Australian Centre for Cyber Security using the IXIA PerfectStorm tool to simulate realistic modern network traffic and attack scenarios. The dataset consists of a total of 1,400,002 network records captured in raw packet form and later processed into structured features using tools such as Argus and Bro-IDS.

Figure 1 shows that the dataset exhibits a significant class imbalance, with normal traffic accounting for approximately 94.6% of the total samples, while attack traffic constitutes only 5.4%.

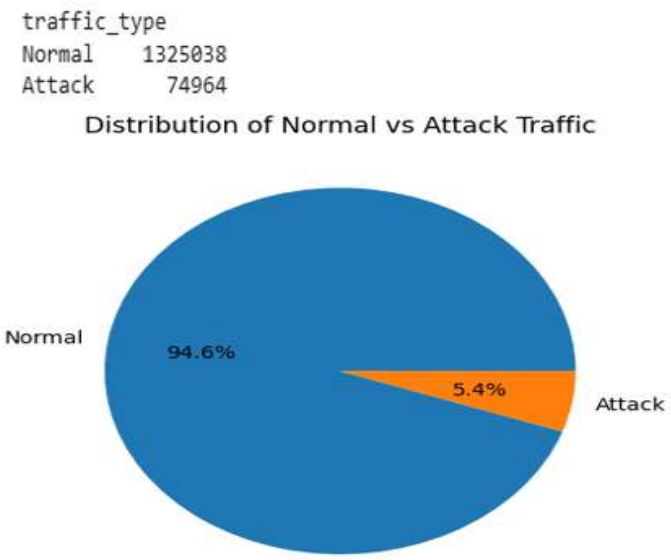

Figure 1: Distribution of Normal and Attack Traffic

In absolute terms, the dataset contains 1,325,038 normal instances and 74,964 attack instances, indicating a strong dominance of benign network activity. This imbalance highlights the need for comprehensive evaluation metrics such as precision, recall, F1-score, and AUC to ensure reliable assessment of intrusion detection performance beyond accuracy alone.

```
[5 rows x 49 columns]
Index(['srcip', 'sport', 'dstip', 'dsport', 'proto', 'state', 'dur', 'sbytes',
       'dbytes', 'sttl', 'dttl', 'sloss', 'dloss', 'service', 'Sload', 'Dload',
       'Spkts', 'Dpkts', 'swin', 'dwin', 'stcpb', 'dtcpb', 'smeansz',
       'dmeansz', 'trans_depth', 'res_bdy_len', 'Sjit', 'Djit', 'Stime',
       'Ltime', 'Sintpkt', 'Dintpkt', 'tcprtt', 'synack', 'ackdat',
       'is_sm_ips_ports', 'ct_state_ttl', 'ct_flw_http_mthd', 'is_ftp_login',
       'ct_ftp_cmd', 'ct_srv_src', 'ct_srv_dst', 'ct_dst_ltm', 'ct_src_ltm',
       'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'attack_cat',
       'label'],
      dtype='object')
```

Figure 2: Dataset Features

As shown in Figure 2, the dataset contains 49 original features describing network flow characteristics, along with class labels indicating normal or attack traffic. The dataset includes nine major categories of attacks, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms, making it suitable for comprehensive intrusion detection studies. The dataset was obtained from a publicly accessible repository, ensuring transparency and reproducibility of the study

3.3 Model Training and Evaluation

To ensure reliable evaluation, the dataset was divided into training and testing subsets using an 80 to 20 split ratio, where 80 percent of the data was used for model training and 20 percent for testing. This split allows sufficient data for learning while maintaining an unbiased evaluation set. In addition, to improve robustness and reduce the effect of random data partitioning, k-fold cross validation with k set to 5 was employed during training. This approach ensures that each model is evaluated across multiple data folds, providing a more stable and generalizable performance estimate. The combination of holdout testing and cross validation strengthens the reliability of the experimental results. Performance was evaluated using accuracy, confusion matrices, ROC curves, zoomed-in ROC analysis and calibration curves. These metrics provide both quantitative and visual insights into detection capability, error distribution, discrimination strength, and probability reliability.

3.4 Machine Learning Models

Multiple supervised ML algorithms were implemented in order to evaluate their effectiveness in intrusion detection. These models include Random Forest, XGBoost, Logistic Regression, Decision Tree, and Naive Bayes. The selection covers tree-based, ensemble, linear, and probabilistic approaches. This allows for a detailed assessment of how different machine learning models handle IoT traffic patterns.

Each model was trained using the same preprocessed dataset and evaluated under the same conditions to ensure consistency and fairness in performance comparison.

3.4.1 Decision Tree

A Decision Tree classifies IoT network traffic by recursively splitting features such as packet size, protocol type, or flow duration into decision rules. Each internal node represents a condition on a feature, while leaf nodes represent the final class label, normal or attack. This makes the model easy to interpret, which is important for security analysts.

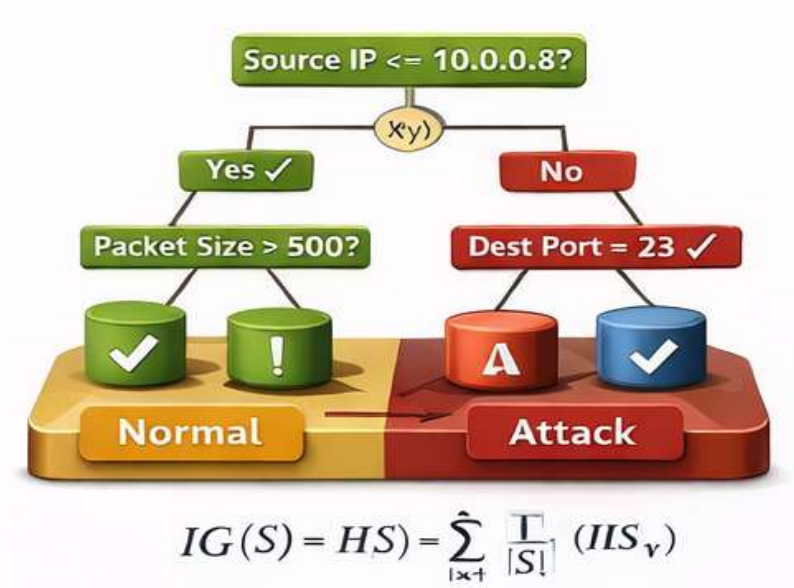

Figure 3: Decision Tree

Mathematically, a split is selected by maximizing information gain

$$IG(S)=H(S)-\sum_{v\in V}\frac{|S_v|}{|S|}H(S_v)$$ Equation 1

Where H S is the entropy of the dataset. In IoT intrusion detection, Decision Trees quickly capture rule based attack patterns but may overfit noisy traffic.

3.4.2 Random Forest

Random Forest improves intrusion detection by combining numerous Decision Trees trained on random subsets of IoT traffic features and samples. Each tree votes, and the final class is selected by majority voting. This reduces variance and improves robustness against diverse attack patterns.

Figure 4: Random Forest

The prediction is given as

$$\hat{y} = mode\{T_1(x), T_2(x), \ldots, T_n(x)\}$$ Equation 2

Where T_i represents individual trees. In IoT environments, Random Forest effectively reduces false alarms and handles high-dimensional traffic data.

3.4.3 XGBoost

XGBoost is a gradient boosting ensemble model that builds trees consecutively, where each new tree corrects the errors of the previous ones. This makes it very powerful for detecting subtle and evolving intrusions in IoT traffic.

Figure 5: XGBoost

The objective function is

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$ Equation 3

Where K = total number of trees and fk = k-th decision tree. XGBoost showed the best performance in your study due to its ability to learn complex nonlinear attack behaviors.

3.4.4 Logistic Regression

Logistic Regression is a linear classifier that estimates the probability of a traffic instance being malicious. It works well when attack patterns are linearly separable but struggles with complex IoT traffic interactions.

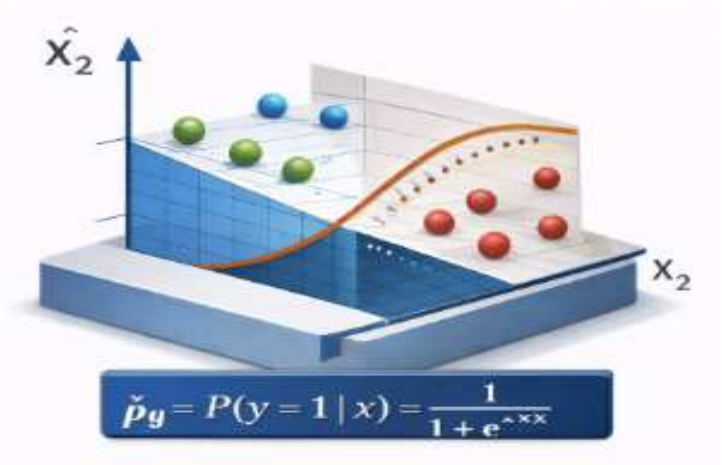

Figure 6: Logistic Regression

The model is defined as

$$\widehat{P}_y = P(y = 1|x) = \frac{1}{1 + e^{-w^T x}}$$ Equation 4

Its simplicity makes it computationally efficient for IoT systems, but it often misses sophisticated intrusion patterns.

3.4.5 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes theorem with the assumption that features are conditionally independent. It estimates the likelihood of traffic being normal or malicious based on observed feature distributions.

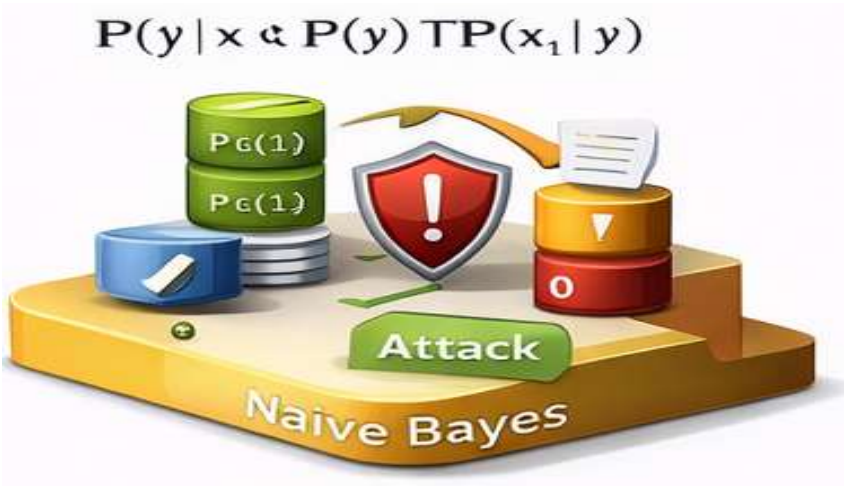

Figure 7: Naive Bayes

The decision rule is

$$P(y \mid x) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$ Equation 5

Although fast and lightweight, the independence assumption limits its effectiveness for IoT intrusion detection, where traffic features are highly correlated.

3.5 Hyperparameters

The machine learning models were implemented using standard configurations. Random Forest and Decision Tree were initialized with default parameters, with the number of trees set to 100 and no restriction on tree depth. XGBoost was configured with a learning rate of 0.3, maximum depth of 6 and 100 estimators, while Logistic Regression was trained using the lbfgs solver with a maximum of 1000 iterations. Gaussian Naive Bayes was applied with default variance smoothing. These settings ensure a fair comparison across models without bias from extensive hyper parameter tuning.

3.6 System Architecture Description

The system architecture is illustrated in Figure 8 with four layers. At the first layer, IoT devices such as sensors, smart appliances, handheld devices, and embedded systems generate network traffic. The traffic logs are forwarded to the data acquisition layer, where packets are captured and logged.

The second layer represents data preprocessing and feature engineering. At this layer, raw traffic is cleaned, transformed, and converted into meaningful features that will be acceptable by machine learning models for analysis.

The third layer consists of the machine learning engine, where the trained models analyze incoming data and classify activities as normal or intrusive. Detected intrusions are then forwarded to layer four, the alert and visualization module, which will enable the administrators to take timely security decisions and actions.

This layered and modular architecture improves system clarity, supports scalability and allows easy integration of new detection models as IoT threats continue to evolve.

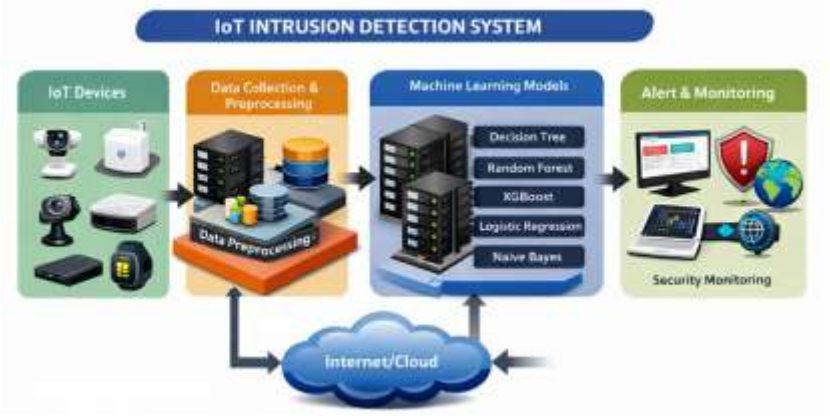

Figure 8: IoT Intrusion Detection System Architecture

4. Results and Discussion

This section presents the results obtained from the ML models applied to intrusion detection in IoT and provides a comprehensive discussion of their performance. The evaluation centers on classification effectiveness, error distribution, discrimination capability, and probability calibration using standard performance metrics and visual analysis tools.

4.1 Overall Model Performance

Table 2 concisely summarizes the performance of all the evaluated models. The results show clearly the variations in detection capability across all the algorithms. Tree-based ensemble models demonstrated top-notch performance over linear and probabilistic models. This trend underscores the importance of capturing complex and non-linear relationships that are common in IoT network traffic patterns.

From the results recorded by the models, Random Forest and XGBoost recorded the highest overall accuracy of 99.78% and 99.76%, respectively, and balanced performance across evaluation metrics. Decision Tree also performed competitively with an accuracy of 99.68%, but showed slightly higher variance, which can be predicted due to its sensitivity to data splits. Logistic Regression and Naive Bayes recorded lower performance, which indicates limitations in modeling complex attack behaviors in IoT traffic.

Table 2: Model Performance

	Accuracy	Precision	Recall	F1-Score	AUC	FPR
Random Forest	99.78	97.59	98.27	97.93	99.99	0.14
XGBoost	99.76	97.72	97.84	97.78	99.99	0.13
Decision Tree	99.68	96.76	97.28	97.02	98.55	0.18
Logistic Regression	99.28	91.07	95.86	93.40	99.92	0.53
Naive Bayes	92.91	40.16	66.30	50.02	93.56	5.59

4.2 Confusion Matrix Analysis

Figures 9 to 13 present the confusion matrices for the individual classifiers. These figures present insight into how each model differentiates between normal and intrusive traffic.

The Decision Tree confusion matrix in Figure 9 shows strong true positive (14585) detection but with noticeable false positives of 448. This points to the fact that while the model is capable of identifying attacks, it may incorrectly flag some normal traffic as malicious, and this could lead to unnecessary alerts in real-world deployments.

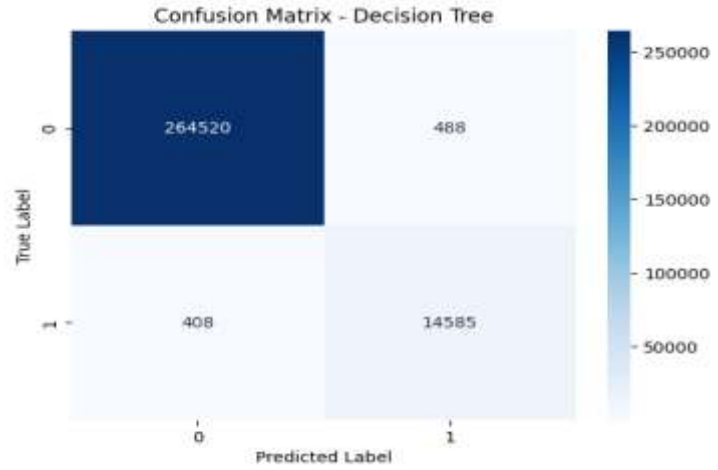

Figure 9: Decision Tree confusion matrix

Figure 10 depicts the Random Forest confusion matrix. The model substantially lowers both false positives to 364 and false negatives to 260. This sign of improvement is because of the ensemble feature of Random Forest, where multiple trees contribute to more stable and generalized predictions.

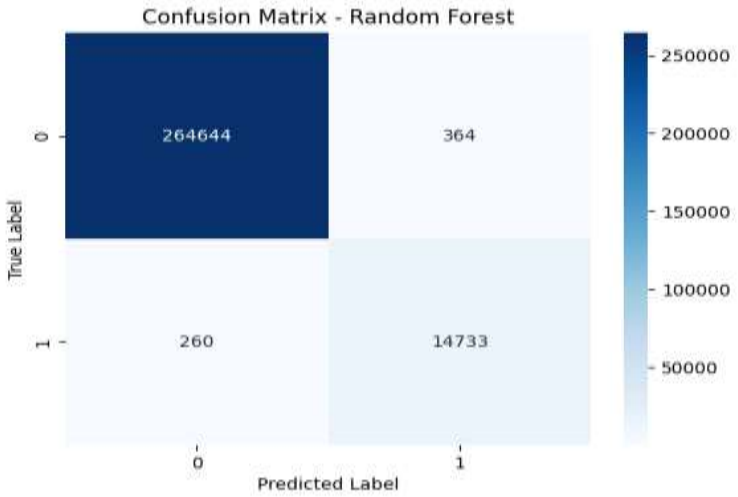

Figure 10: Decision Random Forest Matrix

The XGBoost confusion matrix in Figure 11 presents the best balance between attack detection and normal traffic classification. The lower misclassification rate shows the model's capability to iteratively focus on hard-to-classify instances, making it highly suited for intrusion detection tasks.

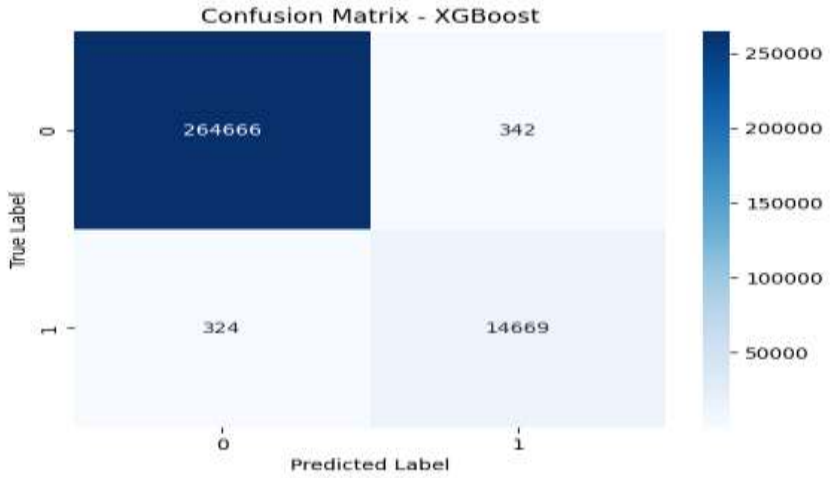

Figure 11: XGBoost confusion matrix

Logistic Regression results shown in Figure 12 unravel a higher number of misclassified samples, particularly false negatives, with the value 620. This behavior is very dangerous in security applications, as undetected attacks can seriously compromise IoT systems.

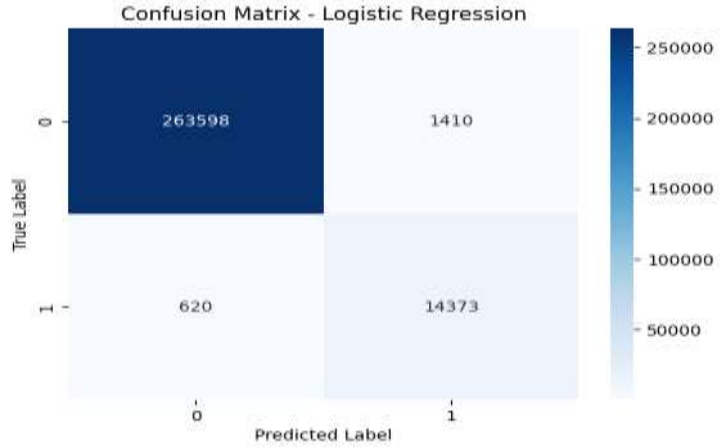

Figure 12: Logistic Regression confusion matrix

The Naive Bayes confusion matrix in Figure 13 further reinforces the performance limitations. Its strong independence assumption does not adequately represent the dependencies among IoT traffic features, resulting in reduced detection accuracy.

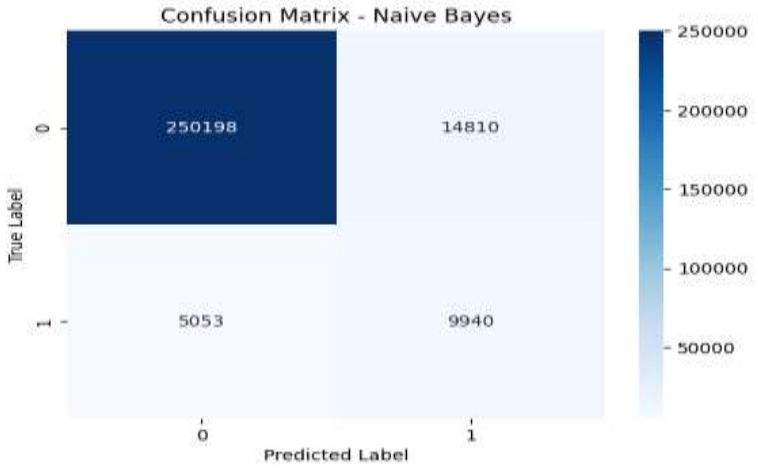

Figure 13: Naive Bayes confusion matrix

4.3 ROC and Discriminative Ability

The Receiver Operating Characteristic curve shown in Figure 14 compares the discriminative power of all models. Models with curves closer to the top left corner exhibit improved separation between attack and normal classes.

Random Forest and XGBoost recorded the highest Area Under the Curve values, showing a strong discrimination capability. Followed by Logistic Regression and Decision Tree, while Naive Bayes showed weaker separation performance.

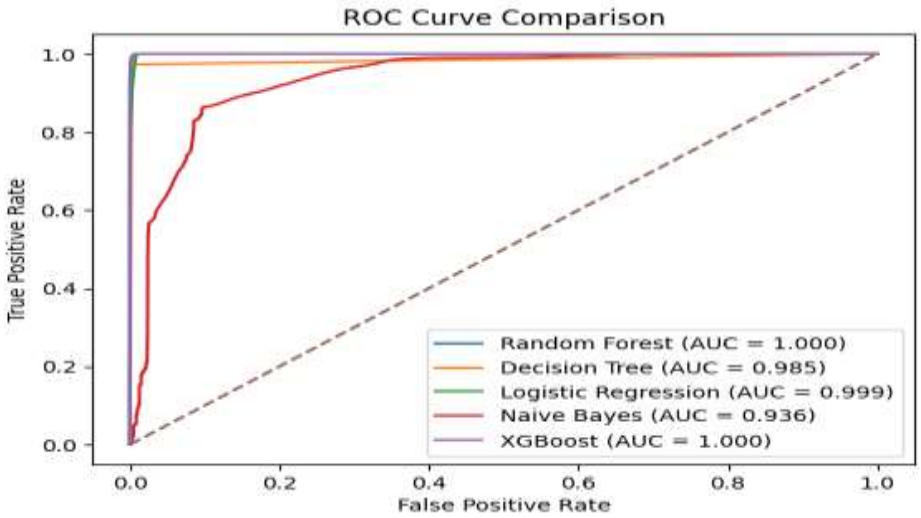

Figure 14: ROC Curve

Figure 15 represents the zoomed in ROC curves which provides a clearer comparison in the low false positive region. This region is very critical for IoT intrusion detection, where excessive false alarms can overpower limited system resources. XGBoost outperformed all other models even in this strict evaluation region.

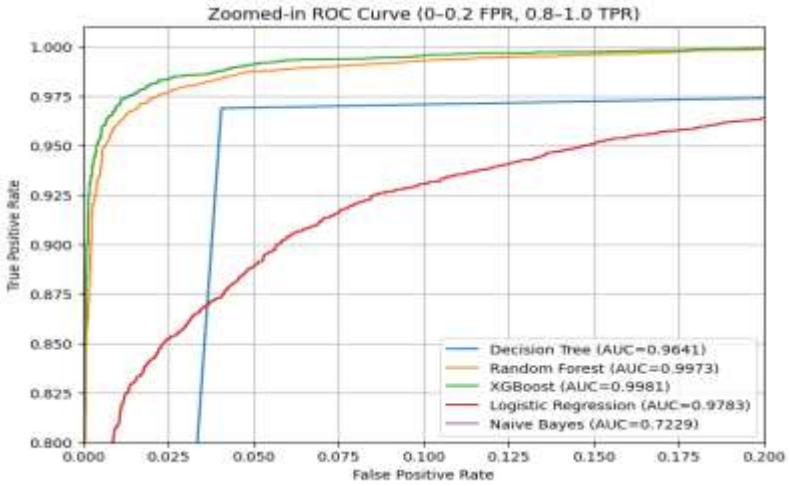

Figure 15: zoomed-in ROC

4.4 Calibration Analysis

Figure 16 shows the calibration curves for the evaluated models. Calibration is used to measure how well predicted probabilities reflect actual outcomes. This is important for risk based decision making in security systems.

The ensemble models recorded a better calibrated predictions, with curves closer to the diagonal reference line. Logistic Regression showed moderate calibration, while Naive Bayes exhibited overconfident probability estimates. Well-calibrated models allow administrators to set more reliable alert thresholds in real world IoT deployments.

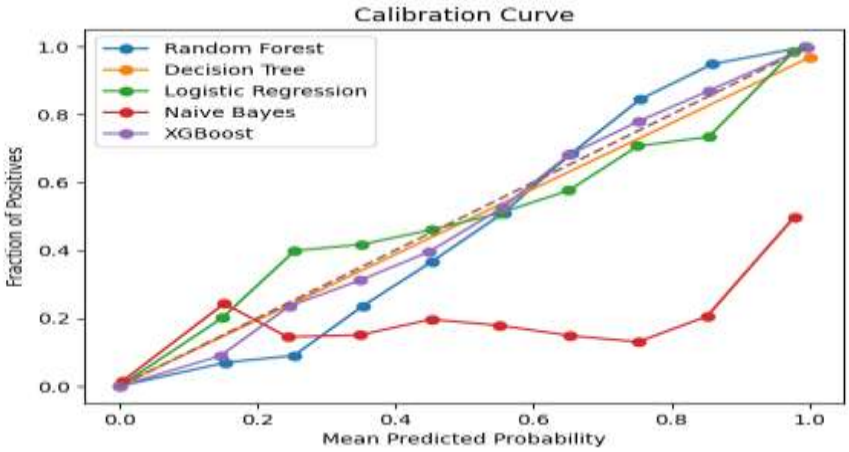

Figure 16: Calibration curve

4.5 Comparative Discussion

Figure 17 presents the general model performance comparison. The results clearly show that ensemble-based models outperform single classifiers in IoT intrusion detection tasks. Ensemble architecture's ability to learn complex patterns and reduce variance makes it more robust and a better choice against diverse and evolving attack types.

Overall, the findings confirm that ML-based intrusion detection is effective for IoT environments, with the emergence of XGBoost and Random Forest as the most reliable models. These models provide a strong balance between detection accuracy, false alarm reduction and probability reliability, making them a better option for practical IoT security solutions.

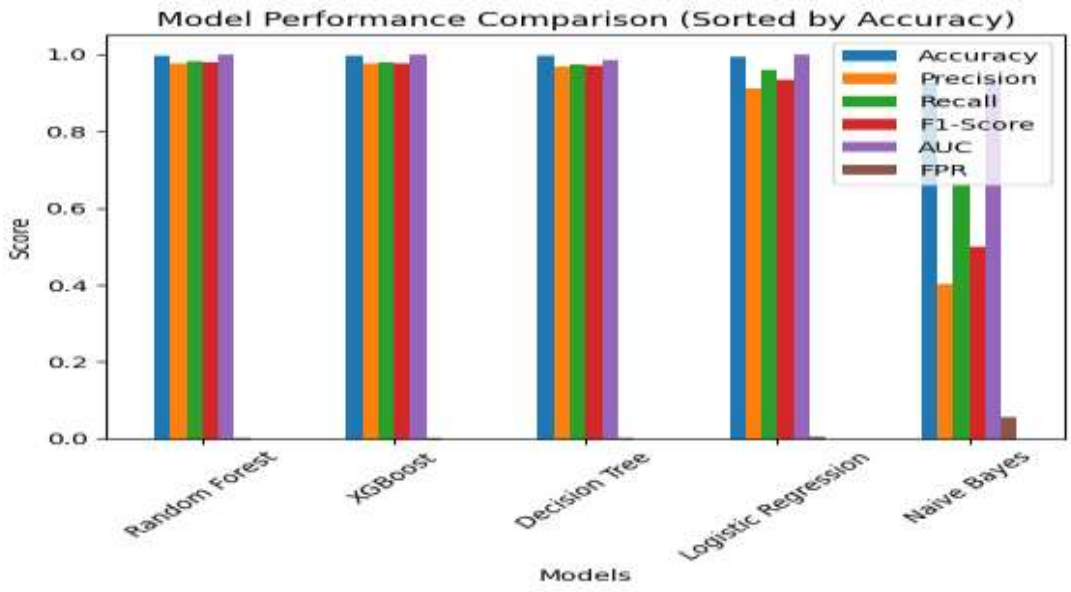

Figure 17: Model Performance Comparison

The experimental results show that ensemble models which include Random Forest and XGBoost perform better than other models. The models achieve their best results because they can model the complex feature interactions that exist in IoT network traffic data. The data from IoT traffic exhibits non-linear behavior because its features which include packet size and protocol type and flow duration and connection state show complex relationships. Models that fail to capture these interactions tend to underperform in intrusion detection tasks.

The Random Forest model performed well because it combines multiple decision trees which use different data and feature sets for their training. The model achieves better performance because it can recognize various traffic patterns while the system maintains its stability and prevents overfitting. Random Forest aggregate multiple decision boundaries that enables it to identify normal and malicious traffic patterns better which explains its low false positive and false negative identification according to the confusion

matrix results. The random feature selection mechanism enables the model to discover hidden interactions between features that may not be obvious when considering features independently.

XGBoost achieves its enhancements through sequential learning method, where each new tree focuses on correcting the errors of the previous ones. The model uses an iterative error correction system to focus more on the difficult-to-classify cases that include subtle and low-frequency attack patterns. As a result, XGBoost achieves superior ability to discriminate between classes especially in the low false positive region of the ROC curve. The built-in regularization system controls model complexity which ensure both high accuracy and strong generalization.

On the other hand, decision Tree can model non-linear relationships yet its single tree structure leads to problems with stability. The system shows higher misclassification rates because its decision boundaries change with minor dataset modifications. The system achieves decent performance yet it fails to deliver the same level of strength that ensemble techniques provide.

Naive Bayes recorded the lowest performance because it depended on its core requirement that features must function as an independent entities. In IoT network traffic, many features show strong correlations for instance, packet count, data volume or protocol type and connection state exhibit this relationship. Naive Bayes treats dependent features as independent which leads to inaccurate data distribution modeling that results in decreased recall and F1-score performance. This limitation becomes more pronounced in imbalanced datasets, where minority attack patterns require more expressive modeling.

Also, the detection of intrusions depends on feature interactions which function as essential components of the process. Ensemble models inherently capture higher-order interactions by combining multiple decision paths, allowing them to detect complex attack behaviors that may not be evident from individual features alone. This explains why both Random Forest and XGBoost demonstrate superior performance to traditional models because they achieve better results on all assessment criteria.

Overall, the research results demonstrate that the success of an intrusion detection systems models in IoT environment depends on their capacity to model non-linear relationships and their feature dependencies. Ensemble learning approaches, by design, provide this capability, making them more suitable for real-world IoT security applications where traffic patterns are dynamic, high-dimensional, and highly interdependent.

5. Conclusion

In conclusion from the study, securing the Internet of Things is not merely a technical hurdle but a fundamental requirement for the safety of our modern infrastructure. Our research into diverse machine learning models has highlighted the clear difference in their performance. The traditional methods like Logistic Regression and Naive Bayes offer simplicity, but they often find it difficult to keep pace with the complexity due to non-linear patterns that are found in modern network attacks. In a security context,

where the failure to detect a single threat can lead to a series of consequences, these constraints cannot be overlooked.

Hence, in our analysis, the ensemble models, especially Random Forest and XGBoost, outperformed other models.

These models reached extreme accuracy of up to 99% in our tests by relying on multiple decision-making paths and thus collecting the "wisdom". Besides, they showed a superior ability to minimize false alarms, which is crucial for preventing "alert fatigue" among system administrators. Random Forest and XGBoost, in particular, was shown to be very robust, indicating that their iterative learning process fits perfectly into the high-stakes environment of intrusion detection.

The development of highly intelligent systems has become the key to unlocking a completely secure Internet of Things ecosystem. Presently, our findings are encouraging, but the coming challenge will be the smooth functioning of these robust models on mobile devices with very little battery life and computing power. Future work will focus on integrating explainable artificial intelligence techniques such as SHAP and LIME to provide interpretable insights into model decisions, enabling security professionals to better understand, trust, and effectively deploy these models in real-world IoT environments. If effort is put in place to keep on closing the gap between state-of-the-art data science and real-world hardware limitations, a digital universe will be create that is not only interconnected but also resilient.

6. Conclusion

1. Ajagbe, S.A., Awotunde, J.B. & Florez, H. (2024). Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset. SN COMPUT. SCI. **5**, 1028. https://doi.org/10.1007/s42979-024-03369-0
2. Alharthi, A., Alaryani, M. & Kaddoura, S. (2025) A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems, Array, Volume 26, 100406, https://doi.org/10.1016/j.array.2025.100406.
3. Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study. Applied Sciences, 15(4), 1903. https://doi.org/10.3390/app15041903
4. Alturki, B., & Alsulami, A. A. (2025). Semi-Supervised Learning with Entropy Filtering for Intrusion Detection in Asymmetrical IoT Systems. Symmetry, 17(6), 973. https://doi.org/10.3390/sym17060973
5. Bibers, I., Arreche, O., Alayed, W., & Abdallah, M. (2025). Ensemble-IDS: An Ensemble Learning Framework for Enhancing AI-Based Network Intrusion Detection Tasks. Applied Sciences, 15(19), 10579. https://doi.org/10.3390/app151910579
6. Ghadami R. (2025). An intrusion detection system in the Internet of Things with deep learning and an improved arithmetic optimization algorithm (AOA) and sine cosine algorithm (SCA). Scientific reports, 15(1), 38156. https://doi.org/10.1038/s41598-025-22074-3
7. Hamidou, S. T., & Mehdi, A. (2025). Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks, Machine Learning with Applications, Volume 22, 100738, https://doi.org/10.1016/j.mlwa.2025.100738.

8. Hosain, Y., & Çakmak, M. (2025). XAI-XGBoost: an innovative explainable intrusion detection approach for securing internet of medical things systems. Scientific reports, 15(1), 22278. https://doi.org/10.1038/s41598-025-07790-0
9. Kalpani, N., & Rodrigo, N. (2026). Securing industry 4.0: a systematic review of AI-driven intrusion detection approaches and emerging trends. Journal of Reliable Intelligent Environments. https://doi.org/10.1007/s40860-025-00264-0
10. Kaddour, H., Das, S., Bajgai, R., Sanchez, A., Sanchez, J., Chiu, S. C., Ashour, A. F., & Fouda M. M. (2024). Evaluating the Performance of Machine Learning-Based Classification Models for IoT Intrusion Detection, 2024 IEEE Opportunity Research Scholars Symposium (ORSS), Atlanta, GA, USA, pp. 84-87, doi: 10.1109/ORSS62274.2024.10697949.
11. Kikissagbe, B. R., & Adda, M. (2024). Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review. Electronics, 13(18), 3601. https://doi.org/10.3390/electronics13183601
12. Kouassi, B. M., Ballo, A. B., Ayikpa, K. J., Mamadou, D., & Coulibaly, M. Z. J. (2025). Top-K Feature Selection for IoT Intrusion Detection: Contributions of XGBoost, LightGBM, and Random Forest. Future Internet, 17(11), 529. https://doi.org/10.3390/fi17110529
13. Mahfouz, A. M., Venugopal, D. & . Shiva, S. G (2020). Comparative Analysis of ML Classifiers for Network Intrusion Detection. *University of Memphis*. https://gtcs.cs.memphis.edu/pub/ahmed_UK.pdf
14. Mallick, C., Nayak, S., Singh, K.N. & Senapati, M. R. (2026). Securing the Internet of Things Through Intrusion Detection System Utilizing Machine Ensemble Learning and Feature Extraction Techniques. SN Computer Science. 7, 66 (2026). https://doi.org/10.1007/s42979-025-04648-0
15. Sharma, A. and Bhushan, K. (2026). A comprehensive survey on IoT security: Challenges, security issues, and countermeasures. Computer Science Review, Volume 59, 100839. https://doi.org/10.1016/j.cosrev.2025.100839.
16. Tekin N., Acar, A., Aris, A. A., Uluagac, S., & Gungor, V. C (2023) Energy consumption of on-device machine learning models for IoT intrusion detection, Internet of Things, Volume 21, 100670, https://doi.org/10.1016/j.iot.2022.100670.
17. Vitorino, J., Andrade, R., Praça, I., Sousa, O., Maia, E. (2022). A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. In: Aïmeur, E., Laurent, M., Yaich, R., Dupont, B., Garcia-Alfaro, J. (eds) Foundations and Practice of Security. FPS 2021. Lecture Notes in Computer Science, vol 13291. Springer, Cham. https://doi.org/10.1007/978-3-031-08147-7_13