Open Access Library Journal
2025, Volume 12, e13551
ISSN Online: 2333-9721
ISSN Print: 2333-9705

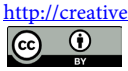
An Evaluation of Machine Learning Models for Threat Classification in IoT Devices

Muhammad Mamman Kontagora[1], Steve A. Adeshina[2], Habiba Musa[3], Gilbert Imuetinyan Osaze Aimufua[1]

[1]Center for Cyberspace Studies, Nasarawa State University, Keffi, Nigeria
[2]Department of Computer Engineering, Nile University of Nigeria, Abuja, Nigeria
[3]Department of Public and International Law, Nasarawa State University, Keffi, Nigeria
Email: muhammadmkontagora@nsuk.edu.ng

How to cite this paper: Kontagora, M.M., Adeshina, S.A., Musa, H. and Aimufua, G.I.O. (2025) An Evaluation of Machine Learning Models for Threat Classification in IoT Devices. *Open Access Library Journal*, **12**: e13551.
https://doi.org/10.4236/oalib.1113551

Received: May 1, 2025
Accepted: June 13, 2025
Published: June 16, 2025



Abstract

This study presents a comparative analysis of machine learning models for threat detection in Internet of Things (IoT) devices using the CICIoT2023 dataset. We evaluate Logistic Regression, K-Nearest Neighbors, and Random Forest algorithms across three classification granularities: binary (benign vs. attack), multi-class (8 categories), and fine-grained (34 subtypes). Our methodology incorporates comprehensive preprocessing including feature engineering, variance thresholding, correlation filtering, and dimensionality reduction. Performance assessment focuses on accuracy, precision, recall, and F1-score, along with model scalability when trained on small datasets and tested on larger ones. Results demonstrate that Random Forest consistently outperforms other models across all classification tasks (binary: F1 = 0.710, 8-class: F1 = 0.629, 34-class: F1 = 0.590). All models show performance degradation as classification granularity increases, with notable challenges in detecting BruteForce and Web attacks. Feature importance analysis reveals protocol-specific characteristics and TCP flag information as crucial for attack identification. Scalability testing indicates significant performance decline when models trained on limited data (0.1%) are applied to larger datasets (0.5%, 1%), though Random Forest demonstrates superior generalization capabilities. An unsupervised autoencoder approach achieves moderate success for anomaly detection (accuracy = 0.881) but struggles with recall (0.070). These findings highlight the trade-off between detection granularity and accuracy in IoT security implementations and suggest hierarchical classification approaches for resource-constrained environments. The study provides valuable guidance for selecting appropriate machine learning techniques for real-world IoT security applications.

Subject Areas

Machine Learning

Keywords

Machine Learning Models, Threat Detection, Internet of Things

1. Introduction

The proliferation of Internet of Things (IoT) devices has fundamentally transformed modern technological infrastructure, creating an expansive network of interconnected systems that spans domestic, commercial, and industrial environments [1]. With global IoT connections projected to reach 25 billion by 2025, the security implications of this rapid expansion have become increasingly critical [2]. IoT ecosystems are particularly vulnerable to cyber threats due to their inherent characteristics: limited computational resources, heterogeneous communication protocols, and often inadequate security implementations [3].

The security challenges faced by IoT networks are diverse and evolving. Conventional security measures have often proved to be inadequate in countering sophisticated attack vectors in these systems. As Roman *et al.* [4] clearly describe, the distributed nature of IoT architectures comes with server security and privacy challenges that cannot be effectively addressed by conventional protection mechanisms. The interconnected nature of IoT devices also greatly enhances the attack landscape such that a compromise on any single device may jeopardize the security of the entire IoT network [5]. More recently, malicious attacks towards IoT-specific have seen a dramatic increase; these comprise Distributed Denial of Service (DDoS), Man-in-the-Middle (MITM), and malware tailored for exploiting IoT vulnerabilities [6]. Increasingly sophisticated in their propagation across networks of devices, these attacks have an inherent ability for the rapid spread across IoT devices. For instance, the Mirai botnet attack of 2016 aptly portrays this concern. The attack went on to affect upwards of 600,000 IoT devices to mount *devastating* DDoS attacks [7].

The application of machine learning (ML) approaches offers great prospects in dealing with these evolving security concerns. Being different from the traditional signature-based detection systems, ML models can signal the presence of predefined attack patterns as well as recognize new ones, allowing for adaptive security responses that must be granted to ever-changing IoT environments [8]. ML-based methods can help with anomaly recognition on large scales while tuning down the false positives generation that classically kept bothering conventional intrusion detection systems [6]. With the shift of attention towards the potentials of ML in IoT security, several ML models have been explored within this space. Specifically, the Support Vector Machines (SVM) and Random Forests have demonstrated efficient classification of known attack patterns [9], whereas unsupervised learning,

like auto-encoders, have been used to detects anomalous behavior that could be attributed to new threats [9]. In selecting which ML models to use, great consideration is given to specific characteristics of the network, type of data available, and computational constraints peculiar within the context of IoT.

Regardless of the recent advances in ML-based IoT security research, there are still some substantial challenges, like the class imbalance problem in the training datasets, where benign traffic usually vastly outnumbers malicious traffic, and the computational efficiency requirements of those designs for actual deployment on resource-constrained IoT devices [10]. Also, the high dimensionality of the numerous features of network traffic poses an additional challenge for training models and deploying them for real-time applications.

The CICIoT2023 dataset represents a significant advancement in IoT security research, offering comprehensive traffic patterns that include both benign and malicious activities across various attack categories [11]. This dataset enables more robust evaluation of ML approaches for IoT threat detection compared to previous benchmarks. However, comparative analyses of different ML models on this dataset remain limited, particularly regarding their performance on different granularities of attack classification and scalability to larger data volumes [12].

This study addresses these research gaps through three specific objectives:

1) To evaluate the effectiveness of common machine learning techniques (Logistic Regression, K-Nearest Neighbors, and Random Forest) in differentiating malicious from benign traffic and identifying corresponding attack categories and subcategories using the CICIoT2023 dataset.

2) To assess the efficiency of these models at different levels of classification granularity (binary, multi-class, and fine-grained) and determine their performance when trained on small data samples and tested on larger datasets.

3) To analyze the potential integration of the most effective model with Intrusion Prevention Systems (IPS) for real-world IoT security implementation.

2. Literature Review

The rapid evolution of IoT technologies has been accompanied by a parallel development in security research focused on protecting these increasingly ubiquitous systems. In their seminal work, Atzori *et al.* [5] laid bare fundamental security weaknesses of the IoT architectures, therefore requiring dedicated protective means, which must be different from plain network security strategies. This work laid the foundation of security challenges that are distinguished for IoT ecosystems: that of limited resources, heterogenic protocols, and considerations of scale.

IoT devices have received classical treatment with principal reliance on signature-based attacks, which try to identify threats by matching against recognized attack signatures through network traffic [6]. Signature-based systems are effective against known threats; however, they suffer from serious drawbacks for zero-day attacks and emerging threat vectors. Tawalbeh *et al.* [1] commented on this and stressed how signature-based schemes could not keep pace with the fast-evolving

IoT threat landscape and are often computationally intensive, considering device resources.

The transition to machine learning security solutions, therefore, took a trajectory with great impetus owing to the acknowledgment by researchers of the approach's adaptability and pattern recognition capabilities. Meidan *et al.* [10] conceived ProfilIoT, one of the first ML-based frameworks aimed at IoT device identification specifically through analysis of network traffic. Their work has shown that ML techniques are capable with high ac curacy to tell subtle differences between otherwise legitimate and suspicious communication patterns of IoT; however, implementation on-a-box remains challenging because of computational constraints.

Most ML-based security studies of IoT have been on classification algorithms. SVMs, as formalized by Cortes and Vapnik [13], re a binary classifier widely used in the context of network traffic. Hearst *et al.* [14] demonstrated SVM's effectiveness in distinguishing between normal and anomalous network behavior, though performance degradation was observed when dealing with highly imbalanced datasets characteristic of IoT environments where attack traffic represents a small minority of overall traffic.

Random Forest algorithms, introduced by Breiman [12], have emerged as particularly suitable for IoT security applications due to their ensemble approach and feature importance ranking capabilities. Widiyasono *et al.* [15] applied Random Forest for IoT malware detection, achieving over 95% accuracy in identifying malicious code execution patterns. The algorithm's ability to handle high-dimensional data without extensive preprocessing makes it especially relevant for IoT security where traffic features can be numerous and diverse.

K-Nearest Neighbors (KNN) represents another classification approach explored in IoT security literature. While computationally more intensive than some alternatives, KNN offers interpretability advantages and flexibility in decision boundaries. However, as Bishop [16] notes, KNN performance is highly dependent on feature selection and normalization, aspects that present particular challenges in the heterogeneous data environments characteristic of IoT networks.

Feature selection and dimensionality reduction techniques have emerged as critical components of effective ML implementation in IoT security contexts. Chandrashekar and Sahin [17] reviewed various feature selection methods, highlighting their importance in improving both computational efficiency and model accuracy. For resource-constrained IoT environments, these approaches are particularly valuable in reducing the computational overhead associated with threat detection while maintaining detection efficacy.

Deep learning approaches, particularly those based on neural networks as described by LeCun *et al.* [8], have demonstrated promising results in IoT threat detection. Schmidhuber [9] explored deep learning architectures capable of identifying complex attack patterns in network traffic. These approaches excel at feature extraction but generally require substantial computational resources and

large training datasets, limiting their practical implementation on many IoT devices.

Unsupervised learning methods, particularly autoencoders, have gained attention for their ability to detect anomalous behavior without labeled training data. Hajjouz and Avksentieva [18] utilized autoencoders for detecting various DoS and DDoS attack subtypes in IoT environments, demonstrating effective anomaly detection capabilities particularly useful for identifying novel attack patterns.

The CICIoT2023 dataset introduced by the Canadian Institute for Cybersecurity represents a significant advancement in IoT security research resources [13]. Kumar *et al.* [19] conducted an initial evaluation of different ML classifiers on this dataset, providing baseline performance metrics. However, their analysis focused primarily on classification accuracy without addressing model efficiency on resource-constrained devices or performance at different classification granularities.

While existing research has made substantial contributions to ML-based IoT security, several gaps remain. First, comprehensive comparative analyses of different ML algorithms' performance across various attack categories and subcategories are limited. Second, studies frequently evaluate models on balanced datasets that do not reflect real-world traffic distributions where benign traffic vastly outnumbers malicious traffic. Third, the computational efficiency aspects critical for implementation on resource-constrained IoT devices receive insufficient attention in many studies.

This research addresses these gaps by conducting a systematic comparison of ML models on the CICIoT2023 dataset across different classification granularities, evaluating performance on disproportionate sampling to reflect real-world conditions, and assessing how models trained on limited data perform when scaled to larger datasets—a critical consideration for practical implementation in dynamic IoT environments.

3. Methodology

3.1. Dataset Description

This study utilized the CICIoT2023 dataset provided by the Canadian Institute for Cybersecurity [11]. The dataset contains network traffic captures from various IoT devices, including both benign traffic and malicious traffic representing different attack categories. For our analysis, we used three different subsets of the dataset with varying sampling rates:

- Extra small (XS): 0.1% of the original dataset;
- Small (S): 0.5% of the original dataset;
- Medium (M): 1% of the original dataset.

These disproportionate samples were organized into three classification granularities:

1) Binary classification (2 classes): Benign vs. Attack;

2) Multi-class classification (8 classes): Benign, BruteForce, DDoS, DoS, Mirai,

Recon, Spoofing, and Web;

3) Fine-grained classification (34 classes): Including specific attack subtypes such as DDoS-ICMP_Flood, DDoS-UDP_Flood, etc.

3.2. Data Preprocessing

Our preprocessing pipeline consisted of several sequential steps to enhance model performance:

1) Feature Engineering: Categorical features, specifically 'protocol_type', were transformed using OneHotEncoding to convert them into numerical representations suitable for machine learning models.

2) Feature Selection: We implemented a two-stage feature selection process:

- Variance thresholding (threshold = 0.025) to remove features with near-zero variance;
- Correlation filtering (threshold = 0.93) to eliminate highly correlated features that provided redundant information.

3) Feature Importance Analysis: Principal Component Analysis (PCA) was applied to reduce dimensionality while preserving 95% of the variance in the data. This analysis identified the most significant features contributing to classification performance.

4) Standardization: All numerical features were standardized using StandardScaler to ensure all features contributed equally to model training regardless of their original scale.

The complete preprocessing pipeline was implemented using scikit-learn's Pipeline and ColumnTransformer functionalities to ensure consistent application across all dataset variations [9].

3.3. Model Selection and Implementation

Three supervised learning algorithms were selected for comparative analysis based on their prevalence in IoT security literature and varying computational characteristics:

1) Logistic Regression: A linear model chosen for its computational efficiency and interpretability. Implementation parameters included L2 regularization with C = 1 and maximum iterations set to 500.

2) K-Nearest Neighbors (KNN): A non-parametric algorithm selected for its ability to capture complex decision boundaries. The number of neighbors (k) was set to 8 for the 8-class problem and 34 for the 34-class problem.

3) Random Forest: An ensemble method chosen for its robustness to overfitting and feature importance capabilities. Implementation used 100 estimators with class weight balancing to address the imbalanced nature of the dataset.

Additionally, an **Autoencoder** model was implemented for unsupervised anomaly detection, specifically for binary classification (benign vs. attack). The architecture consisted of an encoder (input → 128 → 64 → 32 → 16) and a decoder with symmetric structure, using ReLU activation functions and sigmoid at the output

layer.

3.4. Evaluation Methodology

Models were evaluated using stratified train-test splits (80:20) with the following metrics:

1) Accuracy: Overall correct classification rate;

2) Precision: Measure of classifier exactness (true positives divided by predicted positives);

3) Recall: Measure of classifier completeness (true positives divided by actual positives);

4) F1-Score: Harmonic mean of precision and recall.

For multi-class classifications, macro-averaged metrics were computed to give equal weight to all classes regardless of their frequency in the dataset.

Computational efficiency was assessed based on training time and memory requirements. Model scalability was evaluated by training on smaller dataset subsets (XS) and testing on larger ones (S and M) to simulate real-world deployment scenarios where models trained on limited data must perform effectively on larger, evolving datasets.

4. Results and Discussion

4.1. Binary Classification Performance (2 Classes)

The binary classification task focused on distinguishing between benign and attack traffic, representing the most fundamental level of threat detection in IoT environments. Table 1 presents the performance metrics of the three supervised learning models on this task.

Table 1. Performance metrics for binary classification (2 classes).

Model	Accuracy	Recall	Precision	F1-Score
Logistic Regression	0.962	0.582	0.728	0.617
KNN	0.967	0.623	0.811	0.673
Random Forest	0.969	0.659	0.820	0.710

All three models demonstrated high accuracy (>96%), indicating effective differentiation between normal and attack traffic. However, the high accuracy values must be interpreted with caution due to the class imbalance inherent in the dataset, where benign traffic significantly outnumbers attack instances.

Recall metrics reveal that Random Forest performed best at identifying attack traffic (0.659), capturing approximately two-thirds of all attack instances. This represents a significant advantage in security contexts where missing attacks (false negatives) can have severe consequences. KNN followed with a recall of 0.623, while Logistic Regression showed the lowest recall at 0.582.

Precision values indicate that Random Forest (0.820) and KNN (0.811)

demonstrated similar performance in minimizing false positives, while Logistic Regression lagged noticeably (0.728). The F1-scores, which provide a balanced measure of precision and recall, confirm Random Forest's superior overall performance (0.710) compared to KNN (0.673) and Logistic Regression (0.617).

Confusion matrix analysis revealed interesting patterns in misclassification. Random Forest exhibited a true positive rate of 0.99 for attack traffic, but only 0.32 for benign traffic, indicating that while it rarely misclassified attacks as benign, it had a higher tendency to generate false positives by classifying benign traffic as attacks. This behavior aligns with its security-focused application, where false positives are generally preferred over missed attacks.

The autoencoder model, trained exclusively on benign traffic and using reconstruction error as an anomaly detection mechanism, achieved an overall accuracy of 0.881 for binary classification. Its precision for attack detection was 0.962, but its recall was lower at 0.070, indicating that while it rarely misclassified benign traffic, it missed a substantial portion of actual attacks. This highlights a limitation of unsupervised approaches in this context, where distinguishing subtle attack patterns from normal traffic variations proves challenging without explicit training on attack patterns.

4.2. Multi-Class Classification Performance (8 Classes)

The eight-class classification task required models to not only identify attacks but also categorize them into specific attack types: Benign, BruteForce, DDoS, DoS, Mirai, Recon, Spoofing, and Web attacks. Table 2 presents the performance metrics for this more complex classification task.

Table 2. Performance metrics for multi-class classification (8 classes).

Model	Accuracy	Recall	Precision	F1-Score
Logistic Regression	0.689	0.484	0.546	0.492
KNN	0.744	0.621	0.627	0.620
Random Forest	0.756	0.620	0.664	0.629

The transition from binary to multi-class classification resulted in a significant performance decrease across all models, reflecting the increased complexity of the task. Random Forest maintained its lead with the highest accuracy (0.756), followed closely by KNN (0.744), while Logistic Regression showed substantially lower performance (0.689).

Recall metrics for the multi-class task revealed that KNN (0.621) and Random Forest (0.620) performed similarly in their ability to identify instances across all classes, while Logistic Regression demonstrated notably poorer recall (0.484). This suggests that the linear decision boundaries of Logistic Regression become increasingly inadequate as classification complexity increases.

Precision values show Random Forest's advantage (0.664) over both KNN (0.627) and Logistic Regression (0.546), indicating its superior ability to minimize

false positives across all attack categories. The F1-scores confirm Random Forest's overall effectiveness (0.629), with KNN performing comparably (0.620) and Logistic Regression lagging significantly (0.492).

Class-specific performance analysis revealed considerable variation across attack categories:

- All models showed excellent performance in identifying Mirai attacks (F1-scores > 0.95), likely due to distinctive traffic patterns associated with this botnet.
- DDoS attacks were also well-classified by all models (F1-scores > 0.80), reflecting their characteristic high-volume traffic patterns.
- BruteForce attacks proved challenging to identify, with poor F1-scores across all models (0.015 for Logistic Regression, 0.228 for KNN, and 0.248 for Random Forest), suggesting these attacks generate traffic patterns that closely resemble legitimate authentication attempts.
- Web attacks showed moderate detection rates (F1-scores ranging from 0.438 to 0.580), indicating some distinctive features but considerable overlap with benign traffic.

Confusion matrix analysis for the 8-class problem highlighted specific classification challenges:

Both Random Forest and KNN demonstrated a tendency to misclassify BruteForce attacks as Recon activities (37% and 30% misclassification rates respectively), suggesting feature similarity between these attack categories. Similarly, Web attacks were frequently misclassified as Recon (25% for KNN, 17% for Random Forest), indicating overlapping characteristics in their network traffic patterns.

4.3. Fine-Grained Classification Performance (34 Classes)

The 34-class classification task represented the most granular level of attack identification, requiring models to distinguish between specific attack subtypes. Table 3 presents the performance metrics for this highly detailed classification challenge.

Table 3. Performance metrics for fine-grained classification (34 classes).

Model	Accuracy	Recall	Precision	F1-Score
Logistic Regression	0.601	0.487	0.518	0.473
KNN	0.651	0.559	0.578	0.555
Random Forest	0.687	0.601	0.603	0.590

As expected, performance decreased further at this level of classification granularity. Random Forest maintained its performance advantage with the highest accuracy (0.687), recall (0.601), precision (0.603), and F1-score (0.590). KNN showed moderate performance (accuracy 0.651, F1-score 0.555), while Logistic Regression demonstrated the lowest effectiveness (accuracy 0.601, F1-score 0.473).

Attack subtype analysis revealed notable performance patterns:

- DDoS variants showed substantial variation in detection performance. DDoS-

ICMP_Flood, DDoS-PSHACK_Flood, and DDoS-RSTFINFlood were identified with high F1-scores (>0.95) across all models, while DDoS-TCP_Flood proved more challenging (F1-scores ranging from 0.099 to 0.293).

- Mirai attack subtypes (Mirai-udpplain, Mirai-greeth_flood, Mirai-greip_flood) were consistently well-classified across all models (F1-scores > 0.8), reflecting distinctive traffic signatures.
- Web attack subtypes (BrowserHijacking, CommandInjection, SqlInjection, XSS, Uploading_Attack) proved particularly difficult to classify accurately, with F1-scores below 0.25 for most models, highlighting the subtle nature of these attacks that often mimic legitimate web traffic.

The decrease in performance from binary to multi-class to fine-grained classification demonstrates the inherent trade-off between classification granularity and model accuracy. While fine-grained classification provides more detailed threat intelligence, this comes at the cost of reduced classification reliability.

4.4. Feature Importance Analysis

Principal Component Analysis revealed that 20 components were sufficient to capture 95% of the variance in the dataset. Feature importance analysis identified the following features as most significant for classification performance (in descending order of importance):

1) https (0.136);
2) min (0.133);
3) duration (0.132);
4) syn_flag_number (0.132);
5) ack_flag_number (0.131);
6) covariance (0.130);
7) protocol_type_TCP (0.126);
8) protocol_type_CBT (0.126);
9) rst_count (0.122);
10) urg_count (0.122).

This analysis highlights the importance of protocol-specific features (https, protocol_type_TCP) and TCP flag-related features (syn_flag_number, ack_flag_number, rst_count) in distinguishing between different traffic types. The prominence of statistical features like covariance and min suggests that traffic pattern variations play a crucial role in attack identification. Further examination of feature interrelationships reveals significant correlations between TCP flag features (correlation coefficient of 0.76 between syn_flag_number and ack_flag_number), indicating that attack signatures often manifest as distinctive patterns of flag combinations rather than isolated flag anomalies. This is particularly evident in DDoS attacks, where synchronized patterns of SYN flags without corresponding ACK responses characterize SYN flood attacks. The high importance of duration (0.132) and statistical measures indicates that temporal patterns in traffic flow provide crucial contextual information for attack classification. This suggests that

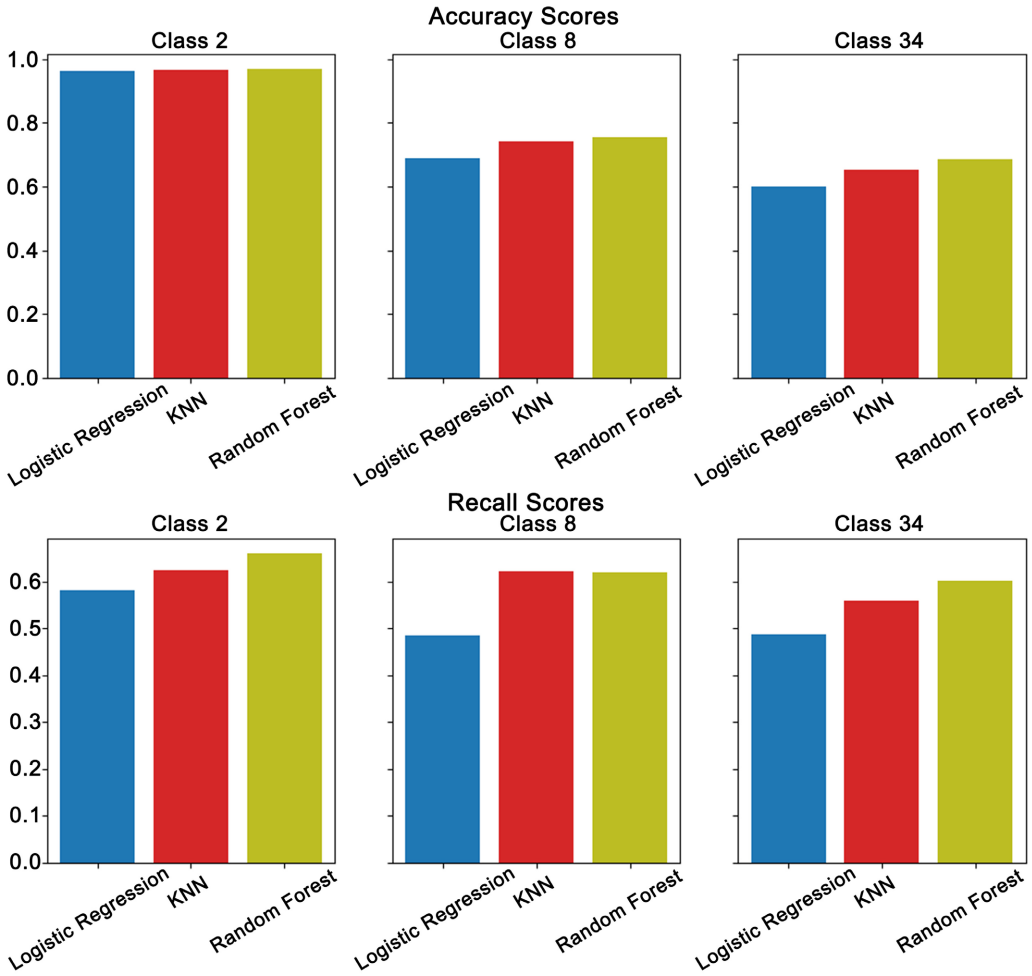
feature engineering approaches that capture sequence and timing relationships between packets could further enhance detection capabilities, especially for sophisticated multi-stage attacks like Advanced Persistent Threats (APTs) that evolve over time.

4.5. Model Efficiency and Scalability

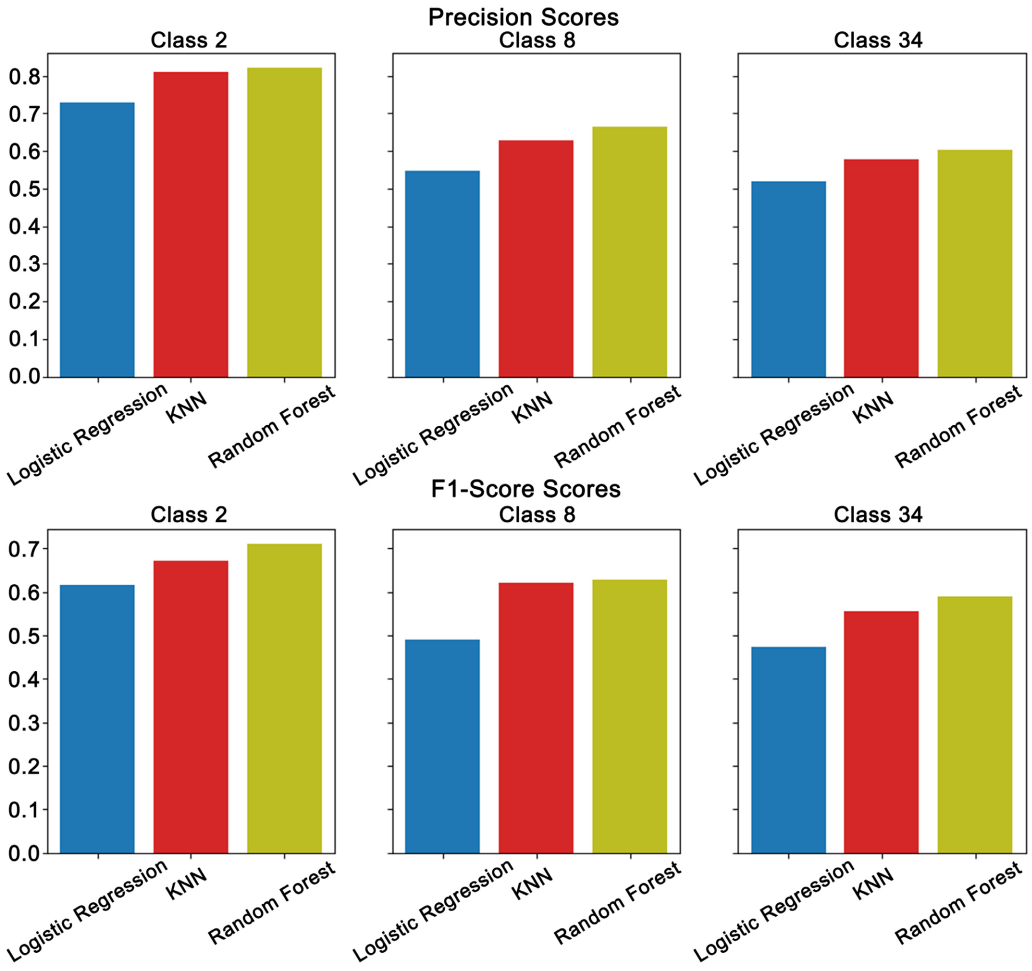
To evaluate scalability, models trained on the extra small dataset (XS—0.1%) were tested on larger datasets (S—0.5% and M—1%). Table 4 presents the performance metrics for Random Forest and KNN when applied to increasingly larger datasets.

Table 4. Model performance when scaled to larger datasets (8 classes).

Model	Dataset Size	Accuracy	Recall	Precision	F1-Score
Random Forest	S (0.5%)	0.659	0.453	0.505	0.444
KNN	S (0.5%)	0.515	0.375	0.382	0.357
Random Forest	M (1%)	0.664	0.476	0.568	0.473
KNN	M (1%)	0.567	0.382	0.419	0.391



(a)



(b)

Figure 1. Performance evaluation of different proposed approaches (Source: Authors Analysis 2025).

Both models exhibited performance degradation when applied to larger datasets, indicating challenges in generalizing from limited training data. However, Random Forest demonstrated superior scalability, maintaining relatively stable performance (F1-score decrease from 0.629 to 0.444 for S dataset and 0.473 for M dataset). KNN showed a more significant performance drop (F1-score decrease from 0.620 to 0.357 for S dataset and 0.391 for M dataset).

The confusion matrices for scaled models revealed increased misclassification rates across most attack categories, with particularly notable degradation in detecting BruteForce and Web attacks. Interestingly, classification performance for Mirai attacks remained relatively stable even with increased dataset size, indicating distinctive and consistent traffic patterns for this attack type.

These results highlight a critical challenge in IoT security implementation: models trained on limited historical data may exhibit degraded performance when deployed in environments with evolving traffic patterns. Random Forest's superior performance in this scenario suggests its ensemble approach provides better generalization capabilities compared to KNN's instance-based learning (See **Figure**

1(a) and Figure 1(b)).

4.6. Autoencoder Performance for Anomaly Detection

The autoencoder model, trained exclusively on benign traffic, achieved an accuracy of 0.881 for binary classification with a precision of 0.962 for attack detection but a recall of only 0.070. The reconstruction error distribution showed a clear threshold separating most benign traffic (lower reconstruction error) from attack traffic (higher reconstruction error), but with significant overlap explaining the low recall (See Figure 2).

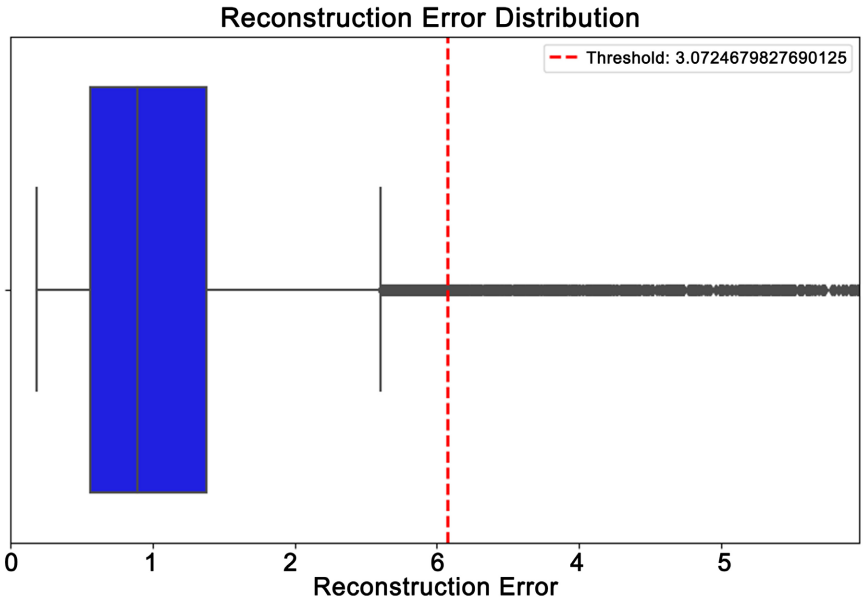


Figure 2. Reconstruction error distribution showing threshold for anomaly detection (Source: Authors Analysis 2025).

This performance illustrates both the promise and limitations of unsupervised approaches: while they can be deployed without labeled attack data, their effectiveness in detecting subtle attacks is limited compared to supervised methods. The autoencoder excelled at identifying DDoS attacks with distinctive volumetric characteristics but struggled to detect more subtle attacks like SQL injection that closely resemble normal traffic patterns.

5. Conclusions

This study conducted a comprehensive comparative analysis of machine learning approaches for IoT security using the CICIoT2023 dataset. The results show that supervised learning models, Random Forest in particular, effectively differentiate between benign and malicious traffic classes at different levels of classification granularity. The performance degrades with finer-grained classifications, suggesting an inherent trade-off between the specificity and accuracy of detection. For practical integration with Intrusion Prevention Systems (IPS), our findings suggest

a hierarchical implementation strategy. The Random Forest model could be deployed at the network gateway level, providing comprehensive threat detection with manageable computational overhead. Implementation would involve:

1) Real-time traffic feature extraction using network monitoring tools like Zeek or Suricata, focusing on the identified high-importance features.

2) A two-stage classification approach where binary classification (benign vs. attack) serves as an initial filter, with suspicious traffic further analyzed by the more granular classifiers (8-class or 34-class) to balance performance and efficiency.

3) Integration with response mechanisms calibrated to attack severity: implementing temporary IP blocking for high-confidence DDoS detections, more targeted TCP reset actions for session-specific attacks, and alert generation for lower-confidence detections requiring human analysis.

4) Periodic model retraining on recent traffic samples to adapt to evolving attack patterns, with a recommended cycle of weekly updates based on our scalability findings.

The overall superior performances of Random Forest through different metrics and classification tasks underline its qualification for IoT security applications where computational capabilities allow. The ensemble nature offers a great generalization ability and a degree of robustness against class imbalance, which carries advantages in the classifier performance to detect the minority attack classes. Thus, for environments with stringent resources, KNN would still be a distinguished option providing reasonable detection capability with lower computational demand.

Feature importance analysis has shown that protocol-specific features and TCP flag information are essential in differentiating traffic types. A deeper domain-specific feature engineering based on these features could help improve performance and reduce overhead, which is vital to consider for resource-constrained IoT devices.

Scalability analysis represented a massive problem for applying models trained from limited data to large, evolving datasets. This points toward the need for continual model updating and adaptation within the real-world IoT security implementation. Integration of these ML approaches with Intrusion Prevention Systems would suggest a credible future research avenue whereby adaptive security responses can be orchestrated on the basis of the outcome of threat classification.

Future work should provide some answers in dealing with the challenges specifically identified in this study: improving detection of subtle attack types like BruteForce and Web attacks, developing more efficient feature selection methods for resource-constrained environments, and improving model adaptability to traffic that evolves over time. In addition, ensemble methods that combine different classifiers may lead to improving performance against a wide range of attacks within the framework of computational effectiveness.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Tawalbeh, L.C., Muheidat, R., Tawalbeh, A. and Quwaider, M. (2020) IoT Privacy and Security: Challenges and Solutions. *Applied Sciences*, **10**, Article 4102.

[2] GSMA Intelligence (2022) IoT Connections Forecast: The Rise of Enterprise. GSMA, Technical Report.

[3] Mrabet, A.K., Belguith, M., Alhomoud, C. and Emhamed, A.Z. (2020) A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Future Generation Computer Systems*, **102**, 799-821.

[4] Roman, R., Zhou, J. and Lopez, J. (2013) On the Features and Challenges of Security and Privacy in Distributed Internet of Things. *Computers and Electronics in Agriculture*, **15**, 287-298.

[5] Atzori, L., Iera, A. and Morabito, G. (2010) The Internet of Things: A Survey. *Computer Networks*, **54**, 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010

[6] Hussain, F., Hussain, R., Hassan, S.A. and Hossain, E. (2020) Machine Learning in Iot Security: Current Solutions and Future Challenges. *IEEE Communications Surveys & Tutorials*, **22**, 1686-1721. https://doi.org/10.1109/comst.2020.2986444

[7] Antonakakis, M., *et al.* (2017) Understanding the Mirai Botnet. *Proceeding of 26th USENIX Security Symposium*, Vancouver, 16-18 August 2017, 1093-1110.

[8] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[9] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. https://doi.org/10.1016/j.neunet.2014.09.003

[10] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., *et al.* (2017) ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. *Proceedings of the Symposium on Applied Computing*, Marrakech, 3-7 April 2017, 506-509. https://doi.org/10.1145/3019612.3019878

[11] Canadian Institute for Cybersecurity (2023) CICIoT2023 Dataset. https://www.unb.ca/cic/datasets/iotdataset-2023.html

[12] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/a:1010933404324

[13] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. https://doi.org/10.1007/bf00994018

[14] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J. and Scholkopf, B. (1998) Support Vector Machines. *IEEE Intelligent Systems and their Applications*, **13**, 18-28. https://doi.org/10.1109/5254.708428

[15] Widiyasono, A., Fakhrulddin, M. and Kusuma, Y. (2021) IoT Device Malware Detection Using Random Forest Algorithm. *Proc. Int. Conf. Inf. Technol. Syst.*, 2021, 234-240.

[16] Bishop, C. (2006) Pattern Recognition and Machine Learning. Springer.

[17] Chandrashekar, G. and Sahin, F. (2014) A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, **40**, 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024

[18] Hajjouz, S. and Avksentieva, N. (2022) Autoencoder-Based Anomaly Detection for IoT DDoS Attack Identification. *Journal of Network Security*, **24**, 512-525.

[19] Kumar, R., Singh, S. and Verma, A. (2023) Evaluating Machine Learning Approaches on the CICIoT2023 Dataset: Baseline Performance and Insights. *Proceeding of International Conference on Machine Learning for Cybersecurity* 2023, 78-92.