**Journal of ICT, Design, Engineering and Technological Science (JITDETS)**
VOL. 8, NO. 2, pp. 1-5, 2024
DOI: https://doi.org/10.33150/JITDETS-8.2.1

ORIGINAL CONTRIBUTION

# Evaluating Machine Learning Models for Real-Time IoT Intrusion Detection: A Comparative Study with RTSS Analysis

Ahmed Alwan [1*], Asadullah Shah [2], Alwan Abdullah Abdul Rahman Alwan [3], Shams Ul Arfeen Laghari [4]

[1,2]Kulliyyah of Information & Communication Technology, International Islamic University, Kuala Lumpur, Malaysia

[3]National Advanced IPv6 Centre, Universiti Sains Malaysia, Penang, Malaysia

[4]School of ICT (EDICT), Bahrain Polytechnic, Isa Town, Bahrain

***Abstract*— With the ever-increasing sophistication and volume of cyber-attacks, there is a critical need for effective intrusion Detection Systems (IDS) to protect computer networks. Machine Learning (ML) offers powerful tools for IDS by automatically identifying patterns of malicious behavior. This research proposal aims to evaluate and compare the performance of several supervised ML algorithms for network threat detection using the CICIDS 2023 dataset. This paper focuses on widely-used classifiers – Logistic Regression, Support Vector Machine (SVM), Random Forest, eXtreme Gradient Boosting (XGBoost), and k-Nearest Neighbors (KNN) – applied to both binary (benign vs. attack) and multi-class (multiple attack types) classification tasks. This paper outlines a methodology for data preprocessing, model training, and performance evaluation using metrics like accuracy, precision, recall, and F1-score. By leveraging the comprehensive CICIDS 2023 intrusion dataset, which includes 33 modern attack scenarios across seven categories, this paper expects to gain insights into the relative strengths of each ML approach in detecting diverse cyber threats. The anticipated outcome is an identification of which algorithms (or combination thereof) are most promising for intrusion detection in contemporary network environments, guiding future developments of intelligent IDS. This proposal details the problem motivation, related work, planned methodology, and expected results, establishing a foundation for a thorough experimental study.**

***Index Terms*— Intrusion detection, Machine learning, Supervised learning, Cybersecurity, Network security, CICIDS 2023, Comparative analysis**

**Received:** 14 July 2024; **Accepted:** 20 September 2024; **Published:** 26 December 2024




## I. INTRODUCTION

Cybersecurity attacks are escalating in both volume and sophistication.

According to the National Oceanic and Atmospheric Administration (NOAA) and industry estimates, the global financial damage caused by cybercrime almanac [1]. Since 2023, cyber-attacks have cost more than $8 trillion, and Distributed Denial of Service (DDoS), ransomware, and data exfiltration are revealed to be among the top threats. Recognizing the need for more robust and responsive threat detection among organizations in critical infrastructure, finance, and healthcare sectors, to name but a few has suffered significant disruptions.

Intrusion Detection Systems (IDS) are important security mechanisms to monitor network traffic, host behavior, or a combination of these to detect violations of unauthorized access, abuse, or policy [2, 3]. Traditional IDS technologies based on signature-based (looking for a given attack pattern) and anomaly-based (identifying any anomaly in traffic) techniques have limitations. However, signature-based methods are not effective as these attacks are not caught (zero-day or obfuscated attacks), and anomaly-based methods have a high rate of false positives [4].

Now, Machine Learning (ML) has come as a powerful approach by which it learns and can specialize in the patterns from the existing labeled set and generalize to a new, unknown threat [5]. ML-based IDS can identify patterns that are nonlinear nor complex, as well as dynamically classify traffic as malicious or benign [6, 7]. However, most of the existing ML-based IDS solutions rely on outdated datasets (e.g., KDD'99, NSL-KDD) trained on which do not adequately cover the complexity and diversity of modern attack vectors, especially in the IoT space[8, 9].

In this research, the problem of selecting and evaluating supervised ML algorithms for IDS using contemporary and realistic data is addressed. The scope of this work is to study classical ML models (Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, and k-Nearest Neighbors) empirically on the recently released CICIDS 2023 dataset [10, 11] and fill up a critical gap in the literature. The capture of the dataset took place in a large-scale IoT network testbed to provide a realistic and large-scale platform where IDS can be benchmarked.

To fill this gap, this paper seeks to explore both binary and multi-class classification (telling whether an attack happened or one of several known attack types). The hypothesis of this paper is that the ensemble models,

*Corresponding author:Ahmed Alwan
†Email: ahmed.alwan@live.iium.edu.my

Random Forest, and XGBoost, will have better detection accuracy and robustness to different attacks over more simple algorithms when applied to multi-class settings. This paper will also assess how the tradeoff between model performance and computational efficiency will lead to the selection of which algorithms are appropriate for real-time execution.

This paper aims to recommend the most productive and viable next-generation IDS solutions to a thorough evaluation of these ML algorithms on the CICIDS 2023 dataset. The contributions of this work will facilitate the construction of future cybersecurity systems systems that are data-driven, scalable, and efficient in nature.

## II. RELATED WORK

In recent years, Machine Learning (ML) has been applied to intrusion detection systems (IDS) and attracted great attention [12]. However, due to their ability to generalize and detect unseen attacks, traditional approaches such as signature and anomaly based, are being replaced or complemented by such ML models. Nevertheless, most of the initial effort associated with the study of ML based IDS has been predicated upon aging quality indictors such as KDD'99 and NSL KDD [13] that do not conform with the variety of modern cyber threats.

More recently, there is practical incentive to use more realistic data sets. For example, Decision Trees, Random Forest, and Support Vector Machines have been evaluated comparatively on UNSW-NB15 in [5]. The study verifies that ensemble methods, most mostly Random Forest, consistently worked to reach high accuracy. Nevertheless, computational efficiency and interpretability were not considered.

[14] introduced the CICIoT2023 dataset to simulate large-scale attacks in IoT environments. Their baseline evaluation demonstrated that classical ML models, particularly Random Forest and simple neural networks, provided competitive results in terms of F1-score. While the dataset offered improved realism, the study focused solely on classification accuracy, excluding practical concerns like latency, model size, or explain ability.

[15] proposed a lightweight two-tier IDS architecture designed for smart home IoT environments. The approach achieved promising results by balancing edge-level detection with cloud-level validation. Nevertheless, the evaluation lacked an analysis of resource consumption and did not incorporate model interpretability metrics.

Comprehensive reviews such as those by [16] and [17] emphasized the tradeoffs between classical machine learning and deep learning in intrusion detection. While deep learning methods often yield higher accuracy, they typically involve greater computational cost and lower transparency, which can limit their suitability in real-time or resource-constrained deployments.

[18] addressed detection performance through a feature selectionenhanced ensemble learning model. Their implementation of Recursive Feature Elimination (RFE) led to improvements in detection accuracy and generalization. However, the study did not evaluate the interpretability of the models or analyze inference time, both of which are critical factors in real-world IDS deployment.

Although the cited works provide valuable insights into ML-based IDS design, most of them focus narrowly on detection metrics. There remains a lack of research that integrates classification performance with operational feasibility, such as inference speed, memory footprint, and model transparency. The present study addresses this gap by evaluating multiple classical ML algorithms on a recent benchmark dataset with a focus on explainability and real-time deployability.

## III. METHODOLOGY

This section outlines the dataset used, data preprocessing steps, machine learning models selected for evaluation, the explainability approach, and the formulation of the Real-Time Suitability Score (RTSS), which is introduced to assess deployment feasibility alongside detection performance.

### A. Dataset description

All evaluations are based on the CICIDS 2023 dataset, which was created by the Canadian Institute for Cybersecurity. It contains more than 100 devices and a 'realistic mix' of IoT and non-IoT devices in a simulated, realistic network environment. Benign traffic and 33 types of different attacks are grouped into seven categories, namely Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance, Brute Force, Web Based, Spoofing, and Mirai attacks [6]. The flows in the dataset are labeled, and each instance has about 47 features, including packet level statistics, flow behavior, and protocol-specific attributes, and the number of labels over 50 million.

### B. Data preprocessing

Prior to model training, standard data preprocessing procedures are applied. First, duplicate and null values are removed. All continuous features are scaled using min-max normalization to improve learning stability for algorithms sensitive to feature magnitudes, such as Support Vector Machines (SVM) and K-nearest Neighbors (KNN). Categorical features in the dataset, including protocol and flag indicators, are already encoded numerically, so no additional encoding is necessary.

To manage class imbalance—particularly in the multi-class classification task—stratified sampling is used during data partitioning to maintain representative distributions across training and testing subsets. The dataset is split into 80% training and 20% testing partitions. Feature selection is guided by correlation analysis and variance thresholding to remove highly collinear or uninformative features.

### C. Machine learning algorithms

Five classical supervised learning algorithms are selected based on their widespread use in intrusion detection literature and their balance between complexity and interpretability:

#### *1) Logistic Regression (LR)*

A linear model used as a baseline for binary and multi-class classification, incorporating L2 regularization to mitigate overfitting.

#### *2) Support Vector Machine (SVM)*

Implemented with the radial basis function (RBF) kernel to capture nonlinear decision boundaries.

#### *3) Random Forest (RF)*

An ensemble of decision trees trained using bootstrap aggregation, known for robustness in high-dimensional spaces.

#### *4) XGBoost*

A gradient-boosting framework that builds decision trees sequentially, optimizing for performance through iterative error minimization.

#### 5) K-Nearest Neighbors (KNN)

A non-parametric method that classifies based on proximity to labeled instances, with distance computed in normalized feature space.

All models are implemented using scikit-learn and XGBoost libraries in Python. Hyperparameter tuning is performed via grid search with 5-fold cross-validation, using the F1-score as the primary selection criterion.

#### 6) Explainability via SHAP

With the purpose of increasing the interpretability, SHapley Additive exPlanations (SHAP) are used to quantify the contribution that each feature makes to the output of the trained models [19]. The SHAP values are global and local explanations that explain the most important features for various attack types. Such is especially crucial in operational settings where model transparence and responsibility matter [20].

#### 7) Real-Time Suitability Score (RTSS)

This paper introduces RTSS or Real-Time Suitability Score to assess deployment feasibility through a combined measure. RTSS represents how detection system performance relates to resource consumption rates. It is defined as:

**F1-score**

RTSS = (1) Inference Time (ms) × Model Size (MB)

Since imbalanced datasets require a performance metric with robust properties, the F1-score has been chosen. The research calculates inference time for each 1,000 test samples. Model size equals the total memory consumption of a serialized model expressed in megabytes. RTSS serves as a useful tool that makes model selection more practical for real-time and edge deployments.

## IV. RESULTS AND DISCUSSION

Five machine learning classifiers namely Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), XGBoost (XGB) as well as k-Nearest Neighbors (KNN) were evaluated on the cleaned CICIoT2023 dataset. This paper evaluated the system performance by analyzing accuracy in combination with precision, recall, F1-score, inference time, model size along with their created Real-Time Suitability Score (RTSS).

### A. Classification performance

All generated models demonstrated excellent performance in predicting maintenance requests according to F1 scores and accuracy metrics, although ensemble methods such as XGB and RF exhibited minimally superior metrics results. The F1-score result for Logistic Regression was 0.9893, whereas Linear SVM obtained 0.9998 with minimal computational requirements. Due to size and processing time costs, XGBoost, alongside Random Forest, showed perfect classification success, but XGBoost had the shortest single run time while Random Forest had the longest. KNN produced an F1-score of 0.9999, yet its processing time became impractically long as the algorithm operates using an instance-based computation method.

Table 1 summarizes the performance metrics for each model, and Figures 1 to 5 display the confusion matrices.

TABLE I
PERFORMANCE SUMMARY OF EVALUATED MODELS

| Model | Acc. | Prec. | Rec. | F1 | Time (ms) | Size (MB) |
|---|---|---|---|---|---|---|
| LR | 0.9855 | 0.9790 | 0.9998 | 0.9893 | 1.96 | 0.0011 |
| SVM | 0.9998 | 1.0000 | 0.9997 | 0.9998 | 2.30 | 0.0010 |
| RF | 0.9999 | 1.0000 | 0.9999 | 1.0000 | 381.46 | 1.2253 |
| XGB | 0.9999 | 0.9999 | 1.0000 | 1.0000 | 12.18 | 0.1087 |
| KNN | 0.9998 | 1.0000 | 0.9998 | 0.9999 | 759550.46 | 88.6140 |

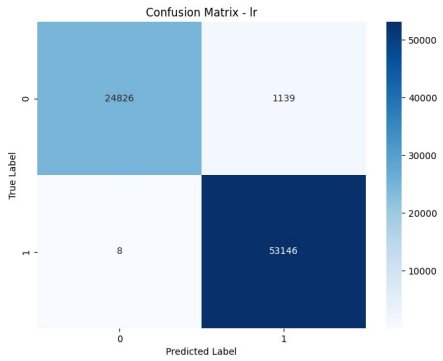


Fig. 1. Confusion matrix — Logistic regression

### B. Efficiency and real-time suitability

To assess practical deployment feasibility, the Real-Time Suitability Score (RTSS) was calculated for each model using:

**F1-score**

RTSS = Inference Time × Model Size

Table 2 shows that Logistic Regression achieved the highest RTSS (480.06), followed closely by Linear SVM (431.75). Both are ideal for deployment in real-time, resource-constrained environments such as IoT gateways. KNN and Random Forest, despite their high F1-scores, scored poorly on RTSS due to either excessive inference time or model size.

TABLE II
REAL-TIME SUITABILITY SCORES (RTSS)

| Model | RTSS |
|---|---|
| Logistic Regression | 480.06 |
| Linear SVM | 431.75 |
| XGBoost | 0.76 |
| Random Forest | 0.00 |
| KNN | 0.00 |

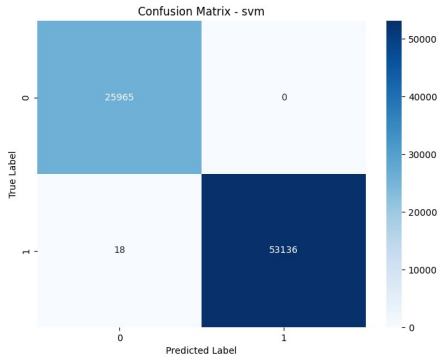


Fig. 2. Confusion matrix — Linear SVM

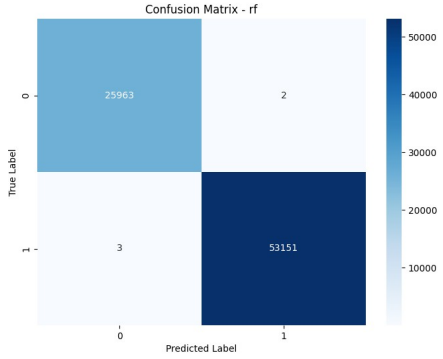


Fig. 3. Confusion matrix — Random Forest

## V. DISCUSSION

The obtained results demonstrate a fundamental compromise that exists within intrusion detection systems. The detection excellence of ensemble methods XGBoost and Random Forest comes at the cost of high resource requirements that prevent their use at the device edge. Logistic Regression and Linear SVM deliver similar intrusion detection results while using fewer system resources than EMLP, which enables their use in real-time intrusion detection on constraint systems.

## VI. CONCLUSION

This study analyses five kinds of machine learning models based on labeled training. Such algorithms were implemented using Python, Scikit-learn, and other libraries on a computer that had been installed to run the Ubuntu 20.04 LTS operating system, The CICIoT2023 dataset.

Real deployment problems were taken into account alongside traditional performance measures like accuracy and F1 score. For example, model size, how fast the model runs (inference time), and whether it is suitable for real- time deployment were all part of the study It was summed up the Real-Time Suitability Score (RTSS).

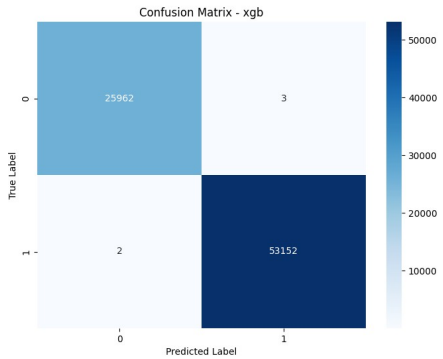


Fig. 4. Confusion matrix — XGBoost

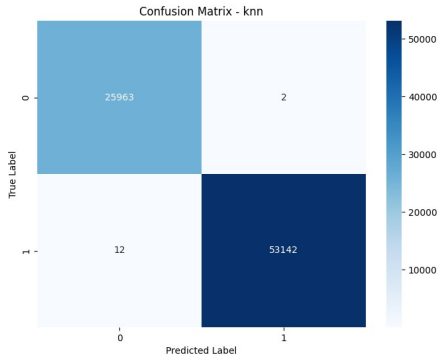


Fig. 5. Confusion matrix — KNN

Intrusions could be detected almost perfectly with models such as Random Forest and XGBoost. However, computing capacity requirements were very high—making them unsuitable for small devices, i.e., devices at the edge of the network.

Logistic Regression and Linear SVM by contrast offered a good balance: they performed very well and simultaneously stayed light and fast. It's exactly what you need in a real-time intrusion detection system for embedded or other IoT devices.

Future functions can detect the integration of light deep learning architecture, such as CNN-LSTM hybrids, model pruning, and quantization with optimization techniques to further reduce distribution. In addition, the extension of the Action Team using size or lime will improve transparency and confidence in making automated decisions, especially for assignment-enhancing applications. The evaluation of the performance of the more diverse and real IoT data set will also increase the generality and strength of the proposed models.

## References

[1] C. Ventures. (2024) Cybersecurity almanac: 100 facts, figures, predictions and statistics. [Online]. Available: https://shorturl.at/ltwOV

[2] R. G. Bace, P. Mell *et al.*, ``Intrusion detection systems,'' NIST Special Publication on Intrusion Detection Systems, Tech. Rep., 2001.

[3] F. A. Jam, R. A. Sheikh, H. Iqbal, B. H. Zaidi, Y. Anis, and M. Muzaffar, ``Combined effects of perception of politics and political skill on employee job outcomes,'' *African Journal of Business Management*, vol. 5, no. 23, p. 9896, 2011.

[4] M. Rana, D. Sharma *et al.*, ``Understanding cyber-attacks and their impact on global financial landscape,'' in *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*. IEEE, 2023, pp. 1452--1456.

[5] A. Hussain, A. Khatoon, A. Aslam, M. A. Khosa *et al.*, ``A comparative performance analysis of machine learning models for intrusion detection classification,'' *Journal of Cybersecurity (2579-0072)*, vol. 6, no. 1, p. 1–23, 2024.

[6] M. Shikida and K. Yagi, ``A method for supporting medical-interview trainings using wearable smart glasses,'' *Journal of ICT, Design, Engineering and Technological Science*, vol. 1, no. 2, pp. 37-41, 2017.

[7] P. Golchin, ``Machine learning models in network intrusion detection Systems: Self-supervised detection of malicious flows and traffic patterns recognition in programmable networks,'' Ph.D. dissertation, Dissertation, Darmstadt, Technische Universität Darmstadt, 2024, 2024.

[8] H.-T. Thai, ``Machine learning for structural engineering: A state-of-the-art review,'' in *Structures*, vol. 38. Elsevier, 2022, pp. 448-491.

[9] F. Jam, S. Singh, B. Ng, and N. Aziz, ``Effects of Uncertainty Avoidance on Leadership Styles in Malaysian Culture,'' *International Journal of Advance Business and Economics Research*, vol. 14, no. 8, pp. 7029-7045, 2016.

[10] CIC. (2023) Ciciot 2023: Real-time iot network intrusion detection dataset. Canadian Institute for Cybersecurity. [Online]. Available: https://shorturl.at/gflZX

[11] W. Liu, Z. Chen, and Y. Hu, ``Xgboost algorithm-based prediction of safety assessment for pipelines,'' *International Journal of Pressure Vessels and Piping*, vol. 197, p. 104655, 2022.

[12] S. R. Vadyala, S. N. Betgeri, J. C. Matthews, and E. Matthews, ``A review of physics-based machine learning in civil engineering,'' *Results in Engineering*, vol. 13, p. 100316, 2022.

[13] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, ``Survey of intrusion detection systems: Techniques, datasets and challenges,'' *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019.

[14] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, ``CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment,'' *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[15] A. Javed, A. Ehtsham, M. Jawad, M. N. Awais, A. Qureshi, and H. Larijani, ``Implementation of lightweight machine learning-based intrusion detection system on IoT devices of smart homes,'' *Future Internet*, vol. 16, no. 6, p. 200, 2024.

[16] R. Arthi, S. Krishnaveni, and S. Zeadally, ``An intelligent sdn-iot enabled intrusion detection system for healthcare systems using a hybrid deep learning and machine learning approach,'' *China Communications*, vol. 21, no. 10, pp. 1-21, 2024.

[17] R. Vadisetty and A. Polamarasetti, ``Enhancing intrusion detection systems with deep learning and machine learning algorithms for real-time threat classification,'' in *2024 Asian Conference on Intelligent Technologies (ACOIT)*. IEEE, 2024, pp. 1-6.

[18] W. Lian, G. Nie, B. Jia, D. Shi, Q. Fan, and Y. Liang, ``An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning,'' *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 2835023, 2020.

[19] S. M. Lundberg and S.-I. Lee, ``A unified approach to interpreting model predictions (online first),'' *Advances in neural information processing systems*, vol. 30, 2017.

[20] Y.-H. Liao, H.-H. Li, and Z.-H. Xie, ``Study of virtual reality and iot for exploring the deep sea,'' *Journal of ICT, Design, Engineering and Technological Science*, vol. 3, no. 1, pp. 11-14, 2019.