applied sciences

MDPI

Article

An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks

Joseph Bamidele Awotunde [1], Sakinat Oluwabukonla Folorunso [2], Agbotiname Lucky Imoize [3,4,*], Julius Olusola Odunuga [2], Cheng-Chi Lee [5,6,*], Chun-Ta Li [7] and Dinh-Thuan Do [8]

1 Department of Computer Science, Faculty of Information and Communication Sciences, University of Ilorin, Ilorin 240003, Nigeria
2 Department of Mathematical Sciences, Olabisi Onabanjo University, Ago-Iwoye 120107, Nigeria
3 Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Lagos, Akoka, Lagos 100213, Nigeria
4 Department of Electrical Engineering and Information Technology, Institute of Digital Communication, Ruhr University, 44801 Bochum, Germany
5 Research and Development Center for Physical Education, Health, and Information Technology, Department of Library and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan
6 Department of Computer Science and Information Engineering, Asia University, Taichung City 41354, Taiwan
7 Bachelor's Program of Artificial Intelligence and Information Security, Fu Jen Catholic University, New Taipei City 24206, Taiwan
8 Department of Computer Science and Information Engineering, College of Information and Electrical Engineering, Asia University, Taichung 41354, Taiwan
* Correspondence: aimoize@unilag.edu.ng (A.L.I.); cclee@mail.fju.edu.tw (C.-C.L.)

Abstract: With less human involvement, the Industrial Internet of Things (IIoT) connects billions of heterogeneous and self-organized smart sensors and devices. Recently, IIoT-based technologies are now widely employed to enhance the user experience across numerous application domains. However, heterogeneity in the node source poses security concerns affecting the IIoT system, and due to device vulnerabilities, IIoT has encountered several attacks. Therefore, security features, such as encryption, authorization control, and verification, have been applied in IIoT networks to secure network nodes and devices. However, the requisite machine learning models require some time to detect assaults because of the diverse IIoT network traffic properties. Therefore, this study proposes ensemble models enabled with a feature selection classifier for Intrusion Detection in the IIoT network. The Chi-Square Statistical method was used for feature selection, and various ensemble classifiers, such as eXtreme gradient boosting (XGBoost), Bagging, extra trees (ET), random forest (RF), and AdaBoost can be used for the detection of intrusion applied to the Telemetry data of the TON_IoT datasets. The performance of these models is appraised based on accuracy, recall, precision, F1-score, and confusion matrix. The results indicate that the XGBoost ensemble showed superior performance with the highest accuracy over other models across the datasets in detecting and classifying IIoT attacks.

Keywords: cybersecurity; industrial internet of things; feature selection; machine learning; ensemble learning; intrusion detection systems; ensemble learning; chi-square statistical algorithm

Citation: Awotunde, J.B.; Folorunso, S.O.; Imoize, A.L.; Odunuga, J.O.; Lee, C.-C.; Li, C.-T.; Do, D.-T. An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks. *Appl. Sci.* **2023**, *13*, 2479. https://doi.org/10.3390/app13042479

Academic Editors: Gyuyeol Kong and Younggeun Hong

Received: 12 December 2022
Revised: 8 February 2023
Accepted: 13 February 2023
Published: 14 February 2023

1. Introduction

Automated network systems have globally adopted the idea of modern technologies in various fields to ease their operations and for the collection of large amounts of big data. The Internet of Things (IoT) is the next level of information technology (IT) development that can be used to connect the world, ranging from a straightforward to a unique application to an IoT-based system. IoT is a collection of integrated devices that are cloud-connected and are used by customers to receive IT services by fusing internet protocol with electronics-related properties [1]. The protocols used in IoT systems may include cybersecurity issues [2]

that could affect the entire system. The devices connected to the Internet Industrial of Things (IIoT) are open to assault by cybercriminals because they do not have the most basic security measures. That suggests they are vulnerable to hacking and botnet attacks, which are used to launch DDoS attacks against industries [3].

However, it is crucial to identify and effectively categorize cyberattacks that cross these security gaps. Therefore, utilizing an ensemble of ML models, this study attempts to develop an accurate and effective Intrusion Detection system (IDSs) to recognize and categorize cyberattacks on an IoT/IIoT network. The learning-based methodology adopted will use tree-based ensemble classifiers, such as eXtreme gradient boosting (XGBoost), Bagging, extra trees (ET), random forest (RF), and AdaBoost, learned on the seven Telemetry data of TON IoT datasets: Fridge, Thermostat, GPS Tracker, Modbus, Motion Light, Garage Door, and Weather devices datasets. For supervised learning issues, tree-based ensemble models are frequently used [2]. The power of ensemble classifiers depends on their capacity to combine many models' predictions to develop an improved model over a single model. When the foundational learners are distinct from one another, tree-based ensemble approaches operate at their best, which can be accomplished through randomization [4] or by employing significantly distinct training procedures for each decision tree.

Greater tree diversity results from randomization in tree growth, which also lowers correlation, i.e., increasing the independence of the decision trees. However, because each classifier in an ensemble technique must be trained, it can be computationally expensive. If there is a huge dataset involved, this cost may increase significantly. As a result, we concentrate on the widely used ensemble of ML models in the literature, particularly XGBoost, due to its efficiency and scalability. There are many different traffic aspects in the IoTs' noisy collected network traffic. Building models for ML-based models takes more time, and because IoT network traffic contains a multitude of features, they have an impact on IDS functionality and performance [5]. Feature selection is required to effectively develop cost-efficient and time-safe models for intrusion detection in IoT [6,7]. The study used criteria, including accuracy, recall, precision, F1-score, and confusion matrix, to evaluate how well the models performed.

Researchers have created and used various machine learning (ML)-based models, frequently combining them with feature selection methods to perhaps enhance their functionality and performance. Promising outcomes for the identification capabilities of ML have been produced using a set of performance metrics, but, for actual industrial IoT networks, these models are not yet trustworthy. This study strategy is to outperform cutting-edge outcomes for a particular dataset instead of learning more about a ML-based IDS application [8]. As a result, there has been far more academic study done than there has been done in other fields where deployments took place. This may result from high errors generated when compared to other fields [8]. Hence, these are unreliable for use in a real-world setting. Furthermore, using a single dataset with various features could be difficult to collect or store in a real-time IoT network connection. Besides, when using ML-based methods, their hyper-parameters, in most cases, require optimization for a better result. The optimization of hyper-parameters and feature selection will generally make the ML-based techniques run more efficiently.

The necessity to minimize risk and potential threats to IIoT systems has recently attracted academic interest. Effective IDSs specifically designed for IoT applications must be created. For training and evaluating such IDSs, a current and comprehensive IIoT dataset is needed. For assessing IDS-enabled IIoT systems, however, there are insufficient benchmark IIoT datasets that can be easily accessed or obtained from the internet freely [9,10]. This study uses brand-new, data-driven IoT/IIoT real-world datasets to solve these issues. It contains a label feature that separates the attack and normal classes and a feature that categorizes the threat subclasses that attack IoT/IIoT network nodes for issues with several classifications [11]. In addition, the TON_IoT dataset contains telemetry information for IoT and IIoT services [12]. With various IIoT-based IDS datasets, this study intends to evaluate the generalizability of feature selection techniques and ensemble classifier combinations.

The following summarizes the main contributions of the study:

- To create the best cyberattack multiclass classification model in IIoT systems, a thorough approach is proposed.
- This study suggests a feature selection strategy for IDS in the IIoT, utilizing ranked features from the Chi-Square Statistical Model and analyzing the link between feature variance and detection accuracy.
- Seven (7) ToN-IoT-based Telemetry datasets were employed to evaluate how well the model performed. In addition, extensive investigations have assessed the performances of an ensemble of ML-based models using these seven datasets.
- The performance of the ensemble models was verified by comparing them with the baseline research, which used the datasets and other existing approaches that used the same datasets.

The remaining sections of this study are structured as follows: Section 2 presents some related work in IoT/IIoT-based IDSs studies. The materials and methods used for the study are covered in Section 3 of the study. Section 4 outlines the outcomes of the experiment that was conducted. Finally, Section 5 concludes the study and provides future perspectives.

2. Related Work

This section describes some state-of-the-art research on ML models and IDSs to classify attacks on IoT networks. The idea of intelligent devices, ranging from refrigerators, doors, GPS trackers, etc., is not new. This section recaps some existing works that are related to smart devices. Many researchers have used the IoT to suggest gadgets that can be remotely monitored for various activities. This IoT-based system has various attacks, such as threats at the device, network, or application layers, which can be exploited by an intruder [11]. Various attacks can be launched against IoT-based networks, such as malware, SQL injection, scanning, DoS, malware, backdoor, ransomware, eavesdropping, and DDoS, among others, which are a few common cyberattack categories [12]. These various attacks can be grouped according to origin and layer.

The processing and analyzing of various methods used for intrusion detection in networks and IoT-based applications play a key role in society. The evaluation of the accuracy and effectiveness of IIoT security solutions relies heavily on the related datasets used, which represent IoT-based operations in the physical realm [12]. However, the major issue and challenge in evaluating IDSs specifically designed for IoT/IIoT purposes is the lack of real-world datasets that represent the IoT/IIoT application in the real world. The creation of IIoT-based IDSs is hampered by the lack of such datasets, considering that such strategies should perform well when empirically validated and evaluated [13,14]. The authors of [15] reviewed publications based on ML-based and data mining models for IDS classification on cybersecurity. They claimed that a large gap in the literature prevents the development of effective anomaly-based intrusion detection methods since tagged datasets are not readily available. This is mostly because of privacy concerns, as most IoT statistics from big businesses are not shared with the academic community [14].

A novel IoT traffic dataset called "Sensor480," presented by the authors in [16], contains 480 cases with three (3) properties of binary class normal and "Man-In-The-Middle" attacks. Based on this dataset, an IDS system was created and examined using various ML-based models. The dataset was split into 80–20% split ratios, and various performance metrics were used to appraise the proposed models, and DT outperforms other models with 100% performance accuracy. Additionally, authors in [17] presented an IDS based on ML-based ensemble models to recognize various forms of IoT cyberattacks. Using the datasets from IoT-23 [18], IoTDevNet [19], DS2OS [20], IoTID20 [21], and IoT Botnet [22], these models are evaluated based on a variety of performance indicators. With the highest accuracy values on the NSLKDD (99.27%), IoTDevNet (99.97%), DS2OS (99.39%), IoTID20 (99.99%), and IoT Botnet (99.991%) datasets, the outcome demonstrates that Bi-LSTM outperformed other models. However, most of the presented datasets are outdated and do not contain the recent IIoT-based intrusion attacks. The Windows 10 dataset from the ToN

IoT [12] was used by authors in [23] to pick the best features. They used the correlation function and the ReliefF method of feature selection schemes. With accuracy scores of 94.12% for the correlation function dataset and 98.39% for the ReliefF dataset, the Medium NN model outperformed other models. The results of the proposed model by the authors show that there is a need for improvement in the areas of IDS accuracy. The model is still very slow and takes a huge part of the computer processor.

To decrease the characteristics of the Linux, Network, and Windows 7 and 10 multiclass datasets of the ToN-IoT dataset, the authors in [24] presented the Chi2 approach and balanced the dataset for the best categorization using the synthetic minority oversampling (SMOTE) approach. They employed various ML-based models, with XGBoost outperforming all others on all datasets according to the numerous performance criteria they used to assess the suggested models. In [25], the authors applied supervised and unsupervised ML over the NF-ToN-IoT-v2 dataset to provide a thorough model of a network IDS (NIDS). It was demonstrated that the technique XGBoost Classifier, which obtained a F-Score of 98.8%, produced the best results when supervised learning was used, as implemented by Azure automated ML (AML). The random forest classifier, with a F-Score of 98.6%, produced the greatest results when a specially designed automated ML (AE2EML) was used. The suggested ML-based NIDS obtained a Silhouette score of 0.553, a Calinski-Harabasz index of 1533106, and a Davies-Bouldin index of 0.631 using clustering with PCA (Principal Component Analysis), performed by PyCaret-automated ML. The proposed model by the study performed excellently, but it used old datasets that did not contain recent IIoT-based network attacks.

By examining the applicability of ML-based algorithms in the detection of abnormalities within the data of such networks, the authors of [26] concentrated on the security element of IoT networks. It investigates ML algorithms that have been effectively applied in circumstances that are comparable to one another and contrasts them using a variety of factors and techniques. The RF algorithm produced the best results, with a 99.5% accuracy rate. The authors of [27] presented an IDS with an ensemble classifier enabled by a feature selection classifier. The study utilized the Correlation Coefficient (CC) method for feature selection before classifying the dataset for the detection of various attacks using various classifiers, such as NB, DT, and ANN. On the UNSW-NB 15 datasets, the system detected DoS assaults with an accuracy of 98.54% with a classifier ensemble that uses a subset of the features. The dataset used to test the model is not an IIoT-based network nodes attacks dataset and does not employ feature selection methods to remove the irrelevant features from the dataset used, and the issue of imbalance data is not considered, thus reducing the performance of the model.

The authors in [28] used the top 13 IG characteristics with the C5 classifier to obtain improved accuracy of 89.76% and a better FAR of 1.68. The study recommended that IG be used for choosing features for IDS. The top ten ranked IG attributes were used in the system to create a greater accuracy of 93.23% with 6.77% FAR. The authors obtained six reduced features in [29] using the multi-objective feature selection method on the CICIDS 2017 dataset. The system delivered an accuracy of 99.90% using an ELM classifier. To detect cyberattacks, the study authors in [30] suggested using LSTM networks enabled with parameter optimization, called Stochastic Gradient Descent (SGD), for the creation of IDS. The study obtained an accuracy of 99.91% for ISCX and 98.22% for AWID datasets, respectively.

The top 10 attributes of the GR technique were used in work by authors in [31], and their layer design was validated on a generated dataset. In contrast to previous rules and tree-based learners, the design performed better with the J48 classifier for recognizing DoS assaults. A decision tree-based multi-layer framework to identify DDoS attacks was provided in the study of authors in [32]. The system recognized ICMP, TCP, and UDP flood attacks on a created dataset, with an accuracy of 99.98%, using eight features that were explicitly picked. The authors in [33] utilized nature-inspired techniques for feature selection with forecasting and chaos methods. The performance of the model was evaluated using the NS-3 created model. For the identification of DoS assaults at the

transport and application layers, the approach obtained a detection rate (DR) of 94.3%. The authors employed the wrapper feature selection approach in [34] for feature selection in IDS. The study performance was tested using the honeypot Cowrie dataset, with various cyberattacks, with an accuracy of 97.4% using the SVM classifier.

The effectiveness of the PCA and the results obtained without it were compared by the authors in [35]. Prior to being used in various ML-based techniques, the dataset was first submitted to Principal Component Analysis (PCA) for feature selection. This experimental investigation demonstrates that utilizing PCA reduces algorithm execution time greatly, with a smaller number of features, while producing the same results as not using PCA. In addition, when compared to SVM, the DT and RF algorithms accurately classified DDoS packets. Matplotlib was used to create a graph to display the results. The IoT-23 dataset was used for our experimental analysis. The authors in [36] developed an IDS model based on a hybrid AI model for the classification of attacks for an IoT-based system. The CIC-IDS2017 and UNSW-NB15 datasets were used to evaluate the performance of the suggested model. The model fared better, with a detection rate of 99.75% and an accuracy of 99.45%.

The authors of [6] presented a hybrid rule-based feature selection DL-based IDS paradigm for IIoT to train and validate data extracted from TCP/IP packets. A hybrid rule-based feature selection and deep feedforward neural network model were used to implement the training procedure. NSL-KDD and UNSW-NB15, two well-known network datasets, were used to test the suggested approach. According to the findings of the performance comparison, the suggested strategy outperforms other pertinent methods in terms of accuracy, detection rate, and FPR by 99.0%, 99.0%, and 1.0%, respectively, for the NSL-KDD dataset, and by 98.9%, 99.9%, and 1.1%, for the UNSW-NB15 dataset. The recommended method is suitable for IIOT intrusion network attack classification, according to simulated trials utilizing a variety of assessment metrics.

The authors of [37] proposed the RDTIDS intrusion detection system (IDS) for IoT networks. The RDTIDS integrates multiple classifier methodologies, such as REP Tree, JRip algorithm, and Forest PA, which are based on decision tree and rules-based principles. The first and second methods specifically classify the network traffic as attack/benign by using features from the data set as inputs. The outputs of the first and second classifiers are used as inputs for the third classifier, together with characteristics from the initial data set. The extensive experiments demonstrate the proposed IDS' effectiveness over existing state-of-the-art schemes in terms of accuracy, detection rate, false alarm rate, and time overhead. These findings were made using the CICIDS2017 dataset and the BoT-IoT dataset.

Authors in [38] suggested a novel ensemble of Hybrid IDSs for IoT device security by fusing a C5 classifier and a One-Class Support Vector Machine classifier. The benefits of Signature IDS and Anomaly-based IDS are combined in HIDS. With high detection accuracy and low false-alarm rates, this system seeks to identify both known intrusions and zero-day threats. The Bot-IoT dataset, which includes legal IoT network traffic and various assaults, is used to assess the proposed HIDS. Studies reveal that, compared to SIDS and AIDS approaches, the proposed hybrid IDS offers a higher detection rate and a reduced percentage of false positives.

To identify out-of-norm actions for cyber threat hunting in the IIoT, the authors of [39] presented an ensemble DL-based model that combines LSTM with the Auto-Encoder (AE) architecture. Additionally, most of the prior literature did not consider the uneven nature of IIoT datasets, which led to low accuracy and performance. The suggested approach takes fresh, balanced data from the unbalanced datasets and feeds these new balanced data into the deep LSTM AE anomaly detection model to resolve this issue. In addition, the advanced related models Stacked Auto-Encoders (SAE), Naive Bayes (NB), Projective Adaptive Resonance Theory (PART), Convolutional Auto-Encoder (C-AE), and Package Signatures (PS) based LSTM (PS-LSTM), are compared to the proposed ensemble model.

In the reviewed literature, it was observed that almost all the studies used ISCX, CICIDS, UNSW-NB15, and KDD Cup 199, which are non-IoT/IIoT-based datasets. They are datasets for network intrusions that contain HTTP DoS assaults. The IEEE 802.11-related

Madiun Access Control (MAC) Layer attacks are part of the AWID dataset. This study acquired datasets containing network traffic, operating system traces, and IoT telemetry data from diverse IoT/IIoT source materials. Additionally, the suggested dataset includes various valid and malicious IoT-related events, incorporating the reality of attacks and legal occurrences.

This study proposes a feature selection-based IDS enabled with various ensemble classifiers for detecting several attacks in an IIoT-based network. Very little research using the TON-IoT dataset is shown in this review. When the ML-based ensemble model was compared with the baseline findings, it was discovered that the frequently misclassified assaults are not discussed. The proposed ensemble classifiers enabled with feature selection will be applied to the IoT telemetry datasets, and the results of the proposed models will be compared with the baseline analysis.

3. Materials and Methods

This section describes a robust framework to detect and classify cyberattacks on IoT network trails. The ensemble classifier process in various successive steps is displayed in Figure 1, and preprocessing is the first step. At this stage, the dataset is explored for the number of instances, the number of features, the relationship between the features, the correlation between the features, etc. The details of the dataset used for the performance evaluation were discussed, followed by the details of the performance metrics used for evaluation purposes. Finally, a training and testing dataset was created from the cleaned dataset. The ensemble models use the training set to learn, while the test set is used to assess the performance of the model.

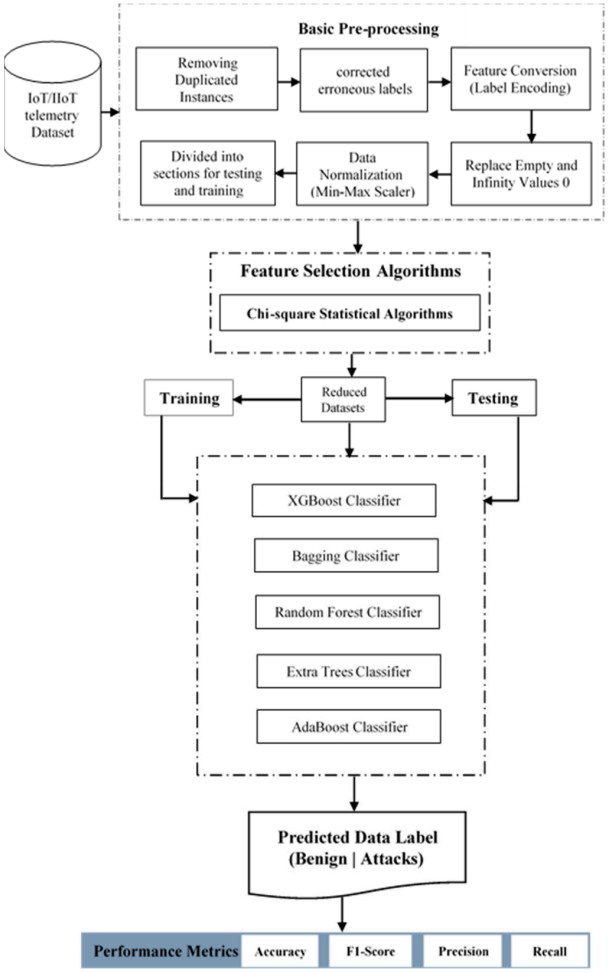


Figure 1. The proposed workflow for the classification of cyberattacks in an IoT network.

3.1. Data Preprocessing

Data preprocessing is a serious first step in streamlining the training of ML models. For the purpose of research, all datasets are openly accessible for download. To minimize storage space requirements and to prevent redundancy, duplicate samples (flows) are eliminated. The flow identifiers, IP addresses, ports, and timestamps are eliminated to eliminate forecast bias against the attackers within end network nodes. Then, using a categorical encoding approach, numerical values are assigned to the strings and non-numeric characteristics. The features in these datasets include protocols and services, which have been compiled as their native string values, as well as ensemble classifiers. However, these are built to function effectively with numerical data.

Hot encoding and label encoding are the two primary methods for encoding the features. The former adds X features to a feature to convert it into X categories, utilizing 0 to indicate that a category is not present and 1 to indicate it is. Nevertheless, this enhances the dataset's dimensionality, which could impact the ML models' effectiveness and performance. Hence, each category is converted to an integer using the label encoding technique.

Categorical features were converted to numerical values for straightforward ML technique application. For instance, the categorical values "open" and "closed" for the door state feature from the GarageDoor dataset were converted into "0" and "1". Furthermore, duplicate, incompatible, and missing values were effectively handled. Additionally, this process permits the equal weighting of all features because network traffic properties are complex, and there are higher numbers than others. This could cause the ensemble model to weigh them more heavily, so it will pay attention to them. The min-max scaler uses Equation (1) to calculate all values for each feature, where X^* is a new feature value between 0 and 1, and X represents the unique feature value, where the feature maximum and minimum values are X_{max} *and* X_{min}, respectively. Segments for training and testing are separated from the dataset, and these components are categorized according to the label features, which are crucial given the class imbalances of the datasets.

$$\widetilde{x_i} = \frac{x_i - min(x)}{max(x) - min(x)}, \tag{1}$$

3.2. The Chi-Square Statistical Feature Selection Model

This method was used in this study to select the most relevant features. The two most prominent variables are usually involved in using this model for feature selection. Typically, they relate to the likelihood of occurrence of category C based on the likelihood of occurrence of feature t. In IDS classification, it is considered whether attributes t and C are independent in the proposed approach. Unless features t and C contradict, characteristic t cannot be used to determine if a label falls under category C. It might be difficult to determine the degree of t and C in training, especially if they are not linked. Therefore, their relevance can be evaluated using the Chi-square test. Using a statistical method called Chi-square, it is possible to quantify the connection between feature t and category C. A bidirectional queue was used to express a label feature called t and a category called C.

Assuming feature t and type C, then the first-order degree of freedom chi-square distribution matches. The higher the category C chi-square score, the more category labels the feature holds. Therefore, t *and* C_j have a lot in common. The feature t category C chi-square score is then defined as follows:

$$X^2 = (t,\ c)\frac{N(AD - BC)^2}{(A + C)(A + B)(B + BD)(C + D)} \tag{2}$$

The solution to Equation (2) demonstrated the relationship between feature t and category C_j. The more autonomous members of the class category, C_j, will be the feature t that matters. When $CHI(t,\ C_i)$, then the label class, C_j, and feature, t, are independent. You can calculate the value for one class, symbolized by $X^2(t,\ c)$, using Equation (3). However, by combining all of the classes of the value in feature label t in $X^2(t,\ c)$, then, for each

characteristic of instance t across all classes, we first determine the $X^2(t,\ c)$. The number of m classes is then determined by testing feature t for each unique $X^2(t,\ c)$ score:

$$XAVG^2(t) = \sum_{i-1}^{m} p(c)X^2(t,\ c) \tag{3}$$

Equation (3) is used to calculate the mean $X^2(t,\ c)$, the score for the feature label t across all classes.

$$\aleph MAX^2(t) = \max_{1 \le i \le m} \left\{ X^2(t,\ c) \right\} \tag{4}$$

For all classes, the maximum $X^2\ (t,\ c)$ of a feature, the label, is determined using Equation (4). The threshold value is used to determine the appropriate number of feature labels after the feature label has been sorted by the X scores.

3.3. Machine Learning Model

This sub-section discusses the ensemble ML-based models used for detecting attacks in IIoT-based networks.

(1) Extreme Gradient Boosting (XGBoost)

XGBoost is a modified gradient tree-boosting algorithm that is efficient and scalable. The optimization problem in ensemble algorithms can be solved using the boosting classifiers, where one weaker learner is added in succession to create a new model to lower the classifier loss function and to progressively reduce the mistakes of earlier models [40]. The exemplary features of the algorithm proposed by the authors in [41] are the regularized model, split-seeking algorithm, column block structure, and cache-aware prefetching algorithm. Some current applications of XGBoost include genre classification of Nigerian songs [42], predicting stock price [43], and forecast gene expression value [44].

(2) Bagging Classifier

A group meta-learner is the Bagging classification algorithm. The approach creates a large number of learners by training each unique base learner on a random subset of the actual dataset. The classifier then estimates the final prediction by averaging the results of all the models [45]. This algorithm averages the probability values of base learners for regression tasks and applies the majority voting scheme to classify labels for the classification tasks. This algorithm starts by resampling the training data with replacements. This means that some instances may be selected again and again, while others may not. The strength of this meta-estimator is the reduction in the variance of the base learner by introducing randomness into the ensemble construction and generation method. Concurrent training is conducted on the randomly selected subset of the training set with the base learners using substitution using the initial dataset. Each base classifier's training dataset is distinct from the datasets of the others.

(3) Random Forest (RF)

RF is a group of weak base learners that functions by building various collections of decision trees to enhance the DTs' effectiveness and resilience [46]. This technique combines the bagging approach of instance sampling with the random selection method for features in creating a collection of DTs with a controlled variation. To complete the classification task of an unlabeled instance, each DT in a set acts as a base learner. The algorithm uses majority voting for the classification task and probability averaging of instance values from the regression task. The RF algorithm is immune to noise and over-fitting and has been applied to several domains, including heart disease classification [47] and label ranking [48].

(4) Extremely Randomized Trees (Extra Trees)

Extra Tree is a collection of ML-based models that combine the classifications from several unpruned DTs on different sub-samples of the target to enhance generalization accuracy, being computationally efficient and preventing over-fitting [49]. The entire

training instance is used to grow trees, and the nodes at each tree are split by selecting the cut points fully at random. These predictions are made by using a majority voting scheme for classification tasks or averaging prediction values for regression tasks.

(5) Adaptive Boosting (AdaBoost)

AdaBoost is an ensemble of ML models adopting the boosting method by joining many weak learners to create a new model using the weighted linear combination method iteratively. To reweight examples of the real train data, it progressively uses a learning algorithm [50]. Firstly, all instances are assigned the same weight. Weights are increased for cases that were incorrectly classified, while they are raised for instances that were correctly classified. This procedure is iterated continually by new weights of the training data on the base model. Finally, a linear combination of all the models generated through the various iterations is used to create the final classification model [51]. This algorithm's weakness is that it is sensitive to anomalies and noisy data.

3.4. Dataset

The study datasets include seven (7) ToN-IoT (https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=%2FProcessed_datasets%2FProcessed_Network_dataset (accessed on 5 June 2022)) datasets obtained from Telemetry. The IoT/IIoT-based network testbed was used to generate various operating systems and Network data. These 7 datasets were generated from various IIoT-based devices, such as GPS_Tracker, Weather, Garage_Door, Modbus, Fridge, Thermostat, and Motion_Light. Table 1 presents all features of the seven (7) datasets, such as the smart fridge device, which measures the temperature and its adjustments below the on-demand. Based on a probabilistic input, the features of a remotely activated garage door when opened or closed. The components and features are based on the Global Positioning System (GPS) device, which tracks the geographical coordinates of a remote object. The features obtained are from the smart sense motion device. This uses a pseudo-randomly generated signal to either "on" or "off" the light. The features generated from the register in the Modbus service device are majorly used for industrial applications. These devices communicate via a master–slave arrangement. The characteristics of a smart thermostat regulate a system's temperature by controlling the heating/cooling system, such as the air conditioner. The dataset of a weather monitoring system creates features, such as temperature, air pressure, and humidity in the data.

Table 1. The IoT Telemetry dataset feature descriptions.

Features	Data Type	Description
		Fridge dataset feature description
Fridge_temparature	float	Temperature estimate of a fridge sensor connected to the network
Temp_condition	categorical	Temperature situations of a fridge sensor connected to the network. The temperature could either be high or low when grounded on a precomputed threshold value
		Garage_door dataset description
door_state	Boolean	State of a door sensor connected to the network, where the door could either be closed or open
sphone_signal	Boolean	Door status signal received on a phone, where the signal could either be true or false
		GPS_Tracker dataset description
latitude	Float	GPS tracker sensor latitude connected to the network
longitude	Float	longitudinal value of the GPS tracker sensor connected to the network
		Motion_Light dataset description
motion_status	Integer	The status of the motion sensor device, which could either be on (1) or off (0)
light_status	Boolean	The status of the light sensor device, which could either be 'on' or 'off'
		Modbus dataset description
FC1_Read_Input_Register	Integer	The Modbus modular code is accountable for accepting an input value from a register.
FC2_Read_Discrete_Value	Integer	The Modbus modular code is accountable for reading in a discrete value.
FC3_Read_Holding_Register	Integer	The Modbus modular code is accountable for accepting a holding register value.
FC4_Read_Coil	Integer	The Modbus modular code is accountable for recording the coil value.

Table 1. *Cont.*

Features	Data Type	Description
		Thermostat dataset description
current_temperature	Float	The present temperature value recorded by the network-connected thermostat sensor gadget
thermostat_status	Boolean	The thermostat sensor device's condition could either be 'on' or 'off'.
		Weather dataset description
temperature	Float	The temperature values recorded based on the weather sensor apparatus attached to the network
pressure	Float	The pressure values recorded by the network-connected weather sensor device
humidity	Float	The humidity values recorded, as determined by the network-connected weather sensor device

These ToN-IoT datasets were commonly labeled into binary categories of 'normal' or under 'attack'. The 'attack' class is also further divided into seven (7) subclasses—Scanning, password, DDoS, injection, ransomware, Cross-site Scripting (XSS), and backdoor. The scanning class occurs at the initial stage, where the information about the target system is obtained by the attackers [8,52] using a scanning tool, such as Nmap [53] or Nessus [54]. The DoS attack [8,52] adopts the flooding strategy, where the attacker blasts off successive malicious attacks against a genuine user to disrupt their right to access service, while DDoS blasts off enormous successive connections to deplete the resources of the device memory, CPU, etc. These two similar attacks are usually blasted off by a vast network of hacked computers known as bots or botnets [8,55]. The Ransomware attack [56] is a classical kind of malware that holds the access right of an authentic user to a system or service to ransom by encrypting their access and attempting to transfer the decryption key to restore the original user's access to the service or system.

The Backdoor attack [57] is a passive attack that uses backdoor software to give an opponent unauthorized remote access. The competitor utilizes this backdoor to manage the infected IIoT devices and to incorporate them into botnets to launch a DDoS attack [57]. The Injection attack [57,58] often attempts to execute vindictive codes or implant vindictive data into the IIoT network to disrupt normal operation. Cross-Site Scripting (XSS) [58] often tries to run vindictive commands on a seb server in the IIoT applications. The XSS lets the attacker insert random web scripts remotely into the IIoT system. The information and the authentication procedure between IIoT devices and the remote web server may be compromised by this attack. A typical Password Cracking Attack [59] occurs when a rival applies password-cracking techniques to figure out an IIoT device's passcode. The attacker will bypass the authentication system and compromise the IIoT devices [57]. A common network attack that might disrupt the communication link between two devices is the MiTM attack [13], which could alter their data. Examples of MiTM attacks include ICMP redirect, ARP Cache poisoning, and port theft [12]. The datasets and their detailed descriptions are presented in Tables 1–7. The seven (7) datasets have login dates for the IoT Telemetry data, login times for the IoT Telemetry data, and the record of the binary label of normal and attacks, where '0′ represents normal and '1′ represents attacks.

Table 2. Record of the multi-class label of various actual attacks and normal.

Dataset	Backdoor	Injection	DDoS	Password	Ransomware	Scanning	XSS	Normal	Total
Fridge	35,568	7079	10,233	28,425	2902	2042	-	500,827	587,076
Garage_door	35,568	6331	10,230	19,287	2902	529	1156	515,443	591,446
GPS_Tracker	35,571	6904	10,226	25,176	2833	550	577	513,849	595,686
Motion-Light	28,209	5595	8121	17,521	2264	1775	449	388,328	452,262
Moldbul	40,036	7079	-	24,269	-	529	577	405,904	454,124
Thermostat	35,568	9498	-	8435	2264	61	449	385,953	442,228
Weather	35,641	9726	15,182	25,715	2865	529	866	539,718	630,242

Table 3. The Confusion Matrix.

	Real Positive ('Normal')	**Real Negative ('Attack')**
Predicted Positive ('normal')	TP	FP
Predicted Negative ('attack')	FN	TN

Where true positive (TP) is the proportion of instances of "attack" that are actually and correctly identified. True negative (TN) is the proportion of legitimately designated "normal" instances that occur. False positive (FP) refers to the percentage of actual "normal" samples that are mistakenly identified as "attack," while a false negative (FN) refers to the percentage of actual "attack" samples that are mistakenly classed as "normal".

Table 4. The classification report for the IIoT Fridge sensor device dataset.

Dataset	**Models**	**Accuracy**	**Precision**	**Recall**	**F1_Score**
Fridge	Bagging	0.9856	0.9795	0.9908	0.9850
	XGBoost	0.9873	0.9801	0.9942	0.9869
	RF	0.9843	0.9777	0.9912	0.9843
	ET	0.9801	0.9754	0.9861	0.9806
	Ada	0.4809	0.1299	0.2589	0.1728

Table 5. The classification report for the IIoT Thermostat dataset.

Dataset	**Models**	**Accuracy**	**Precision**	**Recall**	**F1_Score**
Thermostat	Bagging	0.9858	0.9875	0.9918	0.9896
	XGBoost	0.9883	0.9882	0.9950	0.9915
	RF	0.9865	0.9874	0.9940	0.9907
	ET	0.9838	0.9854	0.9730	0.9787
	Ada	0.5305	0.3557	0.3130	0.2837

Table 6. The classification report for the IIoT GPS_Tracker dataset.

Dataset	**Models**	**Accuracy**	**Precision**	**Recall**	**F1_Score**
GPS_Tracker	Bagging	0.9832	0.9779	0.9829	0.9804
	XGBoost	0.9869	0.9780	0.9895	0.9836
	RF	0.9813	0.9771	0.9838	0.9804
	ET	0.9766	0.9756	0.9753	0.9754
	Ada	0.4738	0.0592	0.1250	0.0804

Table 7. The classification report for the Modbus dataset.

Dataset	**Models**	**Accuracy**	**Precision**	**Recall**	**F1_Score**
Modbus	Bagging	0.9890	0.9870	0.9878	0.9874
	XGBoost	0.9913	0.9864	0.9895	0.9879
	RF	0.9878	0.9861	0.9793	0.9827
	ET	0.9870	0.9843	0.9730	0.9785
	Ada	0.6292	0.1049	0.1667	0.1287

Table 2 gives details of each attack and the normal of the multi-class label of the entire dataset. The datasets are referred to as "ToN IoT", since they comprise a variety of data sources, including Windows 7 and 10 operating system datasets, Ubuntu 14 and 18 TLS, and network traffic datasets, as well as telemetry datasets of IoT and IIoT sensors. The datasets were gathered from a large-scale, realistic network created at the UNSW Canberra @ Australian Defence Force Academy (ADFA) of Cyber Range and IoT Labs, School of Engineering and Information Technology (SEIT). The industrial 4.0 network, which consists of the IoT and IIoT networks, has a new testbed network. To manage the connection between the three levels of IoT, Cloud, and Edge/Fog systems, the testbed was deployed, utilizing several virtual machines and hosts of Windows, Linux, and Kali

operating systems. On the IoT/IIoT network, several hacking methods, including DoS, DDoS, and ransomware, are used against web apps, IoT gateways, and computer systems. Network traffic, Windows audit traces, Linux audit traces, and telemetry data from IoT services were among the datasets collected in parallel processing to capture various regular and cyberattack events.

3.5. Performance Indicators

Many different performance indicators were used to assess the performance and effectiveness of ML models on the different datasets. Some commonly used indicators, which will also be adopted for this study, are confusion matrix, ROC_AUC, F1_score, recall, precision, and accuracy [60]. The confusion matrix is a table shown in Table 3, representing the detection rate of classes of dataset, thereby measuring the performance of an ML model on the test data.

The ROC_AUC indicates the tradeoff between True Positive Rate (TPR), or recall, and FPR, as shown by Equation (5). False Positive Rate is the percentage of 'normal' class instances wrongly classified as an 'attack' class, as shown by Equation (6). The accuracy assessment calculates a model's overall effectiveness as a percentage of all "normal" data and the various "attack" incidents that were correctly classified, as shown by Equation (7). The recall assessor indicates the percentage of 'attacks' instances that were properly detected in the test dataset, as indicated by Equation (8). In contrast, the precision assessor indicates the percentage of properly detected 'attack' instances of all the detected 'attacks', as indicated by Equation (9). Finally, the f1_score estimates the harmonic mean of precision and recall, as indicated by Equation (10).

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{5}$$

$$Recall\,/\,TPR = \frac{tp}{tp + fn} \tag{6}$$

$$Precision = \frac{tp}{tp + fp} \tag{7}$$

$$f1\_score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

$$FPR = \frac{fp}{fp + tn} \tag{9}$$

$$ROC = \frac{1 + (TPR - FPR)}{2} \tag{10}$$

4. Experimental Results and Discussions

This section presents the experimental result performed on five (5) different machine learning models: Bagging, XGBoost, Random Forest, ExtraTrees, and AdaBoost on seven (7) IoT/IIoT-based datasets. The candidate assessment methods on the models are accuracy, F1-score, precision, and recall assessor.

4.1. Experimental Results Based on the Proposed Model

The experimental findings for per-device datasets are presented in this section. The 70–30 train-test data split ratio was applied to all five ensemble models used in the study. The final result was calculated and displayed as the mean value of all evaluation methods.

Table 4 presents the mean values of the accuracy, F1-score, precision, and recall metrics for the proposed ensemble models applied to the IIoT_Fridge sensor device dataset. For this dataset, the XGBoost classifier outperforms all other models, with 0.9873 for accuracy, 0.9801 for precision, 0.9942 for recall, and 0.9869 for F1_Score, respectively. Conversely,

the worst classifier is the AdaBoost 'Ada', with 0.4909 for accuracy, 0.1299 for precision, 0.2589 for recall, and 0.1728 for F1-Score, respectively.

Table 5 demonstrates the mean values of the accuracy, F1-score, precision, and recall metrics for the proposed ML-based ensemble models applied to the IIoT_ Thermostat sensor device dataset. For this dataset, the XGBoost classifier outperforms all other models, with 0.9883 for accuracy, 0.9882 for precision, 0.9950 for recall, and 0.9915 for F1_Score, respectively. Conversely, the worst classifier is the Ada, with 0.5305 for accuracy, 0.3557 for precision, 0.3130 for recall, and 0.2837 for F1-Score, respectively.

Table 6 presents the mean values of the accuracy, F1-score, precision, and recall metrics for the candidate ML models applied to the IIoT_GPS_Tracker sensor device dataset. For this dataset, the XGBoost classifier outperforms all other models, with 0.9869 accuracies, 0.9780 precision, 0.9895 recall, and 0.9836 of F1_Score metrics. Conversely, the Ada classifier is the worst of all the models, with 0.4738 accuracy, 0.0592 precision, 0.1250 recall, and 0.0804 of F1_Score, respectively.

Table 7 illustrates the mean values of the performance metrics evaluation for the ML-based ensemble models applied to the IIoT_Modbus sensor device dataset. For this dataset, the XGBoost classifier still performs excellently when compared with other classifiers, with an accuracy of 0.9913, precision of 0.9864, recall of 0.9895, and F1_Score of 0.9879, respectively, for all the performance metrics used in the study. The worst classifier is the Ada, with an accuracy of 0.6292, precision of 0.1049, recall of 0.1667, and F1_Score of 0.1287, respectively.

Table 8 presents the mean values of the performance metrics evaluation for the ML-based ensemble models applied to the Motion_Light device dataset. For this dataset, the XGBoost classifier still performs excellently when compared with other classifiers, with an accuracy of 0.9719, precision of 0.9531, recall of 0.8957, and F1_Score of 0.9030, respectively. The worst classifier is the Ada, with an accuracy of 0.4695, precision of 0.1144, recall of 0.2241, and F1_Score of 0.1515.

Table 8. The classification report for the Motion_Light dataset.

Dataset	Models	Accuracy	Precision	Recall	F1_Score
Motion_Light	Bagging	0.9601	0.8875	0.8872	0.8869
	XGBoost	0.9719	0.9531	0.8957	0.9030
	RF	0.9541	0.8831	0.8818	0.8823
	ET	0.9537	0.8824	0.8808	0.8815
	Ada	0.4695	0.1144	0.2241	0.1515

Table 9 shows the results of the performance evaluation metrics in terms of mean values for the ML-based ensemble models applied to the IIoT Garage_Door sensor device dataset. For this dataset, the XGBoost classifier outperforms all other models, with an accuracy of 0.9846, precision of 0.9796, recall of 0.9902, and F1_Score of 0.9847, respectively. The worst classifier is the Ada, with an accuracy of 0.4715, precision of 0.0589, recall of 0.1250, and F1_Score of 0.0801, respectively.

Table 9. The classification report for the IIoT Garage_Door dataset.

Dataset	Models	Accuracy	Precision	Recall	F1_Score
Garage_Door	Bagging	0.9804	0.9789	0.9828	0.9808
	XGBoost	0.9846	0.9796	0.9902	0.9847
	RF	0.9773	0.9774	0.9792	0.9783
	ET	0.9772	0.9771	0.9794	0.9783
	Ada	0.4715	0.0589	0.1250	0.0801

Table 10 presents the mean values of the accuracy, F1-score, precision, and recall assessor for the candidate ML models applied to the Weather device dataset. The XGBoost classifier outperforms all other models for this dataset, with an accuracy of 0.9878, precision of 0.9788, recall of 0.9896, and F1_Score of 0.9840, respectively. The Ada classifier is the

worst among all the classifiers, with an accuracy of 0.5311, precision of 0.0664, recall of 0.1250, and F1_Score of 0.0867 metrics, respectively.

Table 10. The classification report for the Weather dataset.

Dataset	Models	Accuracy	Precision	Recall	F1_Score
Weather	Bagging	0.9850	0.9780	0.9868	0.9823
	XGBoost	0.9878	0.9788	0.9896	0.9840
	RF	0.9842	0.9780	0.9876	0.9827
	ET	0.9807	0.9763	0.9834	0.9798
	Ada	0.5311	0.0664	0.1250	0.0867

Summarily, it is observed from the outcomes presented in Sections 4.2 and 4.3 that the XGBoost classifier gave a superior performance score for all assessment indicators across all the datasets. In contrast, the Adaboost ensemble classifier performed worst based on the assessment indicators across all datasets. Based on the recall rate of all models across all datasets, the XGBoost classifier performed the least on the 'Motion_Light" dataset, with the least value of 0.8957.

Based on the baseline study, the proposed study also evaluates IoT/IIoT-based dataset by combining all of the individual datasets for each device into the collective IoT dataset. Since most real-time apps save their data in a single location, this may represent some real situations. The combined IoT dataset assessed the proposed ensemble classifiers for binary and multi-class classification issues. It is worth mentioning that most studies that use the datasets combine the whole datasets to become one before applying classifiers to them. Table 11 shows the results of the binary classification of the ensemble classifiers enabled with feature selection mothed.

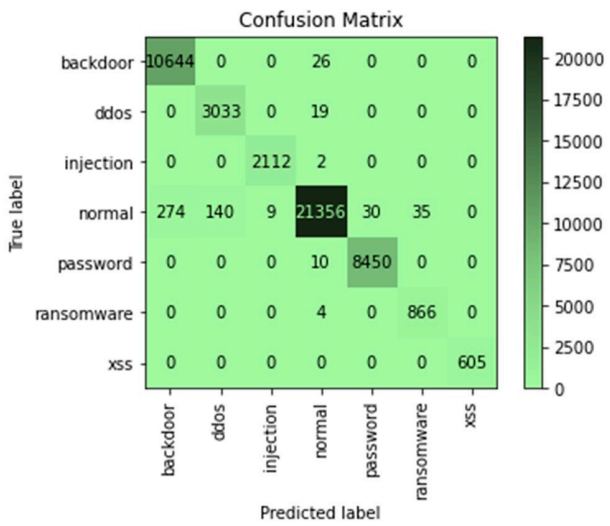
Table 11. Combined_IoT_Dataset Evaluation of Binary Classification Models.

Models	Accuracy	Precision	Recall	F1_Score
Bagging	0.9980	0.9935	0.9931	0.9948
XGBoost	1.000	0.9995	0.9979	0.9975
RF	0.9899	0.9916	0.9904	0.9898
ET	0.9899	0.9883	0.9868	0.9802
Ada	0.6954	0.6912	0.6615	0.5969

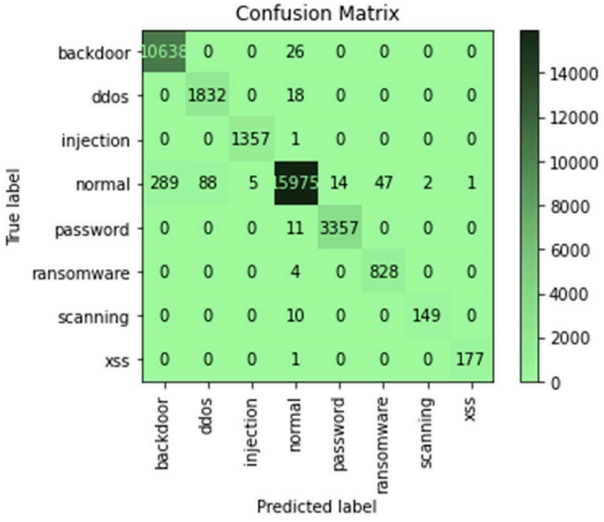
Both the overall IoT dataset and the per-device IoT dataset were used to test the ensemble classifiers. The proposed ensemble classifiers, enabled with the feature selection model, were also evaluated using the performance measures. The findings are summarized in Table 11: XGBoost receives the maximum accuracy rating of 1.0 and scores about 98% across the other performance metrics. From the results obtained, it was noted that all the classifiers scored about 97% across all the performance metrics except Adaboost, which scored less than 70% across all the performance metrics. The performance can be attributed to the application of feature selection to remove the irrelevant features before the classification of the dataset using the ensemble classifiers.

4.2. Results Based on Confusion Matrix

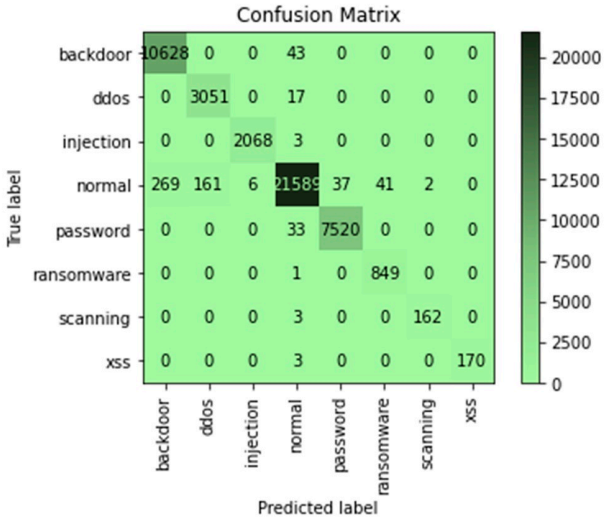
The confusion matrix of the XGBoost algorithm for all datasets was considered in this study. This is the only model considered because of its superior performance. This further analysis is desired to show the commonly misclassified attacks. Only 20% were used as test sets for each of the datasets. It could be observed from Figure 2a–g that the 'normal' class is often misclassified from all classes, and major misclassification occurs between the 'normal' class and the 'backdoor' attack class for all the datasets. For Figure 2a, the misclassification of the 'normal' class also occurs with 'DDoS', 'ransomware', 'password', and 'injection' attack classes, respectively. Incorrect classification also occurs between 'password' and 'ransom' attacks. This pattern could be observed through all the confusion matrices for all datasets.



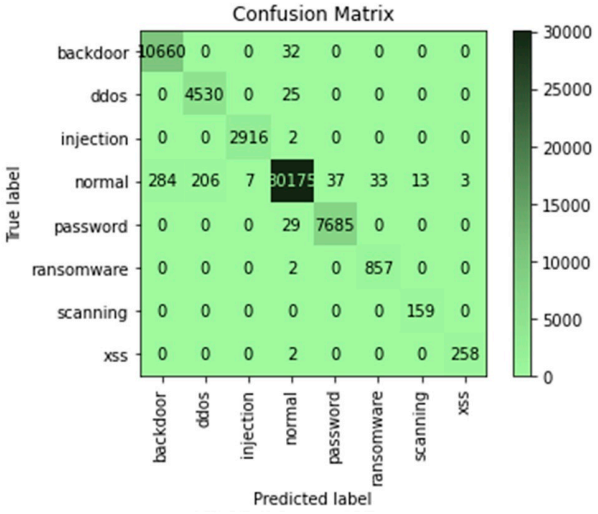
(a) XGBoost Fridge



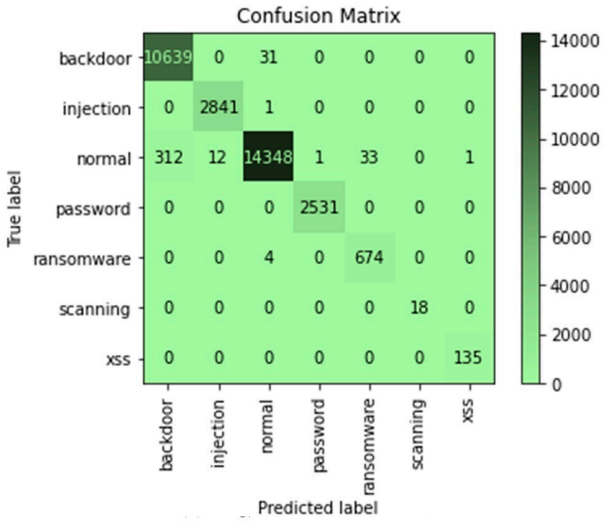
(b) XGBoost-Garage



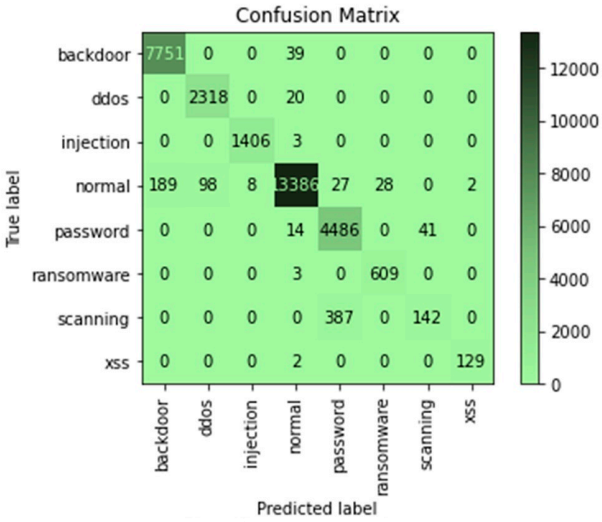
(c) XGBoost-GPS



(d) XGBoost -Weather

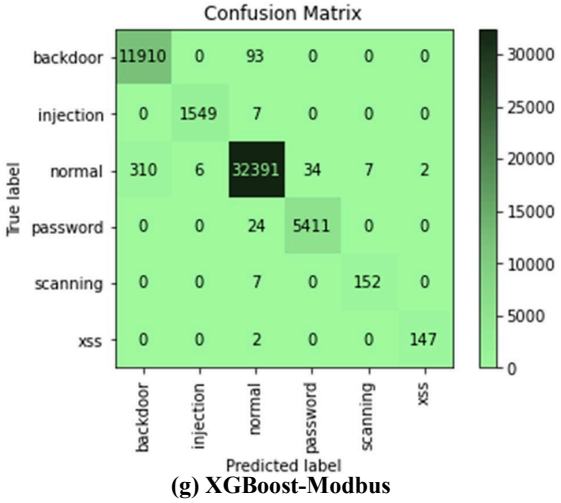


(e) XGBoost -Thermostat



(f) XGBoost - MotionLight

Figure 2. *Cont.*

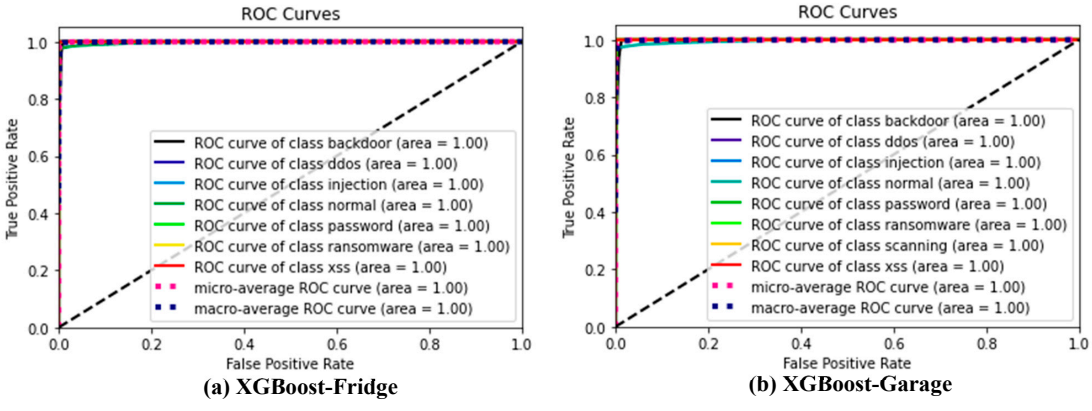


(g) XGBoost-Modbus

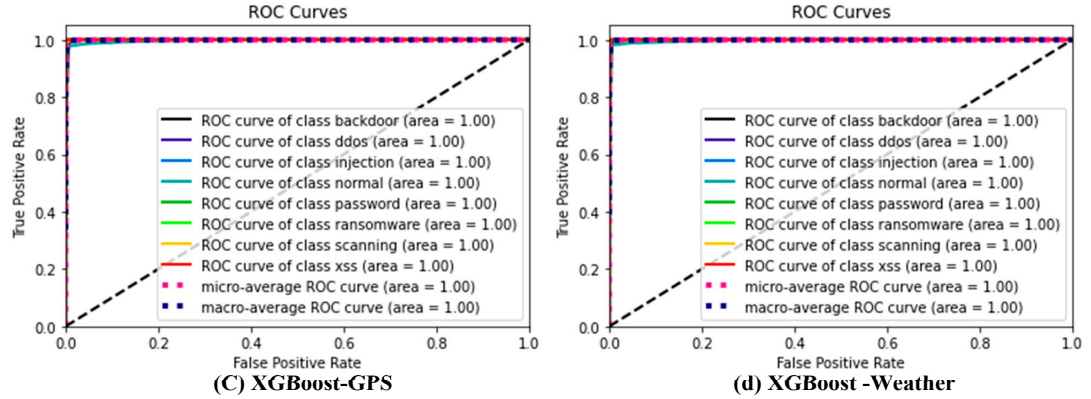
Figure 2. (**a**–**g**) XGBoost confusion matrix for all datasets considered in this study.

4.3. Results Based on ROC Curve

Figure 3a–g show the ROC curve for the XGBoost ensemble ML on the ToN-IoT datasets, respectively. This displays a model ensemble with a ROC curve near the upper left corner and strong separability. The probability value of this assessor ranges between 0 and 1. A good value for this assessor will be closer to 1. Hence, the ROC curve of the XGBoost algorithm for all datasets considered in this study is displayed in Figure 3, respectively. This is the only model considered because of its superior performance. This further analysis is desired to show the commonly misclassified attacks. For each of the datasets, only 20% were used as a test set.



(a) XGBoost-Fridge **(b) XGBoost-Garage**



(C) XGBoost-GPS **(d) XGBoost -Weather**

Figure 3. *Cont.*

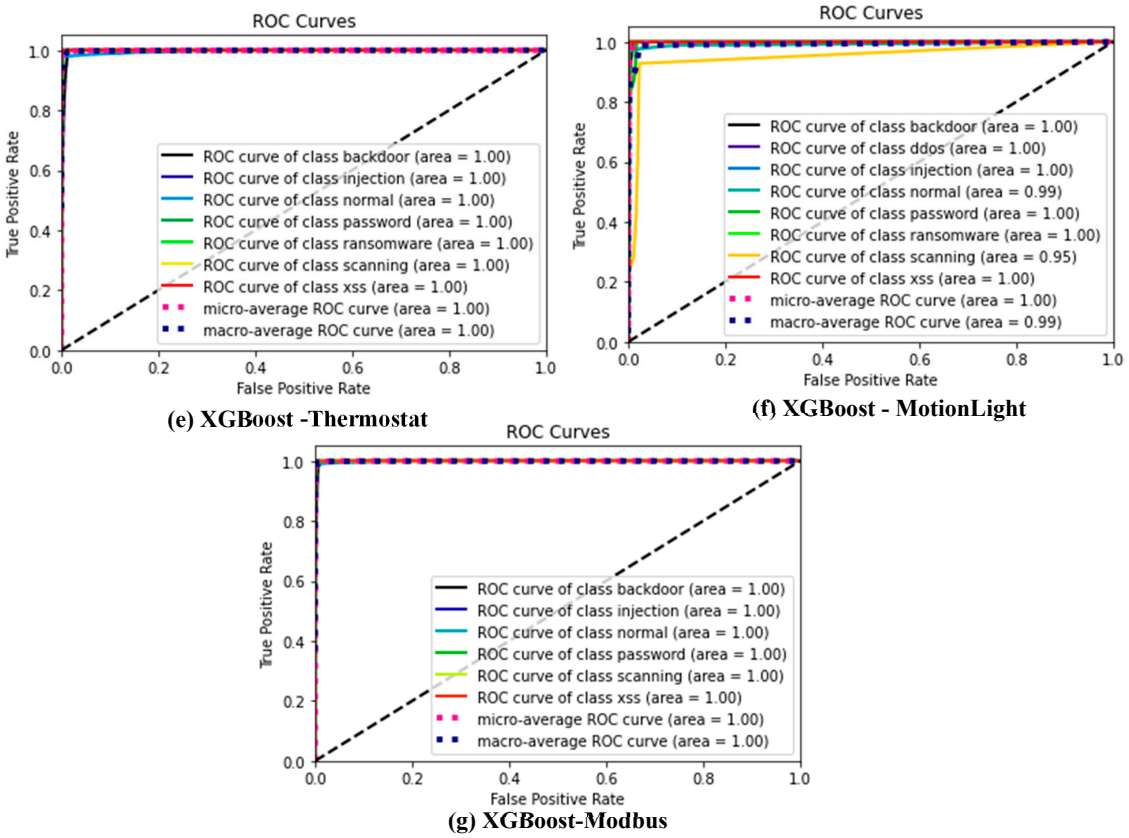


Figure 3. (**a**–**g**) XGBoost ROC curve for all datasets considered for this study.

4.4. Comparative Study with the Baseline Model

To assess how the proposed model performs with the baseline models, the proposed model and baseline models are placed side by side. Table 8 shows the comparison of the proposed model with the baseline model. In [3], CART performs better when compared with other ML models used for the classification of the dataset with 88.0%, and LR and SVM models have the least accuracy, with 61.0% each. CART has the overall best performance across all the performance metrics used to evaluate the datasets. Therefore, it can be said that the proposed model, using feature selection with ensemble classifiers, performs better than the baseline models. In addition, the computational time of the proposed models is very fast, since the number of parameters used is reasonably reduced compared with the baseline model. Table 12 shows the comparison of the proposed model with the baseline model.

Table 12. The proposed model is compared with the baseline model.

	Models	Accuracy	Precision	Recall	F1_Score
Baseline Model [12]	LR	0.61	0.37	0.61	0.46
	LDA	0.68	0.74	0.68	0/62
	KNN	0.84	0.85	0.84	0.84
	RF	0.85	0.87	0.85	0.85
	CART	0.88	0.90	0.88	0.88
	NB	0.66	0.63	0.62	0.51
	SVM	0.61	0.37	0.61	0.46
	LSTM	0.81	0.83	0.81	0.80
Proposed Model	Bagging	0.99	0.99	0.99	0.99
	XGBoost	1.0	1.0	1.0	1.0
	RF	0.99	0.99	0.99	0.99
	ET	0.99	0.99	0.99	0.98
	Ada	0.70	0.69	0.66	0.60

4.5. Comparison with other Existing Models Using the Same Dataset

To emphasize how crucial, it is to use feature selection on the dataset before applying classification models, the baseline and other existing techniques were utilized to compare the proposed model to them. Table 13 compares the outcomes obtained from IDS models proposed with other existing state-of-the-art models based on ToN-IoT datasets. Each row of Table 13 shows a group of various ML-based models from some other notable studies. The use of the ToN-IoT dataset is quite recent in the study of IDSs. Hence, the number of studies associated with network security is very few in this dataset. Worthy of being mentioned is the work of [3], which is the baseline study, where eight (8) ML models were considered. Still, LSTM performed better on four (4) of the datasets. CART performed best on two (2) datasets, and k-NN performed best on one (1) dataset based on accuracy metrics. In [36], the authors used six different ML-based models to classify the dataset, and the study results revealed that the RF classifier performed better on five (5) datasets.

Table 13. Accuracy comparison of existing results based on ToN-IoT datasets.

References	Model	Fridge	Garage_Door	GPS_Tracker	Modbus	Motion_Light	Thermostat	Weather
[61]	DT, RF, Ada, XGBoost, ANN, MLP	RF = 99.56	DT = 99.6	RF = 99.55%	RF = 99.27	DT = 99.54	RF = 99.67	RF = 99.55
[62]	VC, RF, ANN, 1D CNN	–	VC = 99.7	VC = 99.7	VC = 99.7	RF = 99.3	VC = 99.7	RF = 99.3
[63]	CB, RF, Gboost, HBboost	CB = 94.4	CB = 94.5	CB = 94.4	CB = 93.8	CB = 95.4	CB = 93.8	CB = 95.7
[12]	LR, LDA, k-NN, RF, CART, NB, SVM, LSTM	LSTM = 1.00	LSTM = 1.00	k-NN = 0.88	CART = 0.98	LSTM = 0.59	LSTM = 0.66	CART = 0.87
This Study	XGBoost, Bag, RF, ET, Ada	99.73	98.46	98.69	99.13	97.19	98.83	98.78

In contrast, DT performed best on two (2) datasets. The authors in [37] make use of two ML-based models for the classification of the datasets, and the results show that the VC classifier performed better in four (4) datasets, while RF did very well in two (2) datasets. In a similar work by authors in [38], they used six (6) ensemble classifiers for IDS detection. The CB classifier has the best accuracy across all the types of datasets in the TON_IoT Telemetry Dataset used to test the performance of the classifiers used.

Therefore, the comparison in Table 13 revealed that the proposed model performs reasonably better in terms of accuracy when compared with other existing classifiers and the baseline model. As a result, the model performs best when applied to a real-world IIoT ecosystem that contains vast amounts of unstructured and unlabeled datasets. Furthermore, feature selection considerably reduces the computational time needed to process the dataset compared to the baseline. Consequently, data dimensionality is automatically reduced, and high-level functioning is examined efficiently and precisely. Although Table 13 shows that our results appear to be comparable to other research in the field, our suggested approach has been examined on a more pertinent dataset using feature selection to see how the dataset will respond to the model. So, compared to other relevant research, our results would be more trustworthy.

5. Conclusions

In order to protect the IIoT environment from outside attackers and intruders, several IDSs techniques, linked with IIoT-based network traffic, have been proposed and have emerged as essential parts of the technology for proper protection from outsiders. When used, in conjunction with ML-based classifiers, big data as a potent tool for studying massive amounts of data to safeguard IIoT equipment. The technologies have shown to be beneficial for IioT-based systems security measures. Industrial Automation and Control Systems and conventional IT systems are fundamentally different in how they counter

cyberattacks, yet these differences are distinct. Security for the IioT must, therefore, be given specific consideration. Therefore, this study attempts to build an efficient multiclass IDS system based on ML-based ensemble models: XGBoost, Bagging, Random Forest, Extra Trees, and AdaBoost, based on seven (7) paremeters, were used for the Telemetry dataset of ToN_IoT datasets. An empirical experiment is performed on the dataset. The outcome, based on a comparative study, indicates that the proposed model performs excellently, and XGBoost performed superior to other models. The outcomes from the analysis showed that the proposed system could effectively and accurately classify different attacks. One of the major limitations of the proposed model is the inability to deal with the class imbalance that arises from the datasets used to test the performance of the proposed model. Therefore, future work will make use of imbalanced algorithms to balance the dataset. This will enable us to know if the imbalance will affect the performance of the proposed model. Future work will further focus on applying deep learning models to optimize their hyper-parameters to improve the dataset classification performance for the IDS. The proposed model will be applied to other IIoT-based datasets.

Author Contributions: The manuscript was written through the contributions of all authors. Conceptualization, J.B.A., S.O.F. and J.O.O.; methodology, J.B.A., S.O.F. and J.O.O.; software, J.O.O. and J.B.A.; validation, A.L.I., D.-T.D. C.-C.L. and C.-T.L.; formal analysis, A.L.I.; investigation, J.B.A.; resources, C.-C.L., C.-T.L. and D.-T.D.; data curation, J.B.A., S.O.F., A.L.I. and J.O.O.; writing—original draft preparation, J.B.A.; writing—review and editing, A.L.I., C.-C.L., C.-T.L. and D.-T.D.; visualization, J.B.A., A.L.I., S.O.F. and J.O.O.; supervision, J.B.A.; project administration, J.B.A., S.O.F., A.L.I., C.-C.L., C.-T.L. and D.-T.D.; funding acquisition, J.B.A. and A.L.I. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Agbotiname Lucky Imoize is supported in part by the Nigerian Petroleum Technology Development Fund (PTDF) and, in part, by the German Academic Exchange Service (DAAD) through the Nigerian–German Postgraduate Program under grant 57473408.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data related to the outcome of this study are available upon reasonable request from the first author.

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

Abbreviations

ET	Extra Trees
LSTM	Long Short-Term Memory
Bag	Bagging
CB	Cat Boosting
LDA	Linear Discriminant Analysis
GHboost	HistGradient boosting
LGBM	Light Gradient Boosting Method
Gboost	Gradient boosting
XGBoost	eXtreme Gradient Boosting
K-NN	k-Nearest Neighbour
Ada	AdaBoost
CART	Classification and Regression Trees
RF	Random Forest
IoT	Internet of Things
IIoT	Industrial Internet of Things
DNN	Deep Neural Network
DBN	Deep Belief Network
Bi-LSTM	Bidirectional LSTM
ELM	Extreme Learning Machine

References

1. Chifor, B.C.; Bica, I.; Patriciu, V.V.; Pop, F. A security authorization scheme for smart home Internet of Things devices. *Future Gener. Comput. Syst.* **2018**, *86*, 740–749. [CrossRef]
2. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging boosting, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern.* **2012**, *42*, 463–484. [CrossRef]
3. Awotunde, J.B.; Jimoh, R.G.; Folorunso, S.O.; Adeniyi, E.A.; Abiodun, K.M.; Banjo, O. Privacy and security concerns in IoT-based healthcare systems. In *The Fusion of Internet of Things, Artificial Intelligence, and Cloud Computing in Health Care*; Springer: Cham, Switzerland, 2021; pp. 105–134.
4. Folorunso, S.O.; Awotunde, J.B.; Adeniyi, E.A.; Abiodun, K.M.; Ayo, F.E. Heart Disease Classification Using Machine Learning Models. In *Informatics and Intelligent Applications (ICIIA 2021)*; Springer: Cham, Switzerland, 2021; pp. 35–49.
5. Sarhan, M.; Layeghy, S.; Moustafa, N.; Gallagher, M.; Portmann, M. Feature extraction for machine learning-based intrusion detection in IoT networks. *Digit. Commun. Netw.* **2022**, *2022*, 1–17. [CrossRef]
6. Awotunde, J.B.; Chakraborty, C.; Adeniyi, A.E. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 7154587. [CrossRef]
7. Awotunde, J.B.; Misra, S.; Adeniyi, A.; Abiodun, M.; Kaushik, M.; Lawrence, M.O. A Feature Selection-Based K-NN Model for Fast Software Defect Prediction. In *International Conference on Computational Science and Its Applications*; Springer: Cham, Switzerland, 2022; pp. 49–61.
8. Zargar, S.T.; Joshi, J.; Tipper, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun. Surveys Tuts.* **2013**, *15*, 2046–2069. [CrossRef]
9. Nimbalkar, P.; Kshirsagar, D. Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express* **2021**, *7*, 177–181. [CrossRef]
10. AbdulRaheem, M.; Oladipo, I.D.; González-Briones, A.; Awotunde, J.B.; Tomori, A.R.; Jimoh, R.G. *An efficient lightweight speck technique for edge-IoT-based smart healthcare systems." In 5G IoT and Edge Computing for Smart Healthcare*; Academic Press: Cambridge, MA, USA, 2022; pp. 139–162.
11. Miller, L. *IoT Security for Dummies*; Johnson, C.A., Ed.; John Wiley and Sons Ltd.: Hoboken, NJ, USA, 2016.
12. Alsaledi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
13. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network intrusion detection for IoT security based on learning techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
14. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2923–2960. [CrossRef]
15. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
16. Kirana, K.V.V.N.L.S.; Devisetty, R.N.K.; Kalyan, N.P.; Mukundini, K.; Karthi, R. Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques. In Proceedings of the The Third International Conference and Network Communications (CoCoNet'19), Trivandrum, India, 18–21 December 2019.
17. Islam, N.; Farhin, F.; Sultana, I.; Kaiser, M.S.; Rahman, M.S.; Mahmud, M.; Hosen, A.S.M.S.; Cho, G.H. Towards Machine Learning Based Intrusion Detection in IoT Networks. *Comput. Mater. Contin.* **2021**, *69*, 1801–1821. [CrossRef]
18. Parmisano, A.; Garcia, S.; Erquiaga, M. *Iot-23 Dataset: A Labeled Dataset of Malware and Benign Iot Traffic*; Avast-AIC Laboratory, Stratosphere IPS, Czech Technical University (CTU): Prague, Czechia, 2019.
19. Kaggle. Iot Device Network Logs. 2020. Available online: https://www.kaggle.com/speedwall10/iotdevice-network-logs (accessed on 5 June 2022).
20. Pahl, M.; Aubet, F. All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection. In Proceedings of the 14th International Conference on Network and Service Management, Rome, Italy, 5–9 November 2018.
21. Kang, H.; Ahn, D.H.; Lee, G.M.; Yoo, J.D.; Park, K.H.; Kim, H.K. IoT network intrusion dataset. *IEEE Dataport* **2019**, *10*, q70p–q449.
22. Ullah, I.; Mahmoud, Q.H. A technique for generating a botnet dataset for anomalous activity detection in IoT networks. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics, Toronto, ON, Canada, 11–14 October 2020.
23. Mohamed, R.H.; Mosa, F.A.; Sadek, R.A. Efficient Intrusion Detection System for IoT Environment. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2022**, *13*, 572–578. [CrossRef]
24. Gad, A.R.; Haggag, M.; Nashat, A.A.; Barakat, T.M. A Distributed Intrusion Detection System using Machine Learning for IoT based on ToN-IoT Dataset. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2022**, *13*, 548–563. [CrossRef]
25. Karanfilovska, M.; Kochovska, T.; Todorov, Z.; Cholakoska, A.; Jakimovski, G.; Efnusheva, D. Analysis and modelling of a ML-based NIDS for IoT networks. *Procedia Comput. Sci.* **2022**, *204*, 187–195. [CrossRef]
26. Rashid, M.; Kamruzzaman, J.; Imam, T.; Wibowo, S.; Gordon, S. A tree-based stacking ensemble technique with feature selection for network intrusion detection. *Appl. Intell.* **2022**, *52*, 9768–9781. [CrossRef]
27. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet Things J.* **2018**, *6*, 4815–4830. [CrossRef]

28. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Slay, J. Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. In *International Conference on Mobile Networks and Management*; Springer: Cham, Switzerland, 2017; pp. 30–44.
29. Roopak, M.; Tian, G.Y.; Chambers, J. An intrusion detection system against DDoS attacks in iot networks. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 0562–0567.
30. Diro, A.; Chilamkurti, N. Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Commun. Mag.* **2018**, *56*, 124–130. [CrossRef]
31. Anthi, E.; Williams, L.; Słowińska, M.; Theodorakopoulos, G.; Burnap, P. A supervised intrusion detection system for smart home IoT devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
32. Chen, Y.W.; Sheu, J.P.; Kuo, Y.C.; Van Cuong, N. Design and implementation of IoT DDoS attacks detection system based on machine learning. In *2020 European Conference on Networks and Communications (EuCNC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 122–127.
33. Procopiou, A.; Komninos, N.; Douligeris, C. ForChaos: Real-time application DDoS detection using forecasting and chaos theory in smart home IoT network. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 8469410. [CrossRef]
34. Shrivastava, R.K.; Bashir, B.; Hota, C. Attack detection and forensics using honeypot in IoT environment. In *International Conference on Distributed Computing and Internet Technology*; Springer: Cham, Switzerland, 2019; pp. 402–409.
35. Nanthiya, D.; Keerthika, P.; Gopal, S.B.; Kayalvizhi, S.B.; Raja, T.; Priya, R.S. SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset. In *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
36. Awotunde, J.B.; Misra, S. Feature extraction and artificial intelligence-based intrusion detection model for a secure internet of things networks. In *Illumination of Artificial Intelligence in Cybersecurity and Forensics*; Springer: Cham, Switzerland, 2022; pp. 21–44.
37. 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]
38. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics* **2019**, *8*, 1210. [CrossRef]
39. Yazdinejad, A.; Kazemi, M.; Parizi, R.M.; Dehghantanha, A.; Karimipour, H. An ensemble deep learning model for cyber threat hunting in the industrial internet of things. *Digital Commun. Netw.* **2022**, *2022*, 1–10. [CrossRef]
40. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
41. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
42. Folorunso, S.O.; Afolabi, S.A.; Owodeyi, A.B. Dissecting Genre of Nigerian Music with Machine Learning Models. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *8*, 6266–6279. [CrossRef]
43. Ampomah, E.K.; Qin, Z.; Nyame, G. Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information* **2020**, *11*, 332. [CrossRef]
44. Li, W.; Yin, Y.; Quan, X.; Zhang, H. Gene expression value prediction based on XGBoost algorithm. *Front. Genet.* **2019**, *10*, 1077. [CrossRef] [PubMed]
45. Breinmsn, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140.
46. Breiman, L. Random forests. *BMach Learn.* **2001**, *45*, 5–32. [CrossRef]
47. Singh, N.; Bhatnagar, S. Machine Learning for Prediction of Drug Targets in Microbe Associated Cardiovascular Diseases by Incorporating Host-pathogen Interaction Network Parameters. *Mol. Inform.* **2022**, *41*, 2100115. [CrossRef]
48. Zhou, Y.; Qiu, G. Random forest for label ranking. *Expert Syst. Appl.* **2018**, *112*, 99–109. [CrossRef]
49. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
50. Freund, Y.; Schapire, R. Experiments with a new boosting algorithm. In Machine Learning. In Proceedings of the Thirteenth International Conference of Machine Learning (ICML '96), Bari, Italy, 3–6 July 1996.
51. Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: A723 statistical view of boosting. *Ann. Stat.* **2000**, *28*, 337–374. [CrossRef]
52. Krupp, J.; Backes, M.; Rossow, C. Identifying the scan and attack infrastructures behind amplification DDoS attacks. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016.
53. Lyon, G.F. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*; Insecure: Rothwell, IN, USA, 2009.
54. Nessus, Nessus: A Secure Vulnerability Scanning Tool. Available online: https://www.cs.cmu.edu/dwendlan/personal/nessus.html (accessed on 1 July 2022).
55. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.T. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
56. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Comput. Secur.* **2018**, *74*, 144–166. [CrossRef]
57. Kolias, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 184–208. [CrossRef]

58. Zolanvari, M.; Teixeira, M.A.; Gupta, L.; Khan, K.M.; Jain, R. Machin learning-based network vulnerability analysis of industrial Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 6822–6834. [CrossRef]
59. Nelso, T.; Chaffin, M. Common Cybersecurity Vulnerabilities in Industrial Control Systems. In *Control Systems Security Program*; Department of Homeland Security (DHS), National Cyber Security Division: Washington, DC, USA, 2011.
60. Folorunso, S.O.; Awotunde, J.B.; Adeboye, N.O.; Matiluko, O.E. Data Classification Model for COVID-19 Pandemic. In *Advances in Data Science and Intelligent Data Communication Technologies for COVID-19*; Hassanien, A.E., Elghamrawy, S.M., Zelinka, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; Volume 378, pp. 93–118.
61. Tasnim, A.; Hossain, N.; Parvin, N.; Tabassum, S.; Rahman, R.; Hossain, M.I. Experimental Analysis of Classification for Different Internet of Things (IoT) Network Attacks Using Machine Learning and Deep learning. In Proceedings of the International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 23–25 March 2022.
62. Tanzila, S.; Amjad, R.K.; Tariq, S.; Seng-phil, H. Securing the IoT System of Smart City against Cyber Threats Using. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 1241122.
63. Rani, D.; Gill, N.S.; Gulia, P.; Chatterjee, J.M. An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things. *Comput. Intell. Neurosci.* **2022**, *2022*, 1668676. [CrossRef]