INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTER SYSTEMS
E-ISSN: 2180-0650
VOLUME 11, ISSUE 1, 2025, pp. 59 – 69
DOI: https://doi.org/10.15282/ijsecs.11.1.2025.5.0137

RESEARCH ARTICLE

Machine learning for IOT security: Detecting and mitigating cyber anomalies

Hiba Ahmed[1], Razan Alharith[2], Ashraf Osman Ibrahim[3*] and Nada Adam[4]

[1] Department of Information Technology, College of Customs, Medical Science and Technology, Khartoum, Sudan.
[2] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, Sichuan, 611756, China.
[3] Department of Computing, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak, Malaysia.
[4] Department of Computer Science, the Applied College, Northern Border University, Arar 91431, Saudi Arabia.

Abstract - Rapid IoT device proliferation creates critical security vulnerabilities often missed by conventional methods. This vital research evaluates seven machine learning models (Random Forest, Gradient Boosting, Neural Networks, kNN, SVM, Decision Tree, and Naive Bayes) for robust IoT anomaly detection using comprehensive ToN-IoT and BoT-IoT datasets. Random Forest and Gradient Boosting significantly advanced IoT security, demonstrating superior, often perfect, performance on key metrics like AUC, accuracy, and F1. Neural Networks also excelled. SVM and kNN achieved high accuracy but showed varied sensitivity to rare attacks. Naive Bayes struggled with data complexity, while the Decision Tree's operational failure on one dataset stressed the need for careful validation. This study underscores machine learning's potential to enhance IoT resilience. Performance variations and challenges such as class imbalance necessitate tailored solutions. These findings establish a foundation for future work in ensemble methods, explainable AI (XAI), feature engineering, and strategies for managing large-scale imbalanced data to fortify IoT security.

Article History
Received : 22 March 2025
Revised : 17 July 2025
Accepted : 4 August 2025
Published : 8 August 2025

Keywords
Internet of Things (IoT)
Cybersecurity
Cyber Anomalies
Anomaly Detection
Cyber-attack Classification
Machine learning

1. Introduction

The rapid expansion of the Internet of Things (IoT) has transformed multiple industries, including healthcare, agriculture, and manufacturing, by enabling uninterrupted connectivity among billions of devices. However, this increased interconnectedness introduces important safety shortcomings, making IoT systems prime targets for attackers [1]. The reliance upon conventional anomaly detection methods, which generally utilise mathematical and rule-based approaches, has proven inadequate in tackling new risks and the dynamic nature of IoT settings. The current solutions fail to sufficiently address the complexities and breadth of contemporary IoT ecosystems, leading to potential security vulnerabilities and increased risks for consumers [2].

Anomaly detection is critical for safeguarding IoT systems, as it helps identify harmful activities and system malfunctions [3]. Yet, existing techniques often falter under the weight of diverse and voluminous data generated by interconnected devices. Traditional approaches to anomaly detection are not only limited in their ability to recognize new, unknown threats but also face challenges related to integration with heterogeneous IoT infrastructures. This includes difficulties in deploying signature-based intrusion detection systems that are ill-equipped to handle the rapid evolution of attack vectors [4]. Mitigating cyber anomalies in IoT environments is a critical step in maintaining system resilience and ensuring uninterrupted device operation. Once anomalies are detected, machine learning techniques can be employed to intelligently respond to threats through real-time decision-making and automated defence mechanisms [5]. These may include isolating compromised nodes, adjusting access controls, or triggering alerts for further investigation. Unlike traditional rule-based methods, machine learning models can adapt to evolving attack patterns and continuously learn from new data, making them ideal for dynamic IoT ecosystems.

Considering these challenges, machine learning (ML) has emerged as a promising alternative for anomaly detection in IoT networks. ML techniques can analyse vast amounts of data, uncover complex patterns, and continuously adapt to changing conditions. However, the application of ML in IoT security is not without its own set of hurdles. Current implementations struggle with issues such as data segmentation, high network costs, and the resource constraints of many IoT devices [6]. Furthermore, existing algorithms often fail to identify critical data characteristics, leading to missed or misinterpreted anomalies [7]. Despite the advancements in ML algorithms, particularly deep learning and federated learning, significant gaps remain in ensuring effective anomaly detection in IoT environments [8]. The increasing volume and variability of data challenge detection systems, making it challenging to discern between normal and anomalous behaviour. Additionally, the adaptive nature of cyber threats necessitates continuous updates and improvements to detection systems, further complicating the security landscape [9].

This study undertakes an in-depth analysis and comparison of seven supervised machine learning algorithms for anomaly detection within Internet of Things (IoT) networks, aiming to address urgent security concerns. The research evaluates the performance of Random Forest, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Naive Bayes, Decision Tree, Neural Networks, and Gradient Boosting (XGBoost). Utilizing established datasets such as ToN-IoT and BoT-IoT, the objective is to assess these algorithms using various performance indicators, including accuracy, precision, recall, F1-score, and the Receiver Operating Characteristic (ROC) curve, to identify algorithms best suited for specific IoT security requirements. This paper, therefore, presents a comprehensive comparative analysis of these seven machine learning algorithms, focusing on their efficacy in detecting anomalies in IoT settings. Beyond evaluating detection

*CORRESPONDING AUTHOR | A.O. Ibrahim | ✉ ashraf@utp.edu.my
© 2025 The Authors. Published by Universiti Malaysia Pahang Al-Sultan Abdullah Press. This is an open access article under the CC BY-NC 4.0 license

capabilities, the study also considers how these models can support mitigation strategies to reduce the impact of cyber threats. By employing two widely recognized and diverse datasets (ToN-IoT and BoT-IoT), the research provides empirical insights into the strengths and limitations of each model. This leads to practical recommendations for selecting appropriate algorithms, particularly for resource-constrained IoT deployments. Furthermore, the work addresses challenges such as class imbalance and underscores the need for advanced techniques to improve the detection of infrequent attack types. As IoT adoption grows, the necessity for intelligent, adaptable, and efficient security solutions becomes paramount. This study offers a timely contribution, laying the groundwork for future advancements in areas like model efficiency, ensemble learning, and explainable artificial intelligence.

The subsequent sections outline the organisational structure of the remaining portions of this document: The initial section presents an overview of the subject matter. Section 2 contains a survey of related works on the topic. Section 3 summarises the methodology employed in the study. Section 4 provides a description of the performance evaluation metrics that were utilised. Section 5 presents the findings and analyses the implications of these findings. Section 6 concludes the paper and offers recommendations for future opportunities.

2. Related Work

Identifying anomalies in computer systems and networks is essential for ensuring security, especially in the fast-changing environment of the IoT. Numerous machine learning methodologies have been developed to improve the accuracy and effectiveness of anomaly detection, with research demonstrating notable enhancements in detection capabilities [10]. Nevertheless, although numerous approaches have been formulated, they often demonstrate significant strengths and weaknesses that necessitate thorough analysis [11].

Recent advancements in machine learning for anomaly detection have incorporated generative models and neural networks, thereby converting this challenge into a supervised learning task. The Double Adversarial Activation Anomaly Detection (DA3D) technique employs adversarial autoencoders to produce synthetic threats, demonstrating potential in enhancing the detection of unknown security threats [12]. This method indicates advancement; however, its dependence on synthetic data may restrict its effectiveness in practical applications, where data distributions can differ considerably. Autoencoders have demonstrated efficacy in high-performance computing and financial transactions, whereas federated learning methodologies have been utilised to uphold security and privacy [13]. While these methods improve data protection, they may introduce complexities during implementation and might not be appropriate for all IoT environments, particularly those with constrained computational resources.

Multiple studies have effectively integrated investigative methodologies, including the framework established by Novoa-Paradela et al. [14], which provides a distributed online anomaly detection system using machine learning. The effectiveness of hybrid models is contingent upon specific contexts and data characteristics, which raises questions regarding their generalisability. For instance, Nath et al. [15] developed a hybrid model combining Naive Bayes and SVM. However, its performance is subject to variation depending on the dataset utilised, which may restrict its effectiveness in various IoT applications.

The Out-of-Bag anomaly detection strategy exhibits enhanced preprocessing accuracy, especially when applied to multidimensional datasets. Its effectiveness is often contingent upon the context, with performance potentially decreasing in high-dimensional feature spaces. Deep learning techniques, including VGG16 and ResNet, have demonstrated enhancements in anomaly detection via data fusion strategies. However, these approaches can be computationally demanding and often necessitate significant quantities of labelled data for effective training [7].

Investigations into Denial of Service (DoS) attacks in IoT environments have underscored the effectiveness of machine learning models in tackling network-related cybersecurity issues [16]. Research on botnet detection within IoT networks indicates improved F-measures; however, it frequently emphasises particular attack vectors while overlooking a wider spectrum of possible threats [17]. The limited scope may result in deficiencies in the detection capabilities for various categories of cyberattacks.

Feature selection methodologies have been developed to improve the performance of anomaly detection. Abbasi et al. [18] demonstrated significant classification accuracy through the application of Logistic Regression (LR) and Artificial Neural Networks (ANN). Reliance on specific features can introduce biases and may not comprehensively represent the complex nature of anomalies in IoT environments. The integration of federated learning architectures shows potential for aggregating similar communication patterns; however, it is still in the early stages of development and necessitates additional validation [13].

The integration of blockchain technology within machine learning frameworks has been suggested as a means to ensure secure data transmission in IoT applications. However, these solutions frequently encounter scalability issues, especially in extensive deployments, which may impede their practical implementation. Existing surveys offer valuable insights into intrusion detection systems (IDS) in IoT; however, they frequently do not include thorough critical analyses of the methodologies utilised. A recent study presented a taxonomy for Intrusion Detection Systems (IDS) in the IoT, classifying systems according to their deployment strategies and detection techniques [19]. These categorisations do not comprehensively address the nuances of performance trade-offs and the contextual limitations that are inherent in various detection approaches.

This study seeks to address existing gaps by conducting a systematic evaluation and comparison of supervised machine learning, specifically for anomaly detection within IoT networks. We will conduct a thorough evaluation of algorithm performance using established datasets, including ToN-IoT and BoT-IoT. The algorithms under assessment will include Random Forest, k-Nearest Neighbours (kNN), Support Vector Machine (SVM), Naive Bayes, and Decision Tree. This

study will compare the effectiveness of various methods using metrics such as accuracy, precision, recall, and F1-score. Additionally, it will provide a critical analysis of their strengths and weaknesses in relation to IoT security challenges. This examination aims to deliver insights and recommendations for the selection of suitable algorithms tailored to specific IoT applications, thereby contributing to the advancement of more robust and intelligent security solutions.

Table 1 presents a compilation of various studies that concentrate on the detection of anomalies in IoT through the application of diverse machine learning methodologies. Each entry contains a reference, a concise description of the study's main concept, the algorithms employed, and the results achieved. This summary facilitates comparison and highlights advancements in the field of anomaly detection.

Table 1. Summary of anomaly detection studies

Ref.	Main Idea	Algorithm	Outcome	Critical Evaluation
[4]	Introduced federated learning with unsupervised device clustering.	Clustered FL	Grouped devices with similar patterns, enhancing anomaly detection performance.	The approach focuses on improving clustering, but its impact on overall detection accuracy and deployment in real-world IoT systems needs further validation.
[7]	Examined data fusion techniques for improving anomaly detection.	VGG16, Inception, Xception, ResNet	Compared fusion techniques to simpler approaches for IoT data anomaly detection.	Data fusion models can be computationally expensive, making them impractical for resource-constrained IoT devices.
[9]	Enhanced DNN performance by incorporating additional IoT traffic scenarios.	Deep learning technique and SBMO	Validation accuracy reached 0.9989 at epoch 73 during hyperparameter tuning.	DNNs require high computational resources, making them unsuitable for low-resource IoT environments.
[12]	Presented an unsupervised anomaly detection method using adversarial autoencoders.	DA3D	DA3D surpassed current techniques, adapting well to various datasets.	The approach's reliance on unsupervised learning may hinder its ability to detect highly specific or novel attack types.
[13]	Applied federated learning to improve anomaly detection while preserving user privacy.	Federated Learning (FL), Gated Recurrent Units (GRUs), Ensembler Component	FL-based method outperformed traditional ML, enhancing privacy and detection accuracy.	Federated learning is promising but adds significant complexity and communication overhead, limiting real-world deployment in IoT.
[14]	Developed ML algorithms for detecting botnet-driven attacks on IoT networks.	LR, k-NN, and SVM	F-measures reached 98.0%, 99.0%, and 99.0%, demonstrating high accuracy.	The approach doesn't consider real-time detection or scalability for large IoT networks.
[17]	Advocated for ML techniques to detect cybercrime in IoT networks.	BoT-IoT data and ML methods	Successfully identified IoT network attacks, demonstrating ML's value.	The study did not analyse model robustness or its ability to generalize to unseen attack types.
[18]	Employed ANN and LR for feature extraction and classification.	LR and ANN	LR achieved the highest classification accuracy of 99.98%, outperforming deep learning algorithms.	While LR performed well, the generalizability of ANN models in IoT networks needs further exploration.
[20]	Developed a CNN-BLSTM model for network-based anomaly detection, optimizing accuracy through hyperparameter tuning.	CNN and LSTM	Achieved 98.27% accuracy on NSL-KDD and 99.87% on UNSW-NB15, outperforming other approaches.	While the model demonstrated high accuracy, it lacks scalability and computational efficiency for real-time IoT applications.
[21]	Introduced modern ML methods with low false positive rates for IoT anomaly detection.	SVM and RF	SVM achieved 99.9% accuracy, Random Forest 97.9%.	The study did not explore the models' performance on imbalanced datasets, which is critical in IoT environments.

Ref.	Main Idea	Algorithm	Outcome	Critical Evaluation
[22]	A deep learning autoencoder to detect anomalies in HPC systems	Deep machine learning	Their method demonstrates the ability to detect previously unseen anomalies with high accuracy, achieving values between 88% and 96%.	Effectively detects unseen anomalies using autoencoders, but its performance heavily relies on accurate modelling of normal behavior and sufficient, clean training data.
[23]	Compared various ML techniques for anomaly detection in IoT traffic.	SVM, Naive Bayes, LR, Boosting, RF, k-NN	Algorithms detected issues effectively in IoT data, with strong performance across models.	The study didn't address class imbalance, which is prevalent in IoT datasets, affecting model reliability.
[24]	Leveraged Mutual Information and DNN for anomaly detection.	MI and DNN	Improved model accuracy by 0.57-2.6%, reducing the False Acceptance Rate by 0.23-7.98%.	The complexity of DNN models may limit their feasibility in resource-constrained IoT devices.
[25]	Developed a framework combining blockchain with Zero Knowledge Proof for secure data transfer.	DSAE and BiLSTM	Achieved close to 99% accuracy in experiments.	The blockchain-based solution introduces significant overhead, raising concerns about computational efficiency in IoT deployments.

3. Materials and Methods

The methodology utilises a structured approach to facilitate effective data processing and model evaluation, starting with data acquisition to gather pertinent datasets. After collecting the data, the next step involves preprocessing to confirm it is clean and well-organised for analysis. This process comprises handling missing values and normalising the data. Following preprocessing, the dataset is divided into training and test subsets, enabling a thorough assessment of the model's performance. A diverse set of machine learning models is subsequently chosen for training, encompassing Tree, Support Vector Machines (SVM), Random Forest (RF), k-Nearest Neighbours (kNN), and Naive Bayes. After selecting the model, these algorithms undergo training with the training dataset, and their performance is then evaluated using the test dataset. The effectiveness of each model is evaluated using performance metrics including accuracy, precision, recall, and F1-score. The selected models: SVM, k-Nearest Neighbours (kNN), Random Forest (RF), Naive Bayes (NB), and Decision Tree (DT), were chosen for their established efficacy in classification tasks, especially in managing imbalanced datasets and their capacity to separate unique patterns within the data. A thorough comparison of the data is performed to establish the ideal model based on the chosen metrics, guaranteeing that the process is systematic and leads to reliable results, as shown in Figure 1.

3.1 Description

This study utilises two extensive datasets, ToN-IoT and BoT-IoT, to assess the cybersecurity vulnerabilities inherent in the Internet of Things (IoT). A group identified as the Cyber Security Research Centre (CSRC) at the University of New South Wales (UNSW) has produced statistics that offer valuable insights into network traffic associated with IoT devices. The development of these statistics has been completed. The examination of a diverse range of IoT applications is crucial for understanding the interactions among devices and the associated risks.

The ToN-IoT dataset consists of network traffic data collected from a diverse array of one hundred Internet of Things devices. The devices were obtained from diverse real-world environments, including industrial control systems, smart homes, and smart cities. This extensive dataset encompasses 1.2 terabytes of information and includes over 100 million links. The ToN-IoT collection encompasses sensor telemetry data derived from the Internet of Things (IoT) and the Industrial Internet of Things (IIoT). Furthermore, traffic data from the Windows 7 and 10 operating systems, along with Ubuntu versions 14.04 and 18.04, are included as described in [20].

A particular subset of the ToN-IoT dataset, known as the BoT-IoT dataset, was created specifically for the investigation of botnet traffic. The collection, which totals 69 terabytes and over 72 million connections, includes network activity from ten different botnets, including Mirai, Hajime, and Bashlite. Five percent of the original dataset was extracted using MySQL queries for analytical optimization. The segment retrieved for our training and testing datasets consists of four files, each with a size of 0.78 gigabytes and containing approximately 3 million instances. Traffic classifications, including ordinary traffic, denial of service (DoS), distributed denial of service (DDoS), and scanning operations, are precisely detailed in both datasets through careful annotation. Their classification makes them especially appropriate for the training and evaluation of machine learning models within the realm of Internet of Things security [26].

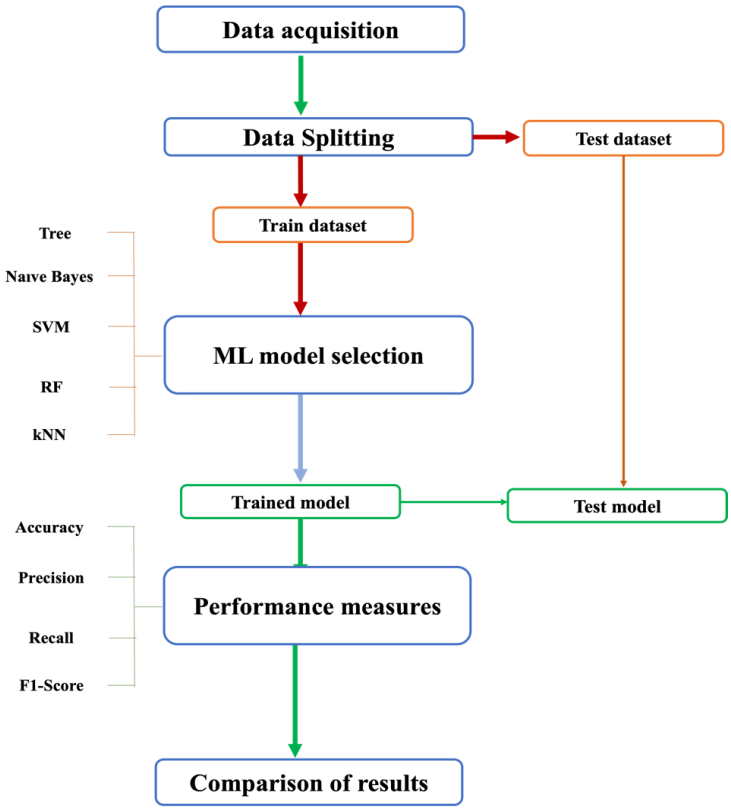


Figure 1. General methodology

3.2 System architecture

The main objective of this system architecture is to evaluate the effectiveness of machine learning algorithms for IoT security using the Orange3 platform. The experimental framework uses two datasets, ToN-IoT and BoT-IoT, while Orange3 provides the machine learning environment for data preprocessing, model training, and performance evaluation.

3.2.1 Datasets

Machine learning techniques were evaluated on the ToN-IoT and BoT-IoT datasets. The chosen methodologies train and evaluate machine learning techniques using various parameters during the preparation of anomaly detection data. Data from ToN-IoT operating systems Windows 7 and 10 were analysed for comparison. Windows 7 encompasses 133 features related to regular operations, attacks, and attack types (DoS, DDoS). The Windows 10.csv file contains 125 properties, categorized as either normal or attack, along with the type of attack (DoS, DDoS). In the BoT-IoT dataset, 5% comprised the "10 best features" version, utilizing 2,934,817 records for training (80%) and 733,705 for testing (20%); [8] provided details regarding the BoT-IoT dataset.

3.2.2 Machine Learning Algorithms

This subsection outlines the machine learning algorithms employed in our comparative analysis for IoT anomaly detection. These algorithms, ranging from probabilistic methods to intricate ensemble and neural network techniques, each provide a unique perspective on data modelling and classification. The following sections provide a concise overview of each method, highlighting their fundamental mathematical formulations:

1) Decision Tree

Decision trees split data into subsets based on feature values to predict target variables. They work by choosing splits that best separate the data, either by maximizing information gain for classification or minimizing mean squared error for regression. The process continues recursively until stopping criteria are met, such as minimum samples per node or maximum depth [27]. The optimal split at node m is found by minimizing the weighted impurity as in Eq. (1):

$$\theta^* = \arg\min_{\theta} \left[\frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \right] \quad (1)$$

Here, H represents the impurity measure (e.g., Gini index, entropy, or mean squared error), and $n_m$is the number of samples at the node m.

2) Multilayer Perceptron (Neural Network)
A multilayer perceptron (MLP) is a type of neural network that models complex relationships by passing inputs through layers of interconnected nodes. Each node applies a weighted sum followed by a nonlinear activation function. The model learns by adjusting weights to minimize prediction error, often using optimizers like Adam [28]. Hidden layer activations as in Eqs. (2-3):

$$h = ReLU(W_1 x + b_1) \tag{2}$$

Output layer:

$$y = g(W_2 h + b_2) \tag{3}$$

where ReLU is the activation function, $ g $ is the output activation, and $W_i, b_i$are weights and biases.

3) Support Vector Machine (SVM)
SVMs classify data by finding the hyperplane that best separates classes with the largest margin. For non-linear data, kernels like the Radial Basis Function (RBF) map inputs into higher-dimensional spaces where separation is easier [29]. The RBF kernel function is in Eq. (4):

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \tag{4}$$

where (γ)controls the kernel width.

4) k-Nearest Neighbours (k-NN)
The k-NN algorithm classifies a data point based on the majority label among its k closest neighbours, measured by a distance metric such as Euclidean distance [30]. Euclidean distance between points p and q in Eq. (5)

$$D(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5}$$

5) Naive Bayes
Naive Bayes is a probabilistic classifier that applies Bayes' theorem assuming feature independence. It calculates the probability that a sample belongs to a class based on feature likelihoods [31] in Eq. (6).

$$P(Y = y_k \mid X_1, \ldots, X_n) = \frac{P(Y=y_k) \prod_{i=1}^{n} P(X_i|Y=y_k)}{\sum_j P(Y=y_j) \prod_{i=1}^{n} P(X_i|Y=y_j)} \tag{6}$$

6) Random Forest
Random Forest combines multiple decision trees trained on different random subsets of data and features. Predictions are aggregated by averaging (regression) or majority vote (classification), improving accuracy and reducing overfitting [32] in Eq. (7).

$$\hat{y} = \frac{1}{M} \sum_{j=1}^{M} T_j(x) \tag{7}$$

where $T_j$is the prediction from the j -th tree.

7) Gradient Boosting (XGBoost)
XGBoost builds an ensemble of trees sequentially, where each new tree corrects errors made by the previous ensemble. It optimizes a regularized objective function to balance model fit and complexity [33], as in Eq. (8).

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{8}$$

where l is the loss function and Ω is the regularization term.

3.2.3 Implemented approach

Orange is a compiler for object-oriented programming modules, utilizing the C++ object library. Free, open-source software is accessible to all users. The Orange widget offers a graphical user interface for data mining and machine learning. Orange obviates the necessity for proficiency in programming languages. Implementing the strategy in machine learning algorithms is straightforward. This study utilized a suite of thirteen distinct widgets to facilitate various stages of data analysis for evaluating machine learning models:

1) File Widget: Streamlined data integration by enabling the loading of the ToN-IoT and BoT-IoT datasets. This widget processes input data and provides access to recent files for efficient workflow initiation.
2) Distributions Widget: Offered visual insights into data characteristics by displaying value distributions for both discrete and continuous features. Users could condition these distributions based on a selected class variable, allowing for targeted exploratory analysis.
3) Select Columns Widget: Provided precise control over the data domain by enabling manual selection of a feature subset. This widget allowed attributes to be designated as regular features, an optional class variable, or metadata, offering flexibility in data preparation.
4) Sampler Widget: Addressed class imbalance within the datasets by offering various resampling techniques, including over-sampling, under-sampling, and balanced sampling. This ensured that all classes were adequately represented,

thereby enhancing the learning process for the models. In this research, a fixed sampling proportion of 70% was applied.

5) Tree Widget: Implemented a decision tree algorithm to partition data based on information gain for categorical targets or mean squared error for numerical targets. Key hyperparameters for this widget included a minimum of one instance in leaf nodes, five instances in internal nodes, a maximum tree depth of 100, a 95% majority stopping criterion, and the enforcement of binary splits for effective decision-making.
6) Neural Network Widget: Deployed a multilayer perceptron for advanced pattern recognition. Hyperparameters were meticulously configured, featuring one hidden layer with 100 units, the ReLU activation function, and the Adam solver (optimizer). The learning rate was set to 0.0001, with a maximum of 200 iterations for model training.
7) Support Vector Machine (SVM) Widget: Optimized classification tasks using a support vector machine. This widget was configured with a Radial Basis Function (RBF) kernel, and its critical hyperparameters included a regularization parameter C set to 1.0, an epsilon ε of 0.1 for the loss function, a numerical tolerance of 0.001, and a maximum of 100 iterations.
8) k-Nearest Neighbors (k-NN) Widget: Executed the k-Nearest Neighbors algorithm for straightforward proximity-based classification. The key hyperparameters for this widget were five neighbors (k=5), the Euclidean distance metric, and uniform weighting for neighbor contributions.
9) Naive Bayes Widget: Constructed a probabilistic classifier based on Bayes' theorem. This widget provided a rapid and efficient method for classifying categorical data, relying on the assumption of feature independence.
10) Random Forest Widget: Leveraged an ensemble of decision trees to improve predictive accuracy and robustness against overfitting. This was achieved by aggregating the outputs of multiple individual trees, each trained on different data subsets. Key hyperparameters typically include the number of trees in the forest and individual tree depth.
11) Gradient Boosting (XGBoost) Widget: Implemented the XGBoost algorithm, an advanced and efficient gradient boosting framework renowned for its high performance. Critical hyperparameters for this widget were explicitly set: the learning rate was 0.300, the number of trees (estimators) was 100, and the maximum depth for each tree was limited to 6.
12) Test and Score Widget: Assessed the performance of the trained classifiers using stratified shuffle splitting. For the BoT-IoT dataset, evaluation involved three random samples (each using 70% of the data), while the ToN-IoT dataset utilized twenty random samples (also 70% of the data). Results were averaged across classes to ensure a consistent and reliable evaluation.
13) Confusion Matrix Widget: Provided a quantitative and visual assessment of classifier performance by comparing predicted class distributions against actual class labels. This enabled a detailed analysis of misclassifications, offering insights into the strengths and weaknesses of each model.

3.2.4 Performance evaluation metrics

Following the implementation of machine learning algorithms, it is essential to evaluate their effectiveness with designated techniques referred to as performance evaluation measures. A diverse array of metrics has been developed in research, each targeting distinct facets of algorithm performance. Consequently, choosing the correct metrics is crucial for the proper assessment of any machine learning work. This study utilizes various common measures for classification tasks to evaluate and compare algorithm performance.

The measurements encompass accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC score, as in Eqs. (9-11). *Precision* assesses the pertinence of chosen data elements. It specifically denotes the proportion of observations projected as positive by the algorithm that are genuinely positive. The calculation is performed using the following formula:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{9}$$

where TP denotes true positives, while FP signifies false positives.

Recall evaluates the algorithm's efficacy in identifying pertinent data items. It indicates the number of real positive observations accurately predicted by the algorithm. The equation for recall is:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{10}$$

$F1$ is referred to as the F-score, which integrates precision and recall to assess algorithm efficacy. It is calculated as the harmonic mean of precision and recall, represented by the formula in Eq. (11) as follows:

$$F1 = 2 * \frac{Prec * Recall}{Prec + Recall} \tag{11}$$

Accuracy is a frequently employed parameter for assessing algorithm performance in classification tasks. It is defined as the proportion of accurately identified data points to the total number of observations, as in Eq. (12):

$$\text{Accuracy} = \frac{TP + T\,N}{TP + TN + FP + FN} \tag{12}$$

Despite its widespread application, accuracy may not be the most suitable statistic when the target variable classes are uneven. Confusion Matrix: It is a clear and insightful instrument for assessing the precision and validity of a machine

learning system. It is especially beneficial in categorization tasks involving two or more categories. The ROC curve, which shows the connection between the true positive rate (sensitivity or recall) and the false positive rate (1 - specificity), is the source of the ROC-AUC score. For binary classification, the area under the ROC curve (ROC-AUC) is crucial because it shows how well a model can differentiate between positive and negative classifications. When both classes are equally important, this statistic is especially significant.

4. Results and discussion

This research utilized two distinct datasets and applied seven different machine learning algorithms to evaluate their performance in detecting and mitigating cyber anomalies within Internet of Things (IoT) networks. The primary goal was to identify the most effective and efficient strategies for anomaly detection. To ensure a reliable analysis, a data sampler was used to create a representative and balanced dataset, which is crucial for training robust machine learning models. The results indicated that the algorithms varied significantly in their effectiveness at detecting cyber anomalies, revealing both their strengths and weaknesses. This analysis not only highlights the capabilities of each algorithm but also provides valuable insights into their potential applications in real-world scenarios. By addressing both detection and mitigation of cyber anomalies, this research contributes to the development of more effective strategies for enhancing security in IoT systems. The evaluation results are listed in Table 2.

Table 2. Evaluation results of the ToN-IoT dataset demonstrating the average across classes

Model	AUC	CA	$F1$	Precision	Recall	MCC
kNN	0.998	0.991	0.991	0.991	0.991	0.987
Naive Bayes	1.000	0.888	0.940	0.999	0.888	0.856
Random Forest	1.000	1.000	1.000	1.000	1.000	1.000
SVM	1.000	0.998	0.997	0.997	0.998	0.997
Tree	1.000	0.999	0.998	0.998	0.999	0.998
Neural Network	1.000	0.999	0.999	0.999	0.999	0.999
Gradient Boosting	1.000	1.000	1.000	1.000	1.000	1.000

This analysis evaluates seven machine learning models on the ToN-IoT (Table 2) and BoT-IoT (Table 3) network intrusion datasets. Performance was assessed using metrics like Area Under the Curve (AUC), Classification Accuracy (CA), F1-score, Precision, Recall, and Matthews Correlation Coefficient (MCC), averaged across classes. On the ToN-IoT dataset (Table 2), most models effectively differentiated between network activity types. Six models (Naive Bayes, Random Forest, SVM, Tree, Neural Network, and Gradient Boosting) recorded perfect AUCs (1.000), with kNN close behind at 0.998. Random Forest and Gradient Boosting achieved perfect scores across all metrics, demonstrating exceptional threat detection. Neural Network (CA 0.999, F1 0.999), Tree (CA 0.999, F1 0.998), SVM (CA 0.998, F1 0.997), and kNN (CA 0.991, F1 0.991) also showed high effectiveness. Conversely, Naive Bayes, despite a perfect AUC, had lower CA (0.888) and F1 (0.940) scores, suggesting challenges with some classifications. The dataset's significant class imbalance (predominantly 'normal' traffic) likely influenced these varied results. The Bot-IoT dataset (Table 3), kNN, Random Forest, Neural Network, and Gradient Boosting achieved perfect scores (1.000) across all metrics, indicating outstanding attack detection. SVM also performed strongly (AUC 1.000, CA 0.994, F1 0.994). Naive Bayes showed reasonable but more modest results (AUC 0.995, CA 0.948, F1 0.950). Notably, the Tree model evaluation encountered an error on this dataset, providing no metrics for comparison.

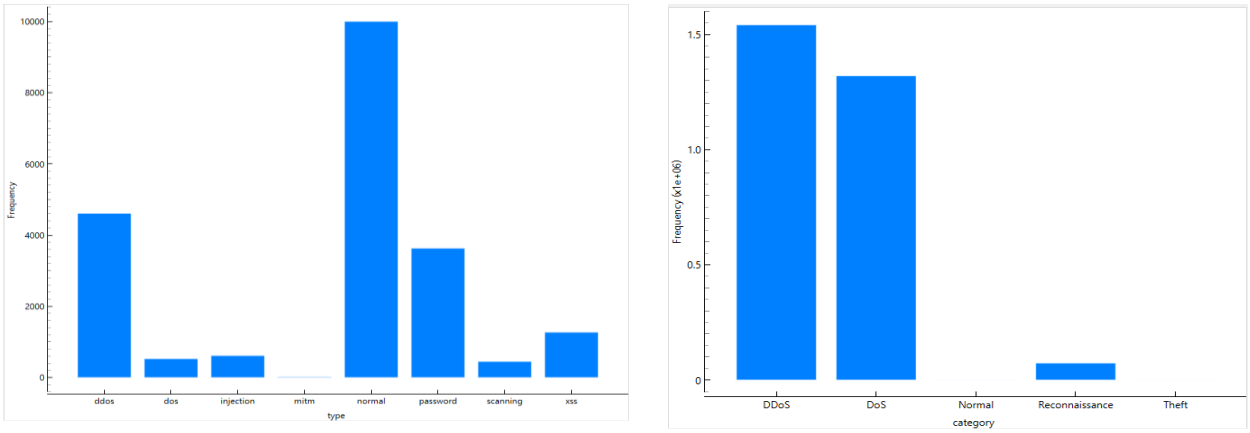
Table 3. Evaluation results of the Bot-IoT dataset demonstrating the average across classes

Model	AUC	CA	$F1$	Precision	Recall	MCC
kNN	1.000	1.000	1.000	1.000	1.000	1.000
Naive Bayes	0.995	0.948	0.950	0.952	0.948	0.902
Random Forest	1.000	1.000	1.000	1.000	1.000	1.000
SVM	1.000	0.994	0.994	0.994	0.994	0.989
Tree	error	error	error	error	error	error
Neural Network	1.000	1.000	1.000	1.000	1.000	1.000
Gradient Boosting	1.000	1.000	1.000	1.000	1.000	1.000

4.1 Discussion

The evaluation of machine learning models on the ToN-IoT and BoT-IoT datasets offers significant insights into their respective capabilities and practical limitations for anomaly detection in IoT environments. A primary challenge evident

across both datasets is class imbalance, where categories such as “normal” and “DDoS” are heavily overrepresented compared to rarer attack types like MITM, Theft, and Reconnaissance. This disparity inherently biases models towards these dominant classes, consequently diminishing their effectiveness in accurately identifying less frequent yet critical anomalies. Notably, even high-performing models such as Random Forest demonstrated occasional misclassification of these minority categories, highlighting the persistent difficulty in achieving robust performance across all classes. The distribution of data types within both the ToN-IoT and BoT-IoT datasets, as depicted in Figure 2, reveals significant class imbalances that are crucial for understanding model performance.



(a) (b)

Figure 2. ToN-IoT dataset type’s distribution and BoT-IoT dataset type’s distribution

In a comparative overview, Random Forest distinguished itself as the most reliable model, attaining perfect scores across all evaluation metrics, which suggests strong generalization and robustness. While Support Vector Machine (SVM) and k-Nearest Neighbors (kNN) also exhibited high accuracy, they displayed reduced sensitivity when detecting infrequent attack types. Conversely, Naive Bayes consistently underperformed, underscoring its limitations in managing complex, multi-class data. These outcomes emphasize the critical need to select models adept at handling both class complexity and dataset imbalance. A specific issue was encountered with the Decision Tree model, which failed to produce valid results on the BoT-IoT dataset. This failure, potentially due to overfitting or an inability to manage the dataset's scale and variability, underscores the necessity for comprehensive model validation and stress-testing across diverse datasets prior to deployment. The analysis further identified limitations in feature representation. As indicated by Figure 2, models frequently confused similar attack types, such as DoS with DDoS. This implies that the selected features may not adequately capture the subtle distinctions essential for precise classification of varied attack behaviours. Beyond mere accuracy, practical IoT deployment demands a careful balance between model performance and resource consumption. While models like Random Forest and kNN achieve high accuracy, their computational demands can be substantial. In contrast, more lightweight models such as Naive Bayes and Decision Tree, despite their comparatively lower accuracy, might offer greater suitability for resource-constrained IoT devices. Consequently, future evaluations should incorporate metrics like runtime, memory usage, and inference latency to thoroughly assess real-world applicability.

Further contextualizing the pursuit of high-performance Intrusion Detection Systems (IDS), recent work by [34] also addressed anomaly detection in IoT networks using the ToN_IoT dataset. Their methodology involved a comparative analysis of eight base classifiers and two ensemble classifiers, from which a stacking ensemble model integrating CatBoost, Extra Trees, and XGBoost was identified as the most effective. The authors reported that their proposed stacking model achieved high efficacy, recording Matthews Correlation Coefficient (MCC) scores of 0.9971 for binary classification and 0.9909 for multiclass classification, and concluded its superiority over other models tested within their study on the same dataset. The success of such sophisticated ensemble techniques, as demonstrated by Guo et al., highlights the potential for advanced methodologies to achieve strong results and provides a valuable contemporary benchmark on the ToN_IoT dataset. To address the identified limitations within our own study and to build upon the current state of the art, future research will prioritize evaluating the computational efficiency of models and exploring advanced methodologies like ensemble learning and eXplainable Artificial Intelligence (XAI). Furthermore, data augmentation techniques, such as SMOTE, will be investigated as a means to mitigate class imbalance and enhance the detection performance for underrepresented attack types.

5. Conclusions

This research evaluated seven machine learning models, Random Forest, Gradient Boosting, Neural Networks, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Decision Tree, and Naive Bayes for detecting anomalies in IoT networks using the ToN-IoT and BoT-IoT datasets. The findings indicate that several models can achieve high efficacy. Notably, Random Forest and Gradient Boosting frequently delivered perfect or near-perfect scores across various metrics on both datasets. Neural Networks also demonstrated consistently strong performance. SVM and kNN showed robust

results, with kNN achieving perfect scores on the BoT-IoT dataset. While the Decision Tree performed well on the ToN-IoT dataset, it encountered operational failures on the BoT-IoT dataset. Naive Bayes generally exhibited lower comparative performance, particularly struggling with class imbalance despite achieving high AUC scores on one dataset. A key challenge identified was class imbalance within the datasets, where dominant classes like "Normal" or "DDoS" often skewed model learning, impacting the detection of rarer yet critical attack types. Limitations in feature representation also emerged, with some models confusing similar attack categories (e.g., DoS and DDoS). These observations underscore that while models like Random Forest, Gradient Boosting, and Neural Networks show high suitability for anomaly detection, practical deployment must also consider model behaviour with imbalanced data and the distinctness of features. The failure of the Decision Tree on one dataset further highlights the need for rigorous testing. Future research will focus on enhancing detection accuracy and practical applicability. This includes exploring advanced methodologies such as deep learning, ensemble techniques, innovative feature engineering, and unsupervised/semi-supervised learning for scenarios with limited labelled data. Strategies to mitigate class imbalance, like data augmentation (e.g., SMOTE), will be investigated. Emphasis will also be placed on evaluating computational efficiency, scalability, implementation costs, and user privacy for real-world IoT environments. Integrating explainable artificial intelligence (XAI) aims to improve trust and understanding, particularly for complex models.

Acknowledgements
The authors want to express their sincere gratitude to the editors and reviewers for their dedicated time and work on our manuscript.

Funding
This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sectors.

Author contribution
Hiba Ahmed: Conceptualisation; Methodology; Validation; Formal analysis
Razan Alharith: Methodology; Data curation; Visualisation; Writing - original draft
Ashraf Osman Ibrahim: Funding acquisition; Supervision; Resources
Nada Adam: Writing - review & editing

Conflict of interest
The authors have no conflicts of interest to declare that are relevant to the content of this article.

References
[1] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, Art. no. 100568, 2022, https://doi.org/10.1016/j.iot.2022.100568.
[2] J. Demšar *et al*., "Orange: Data mining toolbox in Python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013, https://doi.org/10.5555/2567709.2567736.
[3] M. Gao, L. Wu, Q. Li, and W. Chen, "Anomaly traffic detection in IoT security using graph neural networks," *Journal of Information Security and Applications*, vol. 76, Art. no. 103532, 2023, https://doi.org/10.1016/j.jisa.2023.103532.
[4] X. Sáez-de-Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, "Clustered federated learning architecture for network anomaly detection in large-scale heterogeneous IoT networks," *Computers & Security*, vol. 131, Art. no. 103299, 2023, https://doi.org/10.1016/j.cose.2023.103299.
[5] D. Aggarwal, A. B. Saxena, and D. Sharma, "Mitigating cybersecurity risks in IoT: A layered approach to threat detection and prevention," in *Proc. 4th Int. Conf. Sentiment Analysis and Deep Learning (ICSADL)*, 2025, pp. 501–505.
[6] A. Abusitta, G. H. de Carvalho, O. A. Wahab, T. Halabi, B. C. Fung, and S. Al Mamoori, "Deep learning-enabled anomaly detection for IoT systems," *Internet of Things*, vol. 21, Art. no. 100656, 2023, https://doi.org/10.1016/j.iot.2022.100656.
[7] A. Devagopal, V. Menon, S. Ezekiel, and P. Chaudhary, "Exploring the tractability of data fusion models for detecting anomalies in IoT-based dataset," in *Big Data V: Learning, Analytics, and Applications*, vol. 12522, 2023, pp. 82–89.
[8] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, https://doi.org/10.1016/j.future.2019.05.041.
[9] L. Aversano, M. L. Bernardi, M. Cimitile, R. Pecori, and L. Veltri, "Effective anomaly detection using deep learning in IoT systems," *Wireless Communications and Mobile Computing*, vol. 2021, Art. no. 9054336, 2021, https://doi.org/10.1155/2021/9054336.
[10] N. Butt, A. Shahid, K. N. Qureshi, S. Haider, A. O. Ibrahim, F. Binzagr, and N. Arshad, "Intelligent deep learning for anomaly-based intrusion detection in IoT smart home networks," *Mathematics*, vol. 10, no. 23, Art. no. 4598, 2022, https://doi.org/10.3390/math10234598.

[11] S. Zehra, U. Faseeha, H. J. Syed, F. Samad, A. O. Ibrahim, A. W. Abulfaraj, and W. Nagmeldin, "Machine learning-based anomaly detection in NFV: A comprehensive survey," *Sensors*, vol. 23, no. 11, Art. no. 5340, 2023, https://doi.org/10.3390/s23115340.
[12] J.-P. Schulze, P. Sperl, and K. Böttinger, "Double-adversarial activation anomaly detection: Adversarial autoencoders are anomaly generators," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2022, pp. 1–8.
[13] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022.
[14] D. Novoa-Paradela, Ó. Fontenla-Romero, and B. Guijarro-Berdiñas, "Adaptive real-time method for anomaly detection using machine learning," *Proceedings*, vol. 54, no. 1, Art. no. 38, 2020.
[15] M. D. Nath and T. Bhattasali, "Anomaly detection using machine learning approaches," *Azerbaijan Journal of High Performance Computing*, vol. 3, no. 2, pp. 196–206, 2020.
[16] C. Malathi and I. N. Padmaja, "Identification of cyber attacks using machine learning in smart IoT networks," *Materials Today: Proceedings*, vol. 80, pp. 2518–2523, 2023, https://doi.org/10.1016/j.matpr.2021.06.400.
[17] A. D. Khaleefah and H. M. Al-Mashhadi, "Detection of IoT botnet cyber attacks using machine learning," *Informatica*, vol. 47, no. 6, pp. 55–63, 2023, https://doi.org/10.31449/inf.v47i6.4668.
[18] F. Abbasi, M. Naderan, and S. E. Alavi, "Anomaly detection in Internet of Things using feature selection and classification based on logistic regression and artificial neural network on N-BaIoT dataset," in *Proc. 5th Int. Conf. Internet of Things and Applications (IoT)*, 2021, pp. 1–7.
[19] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.
[20] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustainable Cities and Society*, vol. 72, Art. no. 102994, 2021, https://doi.org/10.1016/j.scs.2021.102994.
[21] A. Al Obaidli, D. Mansour, S. M. Abdulhamid, N. Ben Halima, and A. Al-Ghushami, "Machine learning approach to anomaly detection attacks classification in IoT devices," in *Proc. 1st Int. Conf. Advanced Innovations in Smart Cities (ICAISC)*, 2023, pp. 1–6, https://doi.org/10.1109/ICAISC56366.2023.10085349.
[22] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 9428–9433, 2019, https://doi.org/10.1609/aaai.v33i01.33019428.
[23] E. Istratova, M. Grif, and D. Dostovalov, "Application of traditional machine learning models to detect abnormal traffic in the Internet of Things networks," in *Proc. Int. Conf. Computational Collective Intelligence*, 2021, pp. 735–744.
[24] Z. Ahmad, A. Shahid Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, and J. J. P. C. Rodrigues, "Anomaly detection using deep neural network for IoT architecture," *Applied Sciences*, vol. 11, no. 15, Art. no. 7050, 2021, https://doi.org/10.3390/app11157050.
[25] P. Kumar, R. Kumar, G. P. Gupta, R. Tripathi, A. Jolfaei, and A. K. M. N. Islam, "A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system," *Journal of Parallel and Distributed Computing*, vol. 172, pp. 69–83, 2023, https://doi.org/10.1016/j.jpdc.2022.10.002.
[26] A. R. Gad, M. Haggag, A. A. Nashat, and T. M. Barakat, "A distributed intrusion detection system using machine learning for IoT based on ToN-IoT dataset," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 548–563, 2022, https://doi.org/10.14569/IJACSA.2022.0130667.
[27] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, https://doi.org/10.1007/BF00116251.
[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, https://doi.org/10.1038/323533a0.
[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995, https://doi.org/10.1007/BF00994018.
[30] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967, https://doi.org/10.1109/TIT.1967.1053964.
[31] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. AAAI-98 Workshop Learning for Text Categorization*, 1998, pp. 41–48.
[32] L.Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001, https://doi.org/10.1023/A:1010933404324.
[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.
[34] G. Guo, X. Pan, H. Liu, F. Li, L. Pei, and K. Hu, "An IoT intrusion detection system based on TON IoT network dataset," in *Proc. IEEE 13th Annu. Computing and Communication Workshop and Conf. (CCWC)*, 2023, pp. 333–338, https://doi.org/10.1109/CCWC57344.2023.1009914.