applied sciences

MDPI

Article

An Intrusion Detection System Using BoT-IoT

Shema Alosaimi and Saad M. Almutairi *

Faculty of Computers and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
* Correspondence: s.almutairi@ut.edu.sa

Abstract: The rapid growth of the Internet of Things (IoT) has led to an increased automation and interconnectivity of devices without requiring user intervention, thereby enhancing the quality of our lives. However, the security of IoT devices is a significant concern as they are vulnerable to cyber-attacks, which can cause severe damage if not detected and resolved in time. To address this challenge, this study proposes a novel approach using a combination of deep learning and three-level algorithms to detect attacks in IoT networks quickly and accurately. The Bot-IoT dataset is used to evaluate the proposed approach, and the results show significant improvements in detection performance compared to existing methods. The proposed approach can also be extended to enhance the security of other IoT applications, making it a promising contribution to the field of IoT security.

Keywords: internet of things; network anomaly detection; IDS; cyberattacks; machine learning; BoT-IoT dataset

1. Introduction

In the age of the IoT, our world is witnessing unprecedented levels of convenience and efficiency. However, the self-configuring and open nature of the IoT renders it vulnerable to a wide range of attacks. IoT devices often need more manual controls and have limited memory and computational power resources. Despite these limitations, the IoT's high dependence and rapid growth have led to increased security risks, making network security solutions crucial. While detecting some attacks can be challenging, some systems currently do an excellent job [1,2]. The volume of information transmitted across networks is growing quickly, leading to an increase in the number of attacks on networks. This has made it crucial to develop quick and effective ways to detect attacks and reduce the risks associated with the widespread adoption of IoT technology. Denial of Service (DoS) is one of the most damaging attacks, as it prevents legitimate users from accessing services. DoS attacks can have severe consequences for critical applications such as healthcare, leading to fatal delays in medical services. In 2016, the Mirai botnet was launched, which hacked into CCTV cameras using default user IDs and credentials to initiate DDoS attacks on DNS servers, bringing internet access in some parts of the US to a standstill. Another botnet, Mozi, capable of launching various DDoS attacks, was identified in April 2020. The architecture of IoT networks is shown in Figure 1 [3]. There is a need for more innovative approaches to strengthen network security and deliver embedded intelligence in an environment involving the IoT. Intrusion Detection Systems (IDS) monitor the host or network for security breaches and notify the administrator when they are detected. Entry events are entered through sensors into a database and employ a set of criteria to generate alerts for security incidents that have occurred. However, stealth systems are still in the early stages of research, and several issues need to be addressed to achieve such high accuracy and low false alarm rates. Signature, anomaly, and specification are the three types of IDS classified based on their detection methods. Signature-IDS compares network traffic patterns to the previously stored patterns of attack.

Citation: Alosaimi, S.; Almutairi, S.M. An Intrusion Detection System Using BoT-IoT. *Appl. Sci.* **2023**, *13*, 5427. https://doi.org/10.3390/app13095427

Academic Editor: Dimitris Mourtzis

Received: 16 March 2023
Revised: 19 April 2023
Accepted: 19 April 2023
Published: 26 April 2023

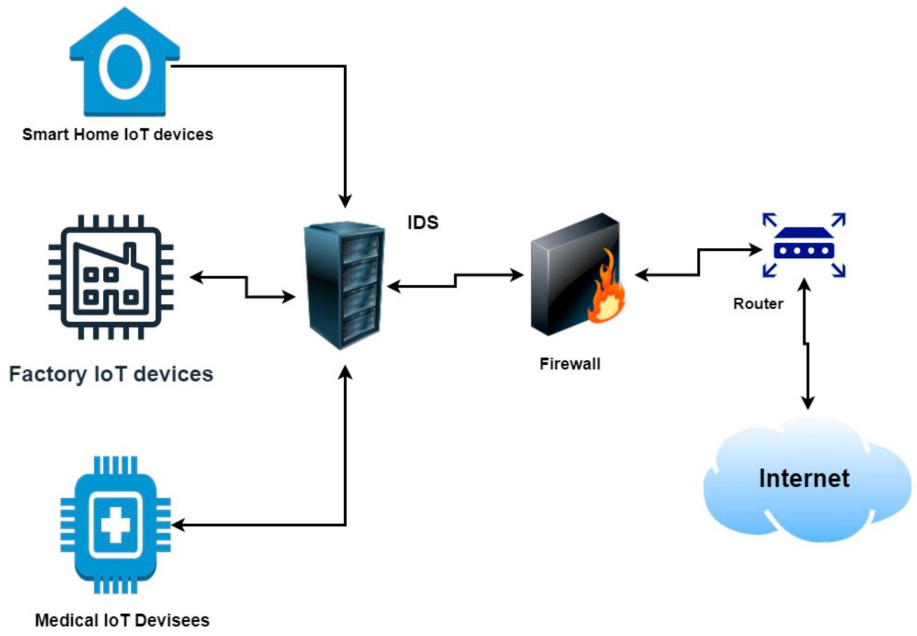

Figure 1. Network Architecture for IoT.

An alert is triggered if a match is discovered. Signature-IDS has high accuracy and a minimum number of false alarms, but it cannot identify new assaults. A specification-based IDS network compares traffic behaviour to the present rule set and values to detect malicious activity. A security expert determines these standards manually [4–6]. IoT devices produce a large volume of information, and conventional data gathering, processing and storage techniques might be unable to handle it. The data heterogeneity the IoT creates is causing issues for current data processing systems. In order to be able to predict, and evaluate the enormous amounts of data, new mechanisms must be developed to handle this overwhelming extent of information. Thus, ML is one of the best computational models for providing intelligence to IoT devices. ML could support systems and intelligent devices in understanding data that machines or humans generate. ML has the capability of automating behaviour for smart devices, based on the knowledge that this is the essential role of IoT solutions [7].

This work is aimed towards developing an ML-based IDS model for IoT applications. In this work, to improve the detection of attacks in IoT networks, the Bot-IoT data set was used to assess the discovery approaches which integrate the original and simulated network traffics of IoT with various attacks. Two databases are derived, and the second database is downsized. For the third database, the problem of imbalance was addressed. Five ML algorithms were applied in the implementation process, and they obtained high performance scores. We used ML algorithms like decision trees, ensemble bag, K-nearest neighbor, linear discriminant, and support vector machine. Essential differences in classifiers are evaluated regarding outstanding metrics accuracy, error rate, recall, specificity, precision, and f-measure. IoT studies using the Bot-IoT data set are still rare in the literature. This work with this data set is an essential contribution to the literature, building an artificial intelligence system based on ML to protect IoT networks and discover attacks against IoT networks, the most famous of which is a DoS attack. This research contributes the following:

- Building an AI system based on three-level algorithms.
- Raising the accuracy and efficiency of our system.
- Discovering attacks against IoT networks, the most famous of which is the DoS attack.

2. Related Works

Several scientific articles have been published on IDS that used data mining and ML techniques. However, primarily, these previous works have only utilized ML approaches to detect intrusions in conventional networks. Hence, in this paper, the field of ML was explicitly used to discover intrusions in the field of the IoT. The applications of ML approaches in the domain of IoT are yet in the initial phases of development, particularly in the IoT security domain, since it has a remarkable ability to identify ideas from IoT information. In the IoT network, ML applications such as flaw detection, pattern recognition, and behavioural analysis could be utilized to identify acute attacks and prevent abnormal activities. For analyzing recent research on detecting attacks utilizing ML in IoT networks, several studies have been analyzed and addressed, as shown in Table 1 [8].

The work in [9] created novel hybrid identifiers to detect DDoS attacks in IoT networks by selecting features of six critical objectives to reduce data from the data set. The shortened data output from the previous step is fed as input data to the model based on the deep learning algorithm. The research in [10] devised a fast and efficient Artificial Neural Network (ANN)-based threat detection mechanism to identify a wide range of attacks on IoT devices and data. The dataset is distributed into sections to validate the structure. ANN technology is implemented in the IoT console to classify malicious packets in the event of an attack. Three layers of input, hidden, and output neural networks were used. The work in [11] used a hierarchical system that detects intrusion through three classifications. The model is made up of three classifiers, the first of which takes the data set's distinct attributes as the input, the second takes distinct data set attributes as the input, and the third input includes all of the raw data set's features, as well as the first and second classifier outputs. Another approach is RDTIDS integration for IoT networks [2]. The extraction of new dataset features can help classifiers improve their prediction skills. The raw data collection was pre-processed to detect anomalies, and then flow-based features were retrieved. The calculations were carried out using two approaches. First, significance weights for each assault type were determined independently. Second, all attacks were gathered into one group, and significance weights for this group were determined. The research in [12] created successful strategies for IoT security and detection of DoS attacks using deep ML algorithms integrating the evaluation of RF, CNN, and MLP algorithms. Hash chains provide a realistic threat model for IoT devices and a secure mechanism for storing and relocating device records. E-Spion uses system information to create three-layer core profiles with varied overheads for IoT devices and detects intrusions based on anomalous behaviour [13].

Nimbalkar et al. introduced a method to select features for intrusion detection systems (IDSs) that identify DoS and DDoS attacks. The proposed approach involves insertion and union operations on subsets of the top 50% of IG and GR features. The approach's effectiveness is evaluated on two datasets, IoT-BoT and KDD Cup 1999, with a JRip classifier. The proposed approach outperforms the original feature set and traditional IDSs, requiring only 16 and 19 features, respectively [14].

Saba et al. developed a deep learning-based approach to enhance IoT security using a CNN-based, anomaly-based intrusion detection system (IDS). The proposed approach can efficiently examine whole traffic across the IoT and detect any possible intrusions and abnormal traffic behaviour. The approach is evaluated using the BoT-IoT datasets, achieving 92.85% [15]. Li et al. provide an ML-based IDS technique that makes use of ensemble trees, comprising DT and RF classifiers, to improve the effectiveness of IDS's attack detection and to explain the predictions made by the ML model. Through the use of the net flow meter feature set, the method is assessed using the NF-BoT-IoT-v2, NF-ToN-IoT-v2, and IoTDS20 datasets [16].

Kumar et al. propose a distributed IDS using fog computing to detect DDoS attacks against mining pools in blockchain-enabled IoT networks. The proposed model is evaluated using Random Forests and an optimized gradient tree boosting system on distributed fog nodes and is tested on the BoT-IoT dataset. Results show that XGBoost outperforms binary

attack detection, while the Random Forest outperforms multi-attack detection. Additionally, the Random Forest takes less time for training and testing on distributed fog nodes than XGBoost [17].

Shareena et al. present a deep-learning-based intrusion detection system for IoT DDoS botnet attacks using a dataset created in a realistic network environment. A highly extensible Deep Neural Network (DNN) is developed and evaluated for headstrong detection of IoT botnet attacks. The results show that the proposed DNN outperforms existing systems with high accuracy and precision, demonstrating its potential for effectively detecting IoT DDoS botnet attacks [18]. Alghanam et al. present an improved PIO feature selection model for intrusion detection. The algorithm uses ensemble learning for detection. The dataset and architecture used in these modes is not complex [19]. Syed et al. discuss an RNN-based model for an intrusion detection system in IoT networks. In comparison to models trained on the full feature set, the models generated on the smaller dataset had higher recall rates without losing the capacity to distinguish between classes [20]. In the past, few researchers applied deep learning algorithms for intrusion detection systems. Many intrusion detection systems currently in use do not take into account the issue of dataset imbalance or model maintenance [21,22]. This can result in high levels of bias, as well as high false-positive and false-negative rates, which in turn can lead to security breaches. It is important to address these issues in order to improve the accuracy and effectiveness of intrusion detection systems, and ultimately reduce the risk of security breaches [23–25]. Mohy-eddine et al. developed an intrusion detection model specifically for IoT environments, utilizing a K-Nearest Neighbors (K-NN) classifier and feature selection techniques. This model involves constructing a NIDS (Network Intrusion Detection System) based on the K-NN algorithm, with the aim of improving the accuracy. PCA is used for the feature selection [26]. Thakkar et al. discuss an approach to addressing the issue of class imbalance in machine learning by utilizing ensemble learning techniques. Specifically, they employed the Bagging classifier, which utilizes a Deep Neural Network (DNN) as its base estimator. In the proposed approach, class weights were introduced during the training process of the DNN, with the aims of creating balanced training subsets for the model [27]. This allows for improving the accuracy and reliability of the model, by ensuring that it is trained on a more representative and balanced dataset, and can better handle the complexities associated with class imbalance in machine learning [28].

Machine learning (ML) and deep learning (DL) algorithms can significantly enhance the effectiveness of intrusion detection systems (IDSs) as a security tool [29]. Douiba et propose an improved IDS that utilizes gradient boosting (GB) and decision tree (DT) algorithms, implemented through the open-source Catboost, for IoT Security. The approach aims to improve the accuracy and reliability of the IDS by incorporating these advanced machine learning techniques, which are designed to better handle the complexities and challenges associated with IoT security [30]. The performance of IDS can enhance and improve the detection of potential threats and vulnerabilities in IoT environments [31]. Table 1 presents a summary of related work.

The reviewed literature highlights the significance and potential of using ML and DL algorithms to improve the accuracy and reliability of IDSs and effectively detect potential threats and vulnerabilities in IoT environments. However, as the existing methods fail to address real-time attacks, we have developed three-level algorithms for Intrusion Detection Systems (IDS) which involve using signature-based detection, anomaly detection, and stateful protocol analysis to detect and classify network intrusions. These levels help improve the accuracy of the intrusion detection system, reduce false positives and negatives, and enhance the security of IoT networks. Signature-based detection identifies known attack patterns or signatures from network traffic, anomaly detection identifies anomalous behavior, and stateful protocol analysis analyzes the protocol state of network traffic to detect any malicious behavior that violates protocol specifications.

Table 1. Analysis of Related Works.

Ref.	Approach	Algorithm	Dataset	Sample	Performance
[8]	ML	KNN RF DT	The authors collected this sample	83 Devices	KNN = 87% RF = 97% DT = 95%
[9]	Deep learning	Neural Network LSTM	CISIDS 2017	225,742 instances	99.03%
[10]	ML	ANN	UNSW-15	175,341 records	84%
[11]	ML	REP Tree JRip Forest PA	CICIDS 2017 and BoT-IoT	40,000 and 5,877,647	96.6% and 96.9%
[2]	ML	NB QDA RF ID3 Adaboost MLP KNN	Bot-IoT	84 network traffic	NB = 79% QDA = 87% RF =97% ID3 = 97% Adaboost = 97% MLP = 84% KNN = 99%
[12]	Machine/deep learning	CNN RF MLP	Bot-IoT	3,264,128	RF & CNN = 90% RF & MLP = 54%
[13]	ML	3 Modules PWM PBM SBM	The authors collected this sample	3973	78% 97% 99% for 3 layers
[15]	Deep learning	CNN	BoT-IoT datasets	3,264,128	92.85%
[17]	Machine learning	Random Forest and XGbooest	BoT-IoT datasets	3,264,128	99%,99%
[19]	Machine learning	Ensemble learning	BoT-IoT datasets	3,264,128	97.37
[20]	Deep learning	RNN	BoT-IoT dataset	3,264,128	99
[26]	Machine learning	KNN	BoT-IoT Dataset	3,264,128	99.99
[27]	Machine learning	Ensemble Learning-based DNN	BoT-IoT Dataset	3,264,128	98.99
[30]	Machine learning	Gradient boosting (GB) and decision tree	BoT-IoT Dataset	3,264,128	99.9%

3. Proposed Approach

The system generally consists of three main components: sensors, a recorder, and an AI-based ID system. IoT networks are devices connected to exchange messages, and these messages are vulnerable to penetration in many forms and at many levels. We have a recorder to capture signals. Any captured signal will be sent to our AI-based system so that we decide whether it is an intrusion on the network by several steps in ML and training the classifiers to make an accurate decision about whether to send it to the IoT system. This is illustrated with our architecture for the system in Figure 2.

The recorder is a critical component of the system as it captures all signals and all traffic on the network, including those that may be malicious or unauthorized. The recorder can store these data for later analysis, allowing the system to detect intrusions in real time or retrospectively. The AI-based ID system is the brain of the intrusion detection system, responsible for analyzing the data collected by the sensors and recorder to identify potential threats. The AI-based ID system uses various machine learning techniques such as deep learning, anomaly detection, and signature-based detection to identify patterns in the traffic and determine whether it takes the form of legitimate communication or an intrusion.

The system may use supervised learning algorithms, where it is trained using labeled data to identify different types of threats, or unsupervised learning algorithms, where it learns to identify anomalous behavior based on deviations from the network's expected behavior. The system uses classifiers to make decisions about the traffic on the network. These classifiers can be trained to identify specific types of attacks, such as denial of service attacks or malware infections, and can be continually updated as new threats emerge. Once the system identifies a potential threat, it can take various actions to prevent the attack or minimize its impact. For example, it can isolate the affected device or block the malicious traffic. Overall, an IoT intrusion detection system is a crucial component in securing IoT networks against cyber-attacks. By monitoring network traffic and using machine learning algorithms to identify potential threats, the system can detect and respond to intrusions quickly, minimizing the risk of damage to IoT devices and infrastructure. This research proposes a working methodology based on six basic steps, as shown in Figure 3 below: selecting the dataset, pre-processing, and dividing the data set into two parts, training and testing. Then we will place a data label at each of the system's three levels and train the classifiers to obtain the results.

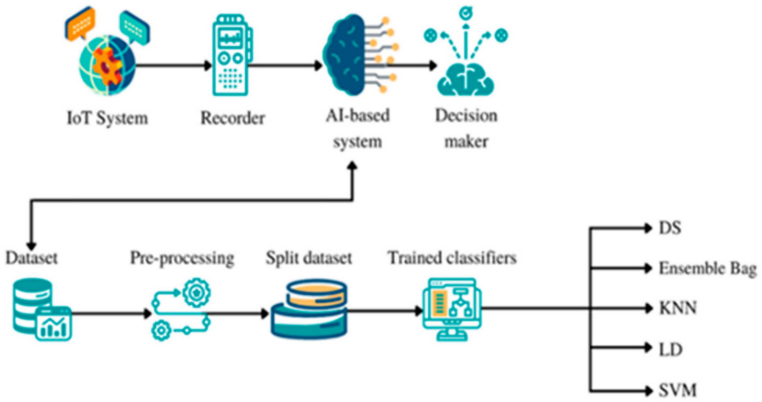

Figure 2. System Architecture.

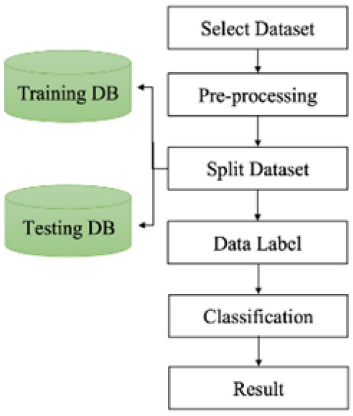

Figure 3. Proposed Approach.

3.1. Datasets

This study uses BoT-IoT as the data set to work on. The BoT-IoT dataset was generated in the Cyber Range Lab at the UNSW Canberra Cyber Center. This dataset simulates a realistic network environment integrating normal and botnet traffic. The dataset contains the following attacks:

- DDoS—Distributed DoS depending on the protocol (TCP, UDP, and HTTP).
- DoS—This attack depends on the protocol (TCP, UDP, and HTTP).

- Information gathering (Service Scanning and OS Fingerprinting).
- Information theft (Keylogging and Data theft).

All the features were selected in the dataset; then we obtained four files that were collected programmatically and worked on them.

The first file contains only a DoS attack, as shown in Figure A1 A DoS attack is an attack whose purpose is to make authorized users unable to access the system. This attack can be performed by flooding the target with HTTP, UDP, and TCP requests. Once the target is saturated with requests, it cannot respond to normal traffic. The second file contains a DoS and DDoS attack, as shown in Figure A2. A distributed DoS attack disrupts normal traffic to a target, whether a service or a network, by flooding the target from multiple sources with flood traffic, whether HTTP, UDP, or TCP requests.

The third file contains two DDoS attacks, as shown in Figure A3. The fourth file contains many attacks, as shown in Figure A4. We discussed some types above, and one of them is an information-gathering attack, which uses tools to collect information about the target. We have two types of this attack: service scanning and OS fingerprinting. Furthermore, another kind of attack is one where information theft occurs, when a target's personal information is stolen. Information is grabbed in several ways, including keyloggers, which are a kind of logger of the target's keystrokes, then sent to a third party. The BoT-IoT dataset is explained in Figure 4 for the three levels and all attacks.

			Total DB (DB1)		
		Classes in DS	Number befor reducing		
Attack	DDOS	DDoSHTTP	989	**1926624**	**3668045**
		DDoSTCP	977380		
		DDoSUDP	948255		
	DOS	DoSHTTP	1485	**1650260**	
		DoSTCP	615800		
		DoSUDP	1032975		
	Reconnaissance	ReconnaissanceOS_Fingerprint	17914	**91082**	
		ReconnaissanceService_Scan	73168		
	Theft	TheftData_Exfiltration	6	**79**	
		TheftKeylogging	73		
Normal	Normal	NormalNormal	477	**477**	**477**
		Total	**3668522**		

Figure 4. BoT-IoT DB 1.

Pre-processing: Two databases were derived from the original database by performing programmatic processing using MATLAB. The classifiers were trained on the three levels of the two derived databases.

Second database: The second database was obtained by taking random samples, as shown in Figure 5, to reduce the data size and work on it. We obtained the database shown in Figure 6 below with a smaller number, decreased from 3,668,522 to 63,030 by random sampling.

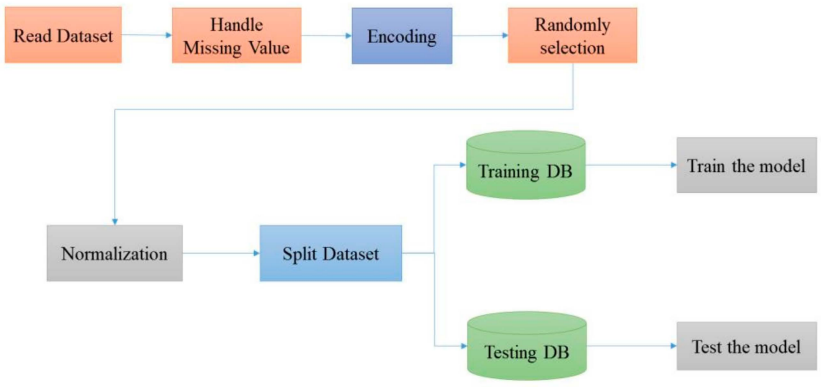

Figure 5. Methodology of DB 2.

			Reduced DB (DB2)		
		Classes in DS	Number after reduced		
Attack	DDOS	DDoSHTTP	989	20989	62553
		DDoSTCP	10000		
		DDoSUDP	10000		
	DOS	DoSHTTP	1485	21485	
		DoSTCP	10000		
		DoSUDP	10000		
	Reconnaissance	ReconnaissanceOS_Fingerprint	10000	20000	
		ReconnaissanceService_Scan	10000		
	Theft	TheftData_Exfiltration	6	79	
		TheftKeylogging	73		
Normal	Normal	NormalNormal	477	477	477
		Total	**63030**		

Figure 6. BoT-IoT DB 2.

3.2. Third Database

One of the factors that cause ML algorithms to perform poorly in classifications is data imbalance. This can happen for a variety of causes:

- First, accuracy is the essential function in many classification jobs, and it is inefficient if the classification model faces data imbalances.
- Next, the class distribution can be an issue, as when the dominant class was keener to enter the region of the minority classes, resulting in decreased generalization ability and an increase in classification errors, and vice versa.

We balanced the third database using the SMOTE function to reduce classification errors and improve the effectiveness of the IDS model, as shown in Figure 7.

			Reduced Balanced DB (DB3)		
		Classes in DS	Number after reduced_Balanced		
Attack	DDOS	DDoSHTTP	10000	30000	80600
		DDoSTCP	10000		
		DDoSUDP	10000		
	DOS	DoSHTTP	10000	30000	
		DoSTCP	10000		
		DoSUDP	10000		
	Reconnaissance	ReconnaissanceOS_Fingerprint	10000	20000	
		ReconnaissanceService_Scan	10000		
	Theft	TheftData_Exfiltration	100	600	
		TheftKeylogging	500		
Normal	Normal	NormalNormal	10000	10000	10000
		Total	**90600**		

Figure 7. BoT-IoT DB 3.

We obtained two other databases from the original data, as shown in Figure 8. A specific problem was solved in each of them, and work was done on these two databases.

In general, real-world data contain an unusable format for the ML model, noise, and missing values. To make the data suitable for ML models, we pre-processed to achieve the highest accuracy and efficiency from these models. Data pre-processing prepares raw data, making it formatted and suitable for an ML model. It is the first and most important step in creating an ML model. We used three pre-processing methods: missing data processing, normalization, and encoding, as shown in Figure 9.

Handle missing value: To fill in the missing values in the data set by learning to model the data set to infer the missing values, computes some simple column statistics are computed in order to obtain the arithmetic mean and make up for the missing value.

Normalization: Normalization converts the columns in a data set to the same scale. We only need it when the property ranges are different. There is more than one way to normalize in ML, and we used the minimum to maximum scale method.

	Total DB (DB1)	Reduced DB (DB2)	Reduced Balanced DB (DB3)
Classes in DS	Number befor reducing	Number after reduced	Number after reduced_Balanced
DDoSHTTP	989	989	10000
DDoSTCP	977380	10000	10000
DDoSUDP	948255	10000	10000
DoSHTTP	1485	1485	10000
DoSTCP	615800	10000	10000
DoSUDP	1032975	10000	10000
NormalNormal	477	477	10000
ReconnaissanceOS_Fingerprint	17914	10000	10000
ReconnaissanceService_Scan	73168	10000	10000
TheftData_Exfiltration	6	6	100
TheftKeylogging	73	73	500
Total	**3668522**	**63030**	**90600**

Figure 8. Explanation of the Three Databases.

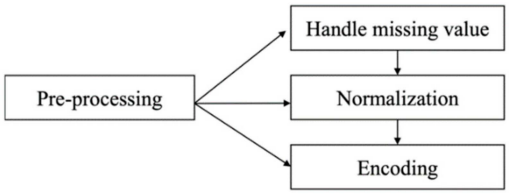

Figure 9. Pre-processing Steps.

Min-Max Scaling: The lowest value is subtracted from the highest value and divided by the range in each column. The columns we obtain will have a minimum value of 0 and a maximum value of 1.

Min-Max Normalization (Equation (1)):

$$X'[:\ ,i] = \frac{X[:\ ,i] - \min\left(X[:\ ,i]\right)}{\max\left(X[:\ ,i] - \min(X[:\ ,i])\right)} \quad (1)$$

Encoding: In an ML model, encoding is a technique for converting categorical variables into numerical values so that they can be used. Encoding is done through the following steps:

- Tokenize every cell in the document.
- Delete punctuation.
- Encode all categorical variables.
- Convert the document to a sequence.

Splitting the Dataset: Once the data are correct and clean, we can divide the data for classification. The dataset is divided into two sections, a training section and a test section, to estimate the performance of ML algorithms when they are used to make predictions about the data. In our work, we divided the dataset into 70% training and the remaining 30% to test for the efficiency of its prediction about the data not used in the training section. In the second database, which has a size of 63,030, we took 44,121 for training and 18,909 for testing. For the third database, which has a size of 90,600, we took 63,420 for training and 27,180 for testing. The results of ML algorithms allow us to compare the performance of each of them to solve the problem of predictive modelling.

Data Labelling: Data labelling is an essential part of the data pre-processing process in ML as data are classified in the task of data detection and labelling. Usually, supervision is manual and can be performed automatically with the help of some tools. The label types are pre-selected by the person who will work on the ML process, and then the ML model

information is used to train the model through the given examples. Then, these labelled data are used to train the ML models.

In the testing section, the "meaning" found in the new data is similar to what the model was trained on. A more accurate characterization through a focus on important factors and a more significant amount of labelled data create more useful deep-learning models. In this study, labels were used, and the label varies according to the level in the dataset:

- The label in the first level is attack or normal.
- The label in the second level is Category, to specify the type of attack (DDoS, DoS, Reconnaissance, and Theft).
- The label in the third level is Subcategory (DDoSHTTP, DDoSTCP, DDoSUDP, DoSHTTP, DoSTCP, DoSUDP, ReconnaissanceOS_Fingerprint, ReconnaissanceService_scan, Theft-Data_Exfiltration, and Theft_keylogging).

Classification: In ML, a classifier is an algorithm that classifies data into one or more sets of "categories". We worked on five classifiers: decision tree algorithm, ensemble bag algorithm, k-nearest neighbor algorithm, linear discriminant algorithm, and support vector machine algorithm. These ML algorithms were selected to fit with the selected data set. They are all supervised algorithms. The ensemble bag algorithm achieved the highest accuracy rate in all the second and third databases and at all levels, which solves a problem by properly integrating weak models until more accurate models are obtained. (Often called "weak learners"). In Figure 10, a flowchart is presented regarding how to train all these classifiers at the three levels in the dataset. The algorithm for our approach is presented in algorithm IDS.

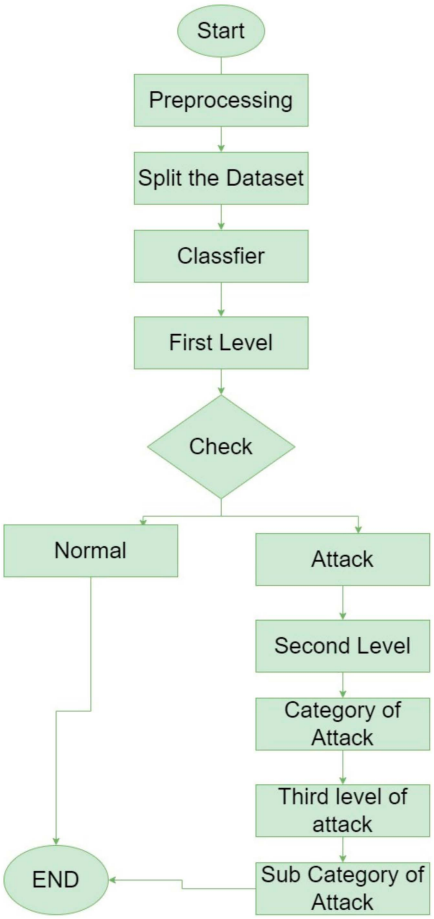

Figure 10. Flowchart of Trained Classifier.

3.3. Algorithm IDS

Input: Dataset of network traffic logs
Output: Ensemble model for detecting intrusions in IoT networks

1. *Split the dataset into training and testing subsets.*
2. *Define the number of base models to be used in the ensemble model.*
3. *For each base model:*
    a. *Pre-process the training data to handle missing values and normalize the data.*
    b. *Apply class weights to balance the training data.*
    c. *Train the base model on the pre-processed training data.*
    d. *Evaluate the accuracy, precision, and recall of the model on the testing data.*
    e. *Save the trained model for use in the ensemble model.*
4. *Load the saved base models.*
5. *For each instance in the testing data:*
    a. *Predict the class probabilities of each base model.*
    b. *Aggregate the predicted class probabilities using a weighted voting scheme.*
6. *Evaluate the accuracy, precision, and recall of the ensemble model on the testing data.*
7. *Save the ensemble model for future use.*

4. Evaluation

Evaluation Metrics

To evaluate the performance of ML models, performance measures must be defined for the tasks to be carried out. In order to assess the results, performance indicators of accuracy, error rate, precision, sensitivity, specificity, and the f-measure were used. As mentioned in the previous section, two databases were derived from the main and second databases. After the numbers were reduced, we trained five ML algorithms on the three levels in the data set. The following tables and confusion matrix show results for all levels. The confusion matrix shows that the ensemble bag classifier obtained the best accuracy compared it to other classifiers. The performance evaluation process was iterated 10 times for every ML model, and the computations in the tabulations were the arithmetic mean for the 10 operations.

Phase 1: In the second database, the proposed ML models were implemented for ten various attacks and on three levels, and the performance analyses are tabulated in Tables 2–4 for the outputs of the proposed models. Observing the outputs in the second database, it can be noted that every algorithm achieved over 99.8% success in detecting each attack. The ensemble bag approach was the best algorithm considered in this analysis.

Phase 2: In the third database, the proposed models were implemented for ten distinct attack types on three levels, and the performance analyses are presented in Tables 5–7 for the results of the algorithms. Observing the outputs in the third database, it can be noted that every algorithm achieved over 99.8% success in detecting every attack type. The ensemble bag approach was considered the best model.

Table 2. Implementation of classifiers from level 1 in the second DB.

Classifiers	Accuracy	Error Rate	Recall	Specificity	Precision	F Measure	Training Time (s)	Testing Time (s)
DT	99,989	0.011	100	100	100	100	0.552	0.006
Ensemble Bag	100	0	100	100	100	100	15.347	1.046
KNN	99,989	0.011	100	99.3	100	100	0.02	27.799
LD	100	0	100	100	100	100	0.85	0.021
SVM	100	0	100	100	100	100	0.775	0.015

Table 3. Implementation of classifiers from level 2 in the second DB.

Classifiers	Accuracy	Error Rate	Training Time (s)	Testing Time (s)
DT	99.989	0.011	0.337	0.011
Ensemble Bag	100	0	14.911	1.353
KNN	99.979	0.021	0.121	28,162
LD	100	0	4.634	0.071
SVM	99.995	0.005	7.339	0.083

Table 4. Implementation of classifiers from level 3 in the second DB.

Classifiers	Accuracy	Error Rate	Training Time (s)	Testing Time (s)
DT	99.995	0.005	0.604	0.01
Ensemble Bag	100	0	18.064	1.864
KNN	99.926	0.074	0.139	27.883
LD	99.921	0.079	8.777	0.515
SVM	99.889	0.111	18.784	0.441

Table 5. Implementation of classifiers from level 1 in the third DB.

Classifiers	Accuracy	Error Rate	Recall	Specificity	Precision	F Measure	Training Time (s)	Testing Time (s)
DT	100	0	100	100	100	100	0.422	0.012
Ensemble Bag	100	0	100	100	100	100	21,455	1436
KNN	100	0	100	100	100	100	0.424	58.303
LD	99.82	0.18	99.8	99.6	100	99.9	3302	0.298
SVM	99.993	0.007	100	100	100	100	2.322	0.369

Table 6. Implementation of classifiers from level 2 in the third DB.

Classifiers	Accuracy	Error Rate	Training Time (s)	Testing Time (s)
DT	100	0	0.512	0.01
Ensemble Bag	100	0	23.79	1.864
KNN	99.996	0.004	0.197	58.144
LD	100	0	4.608	0.151
SVM	99.993	0.007	8.45	0.11

Table 7. Implementation of classifiers from level 3 in the third DB.

Classifiers	Accuracy	Error Rate	Training Time (s)	Testing Time (s)
DT	100	0	1.074	0.018
Ensemble Bag	100	0	28.112	2.903
KNN	99.982	0.018	0.179	57.721
LD	99.989	0.011	11.624	1.335
SVM	99.967	0.033	26.552	0.655

5. Conclusions

This study aimed to discover IoT network attacks using ML. Bot-IoT was used as a dataset for attack diversity and network protocols. Two approaches were used to work

on the data set; the first was to reduce the data size, and the second was to solve the data imbalance problem. Five ML algorithms were trained on the two databases. A test of measures was conducted, including accuracy, error rate, recall, specificity, etc., for each algorithm independently and at all levels. In the second database, the ensemble bag algorithm obtained an accuracy of 100% at all levels of the database, and the decision tree algorithm obtained 99.9% at all levels. In the third database, the ensemble bag algorithm and the decision tree obtained an accuracy of 100% at all levels of the database, and all other ML algorithms did not obtain an accuracy of less than 99.8%. In future work, we aim to expand the range of attacks on networks to cover a more significant number of them to achieve greater security for IoT devices and to develop an IDS to intrusion prevention system (IPS).

Author Contributions: Conceptualization, methodology, software, validation formal analysis, investigation, resources, data curation, writing—original draft preparation, S.A.; Writing—review and editing, visualization, supervision, project administration, funding acquisition, S.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

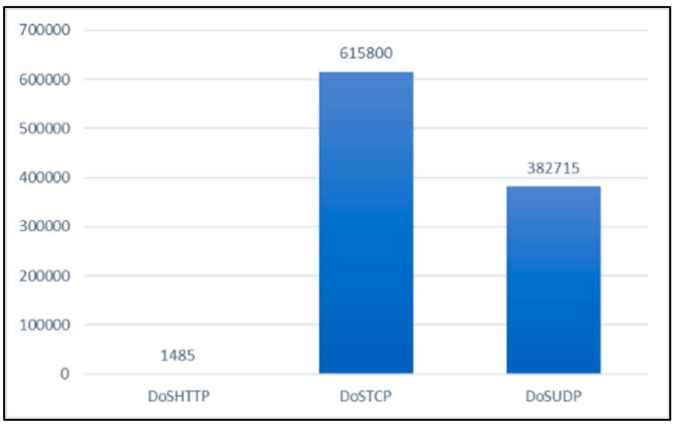

Figure A1. Summary of first file.

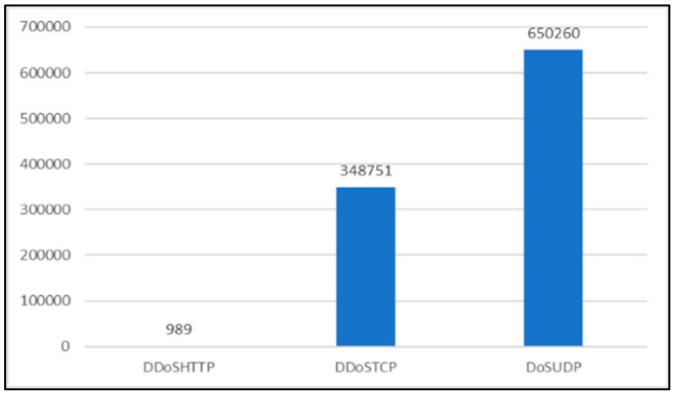

Figure A2. Summary of second file.

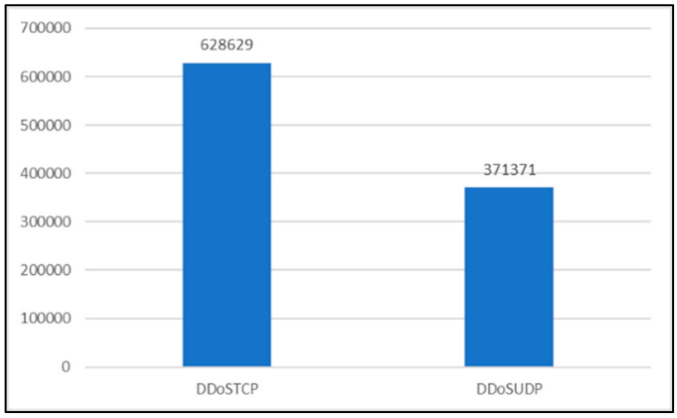

Figure A3. Summary of third file.

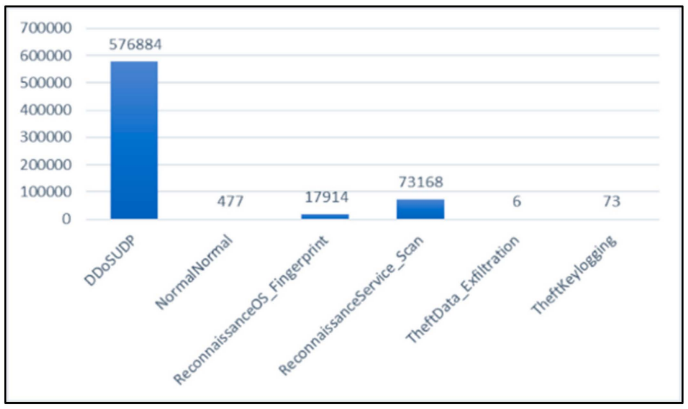

Figure A4. Summary of fourth file.

References

1. Balaji, R.; Deepajothi, S.; Prabaharan, G.; Daniya, T.; Karthikeyan, P.; Velliangiri, S. Survey on Intrusions Detection System using Deep learning in IoT Environment. In Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 7–9 April 2022; IEEE: Manhattan, NY, USA; pp. 195–199.
2. Alsamiri, J.; Khalid, A. Internet of Things cyber attacks detection using machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 627–634. [CrossRef]
3. Velliangiri, S.; Karthikeyan, P. Hybri4d optimization scheme for intrusion detection using considerable feature selection. *Comput. Appl.* **2020**, *32*, 7925–7939.
4. Verma, A.; Virender, R. Machine learning based intrusion detection systems for IoT applications. *Wirel. Pers. Commun.* **2020**, *111*, 2287–2310. [CrossRef]
5. Derhab, A.; Aldweesh, A.; Emam, A.Z.; Khan, F.A. Intrusion detection system for the Internet of Things based on temporal convolution neural network and efficient feature engineering. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 6689134. [CrossRef]
6. Abusafat, F.; Pereira, T.; Santos, H. Proposing a Behavior- Based IDS Model for IoT Environment. In *EuroSymposium on Systems Analysis and Design*; Springer: Cham, Switzerland, 2018; pp. 114–134.
7. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine learning in IoT security: Current solutions and future challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [CrossRef]
8. Guerra-Manzanares, A.; Medina-Galindo, J.; Bahsi, H.; Nõmm, S. MedBIoT: Generation of an IoT Botnet Dataset in a Medium-sized IoT Network. In Proceedings of the ICISSP, International Conference on Information Systems Security and Privacy, Valletta, Malta, 25–27 February 2020; pp. 207–218.
9. Roopak, M.; Tian, G.Y.; Chambers, J. An intrusion detection system against DDoS attacks in IoT networks. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; IEEE: Manhattan, NY, USA; pp. 562–567.
10. Hanif, S.; Ilyas, T.; Zeeshan, M. Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset. In Proceedings of the 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT), Charlotte, NC, USA, 6–9 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 152–156.
11. Ferrag, M.A.; Maglaras, L.; Ahmim, A.; Derdour, M.; Janicke, H. Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. *Future Internet* **2020**, *12*, 44. [CrossRef]

12. Susilo, B.; Sari, R.F. Intrusion detection in IoT networks using a deep learning algorithm. *Information* **2020**, *11*, 279. [CrossRef]
13. Mudgerikar, A.; Sharma, P.; Bertino, E. E-spion: A system-level intrusion detection system for IoT devices. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, Auckland, New Zealand, 9–12 July 2019; pp. 493–500.
14. Nimbalkar, P.; Kshirsagar, D. Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express* **2021**, *7*, 177–181. [CrossRef]
15. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
16. Le, T.T.; Kim, H.; Kang, H.; Kim, H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. *Sensors* **2022**, *22*, 1154. [CrossRef]
17. Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, G.P.; Garg, S.; Hassan, M.M. A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. *J. Parallel Distrib. Comput.* **2022**, *164*, 55–68. [CrossRef]
18. Shareena, J.; Ramdas, A. APH Intrusion detection system for iot botnet attacks using deep learning. *SN Comput. Sci.* **2021**, *2*, 205.
19. Alghanam, O.A.; Almobaideen, W.; Saadeh, M.; Adwan, O. An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. *Expert Syst. Appl.* **2023**, *213*, 118745. [CrossRef]
20. Syed, N.F.; Ge, M.; Baig, Z. Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks. *Comput. Netw.* **2023**, *225*, 109662. [CrossRef]
21. Alhanaya, M.; Ateyeh Al-Shqeerat, K.H. Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique. *Intell. Autom. Soft Comput.* **2023**, *36*, 3709–3724. [CrossRef]
22. Khanday, S.A.; Fatima, H.; Rakesh, N. Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. *Expert Syst. Appl.* **2023**, *215*, 119330. [CrossRef]
23. Srivastav, D.; Srivastava, P. A two-tier hybrid ensemble learning pipeline for intrusion detection systems in IoT networks. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 3913–3927. [CrossRef]
24. Abd Elaziz, M.; Al-qaness, M.A.; Dahou, A.; Ibrahim, R.A.; Abd El-Latif, A.A. Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm. *Adv. Eng. Softw.* **2023**, *176*, 103402. [CrossRef]
25. Lipsa, S.; Dash, R.K. A novel intrusion detection system based on deep learning and random forest for digital twin on IOT platform. *Int. J. Sch. Res. Eng. Technol.* **2023**, *2*, 51–64.
26. Mohy-eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M. An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimed. Tools Appl.* **2023**, 1–19. [CrossRef]
27. Thakkar, A.; Lohiya, R. Attack Classification of Imbalanced Intrusion Data for IoT network Using Ensemble Learning-based Deep Neural Network. *IEEE Internet Things J.* **2023**. [CrossRef]
28. Liu, X.; Du, Y. Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. *Electronics* **2023**, *12*, 1260. [CrossRef]
29. Ramesh Kumar, M.; Sudhakaran, P. Comprehensive Survey on Detecting Security Attacks of IoT Intrusion Detection Systems. *Adv. Sci. Technol.* **2023**, *124*, 738–747.
30. Douiba, M.; Benkirane, S.; Guezzaz, A.; Azrour, M. An improved anomaly detection model for IoT security using decision tree and gradient boosting. *J. Supercomput.* **2023**, *79*, 3392–3411. [CrossRef]
31. Alzahrani, R.J.; Alzahrani, A. A Novel Multi Algorithm Approach to Identify Network Anomalies in the IoT Using Fog Computing and a Model to Distinguish between IoT and Non-IoT Devices. *J. Sens. Actuator Netw.* **2023**, *12*, 19. [CrossRef]