IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 October 2024, accepted 27 October 2024, date of publication 5 November 2024, date of current version 14 January 2025.

*Digital Object Identifier 10.1109/ACCESS.2024.3491831*

RESEARCH ARTICLE

# Cutting-Edge Intrusion Detection in IoT Networks: A Focus on Ensemble Models

**NAJM US SAMA[1], SAEED ULLAH[1], S. M. AHSAN KAZMI[2], AND MANUEL MAZZARA[3]**

[1]School of Computing, University of Derby, DE22 3AW Derby, U.K.
[2]School of Computer Science and Creative Technologies, University of the West of England, BS16 1QY Bristol, U.K.
[3]Institute of Software Development and Engineering, Innopolis University, 420500 Innopolis, Russia

Corresponding author: Najm Us Sama (najmussama@gmail.com)

This work was supported in part by the University of Derby.

**ABSTRACT** As the Internet of Things (IoT) landscape rapidly evolves, robust network security measures are imperative. In particular, Intrusion Detection Systems play a very important role in the preservation of an IoT environment from malicious activities. This paper provides a comprehensive performance comparison of various machine learning classifiers, including K-Nearest Neighbors, Gradient Boosting, XGBoost, Support Vector Machines, Random Forests, Decision Trees, and Extremely Randomized Trees, for intrusion detection in IoT networks. Comparative analysis shows that although all models did very well, the ensemble methods—GB, XGBoost, RF, and ERT—constantly performed better than others in F1-Score, recall, accuracy, and precision. Among them, ERT is turned out to be the most effective model for real-time attack detection on IoT devices, with an accuracy of 99.7% besides excellent precision and recall. XGBoost and RF also turn out to have high reliability and accuracy with F1-Scores of 0.95. These findings further underscore that ensemble methods outperform in intrusion detection for IoT networks and, thus, offer important insights to improve security within networks and protect critical IoT-based infrastructures from a variety of threats.

**INDEX TERMS** Accuracy, Internet of Things (IoT), intrusion detection systems (IDS), machine learning classifiers.

## I. INTRODUCTION

The IoT is believed to be an advanced evolution and a natural extension of the internet through the use of sensors and machine-to-machine connections. Applications for it can varyingly be found in such fields as smart cities, home energy management, medical care, fitness, and classroom automated processes [1]. There are three intertwined layers of the IoT architecture: the perception layer, which includes devices and sensors capturing data from the environment—for instance, security cameras detecting motion; the network/transport layer, which acts between devices and the cloud as an interface, communicating with protocols such as Wi-Fi, Zigbee, and MQTT to enable devices to send information back to servers; and the application layer, using the data provided from devices to give services and operations to users—for instance, motion detection notifications from cameras to a user's mobile application via cloud services [2], [3].

A study has said that the security risks in IoT are at risk at each architectural layer: the perception layer—sensors and devices may be infected with malicious code, eavesdropping, and interference; network layer—attacks like spoofing, denial of service, man-in-the-middle attacks, and routing information manipulation; and at the application layer—viruses, worms, and phishing attacks [4]. Privacy is also a major concern, as many IoT devices lack robust authentication systems due to resource constraints. Researchers have classified IoT attacks into four categories: physical (when an attacker gains physical proximity), software (exploiting vulnerabilities or bugs), network (manipulating IoT networks to compromise devices), and encryption (targeting weaknesses in cryptographic protocols) [5].

The global prevalence of IoT cyber-attacks has escalated alarmingly, with 77.9 million IoT malware attacks recorded globally in just the first half of 2023, a 37% increase compared to the same period in 2022. On average, 54% of organizations worldwide faced nearly 60 attempted IoT cyber-attacks per week in early 2023, translating to over

The associate editor coordinating the review of this manuscript and approving it for publication was Ye Liu.

3,000 attacks per organization in just two months, with the education and research sector being the most targeted. Europe experienced the highest number of weekly IoT attacks per organization, followed by APAC and Latin America, while 80% of organizations reported at least one successful IoT attack in the past year, with an average financial impact of $330,602 per attack [6], [7]. The security risks at each layer should be taken care of to make the IoT system operations safe, reliable, maintaining user privacy, and preserving data with service integrity. Security in IoT devices will need to be based on a multi-faceted approach that develops strong authentication mechanisms, encrypts the transferred data, sets up advanced intrusion detection and prevention systems, and increases awareness among users and organizations about securing IoT devices and networks [8]. The security risks associated with the IoT can be mitigated through proactive security measures during the very beginning of the design of products, establishment of a root of trust, and collaboration across the ecosystem on secure software updates and total cost of ownership considerations [9].

The IDS has become a crucial factor in computer security, safeguarding systems against numerous attacks and vulnerabilities. The goal of IDS is to devise self-learning systems that can generate and update signatures without prior knowledge. In order for such deployed systems to be capable of finding applications in real networking environments and executing operations to enhance the security of a network, they should ensure that the rate of false positives is low [10]. Intrusion detection principally refers to the process of monitoring and identifying the illegal use, exploitation, or even misuse of a network by internal and external intruders; this should be done in real-time. Intrusion detection is getting harder and harder due to increased network connectivity, rapid technological advancement, and the availability of paid hackers. IDSs are therefore crucial security systems that are made to keep an eye out for, identify, and report any illegal activity or policy infractions occurring within computer systems and networks. However, because of the inaccurate identification of IDS alarms, network managers encounter difficulties when processing intrusion notifications. The issue of conventional IDS relying on recognized patterns of attack has led to the development of machine learning-based IDS. These systems train on datasets to predict assaults via categorization, and they learn from both normal and abnormal traffic. Although several machine learning approaches have been effectively employed as classifiers in IDS, they continue to encounter obstacles like high false positive rates [11], [12].

In this study, we conduct a comprehensive evaluation of machine learning classifiers for IDS, focusing specifically on their applicability to IoT environments. We assess the performance of various classifiers including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), XGBoost (XGB), Gradient Boosting (GB), and Extremely Randomized Trees (ERT). Performance evaluation is conducted using key metrics such as accuracy, precision, F1 score, and recall. This analysis provides insights into the effectiveness of these classifiers in detecting intrusions in IoT networks, aiding in the selection of suitable algorithms for enhancing network security in IoT environments.

We can summarize our main accomplishments as below.

- The proposed work deals with the challenge of attack traffic classification in IoT environments, where the unparalleled heterogeneity and dynamicity of IoT networks increase variety and quantity of associated vulnerabilities.
- We utilize sophisticated machine learning models to extract features from network input that has been shown to be informative for classifying attack traffic.
- In this line, we adopt RT-IOT2022, which is a publicly available dataset of attacks against IoT devices [13]. This will also help in making our experimental studies repeatable.
- We present comprehensive empirical findings on attack-traffic classification using DL techniques. The classification performance is evaluated experimentally by comparing them with existing classifiers.
- Our objective is to maximize the performance of attack-traffic classification by assessing various design options.

The proposed work can be structured as: Section II deals with the latest findings in the respective field. Section III provides a brief overview of classification algorithms. Section IV elaborates more on the proposed experimental layout. In Section V, the efficacy of the classifiers and related statistical studies are dealt with. Finally, Section VI summarizes the conclusions of the study.

## II. MACHINE LEARNING BASED IDS

Research into machine learning during the last few decades has equipped IDS with some very powerful tools. The current research into the efficacy of ML strategies for IoT intrusion identification is reviewed in this section, covering different classifiers and their performance metrics.

The most recent article in the set presents a comparative analysis of different machine learning techniques: SVM, Artificial Neural Network, DT, Logistic Regression, and KNN. For example, researchers used ToN-IoT and Bot-IoT to test these methods [14]. Results presented by the authors indicate that the neural network performed better than the other models. The author applies the kernel extreme learning machine not only in binary classification problems but also in multiclass classification problems, such as classifying either benign or malicious traffic flow, or specific types of attacks. Two state-of-the-art datasets, N-BaIoT and UNSW-NB15, are utilized during the process for the evaluation of effectiveness in the proposed anomaly detection method. Results show that the proposed approach improves the detection performance to 99.4% in N-BaIoT and 98.64% in UNSW-NB15, ensuring improvement in efficiency and accuracy of detection [15].

The authors used the famous NSL-KDD dataset in order to reduce the computational overhead [16]. They adopted the metaheuristic algorithms Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Differential Evolution (DE) to find the optimal feature selection. For accurate classification of normal and attack classes based on selected features, some supervised learning algorithms such as KNN and DT were used.

Ahmad et al. propose a feature cluster based on the features extracted from the UNSWNB15 dataset, like flow, Transmission Control Protocol, and Message Queuing Telemetry Transport [17]. The issues of overfitting, the constraint of dimensions, and an imbalanced data set have been resolved. The proposed method used supervised machine learning techniques like SVM, RF, and ANN on the clusters. Employing RF, the algorithm obtains 98.67% and 97.37% accuracy in binary and multiclass classification, respectively. Classifier accuracy of 96.96%, 91.4%, and 97.54% was obtained using cluster-based methods using RF on flow and MQTT features, TCP properties, and the best characteristics across the two clusters. The Synthetic Minority Oversampling Technique Tomek Link (SMOTE-TomekLink) algorithm combines ML techniques in the authors' intrusion detection approach [18]. An evaluation of 374,661 records from the Wireless Sensor Network dataset (WSN-DS) was conducted to identify the optimal model for intrusion detection in WSNs. The proposed model achieved 99.78% accuracy in binary classification scenarios, while in multiclass classification scenarios, it achieved 99.92% accuracy. An IDS utilizing multiple machine learning classifier techniques was evaluated in [19] using the Message Queuing Telemetry Transport - Internet of Things - Intrusion Detection System dataset (MQTT-IoT-IDS2020) to identify multi-class intrusion attacks in IoT environments. Results obtained from the base models are 97.76% using KNN, 97.80% from SVM, 97.58% from NB, 99.98% using RF, 99.98% from DT, and 97.58% from Stochastic Gradient Descent (SGD).

The author proposes an Intrusion Detection System using Deep Learning for IoT devices [20]. Four-layer totally connected network architecture will help this intelligent system to find malicious traffic that might attack connected IoT devices. Since the proposed system is independent of the communication protocols, the deployment complexities are minimized. The intrusion detection had been reliably done in the proposed system for both simulated and real intrusions, and during the experimental performance analysis, it could achieve an average accuracy of 93.74% in identifying various attack types such as Blackhole, Distributed Denial of Service, Opportunistic Service, Sinkhole, and Wormhole attacks. In [21] an IoT-based IDS using several machine learning models and different feature extraction strategies has been introduced. Specifically, it considers various feature extractors such as image filters and transfer learning algorithms including DenseNet and VGG-16. The feature extraction methods investigated were also used in combination with several machine learning algorithms, including RF, KNN, SVM, and others with stacked models. These results turned out that the maximum accuracy of 98.3% was achieved by a combination of VGG-16 and stacking.

The authors have proposed an IDS with the support of a deep learning model called Pearson-Correlation Coefficient-Convolutional Neural Networks for network anomaly detection [22]. The model conducted binary classification for anomaly detection and multiclass classification for various types of attacks. It was tested on three publicly available datasets: NSL-KDD, CICIDS-2017, and IOTID20. First, for performance evaluation, five different machine learning models—Linear Discriminant Analysis, Logistic Regression, KNN, Classification & Regression Tree (CART), and SVM—based on PCC were trained and tested. Among these, the models produced from KNN and CART have given the highest accuracies of 98%, 99%, and 98% on the three datasets, respectively. Sharmila and Nagapadma [23]; provide a Quantized Autoencoder-based approach specifically oriented to low-resource IoT devices for anomaly detection. The paper describes the challenges posed to IoT by resource limitations, with a focus on the RT-IoT2022 dataset. In this regard, the QAE methodology is feasibly applied with quantization techniques that reduce model complexity and resource consumption without affecting detection accuracy [24]. They used the SMOTE algorithm to check how different data pretreatment methods may influence the accuracy rate, handling imbalanced data, data encoding, and data purification. The results of the study prove how good the intelligent IDS is in terms of attack identification.

Yaras et al used the datasets CICIoT2023 and TON_IoT to train and test the model. One dimension CNN and another dimension LSTM are applied to build a hybrid deep learning system [25]. Following the analysis, the "CICIoT2023" dataset displays an accuracy evaluation of 99.995% for classification by binary and 99.96% for multiclassification. A binary classification success rate of 98.75% is attained in the 'TON_IoT' dataset. Based on an adaptive CNN-GRU model, the researchers suggest a powerful deep learning model called AttackNet for the identification and categorization of various botnet attacks in the IIoT [26]. The model performs exceptionally well, especially when applied to the N_BaIoT dataset. Table 1 illustrates the summery of machine learning based IoT intrusion detection from literature.

Various machine learning techniques are proposed over different IoT and intrusion detection datasets. The performance was varied with regard to the classification types-binary and multi-class-as well as with respect to datasets. Previous studies have employed various machine learning techniques such as DT, RF, and SVM, and deep learning approaches like LSTMs and CNNs on NSL-KDD, IoTDevNet, and CICIDS-2017 datasets, among others. Although the performance of different machine learning classifiers has been reported in the literature with respect to IDS in IoT environments, it tends to reveal their potential

**TABLE 1.** Overview of current ML based IDS for IoT.

| Work | Dataset | Used techniques | Classification Type |
|---|---|---|---|
| [27] | NSL-KDD, IoTDevNet, DS2OS, IoTID20, IoT Botnet dataset | DT, RF, SVM, DNN, DBN, LSTM, stacked LSTM, bidirectional LSTM | Multi-class |
| [17] | UNSW-NB15 | SVM, RF, ANN | Binary and Multiclass |
| [22] | NSL-KDD, CICIDS-2017, and IOTID20. | PCC-CNN | Binary and Multiclass |
| [23] | RT-IoT2022 | QAE | Binary and Multiclass |
| [19] | MQTT-IoT-IDS2020 | KNN, SVM, NB, RF, DT, SGD | Multiclass |
| [21] | IEEE Dataport | RF, KNN, SVM | Binary |
| [22] | NSL-KDD, CICIDS-2017, and IOTID20 | PCC-CNN, KNN, CART, SVM | Binary and Multiclass |
| [24] | IoT 23 | KNN, SVM, ANN | Multi-class |
| [15] | N-BaIoT and UNSW-NB15 | hybrid technique (KPCA-KELM) | Binary and Multiclass |
| [25] | CICIoT2023' and 'TON_IoT' | CNN and LSTM | Binary and Multiclass |
| [26] | N_BaIoT | CNN-GRU | Multiclass |

performance in detecting malicious activity in IoT networks. Even though these methods look promising for binary and multi-class classification, there is a gap in the literature regarding the thorough investigation of more recent models-like ERT and state-of-the-art hybrid approaches-in real-world IoT scenarios. Besides, the scalability of those models and their performance on large and diverse IoT datasets is yet to be well-explored. An imminent requirement is for highly efficient and very accurate intrusion detection techniques that could adapt to the dynamic nature of IoT networks. The goal of this paper is to try to fill these gaps by comparing the performance of several machine learning classifiers like KNN, XGBoost, and ERT, while simultaneously recommending some ways in which IDS can be enhanced for practical applications in the IoT context.

The motivation lies in identifying the best models for security in IoT applications, taking into account the factors of accuracy, precision, recall, and F1 score. This paper should contribute to giving insights about the performance of classifiers, which should give direction for the future development of intrusion detection systems for IoT systems in offering better protection against cyber-attacks.

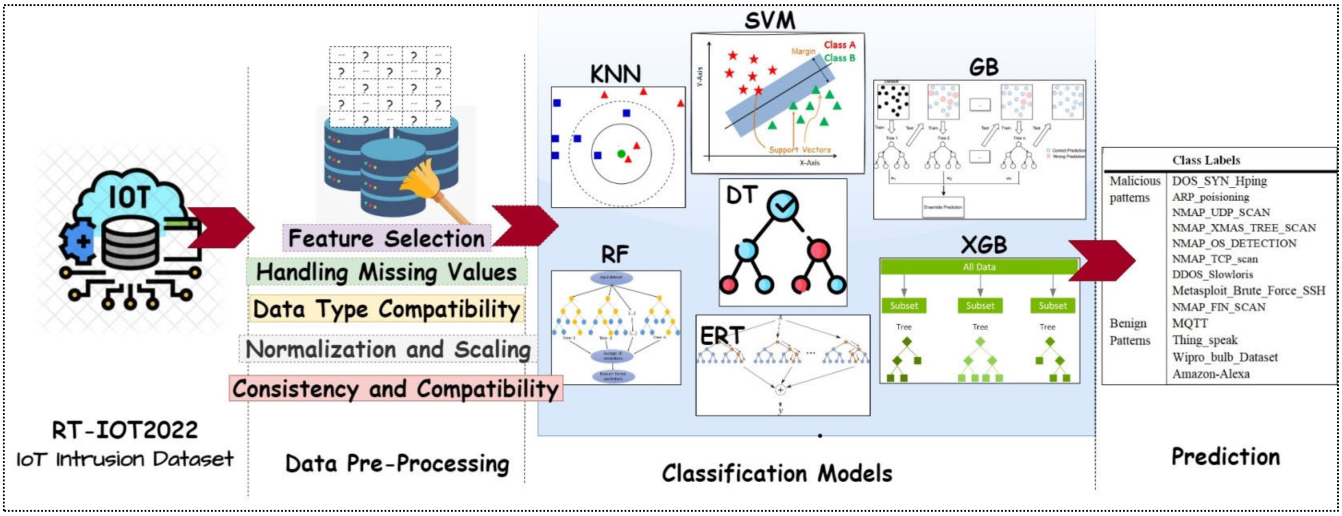
## III. THE PROPOSED IDS FRAMEWORK

The following section discusses our proposed solution, data-set used, and ML algorithms employed to detect malicious network traffic. The Fig. 1 below illustrates the steps involved in this work. Our first step was to select a data-set called RT-IOT2022, whose files are in '.csv' format. The ".csv" files had numerous problems, including an unbalanced data collection, datatypes that weren't compatible with the classification technique and a large number of missing or null values. All of the above-mentioned issues were sorted out at the data pre-processing step. After the cleansing of the dataset, the next task was to develop separate clusters on the network layers. These developed clusters have classification algorithms applied to predict the network traffic to be malicious or benign.

### A. RT-IOT2022 DATASET

The RT-IoT2022 dataset was chosen for this study due to its relevance and comprehensive representation of real-world IoT environments. The RT-IoT2022 dataset is an exclusive collection from a real-time IoT infrastructure [13]. It has a rich source that contains most types of IoT devices and cutting-edge network attack measures.

Unlike most other publicly available datasets, RT-IoT2022 includes IoT-scenario-specific attacks like DDoS, Blackhole, and Sinkhole, which are very common in IoT networks [28]. This dataset, with simulated and real traffic data, is more realistic and applicable compared to some of the earlier datasets like NSL-KDD, which are highly generalized and may not describe the distinct security challenges of IoT devices. Consequently, this RT-IoT2022 dataset represents more modern IoT protocols and architectures and thus becomes a better choice for evaluating intrusion detection systems in contemporary IoT environments. The high quality and well-annotated dataset enabled us to be effective in our analysis, while the IoT security features included therein enabled us to test our security models effectively. With the many forms of attack types and real-looking traffic patterns, it considerably generalizes the model performance to be much more accurate and practical to deploy in realistic settings. This dataset further gives an exact characterization of the real-world setting by holding both abnormal and normal network activities. It contains emulated attack vectors of Brute-Force SSH attacks, DDoS attacks with Hping and Slowloris, and patterns of Nmap, along with IoT devices data like ThingSpeak-LED, Wipro-Bulb, and MQTT-Temp. RT-IoT 2022 makes use of a Flowmeter plugin and the network monitoring tool Zeek for capturing accurate bidirectional attributes of network flows. The dataset is divided into two parts, separated by a router: IoT offender endpoints and target endpoints. The network traffic is collected using Wireshark via a router and then converted into a PCAP file. The target organization consists of 5 departments with 420 machines and 30 servers. The attacker's infrastructure was composed of 50 machines. Besides 80 features extracted from the captured traffic, the dataset includes system logs and network traffic



| | Class Labels |
|---|---|
| Malicious patterns | DOS_SYN_Hping<br>ARP_poisioning<br>NMAP_UDP_SCAN<br>NMAP_XMAS_TREE_SCAN<br>NMAP_OS_DETECTION<br>NMAP_TCP_scan<br>DDOS_Slowloris<br>Metasploit_Brute_Force_SSH<br>NMAP_FIN_SCAN |
| Benign Patterns | MQTT<br>Thing_speak<br>Wipro_bulb_Dataset<br>Amazon-Alexa |

**FIGURE 1.** Proposed framework for detection of cyber-attacks from IoT networks.

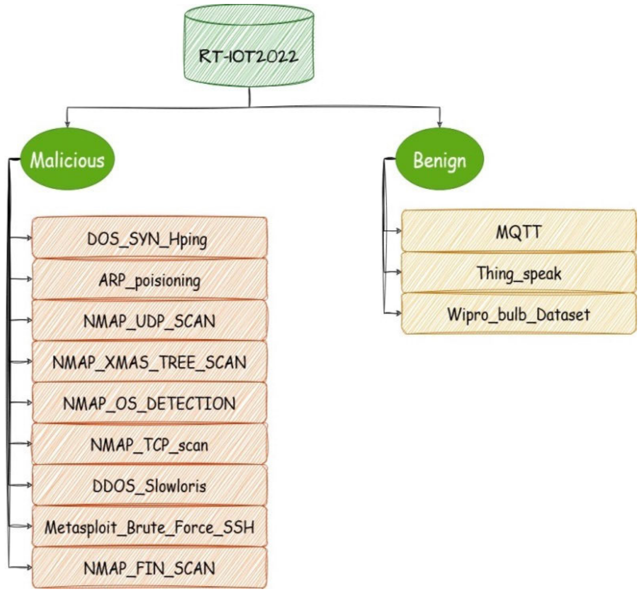


**FIGURE 2.** RT-IOT2022 dataset classes.

**TABLE 2.** Number of instances of each class.

| Segment | Class Number | Class | Number of Instances |
|---|---|---|---|
| **Malicious patterns** | 0 | DOS_SYN_Hping | 94659 |
| | 1 | ARP_poisioning | 7750 |
| | 2 | NMAP_UDP_SCAN | 2590 |
| | 3 | NMAP_XMAS_TREE_SCAN | 2010 |
| | 4 | NMAP_OS_DETECTION | 2000 |
| | 5 | NMAP_TCP_scan | 1002 |
| | 6 | DDOS_Slowloris | 534 |
| | 7 | Metasploit_Brute_Force_SSH | 37 |
| | 8 | NMAP_FIN_SCAN | 28 |
| **Benign Patterns** | 9 | MQTT | 8108 |
| | 10 | Thing_speak | 4146 |
| | 11 | Wipro_bulb_Dataset | 253 |

samples of each machine. This dataset enables the better performance of IDS and promotes the development of robust, flexible security solutions for real-time Internet of Things networks. In total, there are nine different attack scenarios, as shown in Fig. 2. The number of instances in each class is shown in Table 2.

### B. DATA PREPROCESSING FOR THE RT-IOT2022 DATASET

C. Data preprocessing is an important phase of preparation of the RT-IOT2020 dataset for IDS based on machine learning. This section represents procedures under which the dataset is supposed to be cleaned, optimized, and made ready for use with several machine learning methodologies. During the preprocessing of RT-IOT2022, comprehensive analysis led to removing non-contributive features in order to make the dataset lean, which would help improve the performance of intrusion detection models and avoid over fitting. The dataset was checked for missing values and, depending on the degree of absence, rows or columns with the data were removed or imputed with techniques such as mean, median, mode, or KNN imputation. The imputation methods are selected according to the nature of missing data and the characteristics of the IoT dataset. More precisely, for continuous features, mean imputation was chosen to preserve the distribution of the data since this helps retain the central tendency and diminishes the impact of missing values on overall model performance. In the case of skewed data, median imputation was done to reduce the effect of outliers by not allowing extreme values to bias the results. For categorical variables, mode imputation was selected in order to retain most of the frequency; hence, the retention of patterns would be ensured. Such feature compatibility checks ensured that all variables were suitable for the machine learning algorithm. This involved transformations like one-hot encoding for categorical data. These imputation techniques have been tested during the preliminary experiments concerning their impact on the performance of classifiers. Our analysis indicated that these methods consistently yield better model accuracy and stability compared with other more sophisticated imputation techniques like K-Nearest Neighbors imputation or iterative imputation. Moreover, these techniques had lower computational overhead and were very well-suited for real-time IoT applications. In addition, normalization and scaling methods were applied to get all features into a similar range, which is very important for distance-sensitive algorithms. The whole preprocessing regime will ensure consistency in data formatting, enabling seamless integration with the models without further modifications. Therefore, it provides such comprehensive preparation of the dataset that it minimizes the computational complexity and improves the efficiency and accuracy of models developed for intrusion detection.

## *C. CLASSIFICATION MODELS*

Machine learning techniques are generally used as classifiers for IDS and are prone to many challenges like high false positive rates. In this paper, keeping the suitability of these machine learning classifiers for IoT environments in view, we evaluate the machine learning classifiers designed for IDS. This paper evaluates several types of classifiers: KNN, GB, EGB, DT, RF, and ERT. In the case of this heterogeneous set of models, we present how effective these models could turn out to be for categorizing an IoT-based attack. We investigate the different strengths and limitations of each model against various real-time IoT datasets, which help us emphasize exactly which one can be the most appropriate approach for categorizing IoT attacks. Our research puts forward insight into how to select the most appropriate model to detect and classify diverse IoT threats effectively, hence significantly advancing the security of IoT.

### 1) K-NEAREST NEIGHBORS (KNN)

KNN represents one of the most efficient and popular deep learning methods for classification problems, including intrusion detection in IoT networks. The KNN algorithm works well with high-dimensional data, mostly seen in IoT environments where multiple features such as packet size, traffic type, or behavior of devices are analyzed. It requires minimum assumptions regarding the distribution of data, hence making it flexible in dynamic IoT systems [29]. Besides, since it is non-parametric, KNN would be able to pick up new attack patterns or rare attack patterns relying on proximity instead of training or predefined rules, hence highly suitable for real-time IoT attack detection. This makes it a very good choice for IoT attack detection due to its simplicity, interpretability, and the ability to handle multi-class classification problems [30].

KNN can be applied in the domain of IoT intrusion detection to classify the pattern of network traffic as either normal or malicious. In essence, the algorithm calculates the distances between a new instance—in this case, a network traffic sample—and all instances in the training dataset using metrics such as Euclidean or Manhattan distance [11], [31]. The title for the class of the newly created instance is then chosen with the k nearest neighbors of the feature space casting their maximum opinion. KNN has the capability to detect different forms of IoT attacks and attempts of data leakage by efficiently differentiating attack and normal traffic patterns through the accurate selection of the number of k and the optimization of the distance parameter. Further, KNN can handle multi-class classification, thus identifying the exact attack types and giving very good insights to the security analyst for mitigation strategies.

### 2) SUPPORT VECTOR MACHINE (SVM)

SVM aims at building a model that can predict the target values of test datasets according to its property using a training dataset. In its entirety, SVM uses a subset of the training points, making it unique in memory economy. But its performance could be affected if used on noisy datasets with overlapping classes. Applications of SVM can be found in banks, IDSs, image processing, and text classification. Employing an SVM is easier compared to a neural network. SVM also performs very well in high-dimensional spaces [32]. The SVM is suitable for detecting complex attack patterns in IoT networks, since it can handle both linear and nonlinear relationships using kernel functions.

Due to the ability to handle high-dimensional data and also to manage nonlinear relationships, SVM is suitable for IoT networks in identifying abnormal activities with a security threat. The features extracted from IoT data using SVM-based IDS will find the suspicious activity with high accuracy, thereby improving the security of the IoT ecosystem against cyberattacks [33].

### 3) DECISION TREE (DT)

A DT is a tree with leaf nodes that represent categories and inside nodes that can be interpreted as tests (of the patterns in the data). These tests have been refined via the tree to obtain the appropriate output for the input pattern. Decision tree algorithms apply and are useful in a broad spectrum of industries. One of the main advantages of DT algorithms is their interpretability and simplicity, turning them into an attractive solution for intrusion detection in IoT environments [34], [35]. Since the algorithm is capable of handling categorical and numerical data with relatively low computational cost, it will be an appropriate fit for IoT real-time environments where attacks have to be detected with speed. Since the model is more transparent, feature analysis can be more easily done in order to provide an insight into what contributes to an attack, which could be of good value when trying to deduce the nature of the threats.

The DT model employs unusual traffic volume, failed authentication attempts, and unexpected data transmission to identify a family of IoT attacks: DDoS, brute force, and malware-based attacks. This model delivers an interpretable and systematic solution for real-time attack detection in IoT, with high accuracy in distinguishing normal activities from malicious ones; thus, it is practically serviceable for the improvement of security in IoT environments.

#### 4) GRADIENT BOOSTING(GB)

GB works in a systematic way, combining weak learners to increase the accuracy of prediction. It functions quite well on tasks like classification and regression, big data sets management, prevention of overfitting, or complex connections identification in data [36]. Due to this, an ensemble machine learning technique like GB has become a potential option for detecting IoT attacks in real-time. The paper investigates the applicability of GB in the identification of malicious activities over IoT networks, utilizing its capability in building robust predictive models from weak learners. GB improves the performance through iterative refinement and therefore offers great advantages in terms of accuracy and speed, both of which are important in maintaining integrity related to security in IoT systems [37].

IoT generates a vast amount of data with diverse features; hence, Gradient Boosting would seem to be the better technique for detecting patterns in malicious behavior. Furthermore, the algorithm considered has been flexible and efficient in tasks like classification and regression. This makes it suitable for real-time attack detection. Results indicated that GB increased the accuracy of detection while reducing the false positive rates, hence providing a reliable and responsive security framework.

#### 5) EXTREME GRADIENT BOOSTING (XGB)

XGBoost is a very efficient and scalable implementation of the gradient boosting algorithm, which has recently received considerable popularity in a variety of machine learning tasks; for example, intrusion detection in IoT networks [38]. On the other hand, XGBoost represents another ensemble learning technique that combines multiple weak learners into a strong predictive model. This framework relies on gradient boosting, whereby subsequent trees correct mistakes of the former to iteratively come up with better model performance. In the detection of IoT attacks in real-time, XGBoost has a number of advantages. First, the high-dimensional and sparse data represent the varied features found in IoT network traffic, which the algorithm can very well process. Paralleling capabilities and more efficient algorithms within XGBoost allow for efficient training and prediction, thereby empowering real-time analysis of network flows and enabling early detection of a potential attack [39].

It can be trained on labeled datasets of IoT network traffic, where each instance will represent a network flow or packet, including the features such as Ports, Source & Destination IP Addresses, Protocols, Payload Characteristics, and Time-based Features. XGBoost builds up an ensemble of decision trees, wherein each decision tree has focused parts of the data, thus contributing to the final classification decision. By summing up these tree outputs, XGBoost can effectively tell about normal and attack communication structures, therefore enabling the instances to be identified immediately for several IoT attacks. XGBoost has emerged as one of the popular choices for IoT attack detection because it is capable of handling big datasets and complex relationships, which are part of the IoT environment. This algorithm, with its extraordinary ability to reduce overfitting and handle noisy data with ease, will serve perfectly for detecting unseen patterns in attack behaviors while maintaining reliability and accuracy in spotting cyber threats to IoT systems. Feature importance ranking further helps in prioritizing factors of relevance to attack detection, enhancing interpretability in security-related applications.

Our findings provide a view on the real applicability of XGB in real-time attack detection in IoT devices and are of significant importance to practitioners in the industry and cybersecurity professionals seeking proactive measures to predict cyber risks in an era of interconnected IoT devices.

#### 6) RANDOM FOREST (RF)

RF is an ensemble learning method in which many different decision tree structures are made during training and the average of their prediction is returned for regression problems or mode of classes in classification. It has low tuning requirements on hyperparameters, resilience, and is high in performance under high-dimensional data [40]. RF is a powerful, multi-utility, and ensemble learning method that got considerable attention for real-time attack detection. The method creates several decision trees during training and returns the preferred method of the classes for problems in categorization. RF reduces overfitting through an aggregation of predictions from various trees, ensuring generalization with improved accuracy. In this regard, the current study has exploited the ensemble learning capability of RF to classify and detect different types of attacks efficiently and at a very fast speed that can maintain the integrity and security of IoT networks [41].

The RF is capable of handling large and high-dimensional datasets with much accuracy. It makes multiple decision trees out of random subsets of the data and makes its predictions by combining those trees for improved overall accuracy, reducing overfitting [42]. It provides an excellent result in complex patterns of IoT attack detection. It also included feature importance with RF on the key predictors of attacks and it is scalable to handle massive data from IoT devices. Results show its robustness and efficiency: it can work in real time to give immediate alerts and mitigation strategies. Implementation of RF in IoT security frameworks in RT-IoT2022 underlines the potential for improving resilience and reliability against evolving cyber threats in IoT systems.

### 7) EXTREMELY RANDOMIZED TREES (ETC)

ERT is another ensemble method of learning, much like Random Forest. In contrast with Random Forest, which picks the best split based on a subset of features, ERT randomly selects splits from the complete feature space. This increased randomness can lead to faster training times and perhaps improved generalization effectiveness, especially for noisy datasets. ERT can handle high-dimensional data quite well and are resistant to overfitting [43]. ERT, being a cutting-edge ensemble learning technique, has already revealed great potential in the detection of RT-IoT attacks as of RT-IoT2022. Rather than being much like the traditional decision tree algorithms, ERT adds more randomness into this by randomly selecting the cut points for each feature while splitting the nodes. This makes it so that the model can generate a variety of trees independently, ensuring its robustness and generalizing ability. In the domain of IoT attack detection, ERT efficiently deals with high dimensionality and heterogeneity enabling it to identify complex patterns and subtle anomalies, thus giving cues to potential security threats. Its ability to process large volumes of data quickly makes it quite suitable for real-time applications where timely detection is important [44].

In the case of RT-IoT2022, it showed effectiveness with real-time IoT attack detection by providing higher accuracy and speed in malicious activity identification. The model, trained on different varieties of normal and attack traffic datasets. As there are multiple random splits per feature, this will significantly reduce variance without increasing bias, which is critical to improve detection rates in noisy and dynamic IoT environments. Moreover, handling missing data and resistance to overfitting make it quite reliable in real-world scenarios. In IoT attack detection, ERT is preferred due to its efficiency in handling large-sized datasets, computational speed, and good generalization across various attack types for real-time security applications. ERT deployment inside IoT security frameworks increases the intrinsic defense mechanism within them and provides a scalable and effective solution to counteract the rapidly changing landscape of IoT-based cyber threats.

## IV. RESULTS AND DISCUSSION

In this section, we give the details of our experimental setting, including the models and their parameters, followed by datasets. We will specify what settings and configurations have been used in our experiments to be clear and reliable about the presentation. Afterward, we are going to present the results for each model, accompanied by a comparative analysis between them and pointing out some key findings and insight takeaways from the experimental data.

### A. EXPERIMENTAL SETUP

As shown in table 3 for each model, we tried to get an optimal performance for accuracy, F1-score, precision, and recall. Regarding KNN, the best performance was obtained by setting the number of neighbors to 5, the Euclidean distance, and uniform weighting. SVM used the RBF kernel with a regularization parameter C of 1.0 and a gamma coefficient 'scale'. The 'gini' criterion was used in node splitting for DT. Also, in GB, 100 estimators were set, the learning rate was 0.1, and the subsample fraction taken was 1.0. In XGBoost, similarly, 100 boosting rounds, a 0.1 learning rate, and a binary logistic objective function were performed for classification.

**TABLE 3. Experimental parameters for machine learning models.**

| Model | Parameter | Value | Description |
|---|---|---|---|
| **KNN** | n_neighbors | 5 | Specify the number of neighbors |
| | weights | 'uniform' | It specifies contributions of neighbor to the prediction. |
| | algorithm | 'auto' | Procedure for finding the nearest neighbors. |
| | p | 2 | Corresponds to the Euclidean distance. |
| **SVM** | C | 1.0 | Trade-off between reductions of the training error versus the test error. |
| | kernel | 'rbf' | rbf is an abbreviation for Radial Basis Function. |
| | gamma | 'scale' | It determines how much the data points are influenced by each other. |
| | random_state | 42 | Used to control the random seed during data shuffling. |
| **DT** | criterion | 'gini' | Evaluate the quality of split. |
| | splitter | 'best' | Optimal split at each node. |
| | random_state | 42 | Control of randomness |
| **GB** | n_estimators | 100 | The total number of boosting iterations. |
| | learning_rate | 0.1 | Reduces the impact of each tree's contribution |
| | subsample | 1.0 | Fraction of samples to train each base learner. |
| | random_state | 42 | Controls the randomness |
| **XGBoost** | n_estimators | 100 | The total number of boosting iterations to perform. |
| | learning_rate | 0.1 | How much each boosting instance contributes. |
| | min_child_weight | 1 | The minimum sum of instance weights required to be at a child node. |
| | objective | 'binary:logistic | Learning Task and Objective |
| | random_state | 42 | The value to be used as the random number seed. |
| **RF** | n_estimators | 100 | The number of trees in the forest |
| | criterion | 'gini' | The function to measure the quality of a split |
| | random_state | 42 | Controls the randomness of the bootstrapping |
| **ERT** | n_estimators | 100 | The number of trees in the forest |
| | criterion | 'gini' | The function to measure the quality of a split |
| | random_state | 42 | Controls 3 sources of randomness |

The RF used 100 trees with the 'gini' criterion, enabling bootstrap sampling. Finally, the best-performing model, ERT, used 100 trees with the 'gini' criterion but without bootstrap sampling.

These settings helped to ensure that ensemble methods, such as GB, XGBoost, RF, and ERT, always outperform the individual models, and the ERT attains the maximum accuracy and reliability in attack detection over IoT networks.

### B. EVALUATION METRICS

In this section, we evaluate the proposed models and their outcomes. First, accuracy (Ac) measures the proportion of correctly classified network traffic. It is computed by Eq.1, which represents the total number of correctly labeled instances divided by that of the entire dataset. However, high accuracy can sometimes be deceiving, since most datasets are imbalanced, with one class holding a majority, such as benign traffic, dominating the minority classes, making malicious activity obscure to detect. Precision (Pr) Eq.2, measures how many of the true positive instances are among all positive predictions, hence how accurate the model really is at making positive predictions. Recall (Rec) is a measure of the proportion of actual positives that were correctly identified by the model; it is calculated by Eq. (3) and describes how good the model is at capturing all relevant instances.

The F1-score, given by Eq. (4), is the harmonic mean of precision and recall. It provides a balanced measure of both and is therefore more useful in the case of class imbalance. A high F1 score means that a model has high precision and high recall, meaning classifying malicious activity correctly while minimizing false alarms. These metrics will help us get full knowledge of the models' performance to ensure that the results are reliable and informative. We can then comprehensively check the effectiveness of the proposed models for real-time IoT attack detection using this multifaceted approach in evaluation.

$$Ac = (P_T + N_T)/(P_T + N_T + P_F + N_F) \tag{1}$$

$$P_r = P_T/(P_T + P_F) \tag{2}$$

$$Rec = P_T/(P_T + N_F) \tag{3}$$

$$F1S = 2 * (P_r * Rec)/(P_r + Rec) \tag{4}$$

The performance of a model can be assessed using the values provided by the confusion matrix. This matrix reports the number of True Positives ($P_T$), False Positives ($P_F$), True Negatives ($N_T$), and False Negatives ($N_F$) for the predicted classes.

- $\mathrm{P_T}$ are cases correctly predicted as positives
- $\mathrm{P_F}$ are cases incorrectly predicted as positives
- $\mathrm{N_T}$ are cases correctly predicted as negatives
- $\mathrm{N_F}$ are cases incorrectly predicted as negatives

All of these four domains compute performance metrics, including accuracy, recall, precision, and the F1-score. With these measures in place, researchers can now draw well-informed conclusions about models' applicability and dependability for the intended purposes, hence improving the results.

### C. EFFECTIVENESS OF CLASSIFICATION MODELS

Several models for classification were trained and validated using the RT-IOT2022 dataset. The RT-IOT2022 dataset contains a wide variety of IoT network traffic data that involves benign and malicious activities. To make the analysis strong and reliable, this dataset has been divided into two sub-datasets: 80% for the training of the models and 20% for testing and validation purposes. Important features required for intrusion detection were extracted from packet flow characteristics, connection behaviors, and attack patterns by preprocessing the training data. For a fair comparison among different models, all the techniques were applied using constant-sized training data.

This division allowed us to optimize the hyperparameters of each classifier while ensuring that the models could generalize to unseen data. In particular, the larger training set really benefited the ensemble methods, GB, XGBoost, RF, and ERT, since these algorithms had more data to learn from, yielding quite accurate and reliable performances. This helped these models capture the complex attack signatures and network behaviors, hence their superior F1-Scores, recall, accuracy, and precision when compared to other techniques. The results also reaffirm the need to use a sufficiently large and representative training set in an effort toward improvement of detection capability of machine learning models for IoT intrusion detection. Table 4 compares machine learning models using F1-Score, accuracy, recall, and precision for the detection and classification of data. The KNN model achieves an accuracy of 99.4% with balanced precision and recall, indicating that it works fine in general but misses a few positive cases. The SVM shows very good precision, 0.97, with lower recall at 0.88. Its overall F1-Score is therefore 0.92, showing it misses more positive instances. The DT model has accuracy a little higher, 99.5%, and hence delivers a very good performance with balanced metrics. While both GB and XGBoost end with an accuracy of 99.6%, this time, too, XGBoost outperforms GB in terms of precision and recall, which give an F1-Score of 0.95. RF did very near to XGBoost, underpinning the reliability of the results. The ERT model was ranked highest with the best accuracy at 99.7% and very good precision and recall for a lead F1-Score of 0.95, making this model most effective among the comparative models.

Overall, while all models perform well, ERT, XGBoost, and RF exhibit the highest reliability and accuracy. These results suggest that ensemble methods (GB, XGB, RF, ERT) generally perform better in terms of F1 score, precision, and recall, making them highly reliable for real-time IoT attack

**TABLE 4.** Extensive effectiveness analysis of proposed ID models.

| Model | Accuracy | Pr | Rec | F1S |
|---|---|---|---|---|
| KNN | 99.4% | 0.94 | 0.91 | 0.92 |
| SVM | 99.2% | 0.97 | 0.88 | 0.92 |
| DT | 99.5% | 0.94 | 0.92 | 0.92 |
| GB | 99.6% | 0.94 | 0.94 | 0.94 |
| XGBoost | 99.6% | 0.97 | 0.94 | 0.95 |
| RF | 99.6% | 0.97 | 0.94 | 0.95 |
| ERT | 99.7% | 0.97 | 0.94 | 0.95 |

detection. The confusion matrices generated by the ID models are illustrated in Figures 3 to 9.

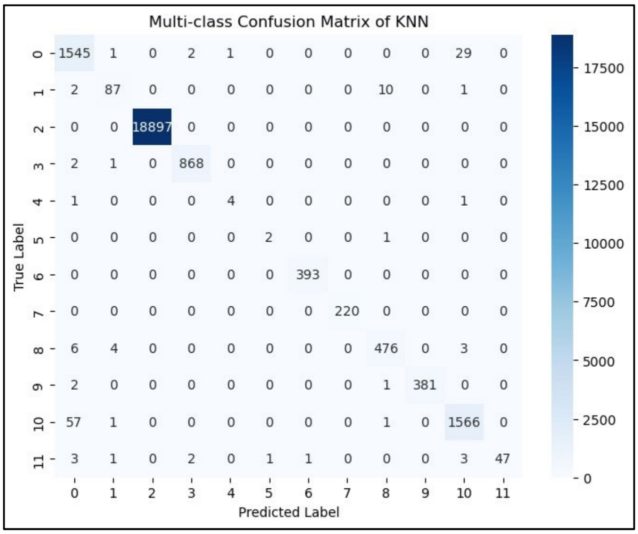


FIGURE 3. Multi-class evaluation confusion matrix of KNN.

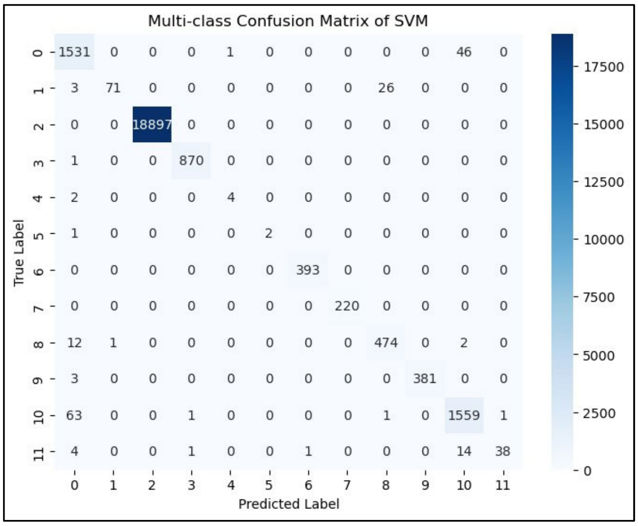


FIGURE 4. Multi-class evaluation confusion matrix of SVM.

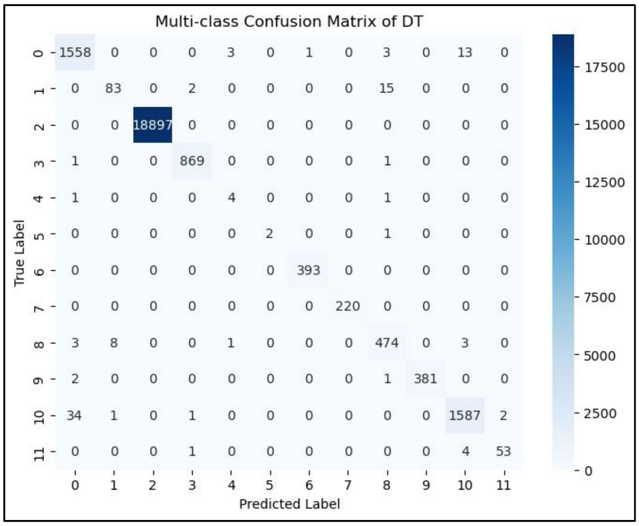


FIGURE 5. Multi-class evaluation confusion matrix of DT.

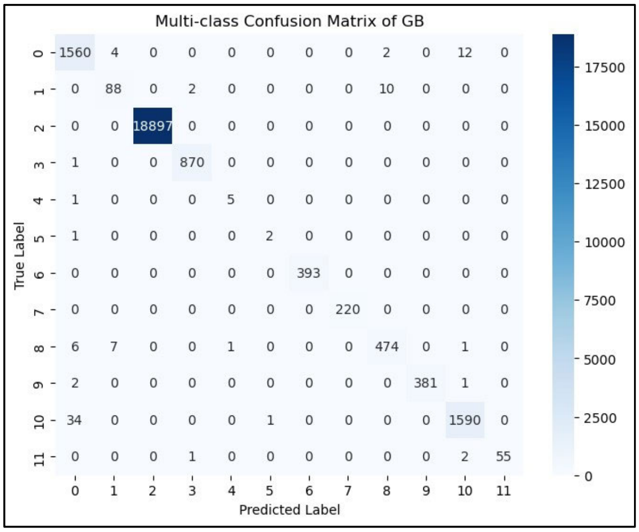
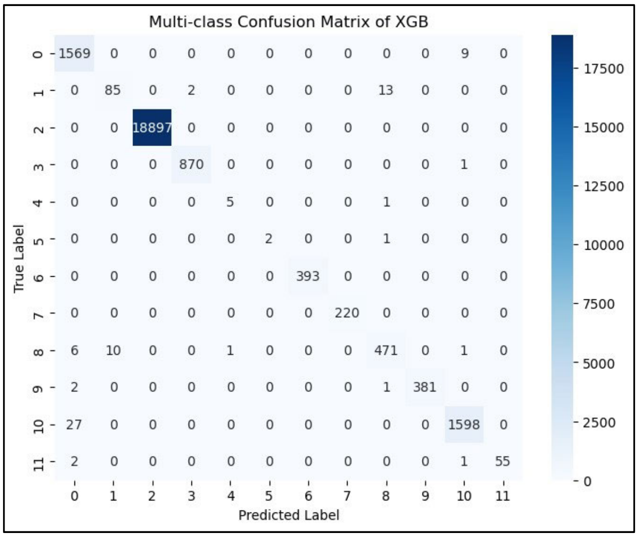


FIGURE 6. Multi-class evaluation confusion matrix of GB.

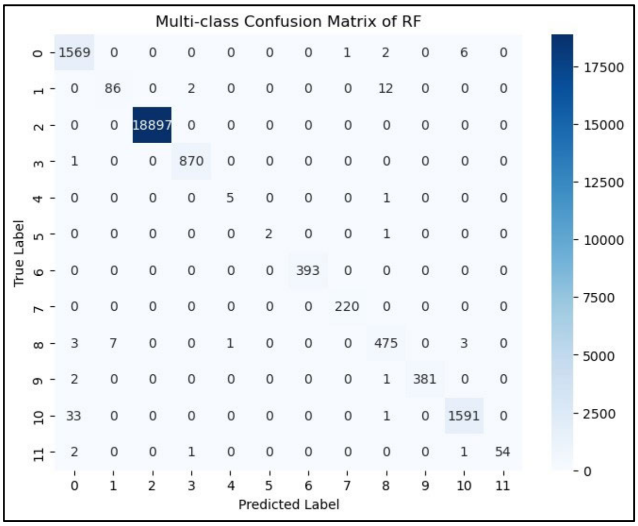
The confusion matrix for the KNN classifier, which can be seen in Fig. 3, indicates a strong predictive performance for specific classes. Notably, it correctly predicted 18,897 instances of class 2 and 1,566 instances of class 10. Nonetheless, the matrix also points out particular misclassification areas; for example, 29 instances of class 0 were inaccurately classified as class 10. Moreover, the matrix identifies a clear disparity in class distribution, with classes 5 and 11 having significantly fewer instances. As depicted in Fig. 4, the SVM demonstrates high accuracy for specific classes, notably class 2 with 18,897 accurate predictions and class 10 with 1,559 accurate predictions. Nevertheless, the matrix also highlights significant misclassifications, such as 46 cases where class 0 was incorrectly identified as class 10 and 26 cases where class 1 was misclassified as class 10. Furthermore, the matrix indicates some class imbalance, particularly evident in classes 1 and 11, which have fewer instances compared to others.

The DT classifier demonstrates strong performance in particular classes as depicted in Fig. 5, especially in class 2 with 18,897 accurate predictions and class 10 with 1,587 accurate predictions. However, there are clear instances of misclassifications, such as 13 cases of class 0 being incorrectly predicted as class 10 and 15 cases of class 1 being predicted as class 10. The GB classifier, as shown in Fig. 6, shows high accuracy for specific classes, particularly class 2 with 18,897 correct predictions and class 10 with 1,590 correct predictions. Despite these achievements, some misclassifications are evident, including 12 instances of class 0 being misclassified as class 10 and 10 instances of class 1 being misclassified as class 10.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1569 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 0 |
| 1 | 0 | 85 | 0 | 2 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 |
| 2 | 0 | 0 | 18897 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 870 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 393 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 220 | 0 | 0 | 0 | 0 |
| 8 | 6 | 10 | 0 | 0 | 1 | 0 | 0 | 0 | 471 | 0 | 1 | 0 |
| 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 381 | 0 | 0 |
| 10 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1598 | 0 |
| 11 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 55 |

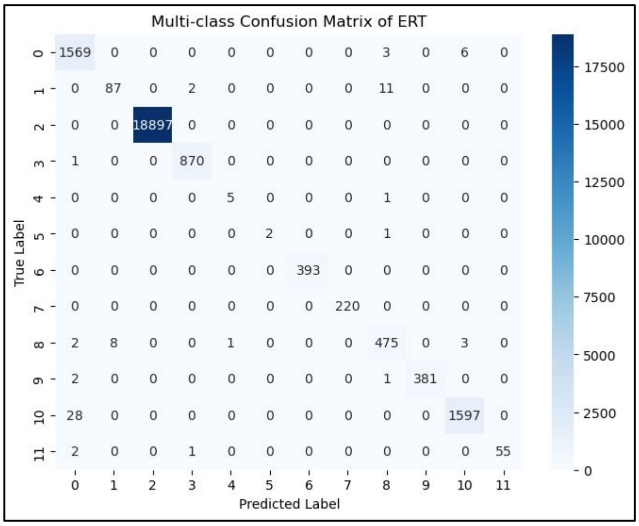
**FIGURE 7.** Multi-class evaluation confusion matrix of XGB.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1569 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 6 | 0 |
| 1 | 0 | 86 | 0 | 2 | 0 | 0 | 0 | 0 | 12 | 0 | 0 | 0 |
| 2 | 0 | 0 | 18897 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 870 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 393 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 220 | 0 | 0 | 0 | 0 |
| 8 | 3 | 7 | 0 | 0 | 1 | 0 | 0 | 0 | 475 | 0 | 3 | 0 |
| 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 381 | 0 | 0 |
| 10 | 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1591 | 0 |
| 11 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 54 |

**FIGURE 8.** Multi-class evaluation confusion matrix of RF.

In the analysis, the XGB classifier, as shown in Fig.7, demonstrates high accuracy for specific classes, particularly class 2 with 18,897 correct predictions and class 10 with 1,598 correct predictions. However, it also exhibits notable misclassifications, such as 9 instances of class 0 being incorrectly predicted as class 10 and 13 instances of class 1 misclassified as class 10. The analysis also highlights class imbalance, with certain classes like 1 and 11 having significantly fewer samples. On the other hand, the RF classifier, depicted in Fig. 8, indicates strong performance in predicting various classes, particularly class 2 with 18,897 correct predictions and class 10 with 1,591 correct predictions. However, the matrix also highlights certain misclassifications, such as 6 instances of class 0 being misclassified as class 10 and 12 instances of class 1 also being misclassified as class 10. ERT classifier, depicted in Fig. 9, demonstrates high accuracy in predicting specific classes, particularly class 2 with 18,897 correct predictions and class 10 with 1,597 correct predictions. Nevertheless, there are numerous instances of misidentification. For example, 6 occurrences of category 0 were mistakenly identified as category 10, and 11 instances of category 1 were incorrectly categorized as category 10.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1569 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 6 | 0 |
| 1 | 0 | 87 | 0 | 2 | 0 | 0 | 0 | 0 | 11 | 0 | 0 | 0 |
| 2 | 0 | 0 | 18897 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 870 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 393 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 220 | 0 | 0 | 0 | 0 |
| 8 | 2 | 8 | 0 | 0 | 1 | 0 | 0 | 0 | 475 | 0 | 3 | 0 |
| 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 381 | 0 | 0 |
| 10 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1597 | 0 |
| 11 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 55 |

**FIGURE 9.** Multi-class evaluation confusion matrix of ERT.

These findings indicate areas for enhancing the classifier, such as improving the model's ability to handle less common categories and refining the algorithm to minimize specific types of prediction mistakes, ultimately leading to an overall improvement in classification performance.

The figure 10 to 16 shows the Pr-Rec curves for DT, ERT, GB, KNN, SVM, RF, and XGB, which provides insight into the model capability of class discrimination. Namely, ERT, RF, and XGB have AP scores of almost 1.00 or 1.00 on most classes. However, class 5 emerged as a difficult class from all models, because their AP scores, especially in DT, GB, and KNN, record significantly lower AP scores, which shows difficulties in maintaining high precision and recall for this particular class. Contrarily, the classes 2, 3, 6, 7 are always performing fine, nearly with an AP score close to perfection in all models.

That among the different models compared here, the ensemble models (ERT, RF, XGB) tend to perform better compared to simple models, DT and KNN, on overall class-wise precision and recall because these techniques can handle class imbalance and complex decision boundaries well. Similarly, SVMs are also quite very well performed for most classes, excluding a small performance decrease for class 5. The Pr-Rec curves point to the fact that while the models do well on certain classes, their precision and recall for under-performing classes (like class 5) may require additional techniques such as more advanced feature engineering or finer class-balancing strategies. Altogether, these figures underpin the power of ensemble methods to achieve high performance across classes.

The ROC curves for all the different models-DT, ERT, GB, KNN, RF, SVM, and XGB indicate high model performance,

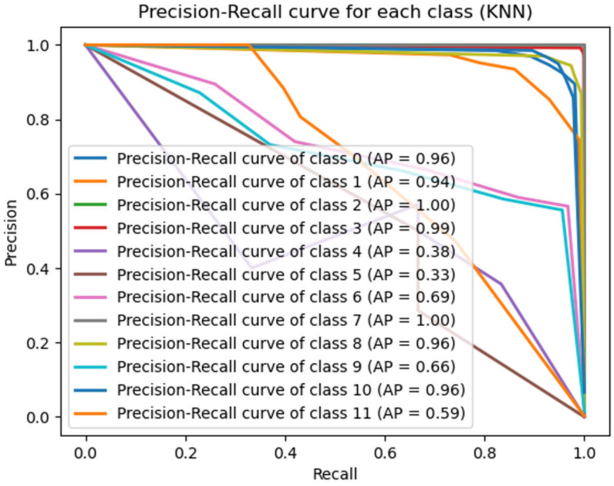


**FIGURE 10.** Precision-Recall curve for KNN.

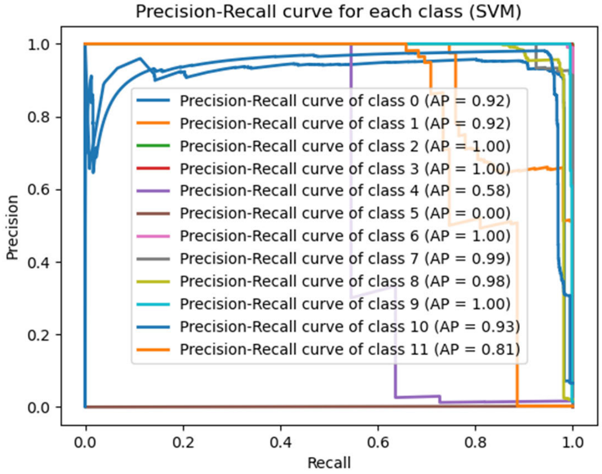


**FIGURE 11.** Precision-Recall curve for SVM.

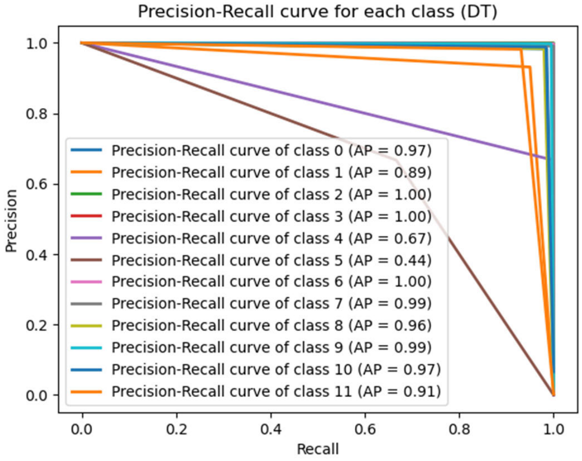


**FIGURE 12.** Precision-Recall curve for DT.

shown in figures 17 to 23. This is since most of these have a high value for AUC. Across all classifiers, there are classes

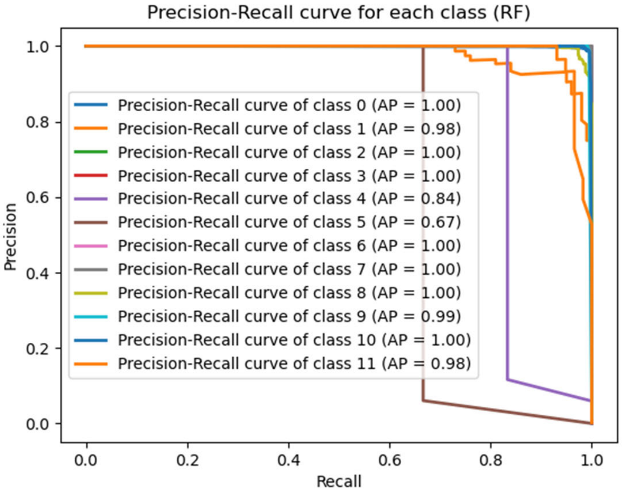


**FIGURE 13.** Precision-Recall curve for RF.

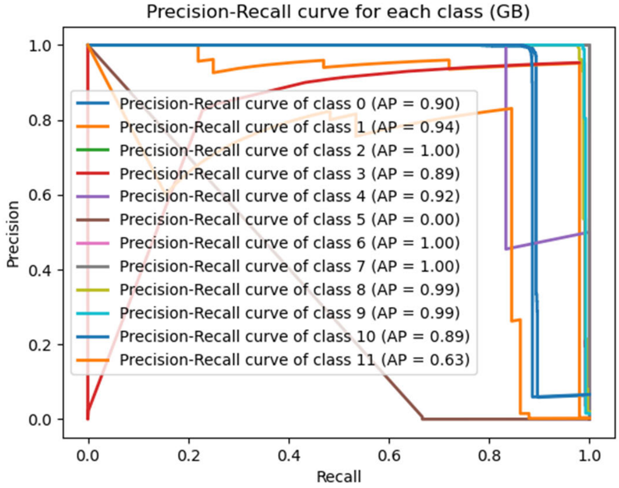


**FIGURE 14.** Precision-Recall curve for GB.

whose AUC is around 1.00, indicating a very good separation of instances between positive and negative. Class 5 is an outlier, with its AUC being as poor as 0.50 in ERT and 0.66 in the GB classifier, which indicates poor recognition. Also, the performance among ERT, RF, and XGB is almost perfect in most classes, with several cases of AUC values equal to 1.00, showing their power in handling both complex decision boundaries and class imbalances. Among these non-ensemble-based systems, the best performances of DT and KNN are related to relatively lower AUC values on some particular classes, such as class 5 and class 11. These allow one to make a conclusion that most of the classifiers tend to demonstrate outstanding results, but additional efforts should be performed either on the feature engineering or on the class rebalancing in segmentation of the worst performing classes.

Table 5 provides individual class performance metrics, as evaluated by the ID models: precision, recall, and F1 score, showing the efficiency of the model in recognizing

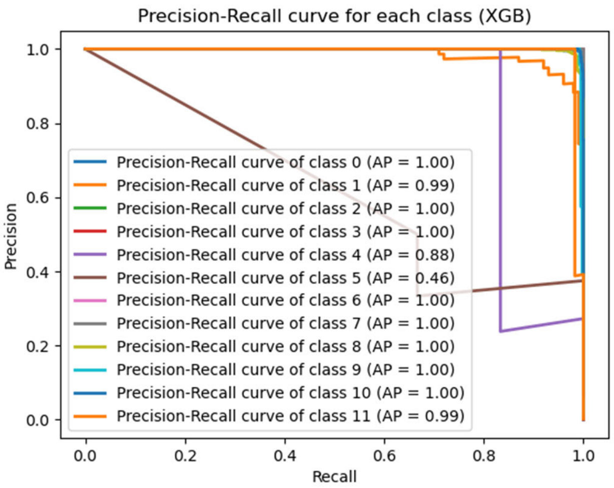


**FIGURE 15.** Precision-Recall curve for XGB.

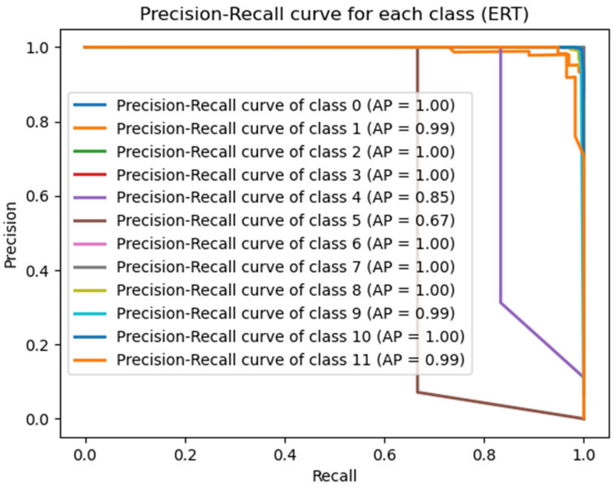


**FIGURE 16.** Precision-Recall curve for ERT.

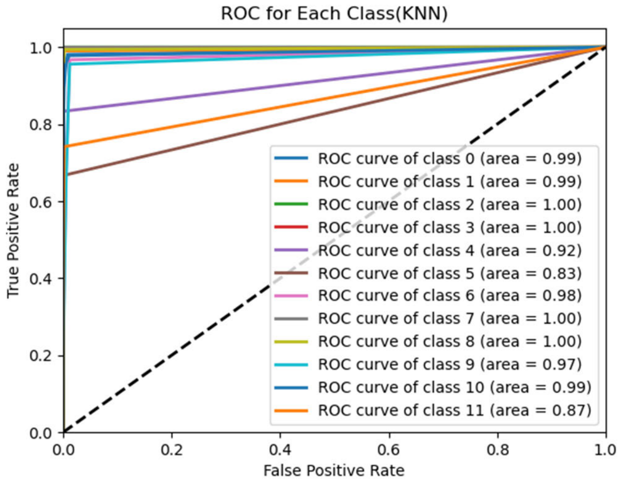


**FIGURE 17.** ROC curve for KNN.

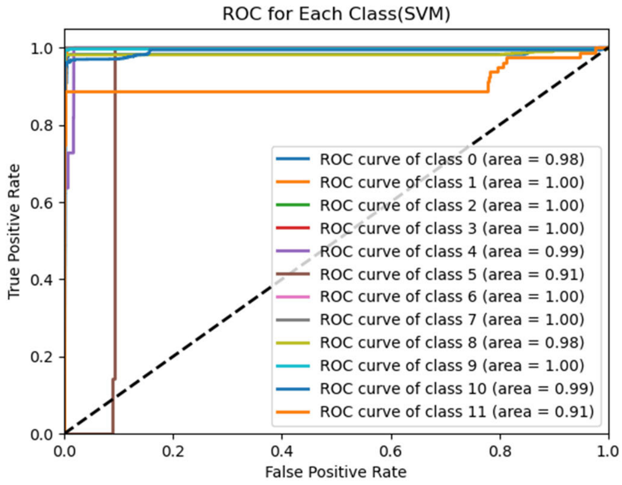


**FIGURE 18.** ROC curve for SVM.

and differentiating with a high degree of accuracy against different classes of intrusions.

KNN provides an accuracy of 99.4% with relatively lower recall and F1-Score compared to other models like XGBoost and ERT. KNN is sensitive to noisy data and computationally expensive during the prediction phase, hence it cannot perform well in real-time IoT networks for which rapid detection is one of the most important factors. KNN also relies highly on distance metrics, which do not work well in high-dimensional space inherent in IoT intrusion data [45]. On the other hand, SVM runs well in precision with an accuracy of 99.2%, though its recall value is lesser compared to the ensemble methods. This is perhaps because SVM are not good at handling large datasets with a lot of overlapping classes and hence might miss the detection of some type of intrusion, which in turn affects the recall. DT, though simple, achieve an accuracy of 99.5% while keeping the other metrics very high too. However, DT models tend to overfit to complex data, and that could be a reason for the slight underperformance in recall and F1-Score compared to the ensemble models such as RF and ERT. Generally, all the ensemble methods reduce overfitting by combining several trees in order to give a better generalization. The performances of GB, XGBoost, and RF are thus higher since they are able to combine several weak learners to form a strong classifier. The ERT model performed best, reaching accuracy as high as 99.7%, with an F1-Score of 0.95. The strength of ERT features in IoT network intrusion detection is due to the random feature selection during tree splits, which helps to avoid overfitting and strengthen its performance across diverse attacks. The ensemble methods-GB, XGBoost, RF, and ERT-outperform the rest because these models are more robust and handle high dimensional, imbalanced data found in intrusion detection tasks. Much of the improved performance of these models is due to a good balance between bias and variance; hence, they will be more deployable in real time for IoT environments.

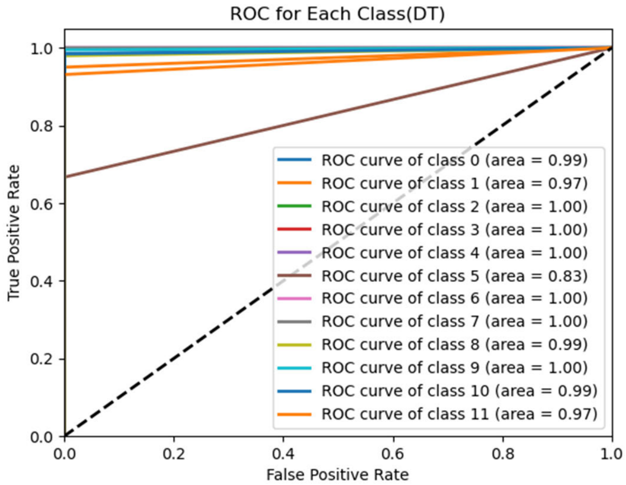


**FIGURE 19.** **ROC curve for DT.**

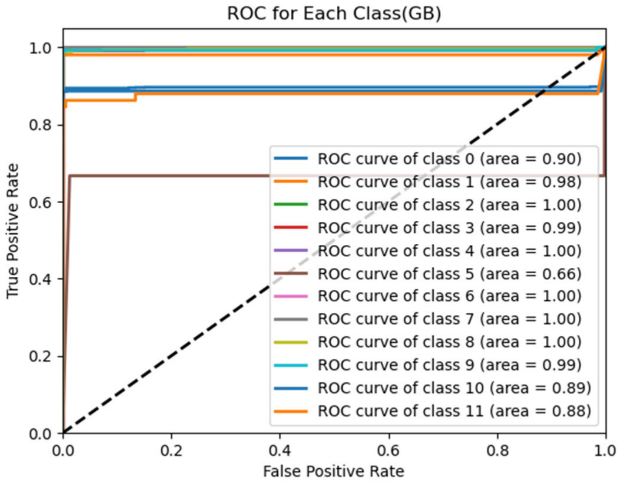


**FIGURE 21.** **ROC curve for GB.**

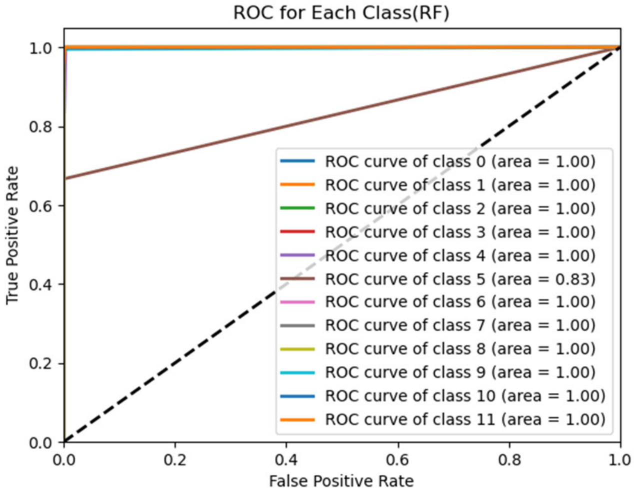


**FIGURE 20.** **ROC curve for RF.**

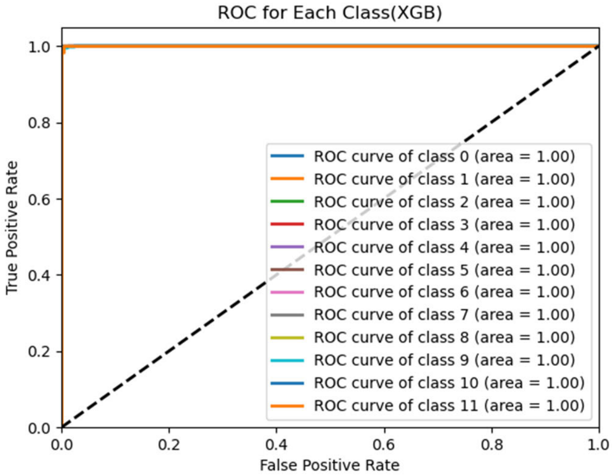


**FIGURE 22.** **ROC curve for XGB.**

### *D. COMPARATIVE ANALYSIS AND INSIGHTS*

Table 6 contains the analysis carried out in relevance to the effectiveness of various machine learning techniques for intrusion detection in IoT environments. Baich et al. formalized an analysis of the performance of machine learning models against the NSL-KDD dataset. In their study, the NB and RF models showed high accuracy rates of 99.26% and 99.13%, respectively, and can thus be considered as effective models for this dataset. Saba et al. proposed a convolutional CNN model against the datasets NID and BoT-IoT and obtained accuracy results of 9.51% on the NID dataset and 92.85% on the BoT-IoT dataset. These findings may support the importance of optimization in respect to each dataset [46], [47]. Sharma et al. applied a DNN on the UNSW-NB15 dataset and got an accuracy up to 91%, which proved the efficiency of the model in complex network traffic environments Keshk et al. applied LSTM models to different datasets, achieving accuracies that ranged between 78.7% and 87.3%. This work therefore underlined how a dataset should be chosen appropriately for any recurrent neural network evaluation. Chaganti et al. applied LSTM models on datasets SDNIoT-focused (DS1, DS2) and obtained accuracy up to 97.1%, thereby showing its effectiveness in software-defined network IoT contexts [48], [49], [50].

Bajpai et al. proposed an Intrusion Detection Framework using Machine Learning, called IDFML, which recorded an accuracy of 98.68% when tested against the IoTID20 dataset, thus proving that a complete machine learning system is reliable for IoT intrusion detection. Arthi et al. combined DNN with SVM on an SDN dataset and got an accuracy of 96.7%. This showed the positive side of hybrid approaches that include deep and traditional machine learning methods. They further introduced quantum-inspired autoencoder models for Sharmila and Nagapadma against the RT-IOT2022 dataset with accuracy of 97.25% and 96.35%, respectively, proving that quantum-inspired approaches are able to improve IDS performance in IoT network security [23], [51], [52].

**TABLE 5.** Detailed findings produced for every class by ID model.

| Model | Parameter | Classes | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| KNN | Pr | 0.93 | 0.92 | 1.00 | 0.98 | 0.57 | 1.00 | 0.97 | 0.99 | 0.96 | 0.98 | 0.95 | 0.83 |
| | Rec | 0.94 | 0.86 | 1.00 | 1.00 | 0.67 | 0.67 | 0.98 | 1.00 | 0.97 | 0.97 | 0.95 | 0.43 |
| | F1S | 0.94 | 0.89 | 1.00 | 0.99 | 0.62 | 0.80 | 0.98 | 0.99 | 0.97 | 0.98 | 0.95 | 0.57 |
| SVM | Pr | 0.95 | 0.99 | 1.00 | 1.00 | 0.80 | 1.00 | 1.00 | 1.00 | 0.95 | 1.00 | 0.96 | 0.97 |
| | Rec | 0.97 | 0.71 | 1.00 | 1.00 | 0.67 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.96 | 0.66 |
| | F1S | 0.96 | 0.83 | 1.00 | 1.00 | 0.73 | 0.80 | 1.00 | 1.00 | 0.96 | 1.00 | 0.96 | 0.78 |
| DT | Pr | 0.97 | 0.91 | 1.00 | 0.99 | 0.50 | 1.00 | 1.00 | 0.99 | 0.97 | 1.00 | 0.99 | 0.95 |
| | Rec | 0.99 | 0.88 | 1.00 | 1.00 | 0.67 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.98 | 0.90 |
| | F1S | 0.98 | 0.89 | 1.00 | 1.00 | 0.57 | 0.80 | 1.00 | 1.00 | 0.97 | 1.00 | 0.98 | 0.92 |
| XGB | Pr | 0.98 | 0.91 | 1.00 | 1.00 | 0.83 | 1.00 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 1.00 |
| | Rec | 0.99 | 0.84 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.98 | 0.95 |
| | F1S | 0.99 | 0.87 | 1.00 | 1.00 | 0.83 | 0.80 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 0.97 |
| GB | Pr | 0.97 | 0.88 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 1.00 |
| | Rec | 0.99 | 0.87 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.98 | 0.95 |
| | F1S | 0.98 | 0.87 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 1.00 | 0.98 | 0.97 |
| RF | Pr | 0.97 | 0.91 | 1.00 | 1.00 | 0.71 | 1.00 | 1.00 | 0.99 | 0.97 | 1.00 | 0.99 | 1.00 |
| | Rec | 0.99 | 0.88 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.98 | 0.93 |
| | F1S | 0.98 | 0.89 | 1.00 | 1.00 | 0.77 | 0.80 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 0.96 |
| ERT | Pr | 0.98 | 0.92 | 1.00 | 1.00 | 0.83 | 1.00 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 1.00 |
| | Rec | 0.99 | 0.86 | 1.00 | 1.00 | 0.83 | 0.67 | 1.00 | 1.00 | 0.97 | 0.99 | 0.98 | 0.95 |
| | F1S | 0.99 | 0.89 | 1.00 | 1.00 | 0.83 | 0.80 | 1.00 | 1.00 | 0.97 | 1.00 | 0.99 | 0.97 |

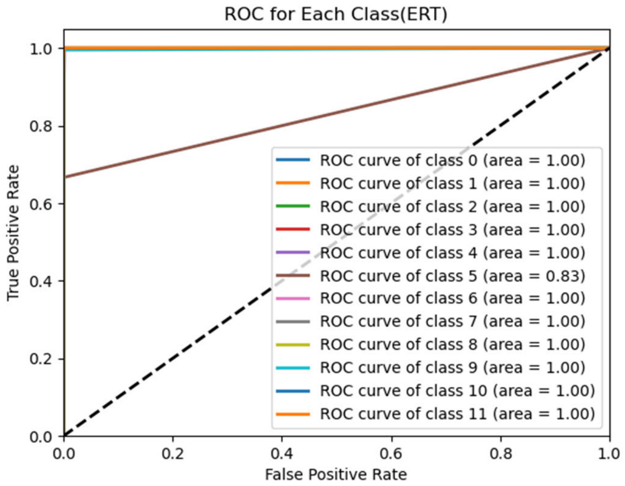


**FIGURE 23.** ROC curve for ERT.

The evaluation of proposed study using the RT-IOT2022 dataset revealed that the models were very accurate, in particular, the KNN, SVM, DT, GB, and RF. Among them, ERT topped 99.7% in accuracy, while ensemble techniques like RF, GB, and XGB underline their strength and reliability for detecting IoT attacks in real time. Our findings suggested that among the various attack types, such as DoS, DDoS, and malware-based attacks, ensemble methods-GB, XGBoost, RF, and ERT-showed consistent superiority over others. Of these, the ERT proved very effective in detecting high-frequency attacks, such as DoS and DDoS, with an accuracy of 99.7%, precision, and recall, thereby emerging as the most effective model for real-time intrusion detection in IoT environments. However, XGBoost and RF are also very competitive among all types of attacks, standing out for low-frequency attacks such as probing and reconnaissance, with F1-scores up to 0.95.

The models were very robust in distinguishing between normal and malicious traffic independently of attack complexity or frequency. On the other side, simpler models such as KNN and SVM, though performing decently well for high-frequency attacks, showed somewhat lower recall and precision for rare sophisticated attacks. This, therefore, underlines the flexibility of ensemble methods particularly in handling the diverse and evolving nature of attacks on IoT networks.

**TABLE 6.** Comparative analysis with alternative ID techniques for IoT attacks.

| Reference# | IDS Model | Dataset | Accuracy |
|---|---|---|---|
| **[46]** | DT | NSL-KDD | 98.65% |
| | RF | | 99.13% |
| | NB | | 99.26% |
| | SVM | | 97.7% |
| **[47]** | CNN | NID | 9.51% |
| | | BoT-IoT | 92.85% |
| **[48]** | DNN | UNSW-NB15 | 91% |
| **[49]** | LSTM | NSL-KDD | 78.7% |
| | | UNSW-NB15 | 85.8% |
| | | TON_IoT | 87.3% |
| **[50]** | LSTM | SDNIoT-focused (DS1, DS2) | 97.1% |
| **[51]** | IDFML | IoTID20 | 98.68% |
| **[52]** | DNN + SVM | SDN dataset | 96.7% |
| **[23]** | QAE-f16 | RT-IOT2022 | 97.25% |
| | QAE-u8 | | 96.35% |
| Proposed IDS Models | KNN | RT-IOT2022 | 99.4% |
| | SVM | | 99.2% |
| | DT | | 99.5% |
| | GB | | 99.6% |
| | XGB | | 99.6% |
| | RF | | 99.6% |
| | ERT | | 99.7% |

An evaluation of the efficiency of different machine learning approaches toward detecting intrusions in IoT environments was investigated. In this research, experiments on a number of models with a number of data sets have returned some interesting findings: RF and NB are very effective in detecting intrusions from the NSL-KDD data set, while LSTM can provide very accurate results when working with SDNIoT-focused datasets. Further, other ensemble learning techniques, such as ERT, RF, GB, and XGB, exhibited better performance in IoT real-time attack detection. The result proves that ERT maintains an accuracy of 99.7%. It is already proven that ensemble learning mostly gives high accuracy, hence the need to emphasize flexibility and efficiency when applying it to intrusion detection in IoT. It also pointed out that dataset-specific optimization could be quite important, and hybrid and ensemble approaches might further improve the performance of IDS. These insights provide valuable guidance on choosing and exploiting machine learning models in order to provide more security for IoT networks.

It includes a specialized dataset, namely RT-IOT2022, in this paper, which is tailored for an IoT environment. This dataset will be able to provide a more representative benchmark for IoT-specific attack detection compared to the NSL-KDD, UNSW-NB15, and SDN datasets that were used in prior research.

More importantly, the strategies of tuning, feature selection, and data preprocessing used in the present work have been optimized with regard to unique features of IoT networks, where low-powered devices are used, including heterogeneous communication protocols. The proposed models are real-time and effective, with least utilization of resources to ensure a detection rate as high as 99.96%. Other works using similar algorithms relate to more traditional network environments or general intrusion detection tasks with different datasets and targets of performance optimization. This is where the contribution of this work to the field differs, since it contributes in the realm of IoT-specific challenges and real-time detection.

## V. DISCUSSION

Similar to other studies, there are various threats to validity that could affect the interpretation of the results and generalizing them. One major internal threat would be the utilization of a biased dataset for both training and testing machine learning models; this does not reflect complete representative diversity in attacks and network traffic as observed in real IoT environments. This leads to overfitting, especially in ensemble models such as RF and ERT, which learn specific patterns in the training data but do poorly when exposed to scenarios that are not encountered before. To handle this, cross-validation was done, but even at that, this does not avoid the risk. Externally, the results might not generalize to every IoT environment due to varied devices, protocols, and network architectures not really captured by the dataset. Also, although GB and XGBoost showed promising performance in this environment, these algorithms do have high computational requests that could seriously compromise scalability on resource-constrained IoT devices. Construct validity in this respect will also be a point of concern because the operationalization of the attacks in the dataset may not truly reflect the real-world constantly changing nature of threats, and feature selection may have missed key attributes of IoT traffic patterns.

Finally, questions of validity might arise when relying on metrics that are related to accuracy, precision, and recall but do not take into account more critical elements such as false negatives or real-time detection capability. Whereas in this work the ensemble methods outperformed other models, context-specific considerations about network complexity and the nature of the attacks in IoT environments may make SVM or KNN a more fitting choice in their respective situations. Addressing these threats, any further research in this respect would make IDS more applicable and reliable in case of IoT networks.

The practical implications of findings from this study indicate that the ensembles, in particular ERT, XGBoost, and Random Forest, proposed here can add significant value to the IoT security frameworks that already exist. These models can be integrated into real-time monitoring systems that detect and respond to potential threats in IoT networks with high precision. Deployment in the IoT environment faces some challenges, pertaining to resource constraints on the IoT devices, scalability concerning huge and complex networks, and real-time processing. Various challenges identified could be addressed by optimizing the models for resource efficiency, distributed processing facilitated through edge computing, and continuous model updates to handle evolving threats. Additionally, practical considerations like model adaptability and minimal latency are crucial for seamless integration into diverse IoT applications.

## VI. CONCLUSION

Efficient network security is a basic requirement of the rapidly changing environment of IoT. One of the pivotal concepts ensuring the security of the IoT environment against a plethora of malicious activities includes IDS. This paper presents the performance evaluation of various machine learning classifiers—KNN, SVM, DT, GB, XGB, RF, and ERT—for intrusion detection in IoT networks using the RT-IoT2022 dataset. The comparative analysis shows that ensemble models such as GB, XGBoost, RF, and ER perform well across various metrics, including F1-Score, recall, accuracy, and precision. In particular, the ERT showed the highest accuracy of 99.7% among all models, also coupled with very good precision and recall, thus making ERT the most effective model for real-time IoT attack detection. Besides, both XGBoost and RF are very reliable and accurate, returning an F1-Score of 0.95 respectively.

These findings highlight the robustness and reliability of ensemble methods in enhancing IoT security. The superior performance of these models suggests their suitability for deployment in real-time IoT IDS, contributing significantly to the ongoing efforts to fortify IoT infrastructures against evolving cyber threats. Results highlight the potential of these approaches in enhancing the security of IoT-based infrastructures by developing intrusion detection solutions that are scalable and efficient. Also, the importance of such insights is not only in the high accuracy obtained but mostly in practical implications regarding the protection of critical IoT networks, increasingly targeted by sophisticated cyber threats. The work therefore provides important knowledge to the IoT security landscape by reinforcing that ensemble methods can serve as strong defensive mechanisms in providing a path toward further advancements in securing IoT ecosystems.

In the future work, this method will be implemented on real network traffic and measuring the performance. In future predicting intruder's next action can be done to protect IoT environment proactively. This method is only limited to detection approach, we can also add mitigation and prevention measure to enhance its effectiveness.

For the future, hybrid models that combine the strengths of different approaches for much higher detection accuracy would be much better to investigate. The incorporation of unsupervised learning methods, targeting anomaly detection, will further improve the system's ability in finding threats that have not been considered before. Future deployments shall include testing the models with real network traffic and, where feasible, proactively predicting intruders' next steps towards better protection of IoT environments.

## REFERENCES

[1] S. Mishra and A. K. Tyagi, "The role of machine learning techniques in Internet of Things-based cloud applications," in *Artificial intelligence-based Internet of Things Systems*. Switzerland: Springer, 2022, pp. 105–135.

[2] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors*, vol. 20, no. 13, p. 3625, Jun. 2020.

[3] M. Lombardi, F. Pascale, and D. Santaniello, "Internet of Things: A general overview between architectures, protocols and applications," *Information*, vol. 12, no. 2, p. 87, Feb. 2021.

[4] Y. R. Siwakoti, M. Bhurtel, D. B. Rawat, A. Oest, and R. Johnson, "Advances in IoT security: Vulnerabilities, enabled criminal services, attacks and countermeasures," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11224–11239, Jul. 2023.

[5] E. Illi, M. Qaraqe, S. Althunibat, A. Alhasanat, M. Alsafasfeh, M. de Ree, G. Mantas, J. Rodriguez, W. Aman, and S. Al-Kuwari, "Physical layer security for authentication, confidentiality, and malicious node detection: A paradigm shift in securing IoT networks," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 347–388, 1st Quart., 2024.

[6] J. Fox, *Top Cybersecurity Statistics for 2024*. USA: Cobalt, 2023. [Online]. Available: https://www.cobalt.io/blog/cybersecurity-statistics-2024

[7] A. Marton and S.Systems, *IoT Malware Attacks up by 37% in the First Half of 2023*. IoTAC Association: EU Research and Innovation Programme, 2023.

[8] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, "A comprehensive review on secure routing in Internet of Things: Mitigation methods and trust-based approaches," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4186–4210, Mar. 2021.

[9] S. A. Hamad, Q. Z. Sheng, W. E. Zhang, and S. Nepal, "Realizing an Internet of Secure Things: A survey on issues and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1372–1391, 2nd Quart., 2020.

[10] O. H. Abdulganiyu, T. A. Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *Int. J. Inf. Secur.*, vol. 22, no. 5, pp. 1125–1162, Oct. 2023.

[11] A. S. Dina and D. Manivannan, "Intrusion detection based on machine learning techniques in computer networks," *Internet Things*, vol. 16, Dec. 2021, Art. no. 100462.

[12] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Proc. Comput. Sci.*, vol. 171, pp. 1251–1260, Jan. 2020.

[13] B. S. A. R. Nagapadma, *RT-IoT2022 2024: UCI Machine Learning Repository*, USA, 2024.

[14] M. M. Inuwa and R. Das, "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks," *Internet Things*, vol. 26, Jul. 2024, Art. no. 101162.

[15] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, "Anomaly-based intrusion detection system in IoT using kernel extreme learning machine," *J. Ambient Intell. Humanized Comput.*, vol. 15, no. 1, pp. 231–242, Jan. 2024.

[16] S. Saif et al., "HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare," *Microprocess. Microsyst.*, 2022, Art. no. 104622.

[17] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in Internet of Things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–23, Dec. 2021.

[18] M. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, "MLSTL-WSN: Machine learning-based intrusion detection using SMOTETomek in WSNs," *Int. J. Inf. Secur.*, vol. 23, no. 3, pp. 2139–2158, Jun. 2024.

[19] N. Saran and N. Kesswani, "A comparative study of supervised machine learning classifiers for intrusion detection in Internet of Things," *Proc. Comput. Sci.*, vol. 218, pp. 2049–2057, Jan. 2023.

[20] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks," *Computers*, vol. 12, no. 2, p. 34, Feb. 2023.

[21] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *J. Sensor Actuator Netw.*, vol. 12, no. 2, p. 29, Mar. 2023.

[22] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet Things*, vol. 3, no. 1, p. 5, May 2023.

[23] B. S. Sharmila and R. Nagapadma, "Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset," *Cybersecurity*, vol. 6, no. 1, p. 41, Sep. 2023.

[24] T. S. Othman, K. R. Koy, and S. M. Abdullah, "Intrusion detection systems for IoT attack detection and identification using intelligent techniques," *Networks*, vol. 5, p. 6, Jan. 2023.

[25] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, p. 1053, Mar. 2024.

[26] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for industrial IoT environment," *Exp. Syst. Appl.*, vol. 249, Sep. 2024, Art. no. 123808.

[27] N. Islam, F. Farhin, I. Sultana, M. Shamim Kaiser, M. Sazzadur Rahman, M. Mahmud, A. S. M. Sanwar Hosen, and G. Hwan Cho, "Towards machine learning based intrusion detection in IoT networks," *Comput., Mater. Continua*, vol. 69, no. 2, pp. 1801–1821, 2021.

[28] V. Choudhary, S. Tanwar, T. Choudhury, and K. Kotecha, "Towards secure IoT networks: A comprehensive study of metaheuristic algorithms in conjunction with CNN using a self-generated dataset," *MethodsX*, vol. 12, Jun. 2024, Art. no. 102747.

[29] V. Choudhary et al., "Towards secure IoT networks: A comprehensive study of metaheuristic algorithms in conjunction with CNN using a self-generated dataset," *MethodsX*, vol. 12, 2024, Art. no. 102747.

[30] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021.

[31] K. G. Sharma and Y. Singh, "Predicting intrusion in a network traffic using variance of neighboring object's distance," *Int. J. Comput. Netw. Inf. Secur.*, vol. 15, no. 2, pp. 73–84, Apr. 2023.

[32] B. S. Bhati and C. S. Rai, "Analysis of support vector machine-based intrusion detection techniques," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 2371–2383, Apr. 2020.

[33] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018.

[34] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 538–566, 1st Quart., 2023.

[35] D. Kumar and N. A. Priyanka, "Decision tree classifier: A detailed survey," *Int. J. Inf. Decis. Sci.*, vol. 12, no. 3, pp. 246–269, 2020.

[36] S. Lee, T. P. Vo, H.-T. Thai, J. Lee, and V. Patel, "Strength prediction of concrete-filled steel tubular columns using categorical gradient boosting algorithm," *Eng. Struct.*, vol. 238, Jul. 2021, Art. no. 112109.

[37] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020.

[38] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021.

[39] T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial Internet of Things intrusion detection systems," *Sustainability*, vol. 14, no. 14, p. 8707, Jul. 2022.

[40] N. S. Akash, S. Rouf, S. Jahan, A. Chowdhury, and J. Uddin, "Botnet detection in IoT devices using random forest classifier with independent component analysis," *J. Inf. Commun. Technol.*, vol. 21, no. 2, pp. 201–232, 2022.

[41] J. B. Awotunde, F. E. Ayo, R. Panigrahi, A. Garg, A. K. Bhoi, and P. Barsocchi, "A multi-level random forest model-based intrusion detection using fuzzy inference system for Internet of Things networks," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 31, Mar. 2023.

[42] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," *Babylonian J. Mach. Learn.*, vol. 2024, pp. 69–79, Jun. 2024.

[43] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wireless Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, Apr. 2020.

[44] M. R. Camana Acosta, S. Ahmed, C. E. Garcia, and I. Koo, "Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks," *IEEE Access*, vol. 8, pp. 19921–19933, 2020.

[45] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 167–185, Jan. 2024.

[46] M. Baich, T. Hamim, N. Sael, and Y. Chemlal, "Machine learning for IoT based networks intrusion detection: A comparative study," *Proc. Comput. Sci.*, vol. 215, pp. 742–751, Jan. 2022.

[47] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107810.

[48] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108626.

[49] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Inf. Sci.*, vol. 639, Aug. 2023, Art. no. 119000.

[50] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep learning approach for SDN-enabled intrusion detection system in IoT networks," *Information*, vol. 14, no. 1, p. 41, Jan. 2023.

[51] S. Bajpai, K. Sharma, and B. K. Chaurasia, "Intrusion detection framework in IoT networks," *Social Netw. Comput. Sci.*, vol. 4, no. 4, p. 350, Apr. 2023.

[52] S. Bajpai, K. Sharma, and B. K. Chaurasia, "Intrusion detection framework in iot networks," *SN Comput. Sci.*, vol. 4, no. 4, p. 350, 2023.

**NAJM US SAMA** received the M.Sc. degree in cyber security from the University of Derby and the Ph.D. degree in computer science from the University of Malaysia Sarawak. She is currently a Distinguished Researcher and academic with a robust background in computer science, machine learning, and cybersecurity. With over nine years of professional experience, she has made significant contributions to the fields of machine learning, cybersecurity, and wireless sensor networks. She has published numerous research articles and conducted workshops on critical topics, such as machine learning applications in healthcare and cybersecurity.

**SAEED ULLAH** received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea. He was a Research Assistant with the School of Electrical Engineering & Computer Science, NUST Pakistan. Currently, he is a Lecturer with the School of Computing and Engineering, University of Derby, U.K. His current research interest includes multimedia communication in future networks.

**S. M. AHSAN KAZMI** received the Ph.D. degree from Kyung Hee University, South Korea, in 2017. He is currently a Senior Lecturer with the Department of Computer Science and Creative Technologies, University of the West of England, Bristol. Prior to this, he was an Assistant Professor with the Institute of Information Security and Cyber-Physical System, Innopolis University, Russia, for three years. He was a Postdoctoral Fellow with the Department of Computer Science and Engineering, Kyung Hee University, from 2017 to 2018. His research interests include applying analytical techniques of optimization, machine learning, and game theory to radio resource management for future networks. He received the best KHU Thesis Award in Engineering, in 2017, and several best paper awards from prestigious conferences.

**MANUEL MAZZARA** received the Ph.D. degree in computer science from the University of Bologna, Italy. He is currently a Professor in computer science with Innopolis University, Russia, with a research background in software engineering, service oriented architecture, concurrency theory, formal methods, and software verification. He has published many relevant and highly cited articles, in particular in the field of service engineering and software architectures and has collaborated with European and U.S. industries and governmental and inter-governmental organizations, such as United Nations, always at the edge between science and software production. The work conducted by him and his team in recent years focuses on the development of theories, methods, tools, and programs covering the two major aspects of software engineering, such as the process side, related to how we develop software, and the product side, concerning the results of this process.