Journal of
Sensor and Actuator Networks

MDPI

Article

Machine Learning-Based Detection for Unauthorized Access to IoT Devices

Malak Aljabri [1], Amal A. Alahmadi [2], Rami Mustafa A. Mohammad [3], Fahd Alhaidari [2], Menna Aboulnour [4], Dorieh M. Alomari [5,*] and Samiha Mirza [4]

1 Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia
2 Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
3 Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
4 SAUDI ARAMCO Cybersecurity Chair, Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
5 SAUDI ARAMCO Cybersecurity Chair, Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
* Correspondence: 2180007089@iau.edu.sa

Abstract: The Internet of Things (IoT) has become widely adopted in businesses, organizations, and daily lives. They are usually characterized by transferring and processing sensitive data. Attackers have exploited this prospect of IoT devices to compromise user data's integrity and confidentiality. Considering the dynamic nature of the attacks, artificial intelligence (AI)-based techniques incorporating machine learning (ML) are promising techniques for identifying such attacks. However, the dataset being utilized features engineering techniques, and the kind of classifiers play significant roles in how accurate AI-based predictions are. Therefore, for the IoT environment, there is a need to contribute more to this context by evaluating different AI-based techniques on datasets that effectively capture the environment's properties. In this paper, we evaluated various ML models with the consideration of both binary and multiclass classification models validated on a new dedicated IoT dataset. Moreover, we investigated the impact of different features engineering techniques including correlation analysis and information gain. The experimental work conducted on bagging, k-nearest neighbor (KNN), J48, random forest (RF), logistic regression (LR), and multi-layer perceptron (MLP) models revealed that RF achieved the highest performance across all experiment sets, with a receiver operating characteristic (ROC) of 99.9%.

Keywords: internet of things; machine learning; deep learning; network security

Citation: Aljabri, M.; Alahmadi, A.A.; Mohammad, R.M.A.; Alhaidari, F.; Aboulnour, M.; Alomari, D.M.; Mirza, S. Machine Learning-Based Detection for Unauthorized Access to IoT Devices. *J. Sens. Actuator Netw.* **2023**, *12*, 27. https://doi.org/10.3390/jsan12020027

Academic Editor: Jordi Mongay Batalla

Received: 17 February 2023
Revised: 12 March 2023
Accepted: 16 March 2023
Published: 20 March 2023

1. Introduction

In recent decades, the Internet of Things (IoT) concept has become widely popular with numerous fields and organizations implementing and investing in its use. IoT refers to the billions of devices that can connect to the internet, thus sharing and collecting vast amounts of data anywhere in the world. This ability of global devices coupled with communication technologies create a system that connects, exchanges, and analyzes data, resulting in faster and more efficient decision making. With the dawn of readily available inexpensive computer chips, and the omnipresence of wireless networks, it has become possible to transform anything into a part of the IoT [1]. Hence, the number of IoT devices has skyrocketed over the past years. According to Statistica [2], the global number of IoT devices has reached 16.4 billion, and by 2025, it is projected to reach more than 30 billion

devices. However, with the rise in the wide use of IoT devices, vulnerabilities have arisen, leading to the breach of confidentiality and integrity of users and systems.

Most IoT devices perform operations on sensitive user data. Thus, various fundamental challenges in designing a secure IoT exist, such as privacy, access control, authentication, confidentiality, trust, etc. As discussed by Kolias et al. [3], various malware botnets, such as Mirai, can take control and quickly spread, exploiting the vulnerabilities of IoT devices. This especially points out that insecure IoT devices can lead to direct risks to all the interconnecting devices in their network. Further, attackers often gain access to users' data and may cause monetary losses and eavesdropping [4,5]. Particularly, IoT devices are prone to network attacks such as phishing attacks, data thefts, spoofing, and denial of service (DoS) attacks. These attacks can cause other cyber security threats, including serious data breaches and ransomware attacks that can cost organizations a lot of money and effort to recover from [5].

Denial of service (DoS) attacks in such devices have also become pervasive, preventing access to services that the user has paid for [6]. Moreover, DoS attacks can negatively affect the services of small networks such as homes, hospitals, educational institutions, etc. [7,8]. What is more is that such attacks quickly spread, leading to exploiting vulnerabilities of a plethora of IoT devices. In 2021, Forbes reported that the attacks on IoT devices skyrocketed and surpassed 300% [9]. Thus, the need to have robust solutions to counter these attacks and prevent their expansion before it is too late is imminent.

The use of artificial intelligence (AI) techniques, mainly machine learning (ML), has become very useful due to their ability to learn from past experiences and prevent cyberattacks before they spread and affect more and more devices [10]. ML is a field in AI that uses data and algorithms to mimic how humans learn, improving over time with experience. Network security, particularly IoT security, is a very challenging field. Utilizing ML's power can lead to more robust solutions to protect the confidentiality, integrity, and availability of IoT networks and users [11,12]. Thus, several research studies focus on attack detection in IoT environments using AI techniques. Accordingly, this research uses a recent dataset to apply ML techniques for detecting attacks in IoT environments.

The main contributions of this paper are as follows:

1. Study the effect of different sets of features on building ML models for detecting various IoT attacks and investigate the models' performance using features selection techniques;
2. Perform a comparative analysis of binary and multiclass experiments on the dataset to detect and classify IoT attacks;
3. Achieve better benchmark results on the utilized IoT attacks dataset.

The paper is divided as follows: Section 2 reviews related works in the field. Section 3 outlines the research methodology, which describes the dataset used, the preprocessing steps carried out, the models applied, and the performance metrics utilized. Section 4 discusses the experimental setup and the results obtained. Finally, Section 5 presents the conclusion and future work.

2. Related Works

Many researchers worked on detecting cyberattacks in IoT networks as they aimed to offer more security to people and cities that use IoT systems. Most studies included in the literature review used the UNSW-NB15 [13] dataset. Following this ideology, Verma et al. [14] aimed to improve the security of IoT systems against DoS attacks by developing several ML models, namely, AdaBoost (AB), RF, a gradient boosting machine (GBM), extreme gradient boosting (XGB), classification and regression trees (CART), extremely randomized trees (ERT), and multi-layer perceptron (MLP). They used three datasets: CIDDS-001 [15], UNSW-NB15 [13], and NSL-KDD [16]. The results showed that CART achieved the best performance with an accuracy level of 96.74%, while XGB resulted in the best performance at the AUC level of 98.77%. Additionally, Khatib et al. [17] aimed to build an ML model for intrusion detection to enhance the accuracy of IoT networks against malicious attacks. For their experiments, the researchers used the UNSW-NB15 [13]

dataset, which consisted of 49 features and contained 2,540,044 samples, and applied seven ML classifiers, namely RF, decision trees (DT), AdaBoost, logistic regression (LR), linear discriminant analysis (LDA), a support vector machine (SVM), and Nystrom-SVM. In multiclass classification, SVM resulted in the best performance with an accuracy of 93%. While in binary classification, Nystrom-SVM, RF, and DT resulted in the best performance with an accuracy level of 95%.

In another study, Rashid et al. [18] proposed ML models for anomaly detection to enhance the security of IoT in smart cities by using two UNSW-NB15 and CIC-IDS2017, having 175,341 and 190,774 instances, respectively. The study used an information gain ratio to select the best 25 features in each dataset for their experiments, and 10-fold cross-validation was used to train the models: LR, SVM, RF, DT, k-nearest neighbor (KNN), and artificial neural network (ANN). Ensemble techniques such as bagging, boosting, and stacking ensemble were also used. CIC-IDS2017 showed better performance among all the classifiers while stacking ensemble models resulted in the best performance with an accuracy level of 99.9%. Similarly, Alrashdi et al. [19] proposed AD-IoT, an ML model for anomaly detection in IoT networks in smart cities using the UNSW-NB15 dataset, taking 699,934 instances for their experiments. For feature selection, the authors used the extra trees classifier to select the best features to train the ML model, and only 12 features were selected to train the model. An RF classifier was used to build the proposed model. The proposed model was developed for binary classification as normal and attack traffic, resulting in an accuracy of 99.34%.

On the other hand, some studies used different datasets for the same purpose as the previous studies. Gad et al. [20] aimed to improve the security of vehicular ad hoc networks against DoS attacks by developing an ML model for intrusion detection using the used ToN-IoT [21] dataset containing 44 features. Synthetic minority oversampling technique (SMOTE) was applied to handle the class imbalance, and the Chi2 algorithm was used for the features selection process resulting in the 20 best features. This study used five ML algorithms: LR, naive Bayes (NB), DT, SVM, KNN, RF, AB, and XGB. The XGB algorithm resulted in the best performance in accuracy levels of 99.1% and 98.3% in both binary and multiclass classification, respectively. In addition, Verma et al. [22] proposed an ensemble ML model for anomaly detection to enhance the security of IoT networks by detecting zero-day attacks using the CSE-CIC-IDS2018-V2 [23] dataset. The SMOTE oversampling technique was used to handle class imbalance. In addition, the authors used the random under-sampling technique to the benign class to reduce its number of instances, and the random search cross-validation algorithm was used to select the best features. To avoid overfitting, the authors split the dataset using a 70–30 train–test split and trained the proposed model 10 times using different instances in each set. This way combined both 10-fold cross-validation and the hold-out splitting method. The ensemble model presented in this study consisted of the RF and GBM classifiers and resulted in an accuracy level of 98.27%.

Similarly, to assess the systems that automate attack detection in industrial control systems (ICS), Arora et al. [24] focused on evaluating different ML algorithms, namely, RF, SVM, DT, ANN, KNN, and NB. The dataset used in their experiment was the SCADA attacks dataset, containing seven features and the label classifying the data samples as normal or attack. Further, the dataset underwent an 80–20 train–test split, and the evaluation metrics utilized were accuracy, false alarm rate (FAR), UN-detection rate (UND), and the receiver operating characteristic (ROC) curve. The results showed that RF achieved the highest accuracy of 99.84% and the highest UND of 84.7%. In another study, Mothkuri et al. [25] proposed a federated learning (FL)-based approach to anomaly detection to detect intrusions in IoT networks by employing decentralized on-device data. They used gated recurrent units (GRU) and kept the data intact on only the local IoT devices by sharing learned rates with the FL. Further, the ML model's accuracy was optimized by aggregating updates from multiple sources. The Modbus-based network dataset [26] was used for building and evaluating the results, and it contained several attack types such as

man-in-the-middle, distributed denial of service (DDoS), synchronization DDoS, and query flood attacks. The experiment results demonstrated that the FL-based approach achieved a better performance average accuracy of 90.286% in successfully detecting attacks than the non-FL-based approach. Table 1 Summarizes the reviewed papers.

Table 1. Literature review summary.

Ref.	Method	Dataset	Features	Results
[14]	AB, RF, GBM, XGB, CART, ERT, and MLP	CIDDS-001, UNSW-NB15, and NSL-KDD	-	Accuracy = 96.74% using CART
[17]	RF, DT, AdaBoost, LR, LDA, SVM, and Nystrom-SVM	UNSW-NB15, 2,540,044 samples	49 features	Binary classification: Accuracy = 95% using Nystrom-SVM, RF, and DT. Multiclass classification: Accuracy = 93% using SVM.
[18]	LR, SVM, RF, DT, KNN, ANN, bagging, boosting, and stacking ensemble	UNSW-NB15, 175,341 samples, and CIC-IDS2017, 190,774 samples.	25 features	Accuracy = 99.9% using stacking ensemble
[19]	RF	UNSW-NB15, 699,934 samples.	12 features	Accuracy = 99.34%
[20]	LR, NB, DT, SVM, KNN, RF, AB, and XGB	ToN-IoT	20 features	Binary classification: Accuracy = 99.1% Multiclass classification: Accuracy = 98.3%, both using XGB.
[22]	RF, and GBM	CSE-CIC-IDS2018-V2		Accuracy = 98.27%
[24]	RF, SVM, DT, ANN, KNN, and NB	SCADA attacks dataset	7 features	Accuracy = 99.84% using RF
[25]	GRU	Modbus-based network dataset	-	Accuracy = 90.286

Although many studies focused on detecting IoT attacks and achieved high performance, a gap and limitations still need to be resolved. From the literature, there is a need to explore new datasets that contain new attack types. Most of the reviewed papers used common datasets that contain old attacks, but many new attacks are being created in the IoT security field. Moreover, most of the datasets used targeted general network attacks. Using a dataset that targets IoT attacks may improve the detection of these attacks. Most reviewed studies used many features to train their models. Exploring feature selection and extracting the most important features will reduce the impact of cure of dimensionality and the time needed for attack detection.

3. Methodology

The primary purpose of our study is to use ML algorithms to detect and classify IoT network security attacks. The models used include bagging, KNN, J48, RF, LR, and MLP. The models were trained using a publicly available dataset from Wheelus and Zhu [27] to detect and categorize IoT network attacks. The dataset underwent several preprocessing steps to convert it into the most suitable format for training the models. Furthermore, we evaluated the performance of these models based on evaluation parameters, including classification accuracy, F-score, recall, precision, and ROC. Moreover, we implemented 10-fold cross-validation to build the models and performed two experiments for detection—binary classification to distinguish between normal and attack sessions—and two experiments for classification—multiclass classification to categorize normal sessions and three types of attack, namely, no shared secret (NSS), query cache (QC), and zone transfer (ZT). Furthermore, the experiments were conducted using a subset of the features to emphasize the significance of the features selection process while maintaining, if not increasing, the models' performance. Figure 1 demonstrates the research methodology steps.

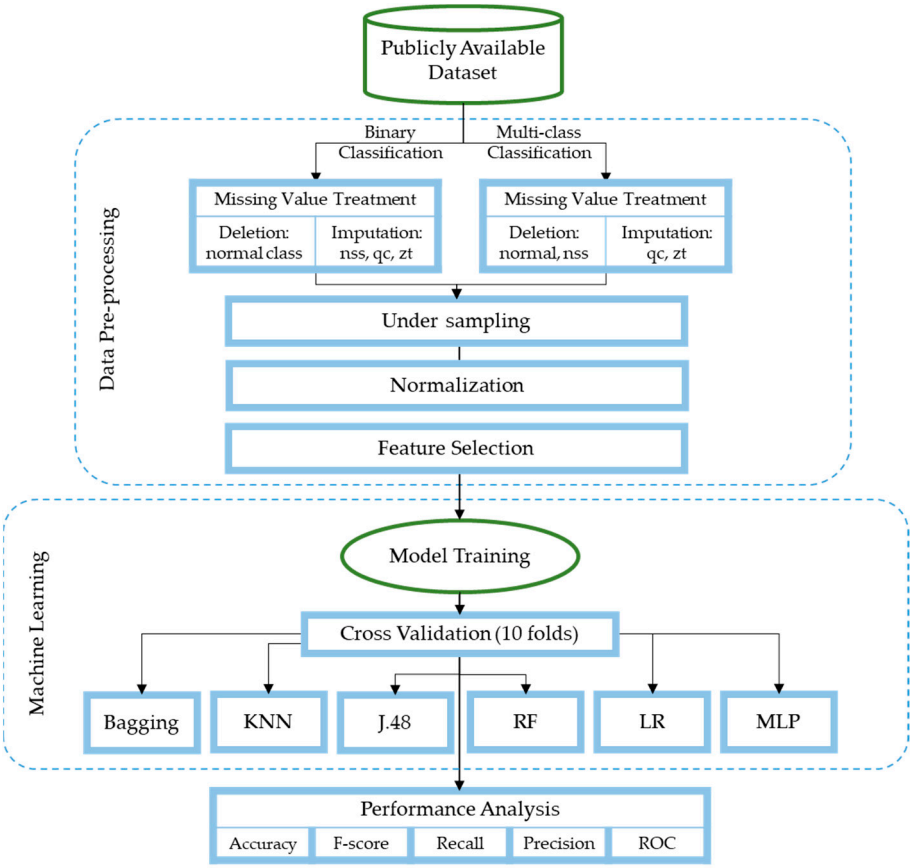


Figure 1. Research methodology steps.

3.1. Dataset Description

The dataset used in this study is a publicly available dataset released by Wheelus and Zhu [27]. The dataset was collected for nine months, and the raw data underwent several preprocessing phases and organizing into sessions based on packet commonalities such as source and destination IP addresses and ports, as well as temporal characteristics. The set of features included in the dataset is depicted in Table 2. For the binary case, the dataset aims to classify samples into attack or normal categories. For multiclass classification purposes, it aims to classify the sample into four categories of unauthorized attacks: normal, query cache (QC), zone transfer (ZT), and no shared secret (NSS). The QC error occurs when an unauthorized request for system data is evidenced by the incorrect sequence of requests made to the remote gate opener (RGO). In a ZT attack, a perpetrator tries to access domain name system (DNS) zone information to scan the IoT system's components. The NSS attack happens when there is an unauthorized attempt to join the IoT network. This attempt is recognized when the shared secret, exchanged during authorization of RGO, is expired or invalid. The binary dataset consists of 212,834 samples, where 178,576 are normal and 34,258 are attacks. On the other hand, in the multiclass dataset, there are four classes: normal, NSS, QC, and ZT, where the number of examples that belong to each class are 178,576, 23,022, 6901, and 4335, respectively.

Table 2. Description of features present in the dataset.

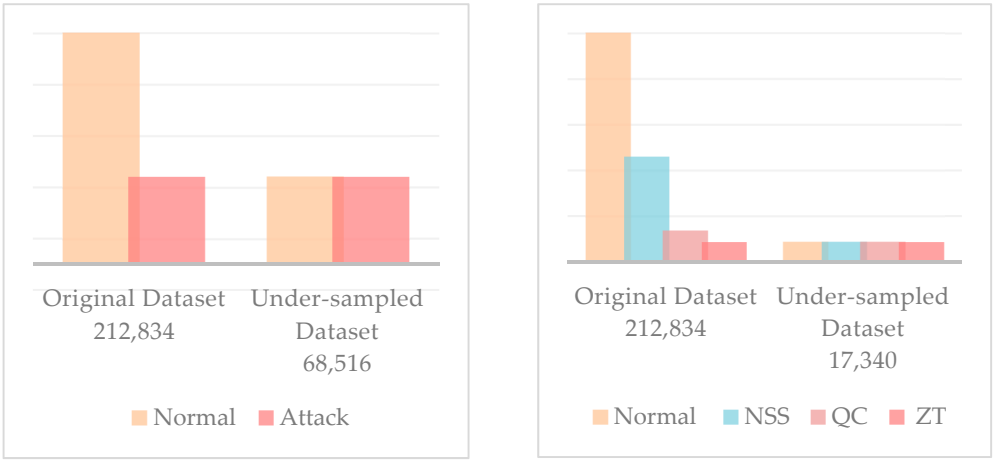
#	Feature Name	Description
1	in_rep	Session repetition—packet count of packets that are of the most common packet size
2	out_rep	
3	in_prdcty	Session periodicity—measure of periodicity in a session, given by the variance of timestamp differences between packets
4	out_prdcty	
5	in_conv	Session convergence—self-similarity of the packets in the session, determined by examining the variance in the size of the packets
6	out_conv	
7	invel_pps	Packets per second—velocity of the traffic measured in packets per second
8	outvel_pps	
9	invel_bps	Bits per second—velocity of the traffic measured in bits per second
10	outvel_bps	
11	invel_bpp	Bytes per packet—velocity of the traffic measured in bytes per packet
12	outvel_bpp	
13	riotp	RIOT packets—ratio of inbound to outbound traffic measured in packets (inbound and outbound combined)
14	riotb	RIOT bytes—ratio of inbound to outbound traffic measured in bytes (inbound and outbound combined)
15	duration	Duration—the total elapsed time of the session (inbound and outbound combined)
16	orig_bytes	Byte count—session traffic size in bytes
17	resp_bytes	
18	orig_packets	Packet count—session traffic size in packets
19	resp_packets	

3.2. Preprocessing

Preprocessing is performed before using the dataset to ensure the data is in a format appropriate for training and testing the models. This step involves loading, cleaning, treating, and converting the data into a suitable format for the intended tasks. The dataset originally contains 212,834 instances, 178,576 of which represent normal traffic. This represents 83.9% of the dataset, indicating the imbalance in the data as this class is substantially higher than the others. Considering that the dataset suffers from imbalance and the presence of missing values in all classes, we worked on treating the missing values differently for each class depending on the best method for each. Because the normal class was significantly higher than the others, the method chosen to treat its missing values was deletion for both experiments. The missing values in the second class representing attacks were imputed using the mean for the binary classification experiments. This was possible as the features included missing values corresponding to variance values. Hence, using the mean was appropriate to impute those missing values. As for the multiclass experiments, the missing values in the NSS class were deleted as doing so would maintain the number of NSS attack instances to still be more than the lowest class. In other words, as illustrated in Table 3, when the missing values of the NSS class are deleted, the number of instances becomes 5597, which is still more than the number of instances of the ZT class—4335. The missing values in the QC and ZT classes were treated by imputation using the mean because the number of instances was already low in those two classes. Table 3 shows how the dataset appears after treating missing values. Moreover, under-sampling was performed due to the imbalance that remained. In addition, the dataset records were also normalized to the range of −1:1. How the final dataset instances appear after all these steps is visualized in Figure 2 for the binary and multiclass classification experiments.

Table 3. Dataset values before and after missing values treatment.

Exterminate	Class	Number of Instances	
		Before Treating the Missing Values	After Treating the Missing Values
Binary classification	Normal	178,576	163,876
	Attack	34,258	34,258
Multiclass classification	Normal	178,576	163,876
	NSS	23,022	5597
	QC	6901	6901
	ZT	4335	4335



(**a**) (**b**)

Figure 2. Randomized under-sampling of the dataset for (**a**) binary classification and (**b**) multiclass classification.

3.3. Feature Selection

The dataset included 20 features depicting relevant information about the traffic sessions. We performed four different experiments to obtain the highest performance. The class attribute takes the values of normal or attack in the case of the binary experiments, and in the case of multiclass classification, the attack values are further classified into three types of attacks: NSS, QC, and ZT. The same set of features is used for all four experiments. Table 4 shows how the dataset appeared after feature selection. As shown in Table 4, the correlation and information gain of all features were calculated. The top 30% of the features were selected. Therefore, the seven features with the highest correlations were selected, and another feature (duration) was added due to its high information gain value. Later, the correlation among the features was calculated, indicating that the riotb and riotp attributes had a 100% correlation. Only one was to be used; otherwise, we would have a redundant feature. According to the higher information gain value of the riotb feature, it was the one we kept. Hence, we had seven features to use in the four experiments.

Table 4. Correlation and information gain values for each of the selected features.

Feature	Correlation	Information Gain
orig_packets	0.6609	0.658
riotp	0.6391	0.82
outvel_bpp	0.6339	0.814
orig_bytes	0.6277	0.87
resp_packets	0.5999	0.623
resp_bytes	0.5275	0.831
duration	0.2827	0.649

3.4. Evaluation Metrics

The aforementioned model is evaluated in terms of *accuracy*, *F1-score*, *recall*, *precision*, and *ROC*. The following are the potential outcomes of attack prediction:

- True positive (TP): TP refers to attack classes that were correctly predicted;
- False positive (FP): FP signifies the normal classes that were incorrectly predicted as attack;
- True negative (TN): TN reflects the normal classes that were correctly predicted;
- False negative (FN): FN refers to attack classes that were incorrectly predicted as normal.

Accuracy indicates the overall rate of correctly identified instances in the test dataset compared with the total number of instances, defined as Equation (1):

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN'} \tag{1}$$

F1 score measures both the precision and recall at the same time, calculated as Equation (2):

$$F1 - \text{score} = 2 \times \frac{P \times R}{P + R} \tag{2}$$

Recall indicates how correctly the model predicts the true positives, calculated as the ratio of the true positives detected to the total actual positive, shown in Equation (3):

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{3}$$

Precision indicates the quality of prediction made, calculated as the ratio of true positives to the total positives (false and true), represented in Equation (4):

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{4}$$

The ROC curve indicates the model's performance at various classification thresholds. It plots two parameters: the true positive rate and false positive rate.

4. Results and Discussion

4.1. Experimental Setup

Different ML models were used to perform the experiments, including RF, LR, KNN, J48, bagging ensemble, and MLP. The rationale behind using these algorithms is the fact that they were successfully used for building classification models in different domains [14,18,20,24]. Further, these algorithms were selected to compare the performance between single-model and ensemble-based classification algorithms. The dataset used in these experiments contains 19 features and a target class divided into normal and attack traffic in the first set of experiments and divided into normal, QC, ZT, and NSS attacks in the second set of experiments. The number of instances was 212,834, but because of the

imbalance problem, the dataset needed to be under-sampled to obtain a different number of instances for each set of experiments. Moreover, two more sets of experiments were performed with seven features that were chosen after performing feature selection based on the correlation with the target class.

4.2. Parameter Settings

Parameter settings are optimized to ensure obtaining the best possible results from the models. After exploring the best range for the most important parameter of each algorithm, different values were tested within these ranges. The best settings used in the binary and multiclass classification experiments are shown in Table 5.

Table 5. Parameter optimization settings.

Model	Parameter	Optimal Value	
		Binary Experiments	**Multiclass Experiments**
J48	Binary split	False	False
	Confidence factor	0.25	0.2
LR	Maxlts	−1	−1
	Ridge	1.0×10^{-8}	1.0×10^{-8}
RF	Iterations	100	50
	Batch Size	100	100
	Features	0	0
KNN	K value	3	1
	Distance	Euclidean	Euclidean
Bagging	Classifier	REPTree	
	Iterations	50	50
	Bag size percent	100	100
MLP	Hidden layers	3	3
	Activation function	ReLU	ReLU
	Optimizer	Adam	Adam
	Epochs	60	60

4.3. Experimental Results

4.3.1. Binary Class Results

The first set of experiments was performed using a binary class of normal and attack traffic with 19 features. The classifiers used for this set of experiments are RF, LR, KNN, J48, bagging ensemble, and MLP. Table 6 shows binary experiment results before feature selection.

Table 6. Binary experiment results before feature selection.

Model	Precision	Recall	F1-Score	Accuracy	ROC
KNN	0.932	0.924	0.924	92.4%	0.924
LR	0.984	0.984	0.984	98.38%	0.994
J84	0.994	0.994	0.994	99.43%	0.996
RF	**0.996**	**0.996**	**0.996**	**99.59%**	**0.999**
Bagging	0.995	0.995	0.995	99.46%	0.999
MLP	0.992	0.992	0.992	99.16%	0.993

As shown in Table 6, RF resulted in the best performance in all the evaluation metrics. The bagging ensemble model resulted in a ROC area of 0.999. KNN resulted in the lowest performance among all the classifiers, with an accuracy level of 92.4%. The nature of the RF classifier is the reason behind its performance. As RF is an ensemble of decision trees, it simultaneously trains multiple trees and then uses all the trees to make the final prediction. In addition, it applies feature selection during its learning process, and each of the trees could use a different set of features. Therefore, by combining all the learned trees, we obtain a very powerful algorithm with an enhanced prediction performance. The results of the bagging ensemble support the discussion before, as the strength of ensemble techniques is important to predicting the type of traffic. The reason behind overperforming the RF model to the bagging ensemble is the nature of the classifiers included in the bagging model. The nature of tree classifiers is more suitable to the prediction class of this study, as RF overperformed the bagging ensemble, and the J48 model overperformed other single classifiers. As mentioned earlier, the feature selection included in training the tree classifiers could be the reason why they select the more suitable features for detecting the attack traffic from normal traffic.

After performing feature selection, more experiments were performed using the best set of features as mentioned in Section 3.3. Table 7 shows the results after the feature selection.

Table 7. Binary experiment results after features selection.

Model	Precision	Recall	F1-Score	Accuracy	ROC
KNN	0.993	0.993	0.993	99.28%	0.997
LR	0.978	0.978	0.978	97.83%	0.993
J84	0.993	0.993	0.993	99.31%	0.995
RF	**0.995**	**0.995**	**0.995**	**99.59%**	**0.999**
Bagging	0.993	0.993	0.993	99.34%	0.999
MLP	0.991	0.991	0.991	98.94%	0.992

As shown in Table 7, RF resulted in the best performance in all the evaluation metrics. As before, the bagging ensemble yielded the same ROC area of 0.999. The performance of the KNN model was significantly improved with an accuracy of 99.28% and beat the LR model which resulted in an accuracy of 97.83%, the lowest performance among all the models. The possible reason behind the improvement of the KNN performance could be that KNN is suited for lower dimensional data. In other words, the KNN model has benefited from feature selection which reduced the dimensionality of the input feature.

4.3.2. Multiclass Results

The second set of experiments was performed using a multiclass of normal and ZT, QC, and NSS attacks with 19 features. The classifiers used for this set of experiments are RF, LR, KNN, J48, bagging ensemble, and MLP. Table 8 shows the multiclass experiment results before feature selection.

Table 8. Multiclass experiment results before feature selection.

Model	Precision	Recall	F1-Score	Accuracy	ROC
KNN	0.875	0.795	0.799	79.48%	0.864
LR	0.949	0.948	0.948	94.83%	0.991
J84	0.981	0.981	0.981	98.08%	0.991
RF	**0.989**	**0.989**	**0.989**	**98.87%**	**0.999**
Bagging	0.982	0.982	0.982	98.22%	0.999
MLP	0.973	0.973	0.973	96.58%	0.992

As shown in Table 8, RF resulted in the best performance in all the evaluation metrics used. The bagging ensemble model resulted in a ROC area of 0.999. KNN resulted in the lowest performance among all the classifiers with an accuracy level of 79.48%. As mentioned in Section 4.3.1, the strength of ensemble trees and the feature selection included in their training could be the reason for resulting in the best performance among all the classifiers.

Another set of experiments was performed using the best seven features. Table 9 below shows the results after the features selection.

Table 9. Multiclass experiment results after features selection.

Model	Precision	Recall	F1-Score	Accuracy	ROC
KNN	0.979	0.979	0.979	97.88%	0.990
LR	0.904	0.905	0.904	90.46%	0.982
J84	0.981	0.981	0.981	98.06%	0.993
RF	**0.987**	**0.987**	**0.987**	**98.67%**	**0.999**
Bagging	0.981	0.981	0.981	98.06%	0.998
MLP	0.973	0.973	0.973	96.58%	0.990

As shown in Table 9, RF resulted in the best performance in all the evaluation metrics. The performance of the KNN model was significantly improved to result in an accuracy of 97.88% and beat the LR model, which resulted in an accuracy of 90.46%, the lowest performance among all the models.

4.4. Discussion

The experimental results showed that the performance of the models in binary classification is better than their performance in multiclass classification. This means the models can differentiate between normal and attack traffic but face some difficulty distinguishing between the attack types. The reason for that can be the similarity of the features of the attack traffic, which made it hard for the classifiers to find unique features for each attack. In comparison, the highest level of the ROC area was the same in all binary and multiclass experiments, with a level of 0.999. Because RF combines the strength of the decision trees, the ensemble techniques, and the feature selection, it achieved the highest performance among all the experiments. The bagging ensemble was the second model in terms of performance. This shows that ensemble models resulted in better performance compared with single classifiers. Moreover, the performance of all models was slightly decreased after applying feature selection, except in the KNN model, as its performance was significantly improved after feature selection. Tables 10 and 11 show the confusion matrix of the RF model in binary and multiclass classifications.

Table 10. Confusion matrix for the binary class case.

		Prediction	
		Normal	**Attack**
Actual	Normal	34,299	29
	Attack	237	34,021

Table 11. Confusion matrix for the multiclass case.

		Prediction			
		Normal	**ZT**	**NSS**	**QC**
Actual	Normal	4301	2	30	2
	ZT	0	4333	2	0
	NSS	125	1	4208	1
	QC	16	0	16	4303

4.5. Comparison with Benchmark Study

This study used different ML models to create models using a well-known dataset for detecting attacks on IoT devices. The dataset that was used to train the models consists of 212,834 samples and 19 features. The target class is divided into binary classes as normal and attack traffic, and multiclass as normal, NSS, QC, and ZT attacks. To handle the missing values, both removing missing values and imputing them using the mean value were used depending on the number of samples in each class. The dataset suffered from an imbalance problem, and the randomized under-sampling technique was used to balance the data. In the end, feature selection was performed to use the best set of seven features and obtain the same accuracy as when using the whole set of features. It was compared with a benchmark study to compare the proposed model with the previous studies.

The benchmark study [27] compared with this study used the same dataset but with multiclass only. Moreover, the benchmark study used four ML algorithms: naive Bayes, J48, LR, RF, and the MLP algorithm. All the samples with missing values were dropped, and the random oversampling technique was used to solve the imbalance problem. Moreover, the entire set of 19 features was used to build the model. In addition, the binary classification of the dataset was experimented with for the first time in this study. The preprocessing techniques used in the proposed study have a significant impact on retaining improved results compared with the benchmark study. In addition, combining the imputing and removing methods in handling missing values helped save data of minority classes from being lost. Also, under-sampling techniques could be a better option to avoid overfitting caused by random oversampling. Lastly, experimenting with feature selection and reducing the number of the needed features to seven instead of 19 is very important in reducing the time required for both training ML models and using them for classification. In the benchmark study, the RF model achieved the best performance with a ROC area of 0.976. Although the same dataset is used in this study and the previous study, the model proposed resulted in better performance in all evaluation metrics. Table 12 shows the comparison in detail.

Table 12. Comparison with benchmark study.

Study	Model	Precision	Recall	F1-Score	Accuracy	ROC
IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework	RF	0.905	0.894	0.891	-	0.976
Proposed tudy (multiclass)	RF	**0.989**	**0.989**	**0.989**	**98.87%**	**0.999**
Proposed study (binary-class)	RF	**0.996**	**0.996**	**0.996**	**99.59%**	**0.999**

5. Conclusions

With the growing number of IoT devices, the number of cyber threats in IoT networks has drastically surged. This imminent situation requires instant action as most devices

share and process sensitive data. Therefore, AI methods have been widely adopted to counter these threats due to their robustness and efficiency. Many researchers focused on detecting various attacks in IoT by building ML classifiers. In this study, we have focused on features engineering and building ML models using a new dataset as it is necessary to explore new cybersecurity datasets due to the changing nature of cyber threats.

We performed two classes of experiments: one for binary classification into normal and attack class, and another for multiclass classification into normal, QC (query cache), ZT (zone transfer), and NSS (no shared secret) classes. In both cases, the dataset underwent randomized under-sampling to obtain an equal number of classes, followed by normalization, and feature selection using correlation and information gain. Then, for each case, two sets of experiments were performed. One uses all the features, and another uses the best features only. The 10-fold cross-validation technique was applied to the dataset and the models applied were bagging, KNN, J48, RF, LR, and MLP. We evaluated their performance in terms of accuracy, F-score, recall, precision, and ROC. The results of the experiments showed that RF achieved the highest performance in all the experiment sets, obtaining a ROC of 99.9%. Furthermore, binary classification experiments gave better results than multiclass. Additionally, feature selection had little effect in both the experiment sets as most of the classifier performance remained the same except for the case of KNN, where its performance significantly increased. As for future work, we suggest building real-time models capable of identifying attacks in IoT devices and classifying them in real-time to stop malicious activity before it leaks or destroys sensitive data. Furthermore, this dataset can be used as an inspiration to build our dataset and generate more attack types to try our ML models on it.

Author Contributions: Conceptualization, M.A. (Malak Aljabri); methodology, M.A. (Malak Aljabri), A.A.A., R.M.A.M., F.A., M.A. (Menna Aboulnour), D.M.A. and S.M.; software, M.A. (Menna Aboulnour), D.M.A. and S.M.; validation, M.A.(Malak Aljabri), A.A.A., R.M.A.M., F.A., M.A. (Menna Aboulnour), D.M.A. and S.M.; formal analysis, M.A. (Malak Aljabri), A.A.A., R.M.A.M., F.A., M.A. (Menna Aboulnour), D.M.A. and S.M.; investigation, M.A. (Malak Aljabri), A.A.A., R.M.A.M., F.A., M.A. (Menna Aboulnour), D.M.A. and S.M.; resources, M.A. (Menna Aboulnour), D.M.A. and S.M.; data curation, M.A. (Menna Aboulnour), D.M.A. and S.M.; writing—original draft preparation, M.A. (Menna Aboulnour), D.M.A. and S.M.; writing—review and editing, M.A. (Malak Aljabri), A.A.A., R.M.A.M., F.A., M.A., D.M.A. and S.M.; supervision, M.A. (Malak Aljabri), A.A.A., R.M.A.M. and F.A.; project administration, M.A. (Malak Aljabri), A.A.A., R.M.A.M. and F.A.; funding acquisition, M.A.(Malak Aljabri) and A.A.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.kaggle.com/datasets/charleswheelus/iotdatadrivendefense (accessed on 4 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aljabri, M.; Zagrouba, R.; Shaahid, A.; Alnasser, F.; Saleh, A.; Alomari, D.M. Machine learning-based social media bot detection: A comprehensive literature review. *Soc. Netw. Anal. Min.* **2023**, *13*, 20. [CrossRef]
2. Global IoT and Non-IoT Connections 2010–2025 | Statista. Available online: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/ (accessed on 21 February 2022).
3. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer Long. Beach. Calif.* **2017**, *50*, 80–84. [CrossRef]

4. Aljabri, M.; Alhaidari, F.; Mohammad, R.M.A.; Mirza, U.S.; Alhamed, D.H.; Altamimi, H.S.; Chrouf, S.M.B. An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–14. [CrossRef] [PubMed]
5. Aljabri, M.; Aldossary, M.; Al-Homeed, N.; Alhetelah, B.; Althubiany, M.; Alotaibi, O.; Alsaqer, S. Testing and Exploiting Tools to Improve OWASP Top Ten Security Vulnerabilities Detection. In Proceedings of the 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), Al-Khobar, Saudi Arabia, 4–6 December 2022; pp. 797–803.
6. Das, R.; Tuna, A.; Demirel, S.; Yurdakul, M.K. A Survey on the Internet of Things Solutions for the Elderly and Disabled: Applications, Prospects, and Challenges. *Int. J. Comput. Netw. Appl.* **2017**, *4*, 84–92. [CrossRef]
7. Aljabri, M.; Alahmadi, A.A.; Mohammad, R.M.A.; Aboulnour, M.; Alomari, D.M.; Almotiri, S.H. Classification of Firewall Log Data Using Multiclass Machine Learning Models. *Electron* **2022**, *11*, 1851. [CrossRef]
8. Aljabri, M.; Mirza, S. Phishing Attacks Detection using Machine Learning and Deep Learning Models. In Proceedings of the 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 1–3 March 2022; pp. 175–180.
9. MORE Alarming Cybersecurity Stats For 2021 ! Available online: https://www.forbes.com/sites/chuckbrooks/2021/10/24/more-alarming-cybersecurity-stats-for-2021-/?sh=4a9c31b24a36 (accessed on 21 February 2022).
10. Aljabri, M.; Aljameel, S.S.; Mohammad, R.M.A.; Almotiri, S.H.; Mirza, S.; Anis, F.M.; Aboulnour, M.; Alomari, D.M.; Alhamed, D.H.; Altamimi, H.S. Intelligent Techniques for Detecting Network Attacks: Review and Research Directions. *Sensors* **2021**, *21*, 7070. [CrossRef] [PubMed]
11. Aljabri, M.; Altamimi, H.S.; Albelali, S.A.; Al-Harbi, M.; Alhuraib, H.T.; Alotaibi, N.K.; Alahmadi, A.A.; Alhaidari, F.; Mohammad, R.M.A.; Salah, K. Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions. *IEEE Access* **2022**, *10*, 121395–121417. [CrossRef]
12. Alzahrani, R.A.; Aljabri, M. AI-Based Techniques for Ad Click Fraud Detection and Prevention: Review and Research Directions. *J. Sens. Actuator Netw.* **2023**, *12*, 4. [CrossRef]
13. The UNSW-NB15 Dataset | UNSW Research. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 19 February 2022).
14. Verma, A.; Ranga, V. Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wirel. Pers. Commun.* **2020**, *111*, 2287–2310. [CrossRef]
15. CIDDS—Coburg Intrusion Detection Data Sets: Hochschule Coburg. Available online: https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html (accessed on 19 February 2022).
16. Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/index.html (accessed on 19 February 2022).
17. Khatib, A.; Hamlich, M.; Hamad, D. Machine Learning based Intrusion Detection for Cyber-Security in IoT Networks. *E3S Web Conf.* **2021**, *297*, 01057. [CrossRef]
18. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks detection in iot-based smart city applications using machine learning techniques. *Int. J. Environ. Res. Public Health* **2020**, *17*, 9347. [CrossRef] [PubMed]
19. Alrashdi, I.; Alqazzaz, A.; Aloufi, E.; Alharthi, R.; Zohdy, M.; Ming, H. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, Las Vegas, NV, USA, 7–9 January 2019; pp. 305–310.
20. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
21. The TON_IoT Datasets | UNSW Research. Available online: https://research.unsw.edu.au/projects/toniot-datasets (accessed on 19 February 2022).
22. Verma, P.; Dumka, A.; Singh, R.; Ashok, A.; Gehlot, A.; Malik, P.K.; Gaba, G.S.; Hedabou, M. A Novel Intrusion Detection Approach Using Machine Learning Ensemble for IoT Environments. *Appl. Sci.* **2021**, *11*, 10268. [CrossRef]
23. IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 21 February 2022).
24. Arora, P.; Kaur, B.; Teixeira, M.A. Evaluation of Machine Learning Algorithms Used on Attacks Detection in Industrial Control Systems. *J. Inst. Eng. India Ser. B* **2021**, *102*, 605–616. [CrossRef]
25. Mothukuri, V.; Khare, P.; Parizi, R.M.; Pouriyeh, S.; Dehghantanha, A.; Srivastava, G. Federated Learning-based Anomaly Detection for IoT Security Attacks. *IEEE Internet Things J.* **2022**, *9*, 2545–2554. [CrossRef]
26. Frazão, I.; Abreu, P.H.; Cruz, T.; Araújo, H.; Simões, P. Denial of Service Attacks: Detecting the Frailties of Machine Learning Algorithms in the Classification Process. *Lect. Notes Comput. Sci.* **2018**, *11260*, 230–235.
27. Wheelus, C.; Zhu, X. IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework. *IoT* **2020**, *1*, 259–285. [CrossRef]