Journal of Education and Science (ISSN 1812-125X)

Enhancing IoT Security: A Machine Learning-Based Intrusion Detection System for Real-Time Threat Detection and Mitigation

A. A. Ahmed

Department of Computer Science, College of Education for Pure Sciences, University of Mosul, Mosul, Iraq

Article information

Article history:
Received: July 02, 2025
Revised: August 31, 2025
Accepted: September 01, 2025
Available online: October 01.2025

Keywords:
IoT
Intrusion Detection System
Machine Learning
Random Forest
Naive Bayes

Correspondence:
Ammar Adel Ahmed
Ammaraladel@uomosul.edu.iq

Abstract

Rapid growth in usage of Internet of Things (IoT) devices has created a situation where security is highly vulnerable, and people require more sophisticated and evolving solutions. Conventional security solutions cannot overcome the issue of heterogeneity, resource scarcity, and dynamism of IoT environments. This paper suggests the use of a machine learning-based Intrusion Detection System (IDS) to identify and attempt to reduce the presence of real-time threats within IoT networks. The results of different machine learning models which include the Logistic Regression, the Decision Tree, the Random Forest, the XGBoost, the AdaBoost, the Gradient Boosting, Bagging, K-Nearest Neighbors (KNN), and the Naive Bayes are compared based on some of the key performance indicators that are accuracy, precision, recall, F1-score, ROC-AUC, and log loss. Our findings indicate that ensemble algorithms, especially Random Forest, Decision Tree, and Bagging, can be more effective than other models in identifying a large number of detections with low false positives, and Random Forest offers an accuracy of 99.99%, precision of 99.96%, a recall rate of 99.96% and ROC-AUC of 99.99%. By contrast, the results of Naive Bayes were much worse, showing an accuracy rate of 74.28 %, a precision rate of 23.32% and an F1-score of 37.71. These findings underline that ensemble algorithms, in particular Random Forest, are also very successful in real-time intrusion detection on IoT systems. The given approach proves that ensemble learning, which possesses the capability to merge several classifiers, is an effective solution to enhancing the IoT safety of systems.

DOI:10.33899/jes.v34i4.49257,

1. Introduction

As the Internet of Things (IoT) keeps growing, the estimate on connected devices would see tens of billions within the next few years. This high rate of growth has been accompanied by an upsurge of cyber-attacks against IoTs as well as connections to them. These risks include a variety of attacks, starting with rather easy ones, such as denial-of-service (DoS), and more advanced ones, including botnets, ransomware, and massive data breaches. Poor device authentication, encryption, and device management are some of the major security flaws that affect most IoT systems and present various exploitation avenues to cybercriminals. The vulnerabilities are especially disturbing, since IoT solutions are becoming more and more common in critical infrastructure, including medical systems, industrial control systems, and so on, as any security breaches may lead to serious data loss, financial damage, or physical injury [1].

IoT devices face their own unique set of security threats that are quite different from those found in traditional computing. Perhaps one of the biggest challenges is the limited resources of IoT devices. Unlike traditional systems, IoT devices are often configured with the bare essentials of processing power, memory, and storage capacity. These limitations make the application of conventional security measures such as complex encryption techniques, intrusion detection, and robust

authentication challenging. Therefore, IoT devices are vulnerable to various types of attacks such as data interception, unauthorised access, and weak security defences exploitation [2].

Another critical issue is the heterogeneity and interoperability of IoT devices. IoT devices have a heterogeneous population of devices with different hardware, software, and communication protocols. This heterogeneity is the one that brings a fragmented security environment wherein security solutions are tailored based on the specific capabilities and requirements of each device. Moreover, devices from different manufacturers may not be compatible at all times, and this can create vulnerabilities when devices need to communicate with one another. In contrast, traditional computing environments are usually more homogenous, with it being easier to deploy uniform security measures [3].

The scalability and dynamic nature of IoT networks contribute to the security issues. IoT systems are prone to include massive arrays of devices that communicate with one another and are constantly added, removed, or relocated. This dynamic topology increases the network's attack surface, providing more opportunities for cybercriminals to attack vulnerabilities. In contrast to conventional systems, where devices are usually stationary and well-maintained, IoT devices are usually installed in remote or inaccessible locations. This means that possible intrusions may not be detected at all for a long time, and it is more challenging to implement real-time monitoring or security protocol updates [4].

Physical and environmental vulnerabilities also make IoT devices vulnerable to attacks. Some of the IoT devices are deployed in public or less-protected environments, such as homes, factories, or infrastructure locations, where they can be compromised or stolen physically. Physical access to the devices can provide attackers with mechanisms to bypass security controls, gain unauthorised access to a network, or even modify device settings. Additionally, IoT devices in certain settings, like industrial settings, are exposed to environmental conditions like electromagnetic interference, which can compromise their security mechanisms or even present additional vulnerabilities [5].

The lack of standardised security standards and regulations also adds to the security issues of IoT ecosystems. While traditional computing systems have well-developed security standards and regulations, IoT devices lack standardised security frameworks. Functionality and price may be more appealing to producers than security, and this results in devices with deficient or inadequate security features. Non-universal security standards among IoT ecosystems result in devices from different vendors likely having different protection levels, making it harder to enforce total security solutions and making them vulnerable to exploitation further [6].

The majority of IoT devices suffer from ineffective device authentication and data encryption. Most devices are regularly devoid of proper security protocols necessary for authenticating users or encrypting confidential data, which makes them vulnerable to unapproved attacks and data leaks. User authentication and encryption are widely used in computing systems, but in IoT systems, they are usually weak or non-existent, making devices vulnerable to man-in-the-middle attacks as well as other types of cyber-attacks [7].

These unique challenges to IoT devices create a security environment much more complex and exposed than traditional computing systems. The resource limitations, heterogeneous device base, no standardisation, and physical exposures require unique security solutions adapted to the specific requirements of IoT networks. Traditional security solutions are inadequate to address these challenges, making IoT networks highly susceptible to a wide range of cyberattacks [8].

The traditional cybersecurity tools do not get the job done in the IoT environment most of the time because of the peculiarities of these environments. The high level of heterogeneity is the common characteristic of IoT ecosystems, with a very diverse number of devices, communication protocols, and software platforms. Moreover, mobile IoT networks, which constantly add or remove devices, have to support flexible and elastic security solutions. To make matters worse, the majority of IoT devices are resource-limited and cannot have either traditional encryption or authentication measures to prevent hacking [9].

In order to eliminate such constraints, machine learning (ML) and artificial intelligence (AI) of intrusion detection systems (IDS) have become of great concern. ML algorithms, especially the ones intended to detect anomalies, present a potential solution to the problem of real-time tracking and detection of abnormal patterns of behaviour. In contrast to classical, signature-based IDS solutions, the ML-based systems can identify a new or previously unknown threat, as they learn how to recognise deviations in network traffic patterns [10].

The study also showed significant progress in using ML to improve IoT security in recent studies. Using labelled datasets in the network, supervised learning methods including decision trees and support vector machines (SVMs) have been applied, as well as deep learning methods including deep neural networks (DNNs). Conversely, unsupervised learning methods, including clustering and anomaly detection, perform Ineffectively in revealing the hidden threats in unlabeled data. The reinforcement learning (RL) also plays a role by dynamically updating or modifying the defence strategy by interacting continuously with the environment [11].

But there are many issues to tackle so that ML-based security systems can prove successful in IoT networks. Preprocessing and data collection continue to be one of the major obstacles because IoT networks produce large and heterogeneous data. Representativeness and well-balancing of training datasets are the key to ensuring the correct and bias-free performance of models. Also, the issue of scale of ML-based IDS solutions is an urgent issue. As the number of IoT devices and

data increases, such systems need to be both extremely high-performance and lightweight so that they can operate on the limited-resource devices. Edge computing comes as a possible solution because it allows processing the data locally, which alleviates the central servers [12].

Low-latency threat detection and mitigation is another crucial provision, particularly on IoT applications in real-time when instantaneous reaction is critical. The advancement of special databases with labelled traffic of real IoT setups helped move the sphere forward. Other research interests include the hybrid and ensemble mode of learning to achieve improved accuracy and robustness of detection, which indicates the changing face of ML in ensuring the future of IoT systems [13].

This paper suggests a machine learning-based Intrusion Detection System (IDS) that would apply to an IoT environment. It also compares various machine learning algorithms, Random Forest, Decision Tree, Bagging and shows that ensemble techniques are better than others in identifying intrusion with a high precision and low false positive rates. This paper demonstrates how machine learning could be utilised in IoT networks to improve the threat detection capabilities in real-time by considering some IoT-specific security challenges, such as resource constraints and network heterogeneity. The results offer practical guidance on how to design better and scalable security approaches to the IoT ecosystem.

2. Related Work

Within the last few years, the research of machine learning (ML) and deep learning (DL) technologies to improve the intrusion detection system (IDS) as a solution to protecting the IoT and network has experienced high levels of development. The studies cover most of the existing approaches to such types as ensemble learning, deep neural networks, feature selection approaches, as well as hybrid models, and test them on diverse data sets such as NSL-KDD, CIC-IDS2017, UNSW-NB15, and IoTID20. The increasing range of studies proves that an ensemble approach to combining classifiers and an effective feature selection is two approaches that can significantly increase the accuracy of detection, cutting the number of false positives and allowing for the detection of both known and new cyber threats in time. The list of relevant contributions in this area is summarised, which includes a description of the approaches used by researchers, the data they used, and the results with which they quantitatively evaluated their results. It can give a general idea of the current tendencies and directions in the research of IoT and network intrusion detection.

Researchers [14] presented the CFS-BA-Ensemble framework, which is a heuristic approach for dimensionality reduction. They employed a type of learning known as ensemble learning, where they collect algorithms such as C4.5, Random Forest (RF), and Forest to solve issues that are common with IDS systems, notably the problem of handling voluminous irrelevant data. This framework proved to be efficient in comparison to previous research in terms of classification accuracy, F1-measure, and attack detection rate. Three datasets were used to evaluate this study: NSL-KDD, CIC-IDS2017, and AWID. they achieved an accuracy level of 99.81%, 99.52%, and 99.89%, respectively.

Also, [15] proposes an ensemble learning model to address the problem of intrusion detection using the NSL-KDD dataset. The work employed integrates several ML approaches, which comprise DT, RF, KNN, DNN, and adaptive ensemble algorithms. The suggested algorithm has shown its effectiveness in evaluating new attack types within a short time. As for the quantitative aspect, the found accuracy for the DNN equals 81. 6%, the highest accuracy achieved by the MultiTree algorithm was 84.2%, and the voting algorithm achieved 85.2%. It is also important to mention that the results of the application of the ensemble voting technique are higher than those of the MultiTree. The study also outlines several areas for future work enhancements. Based on the study, feature selection needs to be improved, while the detection needs to be made more robust with small-scale data sets.

Authors [16] put forward a BAT-MC approach that combines bidirectional long-term memory (BLSTM) and an attention mechanism to develop a deep learning (DL) model for network intrusion detection. The proposed mechanism extracts fine-grained features from the sequential data composed of packet vectors, while the BLSTM model learns the characteristics of the NSL-KDD data set for analysing each packet in the network traffic. However, before, the data was preprocessed through several layers of convolution. Based on the given data set, the above model had a better performance than conventional approaches with an accuracy of 84. 25%.

Researchers [17] put forward a voting strategy that is based on an ensemble of basic classifiers, such as decision trees, Bayes classifiers, RNN-LSTM, and random forests, that acts with a misbehaviour analysis scheme. When this model was compared with the multiple tree algorithm, it was seen that it performed well, and the accuracy rate of the model was up to 85%, using NSL_KDD as a dataset. The studies showed results that demonstrated that the given method allowed for achieving better detection results. There are future directions that need to be forged to include creating an archive of rules that will enable a computerised detection of these intrusions. Through implementing this measure, it will be possible to improve the provided system's performance and speed regarding real-time threat identification.

Meta-classification model is used by [18] to combine these meta-classifiers, which is made up of a stacking architecture comprising raw and meta classifiers; the LR, the K-NN, the RF, and the SVM. To analyse the performance of the proposed model, two different datasets are used: one is a packet-based dataset called UNSW NB-15 collected from a simulated framework, and the second one is called UGR'16 16 is a flow-based dataset captured from the real environment of network traffic. The

experimental findings support the fact that the proposed model yielded 97% accuracy on the real-time dataset UGR'16 and 94% on the simulated dataset, UNSW NB-15, which indeed defines the proposed model as significantly superior in terms of prediction accuracy.

Researchers [19] have found that Working methods such as machine learning and ensemble classifiers are effective in intrusion detection systems when integrated with feature selection techniques to enhance performance. In the past, gain ratio feature evaluator (GRFE), correlation ranking filter (CRF), and other associated optimal feature selection methods were used with the NSL-KDD dataset in detecting attacks. These results indicated that the proposed method, GRFE, outperformed CRF more efficiently in choosing features for the target. According to the proposed system, the experiment used two widely studied datasets, namely NSL-KDD and UNSW-NB15, for intrusion detection. With the assistance of two methods, Lazy IBK and Random Committee, and taking advantage of sub-related features generated from GRFE, it is found that there is a significant misclassification difference of 0. 969% and 1.19% in the NSL-KDD dataset, and 1.62% and 1.576% in the UNSW-NB15 dataset. This contrast affirms the group methodology in outdoing the individual approach, hence supporting group methods in improving IDS functionality.

The model suggested by [20] comprises three primary stages to put forward a completely new methodology in the field of IoT intrusion detection, which includes the clustering and dimensionality reduction steps using 'k-means++ clustering', the over-sampling step using 'SVM-SMOTE', and the classification step using 'SLFN'. So, it has to be mentioned that this paper has some advantages in its approaches, such as the methods of supervised and unsupervised intrusion detection, as well as using multiple methods for data reduction and oversampling to achieve the data balance of training data sets. The findings indicated that the proposed SLFN was valuable for the process of intrusion detection due to its high classification when combined with the synthetic oversampling technique SVM-SMOTE and employing convenient parameters. The testing was done accurately, as the model achieved an accuracy of 93.51%. In the meantime, the ratio was 0.9.

Researchers [21] used the dataset named IoTID20. As a proposal of several deep learning techniques, this work itself is crucial for identifying intrusion in the IoT setting. There are three DL models employed for classification: convolutional neural networks (CNN), long short-term memory (LSTM), and a combined CNN–LSTM model. In order to better enhance the system's performance, the authors chose the PSO detection technique to choose the most relevant features applicable to the actual problem and perform dimensionality reduction. Bridging the gap has yielded an accuracy rate of 96. 60% for CNN, 98. 20% for LSTM and 98. 0% for the CNN-LSTM. Additionally, comparative research involving other current systems illustrates the effectiveness of the proposed framework in augmenting IoT security enhancement, therefore, illustrating the viability of the proposed framework in real-world intrusion detection practices.

In research [22], supply favourable outcomes showing the effectiveness of various deep learning models for identifying fog-based attacks. The network consists of three levels: cloud, secure fog, and intelligent sensing mechanisms. The following are the phases that make up the framework: (1) establishing the network, (2) classifying data on the network, (3) establishing the network together with implementing deep learning models, (4) determining the attack, (5) modifying the cloud behaviour, and (6) modifying the FN network. Where DL models are applied like DNMLP, LSTM, Bi-LSTM, GRU, CNN + LSTM, and HEM (Hybrid Ensemble Method), one model is employed at every stage and sequentially. The LSTMDL model observed high accuracy on multiple datasets: DDoS-SDN, NSLKDD, UNSW-NB15, and IoTID20 datasets, with accuracy rates of 99.70%, 99.12%, 94.11%, and 99.88% attained. This is an indication that the LSTMDL model is very efficient in solving binary classification problems. Furthermore, analysing the results of the CBDT study to compare with the results of the proposed DNMLP and proposed LSTMDL model for detecting Communication Behaviour (CB) and for predicting attacks effectively and accurately shows that although DNMLP is faster to detect CB than other models, the LSTMDL model is more effective.

Researchers [23] developed a collaborative IDS called MidSiot. Its objective is to mitigate cybersecurity threats in the connectivity of IoT. MidSiot's three-stage operation is: The following objectives of the study were defined: 1- to categorise devices of smart cities IoT, 2- to recognise sorts of attacks, and 3- to distinguish between friendly and hostile traffic both at global Internet gateways and local IoT gateways. MidSiot offers a very good detection rate in seven common IoT cyber threats and leverages edge computational capabilities. In this paper, the average accuracy of three types of IDS datasets, IoTID20, CIC-IDS-2017, and BOT-IoT, is 99%. 68% and in turn is above other IDS. This shows how effective MidSiot is as an applied intervention strategy for protecting IoT networks against threats.

The study of [24] aims to use the machine learning approach to enhance the performance of detecting unwanted data on the internet. In the process of improving the detection performance in multi-domain systems, the proposed approach focuses on the use of two datasets: UNSW-NB15 and IoTID20, which incorporate both local and IoT network traffic. The model ensures equal features by applying PCA to minimise the feature number to 30 after horizontally combining the datasets. Their approach is an Extra Boosting Forest (EBF), which is a stacked ensemble of tree classifiers such as random forest, gradient boosting classifier, and extra tree classifier. From the empirical findings, it can be deduced that EBF works; it scored the following accuracy ratings on the multi-domain dataset: 0.985 and 0.984 for two and four classes, respectively, which is higher than previous techniques.

Researchers [25] introduce a hierarchical intrusion detection (IDS) model, which is based on a meta-heuristic optimisation algorithm to enhance network security. The datasets (UNSW-NB15, CICID2017) were employed to assess the proposed model. The suggested system achieved high detection rates with minimal false alarm rates. He presented a multi-category classification solution that opens the way to building distinguished systems in the field of securing computing environments.

The authors [26] provide a general approach for constructing an intrusion detection model based on ensemble learning. The choice is as follows: DT, LR, and NB for the base levels, and stochastic gradient descent SGD as the meta-level. The real-world datasets that are used to evaluate the performance of the machine learning and ensemble models include the CIC-IDS2017, UNSW-NB15, and KDD Cup 1999 datasets. The model also employs the chi-square disparity as its feature selection tool. The analysis highlighted the fact that the ensemble classifier has an added effect on enhancing the IDS capabilities and minimising the chances of false positives and negatives. Therefore, the model has a precision rate of 99.84%.

The research of [27] describes INFUSE, a learning system that utilises meta-learning specifically for network intrusion detection and is built with the help of deep neural networks. Pros of the suggested architecture concern the issues connected with the limited description of attack types in the prior systems and data modification. It uses feature aggregation on the one hand and decision-level fusion on the other so that the detection rates of new attacks can be improved, besides improving the efficiency of Machine learning models. Deep Meta-Learner for ensemble learning, to learn semantic links and construct a hybrid feature space, is a part of the problem. The system works in three stages. For this project, five classifiers will be employed to train five different decision spaces. Next, a coarse and sparse autoencoder that learns related semantic connections between attacks expands the feature representation. The last decision is with the deep Meta-Learner, where it plays an aggregation function to correlate the mixed feature space. On applying the evaluation for the case of the proposed approach on NSL-KDD, the gained accuracy was equal to 91.6% percent of the effectiveness of INFUSE. The F-score of 0.91 and the recall of 0.94 values demonstrate a high level of generalisation of the knowledge and the ability to notice the threats in the network.

Researchers [28] using a deep learning model, namely Pearson-Correlation Coefficient - Convolutional Neural Networks (PCC-CNN), add a new strategy to intrusion detection in Internet of Things networks. It works for multiple classes of attack types through multiclassification and for anomaly detection through binary classification. To address this problem, the model convoluting significant features extracted using linear-based approaches with convolutional neural networks significantly improves its accuracy in detecting network anomalies. The experiment is performed on three external datasets, including NSL-KDD, CICIDS-2017, and IOTID20 datasets. Five PCC-based machine learning models, Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbour, Classification & Regression Tree CART, and SVM, are trained and tested. Evaluation of the classifiers on the three datasets yields the best similar accuracy of 98%, 99%, and 98% for both KNN and CART.

Research by [29] aims to understand and analyze the performance of deep learning algorithms, as three deep learning algorithms (DNN, LSTM, and CNN) were used in the context of intrusion detection, and using (CIC-IDS 2017) as a data set for this study, it achieved an accuracy of 94.61%,97.67%, and 98.61%, in order.

Researchers [30] have put forward an improved IDS using stacked ensemble learning by employing a group of various machine learning algorithms. However, to optimise the performance, it combines three basic models, RF, DT, and K-NN, and the meta-model is a logistic regression model. It ultimately contributes to improving the design's efficiency. The dataset that has been used for the above-proposed model is UNSW-NB15. The proposed system has been tested, and it has passed this test with a very high percentage of strike rate of 97. 95%. The use of the above approach in machine learning is thus equally significant, as it creates new opportunities.

The authors [31] provide a new approach based on the integration of ensemble learning and subspace clustering approaches aimed at the detection and identification of intrusions in Internet of Things networks. This is because three novel solutions have been proposed in this study aimed at enhancing detection efficiency and model robustness as follows: Iterative Feedback Loop IFL, Two-Level Decision Making TDM, and Clustering Results as Features CRF. In this paper, using four subspace clustering algorithms, that is, CLIQUE, PROCLUS, SUBCLU, and LOF; three base classifiers: NB, LGBM, and XGB; logistic regression as the meta-learner; and mutual information for the feature selection, the proposed framework achieves substantial increases in the F1-score, accuracy, precision, recall, and FP-rate. The dataset UNSW-NB15 used for evaluation yields remarkable results: accuracy 97.05%, precision 96.33%, recall 96.55%, F1-score 96.45%, and a false positive rate of 0.029.

3. Methodology

This methodology in Figure 1 aims at creating an Intrusion Detection System (IDS) that can be used in Internet of Things (IoT) networks through an implementation of machine learning techniques to identify the type of network intrusions. The described procedure is followed to conduct preprocessing of the data, develop, test, and deploy machine learning models to detect an intrusion.

3.1 Dataset Description

The dataset consists of 7,062,606 records and 27 features that provide a detailed description of different characteristics associated with the IoT network traffic and the types of attacks. The data consists of numerical and categorical properties, most of which are floating-point values representing various statistics of network traffic in terms of mean, variance, weight, magnitude, and covariance of different features (including MI, H, HH, and HpHp at different levels (e.g., L0.1)). Such numerical columns as MI_dir_L0.1_weight, H_L0.1_mean, HH_L0.1_magnitude, and HpHp_L0.1_covariance hold the features of the intensity and location of the traffic signals, which are critical to the intrusion detection analysis.

The dataset also contains categorical columns such as Device Name, Attack, Attack sub typeType and label that represent the type of device to be used in the network, the particular attack type, as well as the subtype attack and an attack binary label that represents the presence or absence of the attack. This rich feature set of the dataset makes it quite appropriate when it comes to the application of machine learning algorithms in intrusion detection systems because it will not only contain the raw data on the threats but also have labelled points where supervised learning models can be trained. The sheer amount of data makes it more useful in developing powerful models that can detect existing and emergent IoT security threats.

The data therein comprises a distribution of attacks that have been coded in the column referred to as Attack, which classifies the data into the various types of attacks and a normal class. According to the given pie chart in Figure 1, most of the data is of the type of Mirai attack and comprises about 51.9 per cent of the information. This is then followed by gafgyt, which comprises approximately 40.2 per cent of the dataset. The other 7.9 per cent of the data is normal, non-attack cases, which is essential in determining malicious and legitimate network traffic. The uneven distribution of the types of attacks indicates that the dataset is biased towards certain attack cases, with Mirai and Gafgyt being the most frequent types of attacks.

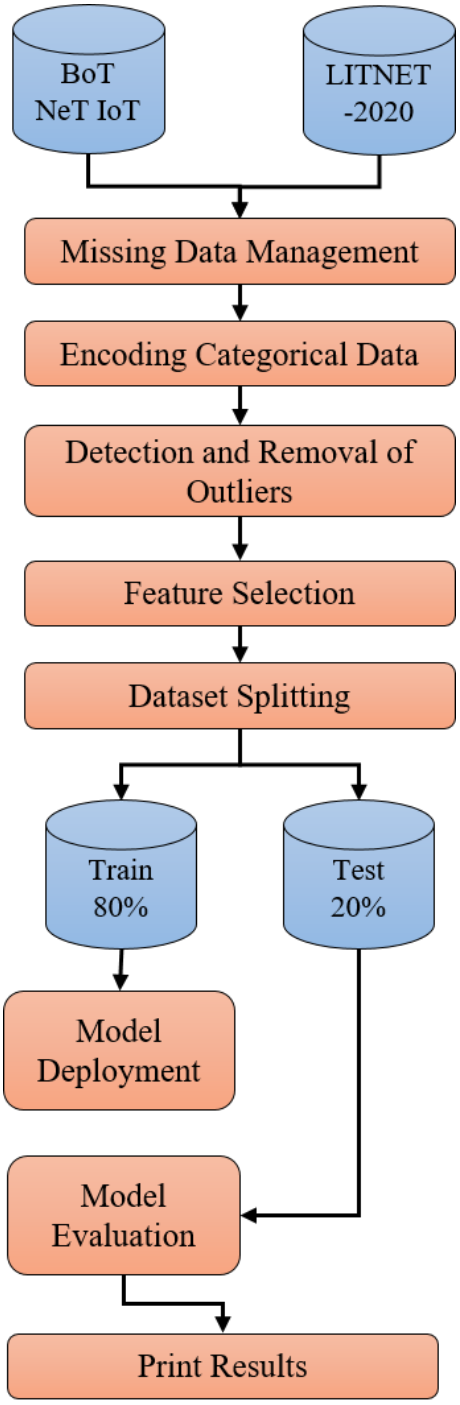

Figure 1. Working Methodology

Such a distribution can presumably influence the performance of machine learning models, because the techniques might be required to focus on the imbalance of classes to prevent the model bias in the more frequently represented classes of attacks.

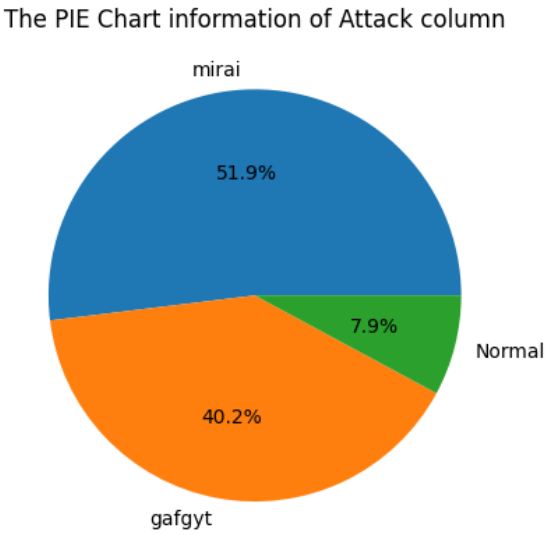

Figure 2. Dataset Classes Distribution

3.2 Dataset Preprocessing

3.2.1 BOT NET IOT DATASET

Data preprocessing is one of the most significant data preparation steps for machine learning algorithms. During our study, we performed various preprocessing steps, including handling missing values, categorical variable encoding, detection and removal of outliers, and feature selection.

Missing Data Management: The next crucial part of data preparation to be used in machine learning was concerned with the problem of missing data. All the features were audited in a comprehensive manner to identify the missing values. Columns that had high percentages of missing data were regarded as unreliable and thus not included in further analysis. In the case of columns having a low percentage of missing values, imputation strategies were used in such a way that the datasets were consistent. In particular, it was either the mean or the median that was used to impute numerical features based on data distribution. Normal data was imputed by the mean, whereas the median was used on skewed data. This was to make sure that the important information is not lost and the dataset can be adequately comprehensive as regards training and testing a machine learning model.

Encoding Categorical Data: As the majority of models of machine learning use numeric data only, categorical variables need to be encoded into numerical values. Label Encoding was used to encode categorical attributes like Device Name, Protocol Type and Attack Category. The approach is to give a distinct integer to each category of a feature. So, assuming that the feature in question, the Attack Category, has DoS, Ransomware and Botnet, then it will be encoded like that: 0, 1 and 2, correspondingly. Label encoding would be particularly useful when the ordinal feature is a categorical feature or the number of distinct values is not too big. This was necessary to allow the algorithms to operate on the categorical data in an interpretable mathematical format, making it compatible with the models that supervised learning employs.

Detection and Removal of Outliers: The presence of outliers might dramatically alter the learning process of a vast majority of machine learning models, and especially those that depend on the scale and distribution of the input data. In response to it, the Z-score analysis was applied to identify and exclude outliers in the dataset. The Z-score indicates the number of standard deviations of a point of data from the mean. Data outside the range of 3 and -3 Z-score was taken as an outlier and discarded. This value was selected according to the empirical rule that in the statistical distribution, 99.7 per cent of the data is within three standard deviations of the mean. This removal of these unusual entries helped to increase the generalisation capability of the model and avoid breaking down the performance due to the presence of extreme values.

Feature Selection: In order to maximise the performance and efficiency of the model, feature selection was used. This occurred through statistical examination of the relationship between each input feature and the target attribute; in this case, the Attack class. A correlation matrix was created, and features that had a Pearson correlation coefficient of greater than 0.25 with regard to the target were then picked. This value was selected in order to strike a balance between keeping useful information and eliminating noise brought by irrelevant or loosely connected features. The model was trained on the basis of the selected

features. The dimensionality reduction not only achieves more efficient computation but also decreases the probability of overfitting and makes the model more interpretable because it concentrates on the most influential variables.

3.2.2 LITNET-2020 DATASET

The LITNET-2020 dataset targets the research contribution in the area of intrusion detection systems (IDS) in the Internet of Things (IoT) networks. It encompasses network traffic information, such as regular and malicious network activities, to allow training and testing of machine learning models in security threat identification. It has 7,062,606 rows and 27 columns and includes different features of IoT network traffic and attacks on these systems. The attributes contained in the dataset are both numerical and categorical. Numerical features indicate the different statistics of network traffic; these include the mean, variance, weight, magnitude, and covariance of different attributes of traffic. These characteristics play a pivotal role in identifying the unusual activity of network communications. Among the most important numerical characteristics are MI_dir_L0.1_weight, H_L0.1_mean, HH_L0.1_magnitude, and HpHp_L0.1_covariance that reflect the intensity of the network traffic, its location, and its variation with the measures at different levels.

Some other attributes of the dataset are categorical, such as Device Name, Attack Type, Attack Subtype, and labels that correspond to the type of device in the network, the type of attack being used, and a binary label that specifies whether an attack is used or not. The data is highly skewed, with most of the traffic caused by only two kinds of attacks, namely Mirai and Gafgyt, that make up more than 92 per cent of the data. The data containing non-attack traffic, i.e. normal traffic, comprises approximately 7.9% of the data set, and is important in isolating malicious behaviour from normal network traffic. The given dataset can perfectly train machine learning models since it covers labelled objects used in supervised learning that would help detect known and emerging IoT security threats. The large extent and wide variety of the data allow it to be used to create IDS models that can be used to develop scalable and efficient models suitable for use in real-life IoT security challenges.

3.3 Model Deployment

Once we pre-processed the data, we proceeded to the model-building process. The primary objective was to build a machine learning model that would effectively classify whether traffic on a network is normal or an attack. A variety of models were selected so that we would be able to identify the best algorithm for this process.

For each of the machine learning models, the hyperparameters in Table 1 play a crucial role in the model's performance. The following table summarises the key training parameters used for each model:

Table 2. Machine Learning Models

Model	Description	Reference
Logistic Regression	Binary classification algorithm for linear data.	[6]
Decision Tree	A non-linear classifier that splits data based on features. Prone to overfitting.	[32]
Random Forest	Ensemble method using multiple decision trees to reduce overfitting and improve accuracy.	[33]
XGBoost	Efficient gradient boosting method for classification with regularisation and handling missing data.	[34]
LightGBM	Optimised gradient boosting method for large datasets with efficient memory usage.	[35]
AdaBoost	Boosting algorithm that corrects errors of previous models. Sensitive to noise and outliers.	[36].
Gradient Boosting	An ensemble method that minimises residual errors progressively. High predictive power.	[37]
Bagging	An ensemble to reduce variance and prevent overfitting by averaging multiple models.	[38]
K-Nearest Neighbors	A non-parametric classifier that predicts based on nearest neighbours. Sensitive to irrelevant features.	[39]
Naive Bayes	Probabilistic classifier assuming feature independence. Efficient for text classification.	[40]

Table 3 shows the most important hyperparameters to train different machine learning models, which are analysed in this paper. The settings of each of these models were either tuned during the initial tuning procedure or chosen according to the general good practices that have been mentioned in the literature. The mentioned list of parameters directly affects the process of learning, complexity, and generalisation that the model would be able to do. The comparison between the models will be fair and consistent because the same configurations have been standardised on all of the experiments, which makes the task of evaluating the performance of a model fair and reliable in cases such as classification or prediction.

Table 3. Models Training Parameters

Model	Parameter	Value
Logistic Regression	max_iter	500
	Solver	'liblinear'
Decision Tree	Criterion	'entropy'
	max_depth	5
	min_samples_split	2
Random Forest	n_estimators	100
	max_depth	None
	min_samples_split	2
	max_features	'sqrt'
XGBoost	learning_rate	0.1
	n_estimators	100
	max_depth	6
	Gamma	0.5
LightGBM	learning_rate	0.05
	n_estimators	100
	max_depth	6
	num_leaves	31
AdaBoost	n_estimators	50
	learning_rate	1
Gradient Boosting	n_estimators	100
	learning_rate	0.1
	max_depth	3
	Subsample	0.9
Bagging	n_estimators	100
	max_samples	1
	max_features	'sqrt'
SVM	C	1
	Kernel	'rbf'
	Gamma	'scale'
KNN	n_neighbors	5
	Algorithm	'auto'
Naive Bayes	var_smoothing	1.00E-09

3.4 Model Evaluations

Assessing the performance of machine learning models in Table 4, particularly in a security context, such as an Intrusion Detection System (IDS), would need various metrics to measure various properties of the model. The most common are such metrics as accuracy, determining the overall correctness of the conclusions on the classification and may be deceptive in unbalanced datasets in which classes are over-represented. Precision and recall give an indication of how the model encounters the positive classes, which is important in intrusion detection, as the main concern is to ensure that attacks are detected correctly rather than focusing on accuracy. F1-score runs an average of both precision and recall, giving one value that can be used whenever class imbalance exists. ROC AUC is used to measure how well a model separates between classes at all levels of discrimination. Lastly, Log Loss is used to measure the certainty level of probabilistic predictions, and it is more penalising in case of wrong predictions where the model is very confident about it. Collectively, they help to get a clear picture of the predictive capability and the overall strength of a model to identify network intrusions.

Table 4. Evaluation Metrics

Metric	Description	Formula	Importance in IDS & Classification
Accuracy	Measures overall correctness of predictions (both positive and negative).	$Accuracy = \frac{Tp + TN}{TP + TN + FP + FN}$	Useful when classes are balanced; shows general model accuracy.
Precision	Proportion of predicted positives that are truly positive.	$recision = \frac{TP}{TP + FP}$	Important when false positives are costly (e.g., false alarms).
Recall	Percentage of actual positives correctly identified (True Positive Rate).	$Recall = \frac{TP}{TP + FN}$	Critical when missing positives (false negatives) is risky.
F1-Score	Harmonic mean of precision and recall, balancing the two.	$F1 - Score = 2 \times \frac{percision \times recall}{Percison + recall}$	Useful when false positives and false negatives are equally important.
ROC AUC	Measures the model's ability to distinguish classes across thresholds; area under the ROC curve.	$AUC = \int_0^1 TPR(FPR)dFPR$	Indicates discrimination ability; 1 is perfect, 0.5 is random.
Log Loss	Measures the confidence of probabilistic predictions by penalising wrong, confident predictions.	$Log\ Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$	Lower values indicate better probabilistic predictions.

4. Results and Discussion

4.1 BoT Net IoT Results

In the modern world of blistering technological changes, the Internet of Things (IoT) has opened up new avenues to the communication and interaction between devices and instigated the creation and transmission of huge amounts. These types of interconnectivity only increase as more and more is created, which is why there are increased risks of cybersecurity. The use of IoT devices, which have low processing power and lack sufficient embedded security features, is a tempting opportunity for malicious entities. The traditional security measures that were built with the static and homogeneous network scenarios in mind are challenged to deal with the complexity and dynamism of the contemporary IoT ecosystems. Such conventional systems do not offer the flexibility needed to succeed in identifying and addressing the ever-changing trends of cyber threats in IoT systems.

Machine Learning (ML)-based Intrusion Detection Systems (IDS) have evolved as a strong and smart alternative in order to face these pitfalls. In comparison to signature-based IDS that utilises known patterns of attacks, the ML-based approaches are capable of learning data and recognising abnormal behaviour and can detect unseen threats in real-time. The systems can run on network traffic constantly and can adjust to emerging attack vectors, and this makes them very accessible to the shifting environments in which IoT deployments are usually built.

This work is aimed at analysing the success of different machine-learning algorithms in identifying intrusions to IoT networks. Models that will be used in the analysis will consist of both classical classifiers, such as Decision Trees and Logistic Regression, as well as more complicated techniques, including Random Forests and Gradient Boosting. To give an objective impression of the strengths and weaknesses of each of these models, their performance will be evaluated based on a list of common evaluation criteria.

Logistic Regression, as a simpler model, gives stable performance in binary classification. When used in IoT intrusion detection, it achieves 99.11% accuracy, meaning it classifies the majority of normal and anomalous traffic correctly. Its precision of 99.21% reflects the ability to reduce false positives so that not many benign activities are misclassified as intrusions. The recall of 89.44% shows that the model misses some potential threats, although it catches most of the intrusions. The F1-score of 94.07% informs us of an equitable trade-off between precision and recall. With a ROC-AUC of 99.82%, Logistic Regression is effective at class separation. Yet, the log loss of 0.0458 shows that while predictions are accurate, there is room for improvement in calibration.

Decision Trees are a widely used machine learning model since they are simple to comprehend and interpret. The Decision Tree model performs exceptionally well in this study with an accuracy of 99.99%, which demonstrates nearly perfect classification. Its 99.96% accuracy means there are a few false positives, and the 99.88% recall means it is extremely good at detecting almost all intrusions. The 99.92% F1-score confirms this model's well-rounded performance. The 99.94% ROC-AUC score means a very good ability to distinguish between normal and malicious behaviour. Furthermore, the 0.0041 log loss is extremely low, showing well-calibrated predictions. This makes the Decision Tree a strong candidate for real-time intrusion detection in IoT networks.

Table 5. Models Evaluation on BoT Net IoT Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Log Loss
Logistic Regression	0.991073	0.992062	0.894360	0.940681	0.998194	0.045808
Decision Tree	0.999873	0.999552	0.998837	0.999195	0.999356	0.004116
Random Forest	0.999936	0.999642	0.999553	0.999597	0.999973	0.000503
XGBoost	0.999763	0.997812	0.999195	0.998503	0.999971	0.000833
AdaBoost	0.999345	0.993809	0.997943	0.995872	0.999918	0.357086
Gradient Boosting	0.999604	0.995987	0.999016	0.997499	0.999843	0.001966
Bagging	0.999880	0.999418	0.999061	0.999240	0.999824	0.001506
KNN	0.999650	0.997186	0.998390	0.997787	0.999846	0.002584
Naive Bayes	0.742829	0.233242	0.983362	0.377051	0.902696	2.441735

Random Forest combining decision trees is the most efficient one with an accuracy of 99.99%, precision and recall of 99.96% and ROC-AUC 99.99, thus able to identify intrusion with a few false positives. It has confidence in its predictions as its log loss is very low (0.0005); thus, it is an appropriate model to use in IoT intrusion detection.

XGBoost, an advanced Gradient Boosting algorithm, attains an accuracy of 99.98 with 99.78 precision and 99.92 recall. It provides good accuracy and a sufficient balance between accuracy and speed with a 99.99 % ROC-AUC and a bit larger log loss (0.00083). AdaBoost gives 99.93% accuracy and 99.79% recall but a lower precision (99.38%), which indicates a higher number of false positives. It has a high ROC-AUC (99.99%), but the log loss (0.3571) means lower confidence when it comes to predictions.

Gradient Boosting also achieves a high accuracy of 99.96 %, precision of 99.60% percent and recall of 99.90% percent. It has a ROC-AUC of 99.98% and low log loss (0.00197), which implies its correct and accurately calibrated predictions. Bagging gives 99.99 accuracy to Random Forest with precision and recall of 99.94 and 99.91, respectively. It is balanced in performance, and ROC-AUC (99.98%), as well as a low log loss (0.00151).

K-Nearest Neighbours (KNN) provides 99.97% accuracy, 99.72% precision and 99.84 % recall. It is 0.00258 less confident but still useful as a method of intrusion detection. Naive Bayes has 74.28 % accuracy and extremely low precision of 23.32 % thus leaving a high number of false positives. Although the recall is high (98.34), the low reliability of IoT IDS by 2.44 log loss and 90.27 % ROC-AUC is a poor indicator of the reliability of IoT IDS.

4.2 LITNET-2020 Results

The analysis of different machine learning algorithms on the LITNET-2020 dataset (Table 6) shows clear performance trends between the different models. The algorithms were tested based on fundamental metrics like accuracy, precision, recall, F1-score, ROC-AUC, and log loss in order to get a holistic view of how effective the models are in intrusion detection for IoT networks.

Logistic Regression showed decent performance with an accuracy of 92.01%. However, its 81.01% recall indicates that it had failed to detect a vast majority of the intrusions, which could turn out to be crucial in security-sensitive applications. Although excellent in precision (92.66%), its relatively lower recall made it yield a medium F1-score of 86.36%. The ROC-AUC of 0.975 reflects well-balanced overall classification performance, but the model will not be suitable for real-time high-stakes use due to its inability to classify a large number of attacks.

Decision Tree was 98.66% accurate with a precision of 98.56% and a recall of 98.23%, reflecting well-balanced performance in respect of both false positives and false negatives. Its 98.40% F1-score and great ROC-AUC score of 0.992 also point to its usefulness in distinguishing normal from bad traffic. Decision Tree models are prone to overfitting, though, and while this model performed adequately, it might not generalise well across some IoT environments with new or unknown attack patterns.

Random Forest topped Decision Tree with a 99.23% accuracy and with an almost perfect precision score of 99.01%. The recall of 98.87% in turn indicates that the model identified nearly all the attacks correctly with a very impressive F1-score

of 98.94%. The extremely high ROC-AUC of 0.995 and also a low log loss (0.0156) indicate a well-calibrated model whose prediction can be trusted in actual applications where detection accuracy and low false positives are critical. Random Forest is the best choice for IoT intrusion detection since it's robust and offers improved generalisation.

Table 6. Models Evaluation on LITNET-2020 Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Log Loss
Logistic Regression	0.920101	0.926561	0.810122	0.863646	0.975042	0.134578
Decision Tree	0.986582	0.985604	0.98233	0.983959	0.992319	0.024866
Random Forest	0.992271	0.990133	0.988728	0.98943	0.995478	0.015603
XGBoost	0.991017	0.987401	0.989251	0.988324	0.994572	0.013982
AdaBoost	0.989605	0.974712	0.988272	0.981383	0.997145	0.428748
Gradient Boosting	0.990342	0.978232	0.991131	0.984633	0.994285	0.008991
Bagging	0.992738	0.991524	0.990217	0.99087	0.995328	0.010746
KNN	0.991529	0.982978	0.985231	0.984103	0.994084	0.012879
Naive Bayes	0.682393	0.161924	0.955889	0.277073	0.835024	2.657383

XGBoost performed slightly poorer than Random Forest but still had good performance with an accuracy of 99.10%, precision of 98.74%, and recall of 98.93%. The F1-score of 98.83% signifies equal balancing of precision and recall. XGBoost's strength lies in its mechanism of gradient boosting, which allows it to minimise errors by iteratively learning from them. While having slightly higher log loss (0.01398) than Random Forest, it is still highly efficient in intrusion detection and can be considered a reliable option for IoT network security.

AdaBoost's accuracy was 98.96%, precision was 97.47%, and recall was 98.83%. The model's F1-score of 98.14% testifies to good overall performance, but the comparatively high log loss of 0.4287 suggests that it may be less certain in its outputs, particularly in noisy or uncertain situations. Still, despite this weakness, AdaBoost's ROC-AUC value of 0.9971 testifies to its high discrimination capability and suggests that it is a strong contender for IoT IDS usage, particularly where detection of new or unknown attack types is crucial.

Gradient Boosting was outstanding with 99.03% accuracy, 97.82% precision, and 99.11% recall. 98.46% F1-score and ROC-AUC of 0.9943 also validate that the model discriminates well between attack and normal traffic. Low log loss (0.00899) also means that the model is calibrated and has high-confidence predictions. Gradient Boosting is a highly suitable candidate for IoT IDS deployment where high accuracy and recall matter to minimise false negatives and false positives. Bagging was 99.27% accurate with 99.15% precision and 99.02% recall. Its 99.09% F1-score and high ROC-AUC of 0.9953 place it on the same level as Random Forest and Gradient Boosting. Its low log loss of 0.0107 shows that Bagging can decide with high confidence and with the least chances of making a wrong decision. This ensemble method is another robust choice for real-time IoT intrusion detection, particularly when high dependability and equal detection are required.

K-Nearest Neighbours (KNN) also performed a bit below the top models with 99.15% accuracy, 98.30% precision, and 98.52% recall. It had a decent F1-score of 98.41%, but it is more susceptible to noisy data from the moderate log loss of 0.0129. KNN models are usually computationally expensive, and their performance is susceptible to decreasing as the data size grows, so they are not as well-suited for large-scale IoT deployments as ensemble methods like Random Forest or Gradient Boosting. Naive Bayes did very badly with an accuracy of merely 68.24% and a low precision of merely 16.19%. Though its 95.59% recall was good, indicating its ability to catch most intrusions, its low precision came in the form of high false positives. Both the F1-score at 27.71% and extremely high log loss (2.6574) reflect the inefficiency of the model in general in correctly classifying normal traffic and attack traffic. Naive Bayes is therefore unsuitable for IoT IDS use cases where minimising false positives and achieving high classification accuracy are paramount.

4.3 Results Discussions

Figure (3) compares a few machine learning models' performances on key evaluation metrics: accuracy, precision, recall, F1-score, ROC-AUC, and log loss on the BoT Net IoT Dataset and the LITNET-2020 Dataset. Overall, the ensemble models, particularly Random Forest, Bagging, and XGBoost, perform consistently well on both datasets. These models are highly accurate, with values close to 1, indicating that they are good at classifying normal as well as malicious traffic. Naive Bayes is significantly below the others, particularly on the LITNET-2020 Dataset, for which its accuracy is extremely low.

For accuracy, the ensemble models again win the competition, and Naive Bayes does poorly, especially on the LITNET-2020 Dataset, where it fails to distinguish between the normal and attack classes and yields high false positives. The recall metric, which deals with finding all cases of attacks applicable to it, shows that all models except for Naive Bayes have high recall. However, Naive Bayes has high recall but at the cost of precision since it has a tendency to label many common instances as attacks, resulting in many false positives.

As far as the F1-score, which is a balance of precision and recall, is concerned, the ensemble methods work well for both datasets. Naive Bayes works with a much lower F1-score due to its poor precision. The ROC-AUC value, which reflects the ability of the model to distinguish between classes, also shows that ensemble models, notably Random Forest and XGBoost, show great performance with nearly 1 scores, indicating great discriminatory power. Naive Bayes shows a sharp drop in ROC-AUC, especially in the LITNET-2020 Dataset, representing its inability to notably distinguish between normal traffic and attacks.

Finally, log loss, which measures the confidence of the predictions made by the model, is the lowest among the ensemble models, meaning the predictions are high in confidence. In contrast, Naive Bayes has a much higher log loss, particularly in the LITNET-2020 Dataset, showing that its prediction is low on reliability and confidence.

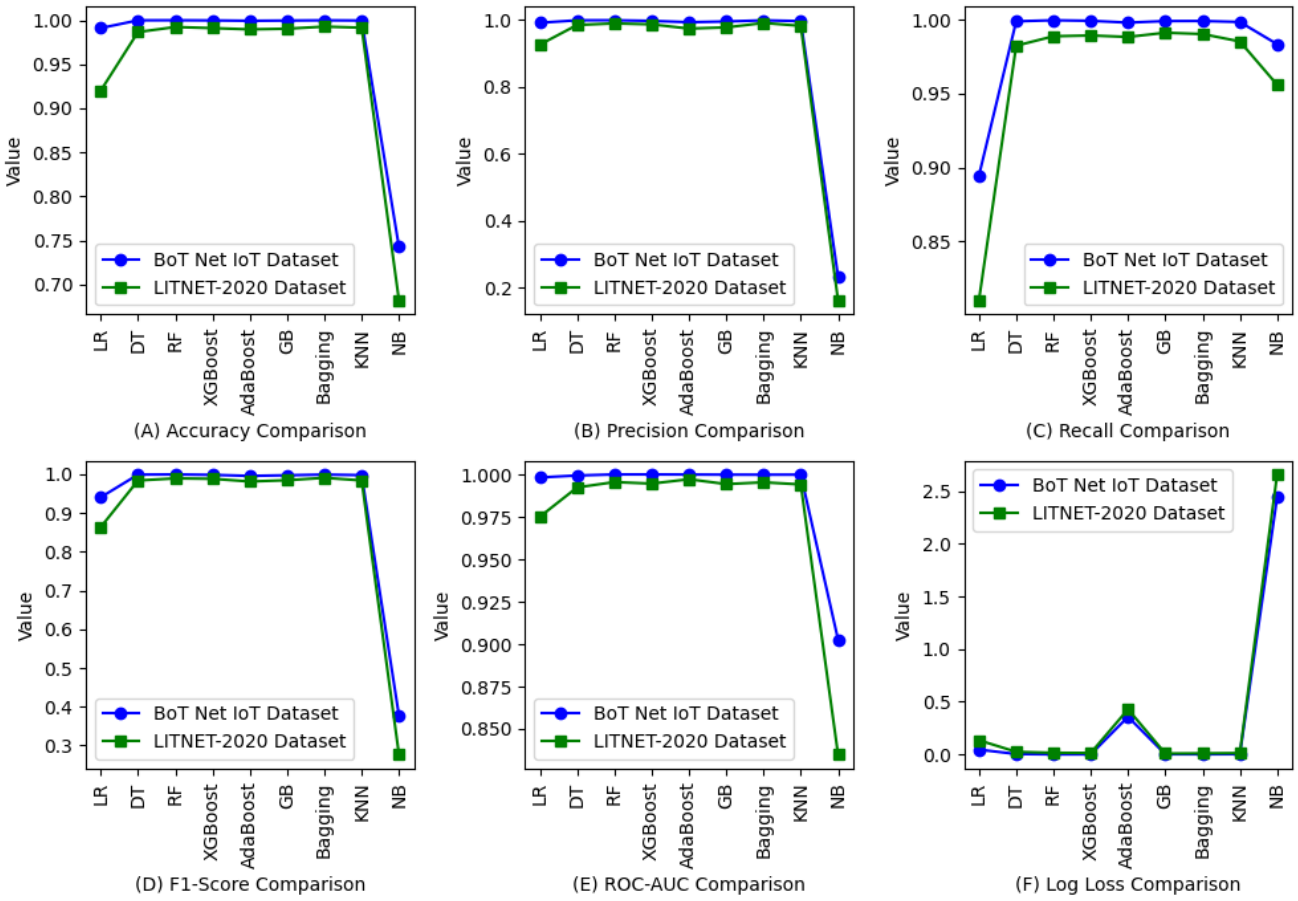

Figure 3. Results Comparison for the Two Datasets

In short, the best-performing ensemble models are Random Forest, Bagging, and XGBoost for all of the metrics and are more suitable for real-world IoT network intrusion detection. Naive Bayes, even with very high recall, does not have a good balance between precision and recall and, therefore, is not a viable choice with its low precision, high log loss, and poor overall performance in such a scenario.

5. Conclusions

This study evaluates the performance of several machine learning models for intrusion detection in Internet of Things (IoT) environments, specifically focusing on two widely used datasets: the BoT Net IoT Dataset and the LITNET-2020 Dataset. Our analysis provides valuable insights into the strengths and weaknesses of various models and highlights the critical importance of choosing the right algorithms for effective security in IoT networks.

The results consistently show that ensemble models, namely Random Forest, Bagging, and XGBoost, yield the best performance on all the metrics considered, including accuracy, precision, recall, F1-score, ROC-AUC, and log loss. These models are shown to be robust in both intrusion detection and in minimising false positives, making them highly suitable for real-time intrusion detection in dynamic IoT environments. Their ability to maintain good detection rates and reduce

misclassifications underscores their usefulness for safeguarding IoT environments, which by design are resource-constrained and vulnerable to every type of cyber-attack.

On the other hand, Naive Bayes, despite high recall, has low precision and a very high log loss, especially in the LITNET-2020 Dataset. This implies a high false positive rate and renders it less suitable for real-time IoT security. The inability to strike a balance between recall and precision and low discriminatory power, as manifested by ROC-AUC, reveals the unsuitability of Naive Bayes for advanced IoT configurations. In addition, the findings underscore the necessity of considering both precision and recall, particularly in IoT intrusion detection, where false positives are costly. Those models that strike a good balance between the measures, i.e., Random Forest and XGBoost, offer the most acceptable trade-offs and will more likely be dependable for high-consequence, real-time monitoring settings.

This study demonstrates the promise of ensemble learning techniques to address the specific challenges posed by IoT security. The results certify that Random Forest, Bagging, and XGBoost are extremely robust and scalable models and thus superb candidates for the deployment of intrusion detection systems in IoT networks. Future work can explore the fusion of these models with other emerging technologies, e.g., edge computing, for additional real-time performance and scalability enhancement in large-scale IoT environments. In addition, the application of hybrid models that combine classical machine learning and deep learning could provide even more advanced and adaptive solutions to the arising threats in IoT..

6. Acknowledgment

I would like to express my sincere gratitude to the College of Education for Pure Sciences at the University of Mosul for its invaluable support and guidance throughout this work. Its academic environment and resources provided a solid foundation that greatly contributed to the successful completion of this study.

7. Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this work. All research was conducted objectively, without any financial, personal, or professional relationships that could inappropriately influence the results or interpretations presented in this study.

8. Funding

The authors declare that they have no known competing financial interests in this paper

9. References

[1] T. Alam, “A Reliable Communication Framework and Its Use in Internet of Things (IoT),” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, no. May, 2018.

[2] I. Coston, E. Plotnizky, and M. Nojoumian, “Comprehensive Study of IoT Vulnerabilities and Countermeasures,” *Appl. Sci.*, vol. 15, no. 6, 2025, doi: 10.3390/app15063036.

[3] A. Zafar, F. Samad, H. J. Syed, A. O. Ibrahim, M. Alohaly, and M. Elsadig, “An Advanced Strategy for Addressing Heterogeneity in SDN-IoT Networks for Ensuring QoS,” *Appl. Sci.*, vol. 13, no. 13, 2023, doi: 10.3390/app13137856.

[4] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "Applied Sciences IoT Privacy and Security : Challenges and Solutions," *Mdpi*, pp. 1–17, 2020.

[5] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, “A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges,” *J. Inf. Intell.*, vol. 2, no. 6, pp. 455–513, 2023, doi: 10.1016/j.jiixd.2023.12.001.

[6] H. Sebestyen, D. E. Popescu, and R. D. Zmaranda, “A Literature Review on Security in the Internet of Things: Identifying and Analysing Critical Categories,” *Computers*, vol. 14, no. 2, 2025, doi: 10.3390/computers14020061.

[7] A. Alotaibi, H. Aldawghan, and A. Aljughaiman, “A Review of the Authentication Techniques for Internet of Things Devices in Smart Cities: Opportunities, Challenges, and Future Directions,” *Sensors*, vol. 25, no. 6, 2025, doi: 10.3390/s25061649.

[8] E. Dritsas and M. Trigka, “A Survey on Cybersecurity in IoT,” *Futur. Internet*, vol. 17, no. 1, 2025, doi: 10.3390/fi17010030.

[9] I. Coston and E. Plotnizky, “Comprehensive Study of IoT Vulnerabilities and Countermeasures,” *Appl. Sci.*, 2025.

[10] A. Alfahaid, E. Alalwany, A. M. Almars, F. Alharbi, E. Atlam, and I. Mahgoub, “Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey,” *Sensors*, vol. 25, no. 11, pp. 1–48, 2025, doi: 10.3390/s25113341.

[11] S. I. Amgbara, C. Akwiwu-uzoma, and O. David, “Exploring lightweight machine learning models for personal internet of things (IOT) device security Exploring lightweight machine learning models for personal internet of things (IOT) device security,” *World J. Adv. Res. Rev.*., no. November 2024, doi: 10.30574/wjarr 2024.24.2.3449.

[12] A. K. Dinkar and A. Haque, “Enhancing IoT Data Analysis with Machine Learning : A Comprehensive Overview Mejora del análisis de datos IoT con aprendizaje automático : Una visión global,” *LatIA*, 2024, doi: 10.62486/latia20249.

[13]M. B. Bankó *et al.*, “Advancements in Machine Learning-Based Intrusion Detection in IoT : Research Trends and Challenges,” *Algorithms Rev.*, 2025.
[14]Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Comput. Networks*, vol. 174, 2020, doi: 10.1016/j.comnet.2020.107247.
[15]X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, “An Adaptive Ensemble Machine Learning Model for Intrusion Detection,” *IEEE Access*, vol. 7, pp. 82512–82521, 2019, doi: 10.1109/ACCESS.2019.2923640.
[16]T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset,” *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.
[17]Y. V. Kumar and K. Kamatchi, "Anomaly-Based Network Intrusion Detection Using Ensemble Machine Learning Technique," *en. Int. J. Res. Eng. …*, vol. 6, no. 4, pp. 216–220, 2020, [Online]. Available: http://www.ijert.org
[18]S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, “A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets,” *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/4586875.
[19]P. Maniriho, L. J. Mahoro, E. Niyigaba, Z. Bizimana, and T. Ahmad, “Detecting intrusions in computer network traffic with machine learning approaches,” *Int. J. Intell. Eng. Syst.*, vol. 13, no. 3, pp. 433–445, 2020, doi: 10.22266/IJIES2020.0630.39.
[20]R. Qaddoura, A. M. Al-Zoubi, I. Almomani, and H. Faris, "A multi-stage classification approach for IoT intrusion detection based on clustering with oversampling," *Appl. Sci.*, vol. 11, no. 7, 2021, doi: 10.3390/app11073022.
[21]H. Alkahtani and T. H. H. Aldhyani, “Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/5579851.
[22]S. P. K. Gudla, S. K. Bhoi, S. R. Nayak, K. K. Singh, A. Verma, and I. Izonin, "A Deep Intelligent Attack Detection Framework for Fog-Based IoT Systems," *Comput. Intell. Neurosci.*, vol. 2022, p. 6967938, 2022, doi: 10.1155/2022/6967938.
[23]N. Dat-Thinh, H. Xuan-Ninh, and L. Kim-Hung, “MidSiot: A Multistage Intrusion Detection System for Internet of Things,” *Wirel. Commun. Mob. Comput.*, vol. 2022, no. December 2017, 2022, doi: 10.1155/2022/9173291.
[24]P. L. Indrasiri, E. Lee, V. Rupapara, F. Rustam, and I. Ashraf, "Malicious traffic detection in IoT and local networks using stacked ensemble classifier," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 489–515, 2022, doi: 10.32604/cmc 2022.019636.
[25]K. A. ElDahshan, A. A. A. AlHabshy, and B. I. Hameed, "Meta-Heuristic Optimisation Algorithm-Based Hierarchical Intrusion Detection System," *Computers*, vol. 11, no. 12, 2022, doi: 10.3390/computers11120170.
[26]N. Thockchom, M. M. Singh, and U. Nandi, “A novel ensemble learning-based model for network intrusion detection,” *Complex Intell. Syst.*, vol. 9, no. 5, pp. 5693–5714, 2023, doi: 10.1007/s40747-023-01013-7.
[27]A. Sohail, B. Ayisha, I. Hameed, M. M. Zafar, and A. Khan, “Deep Neural Networks based Meta-Learning for Network Intrusion Detection,” 2023, [Online]. Available: http://arxiv.org/abs/2302.09394
[28]M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, “Anomaly-based intrusion detection system for IoT application,” *Discov. Internet Things*, vol. 3, no. 1, 2023, doi: 10.1007/s43926-023-00034-5.
[29]J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in Internet of Things using CIC-IDS 2017 dataset," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 1, pp. 1134–1141, 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.
[30]A. Almomani *et al.*, “Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic,” *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 2499–2517, 2023, doi: 10.32604/iasc 2023.039687.
[31]J. Zhu and X. Liu, “An integrated intrusion detection framework based on subspace clustering and ensemble learning,” *Comput. Electr. Eng.*, vol.. 115, no. January, p. 109113, 2024, doi: 10.1016/j.compeleceng.2024.109113.
[32]I. D. Mienye and N. Jere, “A Survey of Decision Trees: Concepts, Algorithms, and Applications,” *IEEE Access*, vol. 12, pp. 86716–86727, 2024, doi: 10.1109/ACCESS.2024.3416838.
[33]M. Kathiravan, V. Rajasekar, S. J. Parvez, V. Sathya Durga, M. Meenakshi, and S. Gowsalya, “Detecting Phishing Websites using Machine Learning Algorithm,” *Proc. - 7th Int. Conf. Comput. Methodol. Commun. ICCMC 2023*, pp. 270–275, 2023, doi: 10.1109/ICCMC56507.2023.10083999.
[34]P. Barnard, N. Marchetti, and L. A. DaSilva, “Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI),” *IEEE Netw. Lett.*, vol. 4, no. 3, pp. 167–171, 2022, doi: 10.1109/lnet.2022.3186589.
[35]T. Mahmood, S. K. Hashemi, S. L. Mirtaheri, and S. Greco, “Machine Learning Techniques for Detecting Fraud in Credit Card Transactions,” *CEUR Workshop Proc.*, vol. 3478, pp. 469–478, 2023.
[36]M. A. Hambali, Y. K. Saheed, T. O. Oladele, and M. D. Gbolagade, "ADABOOST Ensemble Algorithms for Breast Cancer Classification," *J. Adv. Comput. Res.*, vol. 10, no. 2, pp. 31–52, 2019, [Online]. Available: www.jacr.iausari.ac.ir
[37]H. Kim, S. Park, H. J. Park, H. G. Son, and S. Kim, “Solar Radiation Forecasting Based on the Hybrid CNN-CatBoost Model,” *IEEE Access*, vol. 11, no. February, pp. 13492–13500, 2023, doi: 10.1109/ACCESS.2023.3243252.
[38]R. Du, L. Zhen, and Y. Liu, “Physical Layer Authentication Based on Integrated Semi-Supervised Learning in Wireless Networks for Dynamic Industrial Scenarios,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6154–6164, 2023, doi: 10.1109/TVT.2022.3231633.

[39]P. D. Rosero-Montalvo *et al.*, "Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques," *2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018*, no. December, pp. 1–6, 2018, doi: 10.1109/ETCM.2018.8580268.

[40]A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

تعزيز أمان إنترنت الأشياء: نظام كشف التسلل قائم على التعلم الآلي لرصد التهديدات والتصدي لها في الوقت الحقيقي

عمار عادل احمد

قسم علوم الحاسوب، كلية التربية للعلوم الصرفة، جامعة الموصل، الموصل، العراق

المستخلص:

يشهد استخدام أجهزة الإنترنت للأشياء (IoT) نموًا سريعًا مما أدى إلى خلق بيئة حيث تكون الأمانات عرضة للعديد من الثغرات، ويتطلب الأمر حلولًا أكثر تطورًا وتكيفًا. لا تستطيع الحلول الأمنية التقليدية تجاوز مشكلة التباين، نقص الموارد، والديناميكية في بيئات الإنترنت للأشياء. يقترح هذا البحث استخدام نظام كشف التسلل (IDS) القائم على التعلم الآلي لتحديد التهديدات الفورية والحد من وجودها في شبكات الإنترنت للأشياء. تم مقارنة نتائج نماذج التعلم الآلي المختلفة، والتي تشمل الانحدار اللوجستي، شجرة القرار، الغابة العشوائية، XGBoost، AdaBoost، Gradient Boosting، Bagging، الجار الأقرب (KNN)، وNaive Bayes استنادًا إلى بعض مؤشرات الأداء الرئيسية مثل الدقة، الدقة الإيجابية، الاسترجاع، مقياس F1، ROC-AUC، وخسارة السجل. تشير نتائجنا إلى أن الخوارزميات التجميعية، وخاصة الغابة العشوائية، شجرة القرار، وBagging، يمكن أن تكون أكثر فاعلية من النماذج الأخرى في اكتشاف عدد كبير من التهديدات مع تقليل الإيجابيات الكاذبة. فقد حققت الغابة العشوائية دقة تصل إلى 99.99%، ودقة إيجابية 99.96%، ومعدل استرجاع 99.96%، وROC-AUC بنسبة 99.99%. في المقابل، كانت نتائج Naive Bayes أسوأ بكثير، حيث أظهرت دقة 74.28%، ودقة إيجابية 23.32%، ومقياس F1 بلغ 37.71%. هذه النتائج تؤكد أن الخوارزميات التجميعية، وخاصة الغابة العشوائية، تعد فعّالة جدًا في كشف التسلل في الوقت الفعلي على أنظمة الإنترنت للأشياء. وتثبت هذه الطريقة أن التعلم التجميعي، الذي يمتلك القدرة على دمج عدة مصنفات، هو حل فعّال لتعزيز أمان أنظمة الإنترنت للأشياء. يسلط هذا البحث الضوء على إمكانية استخدام التعلم الآلي في الدفاع عن الإنترنت للأشياء كحل قابل للتطوير وفعّال في مواجهة التحديات الأمنية المتزايدة في البيئات المتصلة.