International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 4

Machine Learning Methods for Detecting Internet-of-Things (IoT) Malware

Winfred Yaokumah, University of Ghana, Ghana
https://orcid.org/0000-0001-7756-1832

Justice Kwame Appati, University of Ghana, Ghana
https://orcid.org/0000-0003-2798-4524

Daniel Kumah, Hightel Consults Ltd., Ghana
https://orcid.org/0000-0002-2387-0821

ABSTRACT

This study aims to analyze the performance of machine learning models for detecting internet of things malware utilizing a recent IoT dataset. Experiments on the IoT dataset were conducted with nine well-known machine learning techniques, consisting of logistic regression (LR), naive bayes (NB), decision tree (DT), k-nearest neighbors (KNN), support vector machines (SVM), neural networks (NN), random forest (RF), bagging (BG), and stacking (ST). The results show that the proposed model attains 100% accuracy in detecting IoT malware for DT, SVM, RF, BG; about 99.9% percent for LR, NB, KNN, NN; and only 28.16% for ST classifier. This study also shows higher performance than other proposed machine learning models evaluated on the same dataset. Therefore, the results of this study can help both the researchers and application developers in designing and building intelligent malware detection systems for IoT devices.

KEYWORDS

Classification, Data Theft, Distributed Denial of Service, Internet of Things, IoT Dataset, Keylogging, Machine Learning Algorithm, Malware, OS Fingerprinting, Reconnaissance, Service Scanning

INTRODUCTION

Internet of Things (IoT) is a collection of interconnected devices embedded with light processors and network cards capable of being managed over the Internet (Moustafa, Turnbull, & Choo, 2018a). It represents a network of physical objects (things) embedded with sensors, software, and other technologies to exchange data with other devices and systems (Rouse, 2019). Internet of Things comprises typical network elements (workstations, laptops, and routers), sensors, actuators, smart devices, and radio frequency identification (RFID) devices (Gubbi et al., 2013). Recent developments in IoT technologies have brought about improvements in consumer products,

DOI: 10.4018/IJCINI.286768

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

commercial applications, industrial devices, and other applications for critical infrastructure protection (Khraisat & Alazab, 2021).

In consumer products, IoT devices are used in smart vehicles (networks of moving vehicles), home automation (smart homes), smart cities, wearable devices, and appliances with remote monitoring capabilities (Guhathakurta, 2017). For commercial usage, IoT is applied in medical and health-related applications (Internet of Medical Things, IoMT) for data collection and analysis for research and monitoring (Da Costa et al., 2018; Engineer, Sternberg, & Najafi, 2018). Also known as the Industrial Internet of Things (IIoT), IoT connects industrial devices to acquire and analyze data from connected equipment, operational systems, remote locations, and people equipped with wearable devices. Likewise, in infrastructure, IoT applications monitor and control critical infrastructures like bridges and railway lines (Gubbi et al., 2013). Additionally, the Internet of Military Things (IoMT) applications are deployed in the military domain for national security, reconnaissance, monitoring, and surveillance. Considering its broad scope of applicability, Lu and Xu (2019) predict that the number of IoT devices connected to the internet will reach 25 billion by 2020.

The critical challenge of IoT systems is that they are vulnerable to security threats (Ahmad et al., 2021). The Internet of Things technologies are exposed to severe cybersecurity threats (Singh et al., 2015) and major privacy violations (Howard, 2015). According to Gubbi et al. (2013), the vulnerabilities of IoT networks will significantly increase with complex botnets and denial of service attacks. This problem is compounded because IoT systems are low-powered devices with severe operational limitations and computational power to mitigate malware attacks (Moustafa, Turnbull, & Choo, 2018b). Consequently, malware poses threats to IoT devices' availability and reliable operation, leading to grave security risks (Nakhodchi, Upadhyay, & Dehghantanha, 2020). Furthermore, a series of attacks target the IoT networks to degrade performance and breach their security using malicious software. Therefore, it is crucial to design robust and accurate methods to mitigate the adverse effect of such attacks (Peters et al., 2020).

Though machine learning (ML) techniques are employed to detect and prevent malware in network systems, the design of an effective ML model that can accurately detect and prevent IoT malware is still a challenging task. Besides, the unavailability of new IoT datasets poses challenges to the development and design of IoT malware intrusion detection systems. With few datasets available, it is critical to evaluate them to ascertain their effectiveness in the design of malware intrusion detection systems. A study that evaluates the Bot-IoT dataset with ML algorithms recognizes the importance of appropriate feature selection and engineering approaches to improve ML models for effective malware intrusion detection (Ferrag & Maglaras, 2019). However, the inability to select the optimum features of datasets leads to relatively low-performance outcomes of ML models recently proposed in the literature (Alsamiri & Alsubhi, 2020; Shafiq et al., 2020). Accordingly, the goal of this study is to (a) perform feature selection and engineering on a recent BoT-IoT dataset, (b) evaluate the performance of ML algorithms for IoT malware detection, and (c) compare the results of the study with recent similar studies. Feature selection and engineering are essential factors in machine learning as they can increase the predictive power of ML algorithms. Feature engineering involves understanding the domain knowledge of the dataset to create new features or combine existing features that make ML algorithms perform better (Butcher & Smith, 2020). Hence, the contribution of this study is the feature selection and engineering processes on the BoT-IoT dataset in an attempt to improve the performance of ML algorithms. The feature selection is achieved by the use of four statistical approaches to extract the most relevant features.

In achieving the stated objectives, nine ML algorithms are evaluated using the Bot-IoT dataset. The Bot-IoT dataset combines legitimate and simulated IoT network traffic containing ten different types of attacks. Three ensemble and six non-ensemble machine learning algorithms are assessed. The ensemble algorithms include Random Forest (RF), Bagging (BG), and Stacking (ST). The non-ensemble methods comprise Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), k-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Neural Network (NN). The

study employs the most important performance metrics, namely precision, recall, F-measure, kappa, receiver operating characteristic (ROC), mean absolute error (MAE), and the root mean squared error (RMSE) for the evaluation of the machine learning algorithms.

LITERATURE REVIEW

Machine Learning Algorithms

Machine learning algorithms are programs that learn from data and improve their experience without human intervention (Dataquest Reviews, 2020). There are four primary forms of machine learning algorithms. These are supervised, semi-supervised, unsupervised, and reinforcement learning algorithms. Supervised learning algorithms consist of an outcome variable (or dependent variable), which is predicted from a given set of predictors (independent variables) (Tsiligkaridis & Paschalidis, 2017). In supervised learning, both the input and output variables are labeled. Examples of supervised learning algorithms are Decision Tree, KNN, Linear Regression, Support Vector Machine, and Logistic Regression. In contrast to supervised ML algorithms, semi-supervised learning uses labeled and unlabeled data (Wakefield, 2020). There is no target or outcome variable for unsupervised learning algorithms to predict, but rather the algorithms learn the hidden structure in unlabeled data. Examples of unsupervised learning algorithms are K-Means, Hierarchical clustering, and the Apriori algorithm. In the reinforcement learning algorithms, the machine is trained to make specific decisions. The machine is exposed to an environment where it trains itself continually using trial and error. As it does so, it learns from experience and tries to capture the best possible knowledge to make accurate decisions (Tsiligkaridis & Paschalidis, 2017).

Machine learning approaches can also be classified as linear, nonlinear, and ensemble algorithms. The linear ML algorithms assume that the predicted attribute is a linear combination of the input attributes. Examples are the Linear Regression and Logistic Regression. Linear Regression establishes a relationship between one dependent variable and one independent variable (Simple Linear Regression) or between one dependent variable and two or more independent variables (Multiple Linear Regression). Used for classification problems, Logistic Regression estimates the probability of an event occurring based on the previous data provided with a binary dependent variable (0 or 1). By contrast, the nonlinear ML algorithms do not make strong assumptions about the linear relationship between the input attributes and the output attribute being predicted. Examples are Naive Bayes, Decision Tree, k-Nearest Neighbors, Support Vector Machines, and Neural Networks. The Naïve Bayes is a classification method based on Bayes' theorem assuming independence between predictors. In short, the Naive Bayes classifier assumes that the presence of a particular attribute in a class is not related to the presence of any other attribute (Huang, Zhu, & Siew, 2016). Therefore, a Naive Bayes classifier is very easy to build than others and handy with large datasets. However, the Decision Tree is used for both classification and regression problems. A decision tree is a flow-chart-like tree structure that uses a branching method to illustrate every possible outcome of a decision. Each node within the tree represents a test on a specific variable – and each branch is the outcome of that test (Wakefield, 2020).

Likewise, the k-Nearest Neighbor is an algorithm that is used for both classification and regression problems. It is a straightforward algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors (Tsiligkaridis & Paschalidis, 2018). It uses distance functions for measuring the k-nearest neighbors. The distance function can be Manhattan, Minkowski, Euclidean, or Hamming distance. Hamming functions are used for categorical variables, whiles Manhattan, Minkowski, and Euclidean are used for continuous function. Similar to k-Nearest Neighbors, the Support Vector Machine is used for classification and regression analysis. They essentially filter data into categories, which is achieved by providing a set of training data, each set marked as belonging to one or the other set of the two categories. The algorithm then works to build a model that assigns new values to one category or the other (Wakefield, 2020). Also, the Neural Network is an algorithm

that depicts how the human brain performs its functions. It consists of neurons that provide output from a given input layer. It comes with a hidden layer, where most of the training is done using a sigmoid function.

Besides, the ensemble machine learning algorithms combine the predictions from multiple models to make more robust predictions (Liu, Hao, & Chen, 2020). The main idea behind the ensemble method is to group all weak learners to form a strong learner, thereby increasing the model's accuracy. The common types of ensemble methods are Random Forest, Bagging, and Stacking or Blending. The Random Forest collects a specific amount of decision trees and merges them (Simon, 2011). Random Forest combines multiple algorithms to generate better results for classification, regression, and other tasks. Each classifier is weak, but when combined with others, can produce excellent results. The algorithm starts with a 'decision tree' (a tree-like graph or model of decisions) and an input is entered at the top. It then travels down the tree, with data being segmented into smaller and smaller sets based on specific variables (Wakefield, 2020). In a bagging method, several instances of the same base model are trained in parallel (independently of each other) on different bootstrap samples and then aggregated in some kind of "averaging" process (Reddy et al., 2021). Finally, in stacking methods, different weak learners are fitted independently of each other with a meta-model trained to predict outputs based on the outputs returned by the base models (Rocca, 2019).

Related Works

Previous works in malware intrusion detection applied various machine learning algorithms on malware datasets. As mentioned earlier, IoT devices are lightweight and low-powered devices and have limited computational power (Moustafa, Turnbull, & Choo, 2018b) to run conventional antimalware programs. As a result, studies are ongoing to resolve these challenges. Alhanahnah (2018) sought to overcome this challenge by creating methods of IoT malware detection that could run efficiently irrespective of platform and yet remained lightweight despite resource constraints. The proposed solution was designed by developing lightweight signatures from high-level code. Using analytical approaches, the proposed solution was shown to have a detection rate of 85.2% with zero false positives. The study showed the efficacy of the signature generation system. Also, Ngo and Nguyen (2020) studied the rise of malware targeted at IoT devices and improved the efficiency of existing malware detection methods. The study reviewed several previous works on IoT security issues. Malware detection methods were discussed extensively, with the advantages and disadvantages compared and contrasted. Using tabular comparisons, the study found that the ELF-header method produced a low false detection rate of 0.2%. Also, the findings revealed that using the coding system to group malware samples improved the accuracy of detection to over 98%.

Likewise, Su et al. (2018) proposed a lightweight method for the detection and classification of distributed denial of service (DDoS) malware and benign application on IoT devices. The study conducted tests with a convolutional neural network, allowing for the resource-limited IoT devices to remain unconstrained. A 5-fold validation approach was used to test the accuracy of the proposed design. The proposed design predicted malware with an average accuracy of 94%. Moreover, Nguyen et al. (2018) compared the effectiveness of three deep learning-based approaches in detecting IoT malware. The three models were based on features adapted from 1) fixed-sized byte sequences, 2) fixed-size color images, and 3) variable-sized sequences. The variable-sized sequence and the fixed-size color image approach achieved higher accuracy than the fixed-sized byte sequence approach (90.58%). However, the study was considered preliminary, and the authors suggested that further experiments be made to improve the accuracy.

Similarly, Alasmary et al. (2019) experimented on various datasets to determine the similarities and differences between malware on different systems. Using Control Flow Graphs, a detection system was built and used to categorize Android malware, IoT malware, or non-threatening samples. The performance of these models was based on a 10-fold validation technique. The study found that the convolutional neural network (CNN) model could detect IoT malware from benign samples with an

accuracy of 99.66%. Hasan et al. (2019) also evaluated attack and anomaly detection in IoT sensors in IoT sites artificial neural network (ANN), Logistic Regression, and SVM machine learning approaches. The experiment shows that DF is the better technique to use in IoT for intrusion detection systems, with an accuracy of 99.4%.

A few classic previous studies in machine learning and IoT malware detection are summarized in Table 1. Notably, the datasets used in the studies were mainly Bot-IoT, N-BaIoT, and IoT-23. Furthermore, the predominant ML approaches employed were Support Vector Machines (SVM), Decision Tree, Naïve Bayes, K-nearest neighbors (KNN), Random Forest (RF), Boosting, Bagging, and Blending (Stacking).

Table 1. Summary of ML algorithms for IoT malware detection

Author	Purpose	ML Classifier/Data Mining Approach	Data for Evaluation
Alsamiri & Alsubhi (2020)	Evaluated various machine learning algorithms for detection of IoT network attacks.	K-nearest neighbors (KNN) Iterative dichotomiser (ID3) Quadratic discriminant analysis (QDA) Random Forest AdaBoost Multilayer perceptron (MLP) Naïve Bayes (NB)	Bot-IoT
Shobana & Poonkuzhali (2020)	Deployed detection technique to cluster botnet traffic and the normal traffic.	Support Vector Machines (SVM) Decision Tree Naïve Bayes	N-BaIoT
Shafiq,Tian, Bashir, Du, & Guizani (2020)	Designed and developed a new feature selection algorithm.	Decision Tree (C4.5) Support Vector Machine (SVM) Random Forest (RF) Naíve Bayes	BoT-IoT
Das, Ajila, & Lung (2020)	Analyzed accuracies of machine learning algorithms for network intrusion detection.	Random Forest Naïve Bayes Decision Tree	UNSW-NB15 BoT-IoT
Choudhury & Bhowal (2015)	Categorized network traffic using machine learning	BayesNet, Logistic Regression, IBK. J48, PART, JRip, Random Tree. Random Forest, REPTree, Boosting, Bagging, Blending (Stacking)	NSL-KDD
Chesney, Roy, & Khorsandroo (2021)	Assessed the effectiveness of machine learning algorithms for combatting IoT-related cyber-attacks.	Logistic Regression	CICDoS2019
Sarumi, Adetunmbi, & Adetoye (2020)	Compared two intrusion detection systems.	Apriori Support Vector Machine (SVM)	NSL-KDD UNSW-NB15
Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020)	presents an ensemble method that leverages deep models	Deep Neural Network (DNN) Long Short-Term Memory (LSTM) Meta-classifier (i.e., logistic regression)	IoT-23 LITNET-2020 NetML-2020
De La Torre Parra, Rad, Choo, & Beebe (2020)	Propose a cloud-based distributed deep learning framework for phishing and Botnet attack detection and mitigation.	Distributed Convolutional Neural Network (DCNN) Long-Short Term Memory (LSTM)	N_BaIoT

METHODOLOGY

The study makes use of the design science research method with an experimental approach. The implementation consists of three main steps that include (a) selection of IoT dataset (Bot-IoT) and set of machine learning tools, (b) feature engineering involving feature extraction and encoding - attribute conversion and feature combination to optimize learning algorithm and to improve detection performance and the execution time, (c) ML evaluation through classification and comparative analysis of the model with existing similar works.

Dataset Selection and Description

The BoT-IoT dataset was selected to evaluate ML algorithms since it was relatively new and created purposely for the IoT environment. It was created by designing a realistic network environment in the Cyber Range Lab of the Center of UNSW Canberra Cyber (Koroniotis et al., 2019). The dataset contains more than 73 million records, consisting of 46 features. The testbed for the creation of the dataset used five IoT scenarios, namely, weather station (which generates information on air pressure, humidity, and temperature), smart fridge (which measures the fridge's temperature and, when necessary, adjusts it below a threshold), motion-activated lights (which turn on or off, based on a pseudo-random generated signal), remotely activated garage door (which opens or closes, based on a probabilistic input), and smart thermostat (which regulates the temperature a house by starting the air-conditioning system)(Koroniotis et al., 2019). The dataset contains three major attack categories (information gathering, denial of service, and information theft) with ten attack types. The attack types include distributed denial of service (DDoS), denial of service (DoS), Operating system (OS) and Service Scan, Keylogging, and Data exfiltration attacks (see Table 2). The DDoS and DoS attacks were further organized into three protocols – hypertext transfer protocol (HTTP), user datagram protocol (UDP), and transmission control protocol (TCP).

Information gathering (Reconnaissance) attacks are malicious activities that gather information about victims through remote scanning systems. It is subdivided into two types, service scanning and OS fingerprinting. In service scanning, a scanner identifies the services which run behind the

Table 2. Statistics of attacks in IoT-Bot dataset

Attack category	Attack type	No. of records
Information gathering		
	Service scanning	1,463,364
	OS Fingerprinting	358,275
Denial of Service		
	DDoS TCP	19,547,603
	DDoS UDP	18,965,106
	DDoS HTTP	19,771
	DoS TCP	12,315,997
	DoS UDP	20,659,491
	DoS HTTP	29,706
Information theft		
	Keylogging	1,469
	Data theft	118
Total		73,360,900

system's ports by sending request packets. In OS fingerprinting, a scanner gathers information about the remote system's OS by comparing its responses to pre-existing ones or based on differences in transmission control protocol/Internet protocol (TCP/IP) stack implementations (Hoque et al., 2014). Denial of service (DoS) is a malicious activity that attempts to disrupt a service, making it unavailable to legitimate users. The purpose of such attacks is to disrupt the services accessible by legitimate users. DoS attacks can generate many network traffic, which either forces the victim to process these attack-generated requests or cause the machine to crash, thus making the provided service unavailable (Kolias et al., 2017). The attacks can also abuse the mechanics of the internet protocols, which cause the central processing unit (CPU) and memory resources to be depleted, hence rendering the targeted machine unable to respond to requests (Behal & Kumar, 2017).

Additionally, information theft is a group of attacks where an adversary seeks to compromise a machine's security to obtain sensitive data. Information theft attacks can be split into subcategories based on the target of the attack (Jesudoss & Subramaniam, 2014). The first subcategory is data theft. During data theft attacks, an adversary targets a remote machine and attempts to compromise it, thus gaining unauthorized access to data downloaded to the remote attacking machine. The second subcategory is keylogging. In keylogging activities, an adversary compromises a remote host to record a user's keystrokes, potentially stealing sensitive credentials (Tankard, 2011).

Proposed Feature Engineering Approach

As one of the goals of this study, the feature engineering technique was performed to improve the prediction performance of machine learning methods on the IoT-BoT dataset. Feature engineering is the process of selecting, creating, or modifying features (variables) in a dataset to improve predictions made by ML learning algorithms (Domingos, 2012). In this study, a feature engineering process was carried out in stages. In the first stage, four statistical measures based on correlation were used. According to Butcher and Smith (2020), inferential statistical approaches provide a better solution to appraising the contribution of a predictor to the underlying model or the dataset. These measures are Analysis of Variance (ANOVA), Pearson's Correlation Coefficient, Mutual Information, and Chi-Squared. However, due to the dynamics of the data and the varying assumptions of these statistical measures, different results are expected, hence the second stage.

In this second stage, the best k=25 features were selected from each statistical measure. The majority vote was then applied across features resulting in 19 features that best explain the given data. Experimentally, k=40, 35, 30, 25, and 20 were tested with, k=25 showing improved performance. Next is the creation of a variable. Feature creation was performed by modifying variables and creating a new one by combining multiple different variables (Kern, 2014). An example of such an operation is the combination of "attack" and "sub-category" to create a new variable (combined) so that it can be used for classification. The study further converted *sport* and *dport* features in the engineering process, originally string data types, to numeric data types. Table 3 presents the 19 features extracted through the feature selection process and an additional feature obtained through a feature creation process.

Performance Metrics

The following performance metrics were used to evaluate each of the nine ML algorithms on the dataset:

1. Accuracy, TPR (True Positive Rate), Recall, FPR (False Positive Rate), Precision, and F1-score.

$$\text{Accuracy} = \frac{Tp + Tn}{(Tp + Tn + Fp + Fn)}$$
$$TPR = \frac{Tp}{(Tp + Fn)}$$
$$FPR = Fp$$

Table 3. Selected features of Bot-IoT dataset

No.	Feature	Description	No.	Feature	Description
1	pkSeqID	Row Identifier	11	state_number	Numerical representation of feature state
2	Proto	Textual representation of transaction protocols present in network flow	12	Mean	Average duration of aggregated records
3	saddr	Source IP address	13	N_IN_Conn_P_DstIP	Number of inbound connections per destination IP.
4	sport	Source port number	14	srate	Source-to-destination packets per second
5	daddr	Destination IP address	15	drate	Destination-to-source packets per second
6	dport	Destination port number	16	Max	Maximum duration of aggregated records
7	seq	Argus sequence number	17	Attack	Class label: 0 for Normal traffic, 1 for Attack Traffic
8	stddev	Standard deviation of aggregated records	18	Category	Traffic category
9	N_IN_Conn_P_SrcIP	Number of inbound connections per source IP.	19	Subcategory	Traffic subcategory
10	Min	Minimum duration of aggregated records	20	Combine	Combination of category and subcategory

$$\frac{(Tn + Fp)}{\text{Precision} = Tp}$$
$$\frac{(Tp + Fp)}{F1 = 2Tp}$$
$$\frac{(2Tp + Fp + Fn)}{}$$

where Tp, Fp, Tn, Fn indicate true positives, false positives, true negatives, and false negatives, respectively.

2. Receiver Operating Characteristic (ROC) curve – which is a curve with the false positive rate (FPR) as the abscissa and true positive rate (TPR) as the ordinate. The closer the curve is to the (0,1), the more accurate the model classification.
3. The Area under Curve (AUC) indicates the area under the ROC curve, between 0.1 and 1.0. AUC score is the probability that a classification model will rank a randomly chosen positive sample higher than a random negative one. A larger value indicates a better classification model.

Experimental Setup

The experimental setup comprised a laptop with a 64-bit Intel Core-i7 CPU running at 2.20GHz with 16GB RAM. The simulation was carried out with Waikato Environment for Knowledge Analysis (WEKA) to run the selected machine learning algorithms. WEKA is a popular suite of machine

learning software written in Java, developed at the University of Waikato, New Zealand. The Weka workbench contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality (University of Waikato, 2018). WEKA has been used in earlier studies to detect intrusion in network systems (Alsamiri & Alsubhi, 2020; Ferrag & Maglaras, 2019).

RESULTS AND DISCUSSION

This study evaluates nine machine learning algorithms on the Bot-IoT dataset with the best nineteen features from a feature selection and engineering process. The nine selected ML algorithms comprise three ensemble and six non-ensemble machine learning algorithms. The ensemble algorithms include Random Forest (RF), Bagging (BG), and Stacking (ST). The non-ensemble methods are the Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), k-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Neural Network (NN). The ML algorithms are applied to ten different attack types of the Bot-IoT dataset, and the results are presented in the following section. The attack classes and types are information gathering (Service scanning, OS Fingerprinting), denial of service (DDoS TCP, DDoS UDP, DDoS HTTP, DoS TCP, DoS UDP, and DoS HTTP), and information theft (Keylogging and Data theft). The prominent metrics commonly used to assess the performance of ML algorithms are employed. These include precision, recall, F-measure, kappa, receiver operating characteristic (ROC) value, and error rates measured by the mean absolute error (MAE) and the root mean squared error (RMSE).

Evaluation of Performance of ML Algorithms

Firstly, the study evaluated the performance of the ML algorithms using the F-measure and the results presented in Table 4. The F-measure is a measure of the test's accuracy. The highest possible value of an F-measure is 1, indicating perfect precision and recall, and the lowest possible value is 0 if either the precision or the recall is zero. As can be observed in Table 4 for the F-measure scores,

Table 4. Distribution of F-measures results according to the type of attack

ML Algorithm	F-Measures								
	Non-Ensemble ML Algorithms						Ensemble		
	Linear	Non-Linear algorithms							
	LR	NB	DT	KNN	SVM	NN	RF	BG	ST
Normal	0.981	0.982	1	1	1	0.991	1	1	-
DoS-UDP	1	1	1	1	1	1	1	1	0.439
DDoS-TCP	1	0.999	1	1	1	1	1	1	-
DDoS-UDP	1	1	1	1	1	1	1	1	-
DoS-TCP	1	1	1	1	1	0.999	1	1	-
Reconnaissance-Service_Scan	1	0.994	1	1	1	0.999	1	1	-
DoS-HTTP	1	0.998	1	1	1	-	1	1	-
Reconnaissance-OS_Fingerprint	0.998	0.987	1	0.999	1	0.999	1	1	-
DDoS-HTTP	0.999	0.778	1	1	1	-	1	1	-
Theft-Keylogging	1	0.963	1	1	1	-	1	1	-

it can be seen that Decision tree (DT), Support vector machine (SVM), Random Forest (RF), and Bagging (BG) algorithms were the most successful algorithms with the highest scores, obtaining 100% detection for all the attack types (Normal, DoS-UDP, DDoS-TCP, DDoS-UDP, DoS-TCP, service scan, DoS-HTTP, OS fingerprint, DDoS-HTTP, and keylogging). This shows that these models can correctly identify the IoT malware and effectively protect an IoT network against malware. On the other hand, the Naive Bayes and Stacking algorithms show lower F-measure scores. In particular, the Naive Bayes had a relatively low score for DDoS-HTTP (0.778), and the Stacking algorithm had worse performance on the DoS-UDP attack (0.439). Moreover, out of the three ensemble methods, two (RF and BG) performed well, achieving 100% accuracy. However, contrary to studies that suggest higher performance for the ensemble approach (Gautam & Doegar, 2018), the Stacking ensemble method performed poorly on the dataset.

A similar observation can be made from Table 5 for the precision scores. Except for LR, NB, NN, and ST, all the algorithms obtained a 100% detection rate for all the attack types. Again, ST recorded the worse score for DoS-UDP, and BN achieved as low as 0.643 for DDoS-HTTP. Moreover, a comparison of all the algorithms on the performance metrics can be seen in Table 6. It can be observed that four ML algorithms (DT, SVM, RF, and BG) were the best performing algorithms, followed by NB. The lowest scoring algorithm across all the performance metrics was ST. The error rates for all nine algorithms are presented in Table 7. As can be observed from the table, the error rates for DT and BG were the lowest. In general, ST had the worst performance among the algorithms in terms of error rates.

Overall, two ensemble techniques, RF and BG, yield better performance compared to other classifiers. Among the non-ensemble methods, DT and SVM also achieve 100% performance for accuracy scores. However, contrary to Alahari and Yalavarthi's (2020) findings, the stacking ensemble method shows poor and least performance among all the classifiers evaluated. This suggests that ensemble methods do not always perform better than other methods, but rather the performance of ensemble methods may depend on the base classifiers. Besides, the NB also records a low detection

Table 5. Distribution of precision results according to the type of attack

ML Algorithm	Precision								
	Non-Ensemble						Ensemble		
	Linear	Non-Linear							
	LR	NB	DT	KNN	SVM	NN	RF	BG	ST
Normal	1	0.964	1	1	1	0.982	1	1	-
DoS-UDP	1	0.999	1	1	1	0.999	1	1	0.282
DDoS-TCP	1	0.999	1	1	1	0.999	1	1	-
DDoS-UDP	1	1	1	1	1	1	1	1	-
DoS-TCP	1	1	1	1	1	0.999	1	1	-
Reconnaissance -Service_Scan	1	0.995	1	1	1	0.997	1	1	-
DoS-HTTP	1	0.997	1	1	1	-	1	1	-
Reconnaissance -OS_Fingerprint	0.999	0.997	1	1	1	0.998	1	1	-
DDoS-HTTP	0.995	0.643	1	1	1	-	1	1	-
Theft-Keylogging	1	1	1	1	1	-	1	1	-

Table 6. Performance metrics

ML	TP Rate	FP Rate	Accuracy (%)	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
LR	1	0	99.999	1	1	1	1	1	1
NB	0.999	0	99.931	0.999	0.999	0.999	0.999	1	0.999
DT	1	0	100	1	1	1	1	1	1
KNN	1	0	99.999	1	1	1	1	1	1
SVM	1	0	100	1	1	1	1	1	1
NN	0.999	0	99.929	-	0.999	-	-	1	1
RF	1	0	100	1	1	1	1	1	1
BG	1	0	100	1	1	1	1	1	1
ST	0.282	0.282	28.162	-	0.282	-	-	0.500	0.246

Table 7. Errors metrics

ML Algorithm	Kappa statistic	MAE	RMSE	RAE (%)	RRSE (%)
LR	1	0	0.0015	0.0014%	0.5377%
NB	0.9991	0.0001	0.0113	0.0902%	4.1022%
DT	1	0	0	0%	0%
KNN	1	0	0.0015	0.0032%	0.5377%
SVM	1	0	0	0%	0%
NN	0.9991	0.0002	0.0101	0.1213%	3.6782%
RF	1	0	0.0006	0.0031%	0.2229%
BG	1	0	0	0%	0%
ST	0	0.1509	0.2746	100%	100%

rate. This is not surprising, as a similar performance of NB has been reported in an earlier study by Yaokumah and Wiafe (2020).

While DT, SVM, RF, and BG achieve 100% precision in classifying all the attack types, the LR, NB, NN, KNN, and ST achieve various degrees of precision. Noticeably, LR does not classify OS fingerprint and DDoS-HTTP attack types correctly. Also, NB does not classify as many as six attack types correctly out of the nine attack types present in the dataset. These include DoS-UDP, DDoS-TCP, service scan, DoS-HTTP, OS fingerprint, and DDoS-HTTP. A similar result is observed with the NN classifier, unable to accurately classify DoS-UDP, DDoS-TCP, DoS-TCP, service scan, and OS fingerprint. In general, it can be observed that these four algorithms (DT, SVM, RF, and BG) do not classify precisely DDOS and reconnaissance variants of attacks. In a recent study, Kushwah and Ranga (2021) identify distributed denial of service attacks as a serious security threat to cloud computing. Moreover, Kushwah and Ranga (2021) observed that, in most cases, the first step of an attack is gathering information (reconnaissance) to identify the weak points of the targeted system. Therefore, based on our model, the LR, NB, NN, and ST are not appropriate for designing IoT malware intrusion detection systems since they fail to classify reconnaissance and DDOS-related attacks accurately.

Comparison of Performances of Models

Finally, the results are compared with two similar recent studies conducted by Ferrag and Maglaras (2019) and Alsamiri and Alsubhi (2020). The reason for choosing these studies is that they used the same dataset and three machine learning methods similar to those in this current study. The machine learning algorithms are RF, NB, SVM (Ferrag & Maglaras, 2019) and RF, NB, KNN in Alsamiri and Alsubhi (2020). In general, when the results are compared, it can be observed that the Random Forest and Naive Bayes algorithms used in this study have higher F-measure scores than those used by Ferrag and Maglaras and Alsamiri and Alsubhi for all the attack types (see Table 8). Also, the current study outperformed that of Ferrag and Maglaras in terms of SVM for all the attack types and recorded higher scores for KNN compared with that of Alsamiri and Alsubhi.

Specifically, the F-measure values for RF, NB, and SVM were consistently higher than that of Ferrag and Maglaras (2019) (see Figure 1). For example, the current study achieved 100% F-measures for RF, which correctly classified all the ten attack types, but the highest recorded score for Ferrag and Maglaras was 82.26% for DDOS-TCP attack and as low as 55.26% for DDOS- UDP attack. A similar pattern was observed from Figure 1 with the scores of SVM between the current study (score of 100%) and that of Ferrag and Maglaras. However, observably, Ferrag and Maglaras recorded 100% scores for SVM in the classification of DOS-UDP and DDOS-UDP. Moreover, though the scores of NB in this study for the attack types were relatively low, they appear to be far higher than that of Ferrag and Maglaras, which recorded the lowest score of 50.78% for DDOS- HTTP attack.

Moreover, the results of Alsamiri and Alsubhi (2020) appear relatively higher than that of Ferrag and Maglaras but lower than the current study. Both RF and KNN achieved 100% for all the attack types for F-measure scores than Alsamiri and Alsubhi, which obtained the highest score of 100% for DOS-TCP attack and 65.12% for Keylogging attack. Also, it can be observed from Figure 2 that the NB scores were generally low for Alsamiri and Alsubhi, ranging between 63% and 72%. These may be attributed to the differences in the dependent variables. In the current research, the dependent variable was obtained from the attack and the attack subcategory. Moreover, this study's feature selection and engineering protocol may explain the highest scores obtained compared with the earlier studies conducted by Ferrag and Maglaras (2019) and Alsamiri and Alsubhi (2020). Ferrag and Maglaras (2019) used the original feature set of the IoT-BoT dataset. Alsamiri and Alsubhi

Table 8. Comparison of performance of algorithms using F-measure

Attack Names	Ferrag and Maglaras (2019)			Alsamiri and Alsubhi (2020)			This study's results			
	RF(%)	*NB(%)*	SVM(%)	*RF(%)*	*NB(%)*	*KNN(%)*	*RF(%)*	*NB(%)*	*KNN(%)*	SVM(%)
DDOS HTTP	82.26	50.78	62.24	96	71	96	100	77.8	100	100
DDOS TCP	88.28	78.67	71.26	99	70	99	100	99.9	100	100
DDOS UDP	55.26	78.50	100	98	72	98	100	100	100	100
DOS HTTP	82.20	68.68	70.14	95	71	96	100	99.8	100	100
DOS TCP	81.77	65.56	71.26	100	63	99	100	100	100	100
DOS UDP	82.99	100	100	97	71	97	100	100	100	100
Data exfiltration	86.55	66.55	89.67	96	71	97	100	100	100	100
Keylogging	70.12	65.62	65.12	95	71	98	100	96.3	100	100
OS Scan	82.20	68.68	70.14	94	70	99	100	98.7	99.9	100
Service Scan	69.82	65.21	72.82	95	72	94	100	99.4	100	100

Figure 1. Comparison of Models with Ferrag and Maglaras (2019) using F-measure

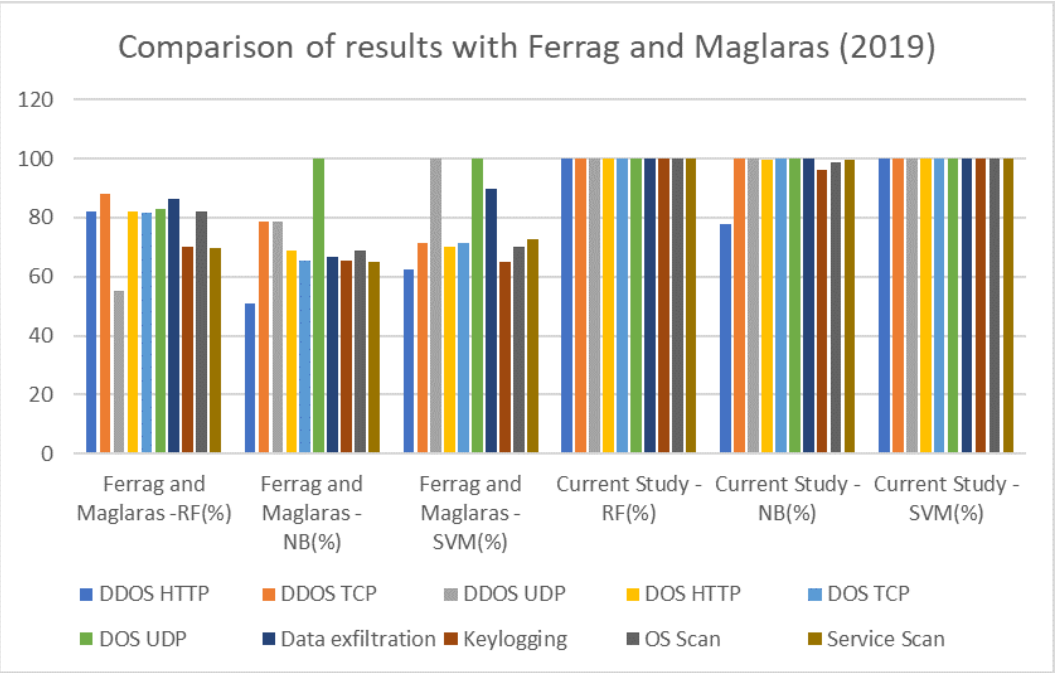
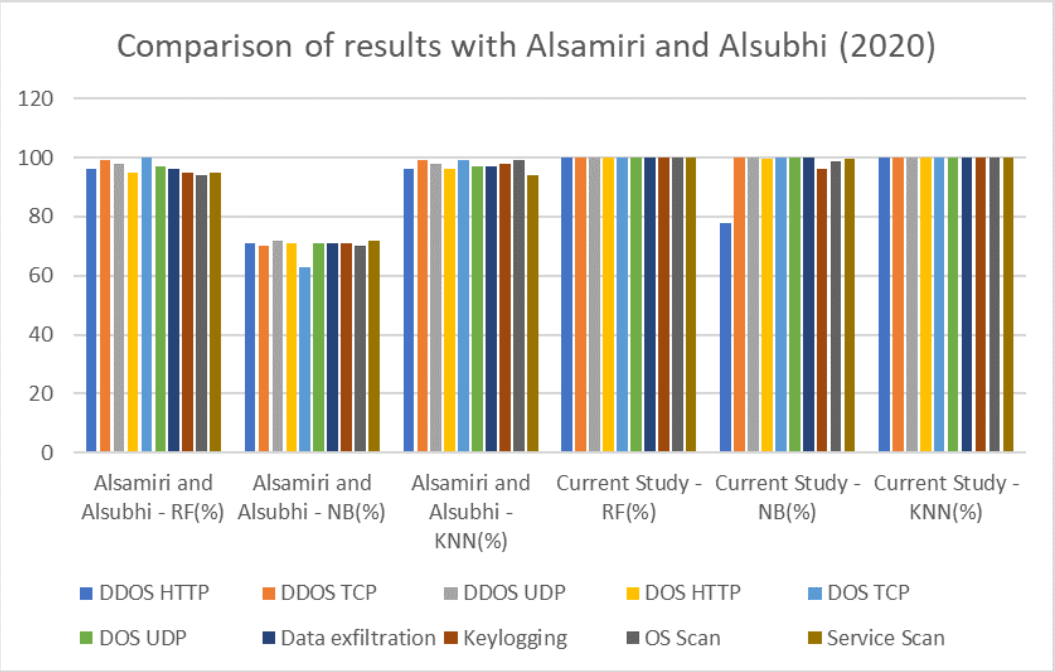


Figure 2. Comparison of Models with Alsamiri and Alsubhi (2020) using F-measure



(2020) did feature engineering by adding new features. However, this study's feature selection and engineering techniques yielded optimum results, comparable to Alsamiri and Alsubhi (2020). Another key difference between this current study and the earlier works is the hardware platforms and software tools used in the experiment. However, it should be noted, that the differences in platforms, particularly the software tools used in the studies, may contribute to the differences in performance indicators of the learning algorithms.

CONCLUSION

The current study performed feature selection and engineering on a BoT-IoT dataset and evaluated the performance of ML algorithms for IoT malware detection. It evaluated nine supervised machine learning algorithms on the Bot-IoT for the detection of malware. Feature selection was performed on the Bot-IoT dataset using correlation to extract a set of features. Four ML algorithms, consisting of Random Forest, Bagging, Decision Tree, and Support Vector Machines, attained the highest score of 100% with all the attack types (DoS-UDP, DDoS-TCP, DDoS-UDP, DoS-TCP,-Service Scan, DoS-HTTP, OS Fingerprint, DDoS-HTTP, and Keylogging). The study further compared its results with two recent studies that employed the same dataset. The current study results outperformed the two related works on the Bot-IoT dataset with all the attack types; in Random Forest, Naive Bayes, Support Vector Machines, and k-Nearest Neighbours.

This study's findings will benefit society and organizations, considering the broad scope of applicability of IoT systems. The IoT applications are deployed in consumer products (smart homes and wearable devices), commercial applications, industrial devices, critical infrastructure (bridges and railway lines), and military applications for monitoring and surveillance. The greater demand for privacy and security of these devices justifies the need for more effective intrusion detection systems to help mitigate cyber-attacks and reduce system vulnerabilities. With the proposed model of this study, the IoT malware intrusion detection rates of ML algorithms on the Bot-IoT dataset are significantly improved. The high intrusion detection accuracy rate of 100% achieved by Random Forest, Bagging, Decision Tree, and Support Vector Machines will help protect data traversing IoT networks from information gathering (service scanning and OS fingerprinting), information theft (keylogging and data theft), and denial of service attacks. Thus, the IoT systems designers and developers that apply the recommended approach will produce more effective security systems against cyber-attacks. Furthermore, an individual's privacy will be protected as the proposed approach will detect data theft and keylogging attacks. For the researcher, the feature selection and engineering approach employed in this study will form the foundation for further investigation into similar IoT datasets to improve ML algorithms' performance.

Despite achieving high performance, the limitation of this study is the use of one IoT dataset for the evaluation of the ML models. As different IoT datasets may have various features and attack types, experimentation with multiple datasets with a deep learning approach to feature selection will be insightful. Therefore, a future study will employ multiple IoT datasets, specifically Bot-IoT, N-BaIoT, and IoT-23, to assess their comprehensiveness; compare and contrast their appropriateness for the effective design of IoT intrusion detection systems. A future study will also look into deep learning techniques to perform automatic feature engineering and compare its performance in terms of accuracy and computational efficiency.

REFERENCES

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., & Khan, M. S. (2021). Intrusion detection in the internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *Eurasip Journal on Wireless Communications and Networking, 2021*(1). 10.1186/s13638-021-01893-8

Alahari, H. P., & Yalavarthi, S. B. (2020). Ensemble DOS attack detection with IoT integration algorithm. *International Journal of Emerging Trends in Engineering Research*, *8*(9), 6342–6346. doi:10.30534/ijeter/2020/230892020

Alasmary, H., Khormali, A., Anwar, A., Park, J., Choi, J., Abusnaina, A., Awad, A., Nyang, D., & Mohaisen, A. (2019). Analyzing and Detecting Emerging Internet of Things Malware: A Graph-Based Approach. *IEEE Internet of Things Journal*, *6*(5), 8977–8988. doi:10.1109/JIOT.2019.2925929

Alhanahnah, M., Lin, Q., Yan, Q., Zhang, N., & Chen, Z. (2018). Efficient Signature Generation for Classifying Cross-Architecture IoT Malware. *2018 IEEE Conference on Communications and Network Security (CNS)*, 1-9. doi:10.1109/CNS.2018.8433203

Alsamiri, J., & Alsubhi, K. (2019). Internet of things cyber-attacks detection using machine learning. *International Journal of Advanced Computer Science and Applications*, *10*(12), 627–634. doi:10.14569/IJACSA.2019.0101280

Behal, S., & Kumar, K. (2017). Detection of DDOS attacks and flash events using information theory metrics – an empirical investigation. *Computer Communications*, *103*, 18–28. doi:10.1016/j.comcom.2017.02.003

Butcher, B., & Smith, B. J. (2020). Feature Engineering and Selection: A Practical Approach for Predictive Models. *The American Statistician*, *74*(3), 308–309. doi:10.1080/00031305.2020.1790217

Chesney, S., Roy, K., & Khorsandroo, S. (2021). *Machine learning algorithms for preventing IoT cybersecurity attacks.* 10.1007/978-3-030-55190-2_53

Choudhury, S., & Bhowal, A. (2015). Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection. *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. doi:10.1109/ICSTM.2015.7225395

Da Costa, C. A., Pasluosta, C. F., Eskofier, B., da Silva, D. B., & da Rosa Righi, R. (2018). Internet of Health Things: Toward intelligent vital signs monitoring in hospital wards. *Artificial Intelligence in Medicine*, *89*, 61–69. doi:10.1016/j.artmed.2018.05.005 PMID:29871778

Das, A., Ajila, S. A., & Lung, C. (2020). *A comprehensive analysis of accuracies of machine learning algorithms for network intrusion detection.* 10.1007/978-3-030-45778-5_4

Dataquest Reviews. (2020). *An Analysis of 656 Learner Outcomes*. Retrieved from https://www.dataquest.io/blog/topics/dataquest-updates/

De La Torre Parra, G., Rad, P., Choo, K.-R., & Beebe, N. (2020). Detecting internet of things attacks using distributed deep learning. *Journal of Network and Computer Applications*, *163*, 102662. Advance online publication. doi:10.1016/j.jnca.2020.102662

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87. doi:10.1145/2347736.2347755

Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). A deep learning ensemble for network anomaly and cyber-attack detection. *Sensors (Basel)*, *20*(16), 1–20. doi:10.3390/s20164583 PMID:32824187

Engineer, A., Sternberg, E. M., & Najafi, B. (2018). Designing interiors to mitigate physical and cognitive deficits related to aging and to promote longevity in older adults: A review. *Gerontology*, *64*(6), 612–622. doi:10.1159/000491488 PMID:30130764

Ferrag, M. A., & Maglaras, L. (2019). DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids. *IEEE Transactions on Engineering Management*, 1–13. doi:10.1109/TEM.2019.2922936

Gautam, R. K. S., & Doegar, E. A. (2018). An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms. *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 14-15.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660. .S2CID20498203210.1016/j.future.2013.01.010

Guhathakurta, R. (2017). *How IoTs are changing the fundamentals of retailing*. Retrieved from https://trak.in/tags/business/2016/08/30/internet-of-things-iot-changing-fundamentals-of-retailing

Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, *7*(1), 100059. doi:10.1016/j.iot.2019.100059

Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, *40*, 307–324. doi:10.1016/j.jnca.2013.08.001

Howard, P. N. (2015). *Pax Technica: How the internet of things may set us free, or lock us up*. Yale University Press.

Huang, G., Zhu, Q. Y., & Siew, C. K. (2016). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1–3), 489–501. doi:10.1016/j.neucom.2005.12.126

Jesudoss, A., & Subramaniam, N. (2014). A survey on authentication attacks and countermeasures in a distributed environment. *Indian Journal of Computer Science Engineering*, *5*, 71–77.

Kern, R. (2014). *Feature Engineering, Knowledge Discovery and Data Mining*. Retrieved from http://kti.tugraz.at/staff/denis/courses/kddm1/featureengineering.pdf

Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, *4*(1), 18. Advance online publication. doi:10.1186/s42400-021-00077-7

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDOS in the IoT: Mirai and other botnets. *Computers*, *50*(7), 80–84. doi:10.1109/MC.2017.201

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: BoT-IoT dataset. *Future Generation Computer Systems*, *100*, 779–796. doi:10.1016/j.future.2019.05.041

Kushwah, G. S., & Ranga, V. (2021). Optimized extreme learning machine for detecting DDoS attacks in cloud computing. *Computers & Security*, *105*, 102260. Advance online publication. doi:10.1016/j.cose.2021.102260

Liu, S., Hao, X., & Chen, X. (2020). A semi-supervised dynamic ensemble algorithm for IoT anomaly detection. *Proceedings - IEEE Congress on Cybermatics*, 264-269. doi:10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00058

Lu, Y., & Xu, L. D. (2019). Internet of Things (IoT) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal*, *6*(2), 2103–2115. doi:10.1109/JIOT.2018.2869847

Mazurczyk, W., & Caviglione, L. (2021). Cyber reconnaissance techniques. *Communications of the ACM*, *6*(3), 86–95. doi:10.1145/3418293

Moustafa, N., Turnbull, B., & Choo, K.-K.R. (2018a). Towards automation of vulnerability and exploitation identification in IIoT networks. *2018 IEEE International Conference on Industrial Internet, ICII,* 139–145.

Moustafa, N., Turnbull, B., & Choo, K.-K. R. (2019, June). Turnbull, B., & Choo, K.-K.R. (2018b). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, *6*(3), 4815–4830. Advance online publication. doi:10.1109/JIOT.2018.2871719

Nakhodchi, S., Upadhyay, A., & Dehghantanha, A. (2020). A Comparison Between Different Machine Learning Models for IoT Malware Detection. In H. Karimipour, P. Srikantha, H. Farag, & J. Wei-Kocsis (Eds.), *Security of Cyber-Physical Systems*. Springer. doi:10.1007/978-3-030-45541-5_10

Ngo, Q. D., Nguyen, H. T., Nguyen, L. C., & Nguyen, D. H. (2020). *A survey of IoT malware and detection methods based on static features*. ICT Express. doi:10.1016/j.icte.2020.04.005

Nguyen, K. D. T., Tuan, T. M., Le, S. H., Viet, A. P., Ogawa, M., & Minh, N. L. (2018). Comparison of Three Deep Learning-based Approaches for IoT Malware Detection. *10th International Conference on Knowledge and Systems Engineering (KSE)*, 382-388. doi:10.1109/KSE.2018.8573374

Peters, W., Dehghantanha, A., Parizi, R. M., & Srivastava, G. (2020). A Comparison of State-of-the-Art Machine Learning Models for OpCode-Based IoT Malware Detection. In K. K. Choo & A. Dehghantanha (Eds.), *Handbook of Big Data Privacy*. Springer. doi:10.1007/978-3-030-38557-6_6

Reddy, D. K. K., Behera, H. S., Pratyusha, G. M. S., & Karri, R. (2021). *Ensemble bagging approach for IoT sensor based anomaly detection*. 10.1007/978-981-15-8439-8_52

Rocca, J. (2019). *Ensemble methods: bagging, boosting and stacking - Understanding the key concepts of ensemble learning*. Retrieved from https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Rouse, M. (2019). Internet of things (IoT). *Agenda (Durban, South Africa)*, *2019*. https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Sarumi, O. A., Adetunmbi, A. O., & Adetoye, F. A. (2020). Discovering computer networks intrusion using data analytics and machine intelligence. *Scientific American*, *9*, e00500. Advance online publication. doi:10.1016/j.sciaf.2020.e00500

Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). IoT malicious traffic identification using wrapper-based feature selection mechanisms. *Computers & Security*, *94*, 101863. Advance online publication. doi:10.1016/j.cose.2020.101863

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems*, *107*, 433–442. doi:10.1016/j.future.2020.02.017

Shobana, M., & Poonkuzhali, S. (2020). An efficient botnet detection approach for green IoT devices using machine learning techniques. *Journal of Green Engineering*, *10*(3), 1053–1076.

Simon, M. K. (2011). *Assumptions, limitations and delimitations*. Retrieved from http://dissertationrecipes.com/wp-content/uploads/2011/04/AssumptionslimitationsdelimitationsX.pdf

Singh, J., Pasquier, T., Bacon, J., Ko, H., & Eyers, D. (2015). Twenty cloud security considerations for supporting the Internet of Things. *IEEE Internet of Things Journal*, *3*(3), 1. doi:10.1109/JIOT.2015.2460

Su, J., Vasconcellos, D. V., Prasad, S., Sgandurra, D., Feng, Y., & Sakurai, K. (2018). Lightweight Classification of IoT Malware Based on Image Recognition. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC),* 664-669. doi:10.1109/COMPSAC.2018.10315

Tankard, C. (2011). Advanced persistent threats and how to monitor and deter them. *Network Security*, *8*(8), 16–19. doi:10.1016/S1353-4858(11)70086-1

Tsiligkaridis, A., & Paschalidis, I. C. (2017). Anomaly detection in transportation networks using machine learning techniques. *IEEE MIT Undergraduate Research Technology Conference (URTC)*. doi:10.1109/URTC.2017.8284194

University of Waikato. (2018). *The workbench for machine learning*. University of Waikato. Retrieved from https://www.cs.waikato.ac.nz/ml/weka/

Wakefield, K. (2020). *A guide to machine learning algorithms and their applications*. Retrieved from https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

Yaokumah, W., & Wiafe, I. (2020). Analysis of machine learning techniques for anomaly-based intrusion detection. *International Journal of Distributed Artificial Intelligence*, *12*(1), 20–38. doi:10.4018/IJDAI.2020010102

Winfred Yaokumah is a researcher, cybersecurity expert, and senior faculty at the Department of Computer Science of the University of Ghana. His work appears in several local and international reputable journals. He is an editor of the Modern Theories and Practices for Cyber Ethics and Security Compliance. He also serves on an International Review Board for the International Journal of Technology Diffusion. His research interest includes Cybersecurity, Machine Learning, Internet of Things, Network Security, and Information Systems Security.

Justice Kwame Appati is a lecturer in the School of Physical and Mathematical Science (SPMS) and the Department of Computer Science. He began his teaching career at Kwame Nkrumah University of Science and Technology in Kumasi as a graduate assistant and then later moved to the University of Ghana in 2017 as a lecturer. Justice earned a PhD, in Applied Mathematics from Kwame Nkrumah University Science and Technology in 2016. He also graduated in 2010 and 2013 with a BSc. Mathematics and MPhil Applied Mathematics from the same institution. His current research includes data science, mathematical intelligence, image processing and scientific computing, He has singly and jointly supervised undergraduate and postgraduate students from Kwame Nkrumah University of Science and Technology (KNUST), National Institute of Mathematical Sciences (NIMS), African Institute of Mathematical Sciences (AIMS) and the University of Ghana Currently, Justice handles course like Design and Analysis of Algorithm, Artificial Intelligence, Formal Methods and Computer Vision. He looks forward to working with everyone interested in his field of study more especially, Intelligence and Data Science.

Daniel Kumah is a post-graduate student at the Department of Computer Science at the University of Ghana. His research interests include smart computing, internet of things and their effects in driving industries. His favourite pastime activities are reading and building things.