Internet of Things Cyber Attacks Detection using Machine Learning

Jadel Alsamiri[1], Khalid Alsubhi[2]
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, KSA

***Abstract*—The Internet of Things (IoT) combines hundreds of millions of devices which are capable of interaction with each other with minimum user interaction. IoT is one of the fastest-growing areas in of computing; however, the reality is that in the extremely hostile environment of the internet, IoT is vulnerable to numerous types of cyberattacks. To resolve this, practical countermeasures need to be established to secure IoT networks, such as network anomaly detection. Regardless that attacks cannot be wholly avoided forever, early detection of an attack is crucial for practical defense. Since IoT devices have low storage capacity and low processing power, traditional high-end security solutions to protect an IoT system are not appropriate. Also, IoT devices are now connected without human intervention for longer periods. This implies that intelligent network-based security solutions like machine learning solutions must be developed. Although many studies in recent years have discussed the use of Machine Learning (ML) solutions in attack detection problems, little attention has been given to the detection of attacks specifically in IoT networks. In this study, we aim to contribute to the literature by evaluating various machine learning algorithms that can be used to quickly and effectively detect IoT network attacks. A new dataset, Bot-IoT, is used to evaluate various detection algorithms. In the implementation phase, seven different machine learning algorithms were used, and most of them achieved high performance. New features were extracted from the Bot-IoT dataset during the implementation and compared with studies from the literature, and the new features gave better results.**

***Keywords*—Network anomaly detection; machine learning; Internet of Things (IoT); cyberattacks; bot-IoT dataset**

I. Introduction

Concerns over security and privacy regarding computer networks are increasing in the world, and computer security has become a requirement as a result of the spread of information technology in daily life. The raise in the amount of Internet applications and the appearance of modern technologies such as the Internet of Things (IoT) are followed with new and recent efforts to invade computer networks and systems. The Internet of Things (IoT) is a set of interrelated devices where the devices have the ability to connect without the need for human intervention. With IoT, many things that have sensors (such as coffee makers, lights, bicycles, and many others) in areas like healthcare, farming, transportation, etc. can connect to the Internet[1]. By saving time and resources, IoT applications are changing our work and lives. It also has unlimited advantages and opens numerous opportunities for the exchange of knowledge, innovation, and growth.

Every security threat within the Internet exists within the IoT as well because the Internet is the core and center of the IoT. Compared to other traditional networks, IoT nodes have low capacity and limited resources, and do not have manual controls. Also, the rapid growth and broad daily-life adoption of IoT devices makes IoT security issues very troublesome, raising the need to develop security solutions based on networks. While current systems perform well in identifying some attacks, it is still challenging to detect others. As network attacks grow, along with a massive increase in the amount of information present in networks, faster and more effective methods of detection of attacks are required [2] and there is no doubt that there is scope for more progressive methods to improve network security. In this context, in order to provide embedded intelligence in the IoT environment, we can consider Machine Learning (ML) as one of the most effective computational models. Machine learning approaches have been used for different network security tasks such as network traffic analysis [3],[4],[5], intrusion detection[6], and botnet detection [7].

Machine Learning can be described as an intelligent device's ability to modify or automate a knowledge-based state or behavior, which is considered a critical part of an IoT solution. ML has the ability to infer helpful knowledge from data generated by devices or humans, and ML algorithms are used in tasks such as regression, and classification. Likewise, in an IoT network, ML can be used to provide security services. The use of machine learning in attack detection problems is becoming a hotly pursued subject, and ML is being used more and more in different applications in the cybersecurity field. Although many studies in the literature have used ML techniques to discover the best ways to detect attacks, only limited research exists on efficient detection methods suitable for IoT environments.

Machine learning can be applied to the attack detection task via two main types of cyber-analysis: signature-based (sometimes also called misuse- based) or anomaly-based. Signature-based techniques are designed to detect known attacks by using specific traffic characteristics (also known as "signatures") in those attacks. One of the advantages of this class of detection technique is its ability to detect all known attacks effectively without generating an overwhelming number of false alarms. In the literature, some works use signature-based techniques to detect attacks [3], [7]; for instance, in the domain of network traffic analysis, [3] applied four different machine learning techniques as preliminary tools to learn the features of some known attacks. Signature-based techniques were also used in [7] to identify compromised machines by identifying botnet

network traffic patterns. The main drawbacks of signature-based approaches are that the efficient use of these approaches requires frequent manual updates of attack traffic signatures and that these approaches cannot detect previously unknown attacks. The second class of detection methods is anomaly-based detection. This class models normal network behavior, and anything abnormal is considered an attack. The ability of this class to detect unknown attacks makes it appealing to use. The essential issue with anomaly-based techniques is the possibility of high false alarm rates (FARs), as previously unknown (even though legal) behaviors can be considered as anomalies. Signature and anomaly detection techniques can be combined as a hybrid technique. One of the hybrid technique examples is presented in [8] where this technique is used to increase the detection rates of known attacks and reduce the false positive (FP) rate for unknown attacks.

In this study, we contribute to the literature as part of a defense against IoT attack behavior by investigating the efficacy of using machine learning approaches to detect IoT network attacks. The detection algorithms are evaluated using a recent dataset, Bot-IoT, that combines legitimate and simulated IoT network traffic along with different types of attacks [9]. Using the Random Forest Regressor algorithm, features were selected from this dataset. In the implementation phase, seven different machine learning algorithms were used, and high performance achieved. The following are the machine learning algorithms that we used: K-nearest neighbours (KNN), ID3 (Iterative Dichotomiser 3),Quadratic discriminant analysis (QDA), Random Forest, AdaBoost, Multilayer perceptron (MLP), and Naıve Bayes (NB).

We can summarize our contributions through this research as:

- Improvement in attack detection in IoT networks by evaluating the performance of machine learning algorithms on a recent IoT dataset.
- Extract new features from the dataset and select the most appropriate features to improve machine learning algorithm performance.
- Contribute to the IoT literature. Since the number of studies done with the Bot-IoT dataset are still few, working with this dataset could be considered to be a possible significant contribution to the literature.

The remainder of the paper is organized as follows: Section II we review related work and discuss the background in this domain; Section IV show our proposed approach, followed by the implementation details in Section V. Experimental results with evaluations are presented in Section VI; and finally, we conclude this paper with a summary in Section VII.

II. Related Work

The domain of using machine learning has been extensively researched in the past [6], and several scholarly papers on intrusion detection by data-mining techniques and machine intelligence have been published [10]. However, most of these prior studies have only used machine learning techniques for intrusion detection in traditional networks. We are therefore extending this area of research in this study by specifically applying machine learning to detect attacks in the context of IoT. The application of machine learning techniques to the IoT field is still in the early stages of research, specifically in IoT security, but it has a huge possibility to discover insights from IoT data [11]. In IoT networks, machine learning principles like pattern recognition, anomaly detection, and behavioral analysis can be used to detect potential attacks and stop abnormal behaviors.

To review recent research on the topic of attack detection using machine learning in IoT networks, we examined various studies and summarized them in Table I. In each study, the machine learning algorithms, datasets, and detection approaches are given. When selecting these studies, we focused on the use of different machine learning algorithms and datasets. The studies provide evidence that machine learning techniques can achieve success for attack detection. From the works discussing the issue of using machine learning for IoT security, the detection methodologies can be categorized as unsupervised methods [10], [12], [13], [14] and supervised methods [15], [16], [17], [9], [18].

Many studies have indicated that machine learning techniques can be applied to support attack detection tasks, including kmeans, artificial neural networks (ANNs), Random Forest (RF), auto-encoder, and others, and several authors have applied unsupervised machine learning algorithms for detection problems. Auto-encoders are some of the most significant unsupervised algorithms that have been used in many works; for example, Mirsky et al. [10] proposed the use of autoencoders to extract features from datasets in order to improve the detection of cyber threats. They introduced Kitsune, which is an unsupervised network intrusion detection system that has ability to learn to detect attacks on networks efficiently. Kitsune's main algorithm (KitNET) uses a set of neural networks, known as autoencoders, to distinguish between normal and anomalous traffic patterns. In [12], Meidan et al. proposed and evaluated a novel detection method that extracts behavioral snapshots from the network and also uses auto-encoders to detect abnormals network traffic from compromised devices. The major drawback of using unsupervised machine learning algorithms for detection problems is that in network traffic, most of the flows are normal and anomalies like attacks and outliers are rare, which negatively affects success rates and the detection of anomalies. For this reason, better results are expected with supervised techniques. On the other hand, many supervised learning algorithms are used to detect attacks and are trained on datasets with labels indicating whether the instances have been pre-classified as attacks or not. In [19], Elike Hodo used ANN and Support Vector Machine algorithms to detect non-Tor traffic attack by using ML techniques on UNBCIC datasets. In order to accurately identifying IoT device types from the whitelist In [15], Random Forest algorithm, was applied to features extracted from network traffic data. A recent work that has a similar approach to our study was presented by Moustafa et al. [9] in the original paper which proposed the Bot-IoT dataset. They used LSTM, SVM, and RNN machine learning models to evaluate the IoT dataset, but in their analysis they did not determine the adversarial robustness of their models. In our work, while we use the same Bot-IoT dataset presented in [9], we focus on extracting new features from the dataset and evaluating different machine learning algorithms on this dataset. [22] is another study that used the BoT-IoT dataset.

TABLE I. SUMMARY OF RELATED STUDIES

Ref.	Year	Detection approach		Machine learning algorithms used							Data For Evaluation
		Signature	Anomaly	Supervised approaches						Unsupervised	
				ANN	RF	SVM	NB	AdaBoost	LSTM		
[9]	2018		✓	✓		✓			✓		Bot-IoT
[10]	2018		✓							✓	Real dataset
[8]	2016	✓									Simulated dataset
[12]	2018		✓							✓	N-BaIoT dataset
[14]	2019		✓							✓	CICIDS2017-Simulated dataset-Bot-IoT
[19]	2016		✓	✓							Simulated dataset
[13]	2017		✓							✓	KDD /DARPA
[20]	2015		✓								Real dataset
[15]	2017	✓			✓						Real dataset
[21]	2018		✓	✓							Real dataset
[18]	2018		✓			✓					Simulated dataset
[16]	2017		✓	✓	✓		✓				USNW-NB15
[22]	2019		✓	✓							Bot-IoT
[23]	2019		✓					✓			Real dataset
[24]	2019		✓				✓				Bot-IoT
[17]	2019	✓		✓							Bot-IoT

They compared the Self-normalizing Neural Network (SNN) performance with the FNN for classifying intrusion attacks in an IoT network. Based on multiple performance metrics in this experiments , the FNN outperformed the SNN in their experimental results for intrusion detection in IoT networks, offering a bright future in the search of secure deep learning in IoT networks. Ferrag, in [14], used the Bot-IoT dataset to evaluate the DeepCoin framework's performance in traffic generated by IoT. DeepCoin is a novel deep learning and blockchain-based energy framework. Through performance evaluations using the Bot-IoT dataset, they demonstrated the efficiency of the proposed DeepCoin framework. In other research, the authors of [17] used the BoT-IoT dataset to generate the rules for IoT-IDS. They used J48, machine learning algorithms for generating effective rules to support lightweight IDS systems appropriate for IoT devices.

III. PROPOSED APPROACH

This section provides a brief description of the dataset used and our proposed approach to detect attacks in IoT networks. In our proposed approach, various pre-processing and actual applications are performed to detect anomalies by machine learning techniques. First, flow-based features from the raw dataset were extracted by CICFlowMeter [25]. Then, the data pre-processing process was performed in the first step before dividing the dataset into two parts: training and test. Data pre-processing is required to transform the data into a format usable by machine learning algorithms. After these operations, the properties to be used by the algorithms are decided in the feature selection step. Finally, our approach ends with the implementation of machine learning algorithms. An overview of the proposed approach is presented in Fig. 1.

We selected the Bot-IoT dataset for the experiments because of its regular updates, wide attack diversity, inclusion of IoT-generated traffic, and ability to generate new features from the raw dataset. The Bot-IoT dataset [10] was created in the Cyber Range Lab at the Australian Centre for Cyber Security (ACCS). This dataset has three main kinds of attacks, which are based on botnet scenarios such as Probing, DoS, and Information Theft. We used CICFlowMeter to extract flow-based features from the raw traffic traces. CICFlowMeter [26] is a network traffic flow generator distributed by CIC to generate 84 network traffic features.

IV. IMPLEMENTATION

As already noted in the previous sections, the major objective of the experiments is to evaluate the performance of machine learning algorithms in detecting IoT network attacks. In this section, we describe the dataset, machine learning algorithms that we used and present our implementation steps.

A. Datasets

Since the applications for various network security tasks use machine learning methods, large datasets are needed to analyze network flows and distinguish between normal and abnormal traffic. Over the years, several experiments have been conducted to generate network datasets. As shown in Table I, most of the studies using machine learning have tested their work against simulated or real network data. Although a good number of those datasets remain private, primarily due to security concerns, some have become publicly available such as DARPA 98, KDD99, UNSW-NB15, ISCX, CICIDS2017, and N-BaIoT. Although several datasets have been produced, however, the development of realistic IoT and network traffic datasets that include new Botnet scenarios are still few. More importantly, some datasets lack the inclusion of IoT-generated traffic, while others neglect to generate any new features. In some cases, the testbed used was not realistic, while in other cases, the attack scenarios were not diverse enough. For instance, in [12], Meidan et al. created a publicly available IoT dataset named N-BaIoT, and many later studies used this dataset for training and to test their classifier models. While this dataset is relatively large and clean, it is unbalanced, and the ratio of normal data is much lower compared to attack data. Moustafa et al. [9] sought to address the shortcomings by designing the Bot-IoT dataset, which we used for our experiments. The Bot-IoT dataset incorporates legitimate and simulated IoT network traffic along with various types of attacks[14]. The BotIoT database attacks are classified into three types: Probing attacks, DoS, and theft information theft.

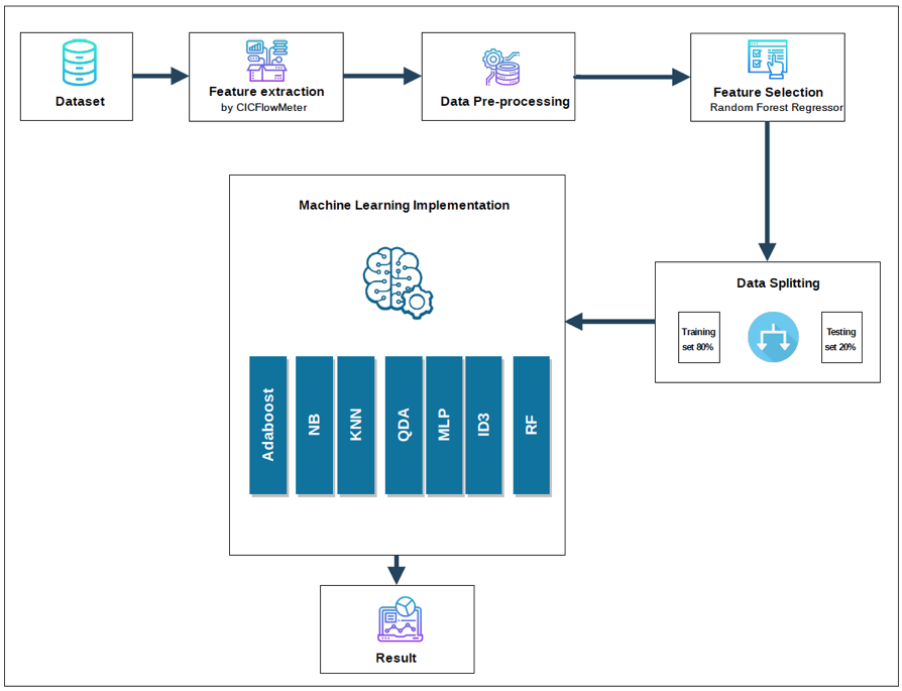

Fig. 1. Proposed Approach Overview.

B. Machine Learning Algorithms

We used the Bot-IoT dataset to evaluate seven well known machine learning classifiers: (K-Nearest Neighbours (KNN), ID3 (Iterative Dichotomiser 3), Random Forest, AdaBoost, Quadratic discriminant analysis (QDA), Multilayer perceptron (MLP), and Naıve Bayes (NB). When choosing these classifiers, the focus is on bringing together popular algorithms with different characteristics. In this context, the algorithms used are briefly examined in the following.

- **K-Nearest Neighbours(KNN):** KNN is one of the simplest and most effective supervised learning algorithm. It is used for searching through the available dataset to associate new data points with similar existing points [24]. KNN, which provides good performance over multidimensional data and is a fast algorithm during the training phase, is relatively slow in the estimation stage.

- **Quadratic discriminant analysis (QDA):** QDA is an ideal algorithm to supervised classification problems. Discriminant analysis is a statistical technique for assigning measured data to one group among many groups. QDA is appropriate to situation where a category is not characterized by much data. In order to be able to apply Quadratic Discriminant Analysis, the number of samples observed must be greater than the number of groups.

- **Iterative Dichotomiser 3(ID3):** ID3 is an algorithm used to create a decision tree from a dataset. It was developed by Ross Quinlan [27]. A decision tree is an algorithm for classification that uses a tree-like decision structure. It is one way to display an algorithm that only contains conditional control statements. The attributes are used as the tree nodes and the criteria are constructed so as to guide from one node to another, with the "leaves" being the class values allocated to the record [16]. ID3 is usually used in the domains of machine learning and natural language processing, and it is the precursor of the C4.5 algorithm.

- **Random Forest(RF):** RF is a machine learning approach that uses decision trees. In this method, a "forest" is created by assembling a large number of different decision tree structures that are formed in different ways[28]. This algorithm has many advantages, such as the ability run on huge datasets efficiently, its light weight compared to other methods, and robustness against noise and outliers when compared to single classifiers.

- **Adaptive Boosting (AdaBoost):** AdaBoost is a machine learning algorithm that focuses on classification issues and tries to convert weak classifiers into efficient ones. It was first proposed by Freund and Schapire in 1996, and can be used in conjunction with many other types of learning algorithm to improve performance. The most important characteristic of the AdaBoost algorithm is its capability to deal with missing values in a dataset.

- **Multilayer Perceptron (MLP):** MLP is a class of feedforward artificial neural network (ANN). Artificial neural networks (ANNs) are a machine learning method that takes inspiration from the way the human brain works, like learning and deriving new information. An MLP include no less than three layers: an input,output and hidden layer. MLP utilizes a

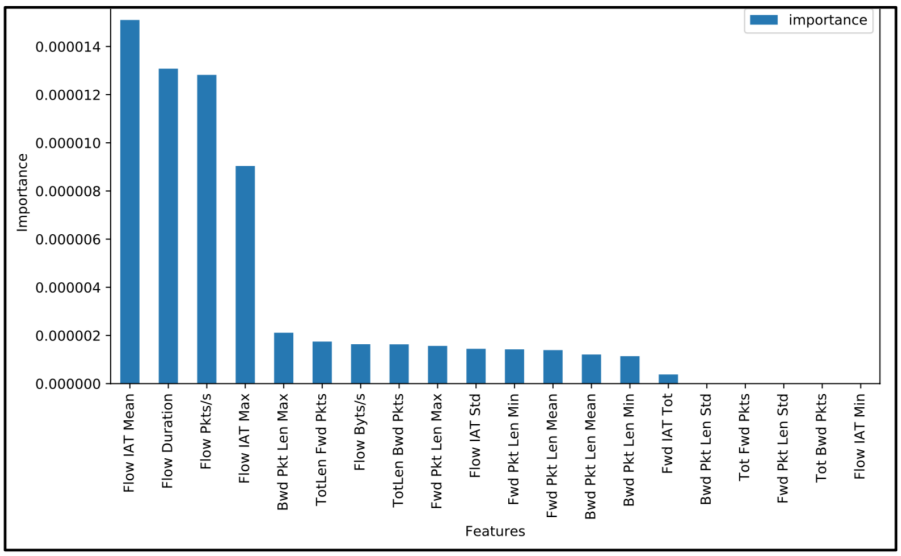

Fig. 2. Graph of Feature Importance of entire dataset

supervised learning technique called back-propagation for training.

- **Naïve Bayes(NB):** The NB is a widely used supervised algorithm, and is famous for its simple principles. The Naïve Bayes method is based on the work of Thomas Bayes [29]. For instance, NB might be utilized to categorize traffic as normal or anomalous for intrusion detection. The traffic classification features used are handled independently by the NB classifier despite the fact that these features may depend on each other. Many attributes make NB user-friendly, like its simplicity, low sample requirement, and ease of implementation[27]. On the other hand, NB deals with features independently and is therefore unable to obtain valuable information from the communication and relationships between features.

C. Implementation Steps

Our method consists of five essential steps:: Feature extraction, data pre-processing, Splitting data, feature selection, and implementation of machine learning algorithms.

- **Feature Extraction:** CICFlowMeter [25] was used to extract flow-based features (in pcap format) from raw network traffic data. CICFlowMeter is a network traffic flow generator distributed by CIC that produces 84 network traffic characteristics. It reads the pcap file and produces a visual document of the features extracted, and also offers a csv file of the dataset. This process was primarily designed to improve classifiers' predictive capabilities by extracting new dataset features.

- **Data pre-processing:** Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning. This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.

- **Splitting Data:** During the machine learning process, data are needed so that learning can take place. In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works. In our study, we considered 80% of the Bot-IoT dataset to be the training data and the remaining 20% to be the testing data.

- **Feature Selection:** It is significant to decrease the count of features and just use the features needed to train and test the algorithms to find a lightweight security solution appropriate for IoT systems [13]. We used the Random Forest Regressor algorithm as features selection technique. The random forest regressor has been proven to be an effective method of reducing the dimensions of a dataset. Decreasing the input data features from more than 80 network traffic features to 7 makes the model train and respond more quickly. The features' importance weights for the full dataset are shown in Fig. 2.

- **Implementation of Machine Learning Algorithms:** All the experiments were done in Python by relying on Python machine learning libraries (scikit-learn, Matplotlib, Pandas, and NumPy). We organised the evaluation of machine learning algorithms for the dataset in three phases: applying the proposed algorithms on each attack in the dataset separately; applying the algorithms on the entire dataset with a set of features combining the best features for each attack (the list of these features can be seen in Table II); and applying the algorithms on the entire dataset with the seven best features obtained in the feature selection step.

TABLE II. THE FEATURE LIST CREATED FOR ALL ATTACK TYPES

Flow IAT Mean	Flow Duration	Flow Pkts/s
Flow IAT Max	Fwd Pkt_Len_Max	TotLen Fwd Pkts
Fwd Pkt Len Mean	Tot Bwd Pkts	Fwd IAT Tot
Flow IAT Std	Flow Bytss	Tot Fwd Pkts
Flow IAT Min		

V. EVALUATION

A. Evaluation Metrics

When evaluating the performance of machine-learning models, it is crucial to define performance measures that are suitable for the task to be solved. In order to evaluate our results, we used the most important performance indicators for accuracy, precision, f-measure, and recall, as shown in the equations below:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$F - measure = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{4}$$

B. Results

As stated in the previous section, we organised the evaluation of machine learning algorithms for the dataset in three phases, as follows. Phase 1: applying machine learning algorithms on each attack in the dataset separately; Phase 2: applying machine learning algorithms on the entire data set with a set combining the best features for each attack; and Phase 3: applying machine learning algorithms on the entire dataset with the best seven features obtained in the feature selection step. The results of all the experiments are given in the following tables. The performance evaluation procedures were repeated 10 times for each machine learning algorithm, and the numbers given in the tables are the arithmetic means of these 10 processes.

Phase 1: applying machine learning algorithms on each attack in the dataset separately. Seven different machine learning methods are applied to 10 different attack types, and the results are presented in Table III. In the results of the algorithms, if there is an equality in the F-measure, the following values are examined in order to eliminate equality: precision, accuracy, recall, and time.

When observing the results in Table III, it can be noted that all the algorithms, except the Naive Bayes (NB) and Quadratic algorithm(QDA), achieved over 90% success in detecting almost all attack types. The ID3 algorithm was the most successful algorithm, completing 6 out of 10 tasks (DDOS-HTTP, DDOS-UDP, DOS-HTTP, DOS-TCP, Data exfiltration, and Service scan) with the highest score. In fact, for all the tasks, ID3 shares its highest score with at least one other algorithm. However, low processing time puts it ahead of the other algorithms. The last algorithm used in all tasks was Naive Bayes, the lowest F-measure algorithm. Especially with the DOS TCP attack, it had a fairly low score. Even though Naive Bayes performed worse than the other algorithms, it was much better than the alternatives when it came to speed. However, it is also necessary to mention the QDA here, because the QDA had the second-worst performance among the algorithms.

TABLE III. DISTRIBUTION OF RESULTS ACCORDING TO TYPE OF ATTACK

	F-Measures						
Attack Names	**NB**	**QDA**	**RF**	**ID3**	**AB**	**MLP**	**KNN**
DDOS_HTTP	*0.72*	0.85	0.96	**0.96**	0.96	0.95	0.96
DDOS_UDP	*0.73*	0.92	0.98	**0.98**	0.98	0.97	0.98
DDOS_TCP	*0.71*	0.85	0.99	0.99	**1.0**	0.74	0.99
DOS_HTTP	*0.72*	0.82	0.95	**0.96**	0.95	0.95	0.96
DOS_UDP	*0.72*	0.83	0.97	0.97	**0.98**	0.98	0.97
DOS_TCP	*0.64*	0.74	1.0	**1.0**	1.0	0.78	0.99
Data exfiltration	*0.72*	0.76	0.96	**0.97**	0.97	0.94	0.97
Keylogging	*0.72*	0.82	0.95	0.95	0.95	0.91	**0.98**
Service_Scan	*0.73*	0.83	0.95	**0.95**	0.95	0.94	0.94
OS_Scan	*0.72*	0.76	0.94	0.97	0.98	0.97	**0.99**

Phase 2: applying machine learning algorithms on the entire dataset with a set of features the combined the best features for each attack. The whole dataset is used in this phase. Seven different methods of machine learning were implemented on the entire dataset, and we used feature sets that were extracted for each attack separately. Table IV shows the results obtained by using 13 features extracted for the attacks.

TABLE IV. IMPLEMENTATION OF FEATURES OBTAINED FROM PHASE1

ML Algorithm	Accuracy	Precision	Recall	F-Measure	Time
NB	0.78	0.84	0.78	*0.75*	**5.056**
QDA	0.88	0.89	0.88	0.87	6.1964
RF	0.98	0.98	0.98	0.98	27.0328
ID3	0.99	0.99	0.99	0.99	19.3447
Adaboost	1.0	1.0	1.0	**1.0**	308.9403
MLP	0.84	0.88	0.84	0.83	1011.5001
KNN	0.99	0.99	0.99	0.99	*2052.1801*

When observing Table IV, it can be seen that Adaboost was the best performance algorithm, followed by KNN and ID3. ID3 is noticeably faster than KNN, so it takes precedence with this feature. The lowest scoring algorithm was Naive Bayes, with a score of 0.75. From the speed perspective, NB and QDA were the fastest. Although KNN had a high performance score, it was still noticeably slower than the other algorithms.

Phase 3: applying machine learning algorithms on the entire data set with the seven best features obtained in the feature selection step.

From the F-measure perspective, there was no significant change in the algorithms' performance, but from the speed perspective, the running times of all the algorithms were noticeably reduced. The reason for this reduction in execution time is that 13 attributes are used in the method applied in Table V, whereas only 7 attributes are used in Table IV. This reduction in the feature count reduced the running time of the machine learning algorithms.

TABLE V. IMPLEMENTATION OF FEATURES OBTAINED USING RANDOM FOREST REGRESSOR FOR ALL DATASET

ML Algorithm	Accuracy	Precision	Recall	F-Measure	Time
NB	0.79	0.85	0.79	*0.77*	**4.0472**
QDA	0.87	0.89	0.87	0.86	4.4056
RF	0.97	0.97	0.97	0.97	28.9246
ID3	0.97	0.97	0.97	0.97	17.0899
Adaboost	0.97	0.97	0.97	0.97	238.8618
MLP	0.84	0.87	0.84	0.83	949.6977
KNN	0.99	0.99	0.99	**0.99**	*1615.9852*

The final results of the implementation (see Table VI) are compared with a study in the literature. For this comparison, the study conducted by Ferrag et al. [14] in 2019 was chosen. The reason for this is that the mentioned work used the same dataset as well as two machine learning methods similar to the ones we used. These similar machine learning algorithms are Random Forest and Naive Bayse. The key difference between our work and theirs is the feature set used. They used the original feature set while we used a new feature set extracted by CICFLOWMETER. The detection rate (Recall) was determined as the main evaluation criterion. Table VI shows the comparison of the results obtained from the two studies. When the results are examined, it can be seen that the Random Forest algorithm used in our study is higher than that used in [14], and the same thing can be seen for most attack types with the NB algorithm. So, we can see that the new features used in our work increased the performance of both algorithms.

TABLE VI. COMPARISON OF PERFORMANCE OF THE TWO ALGORITHMS

Attack Names	Ferrag et al[14]		Our Work	
	RF	NB	RF	NB
DDOS_HTTP	82.26%	50.78%	96%	71%
DDOS_TCP	88.28%	78.67%	99%	70%
DDOS_UDP	55.26%	78.50%	98%	72%
DOS_HTTP	82.20%	68.68%	95%	71%
DOS_TCP	81.77%	65.56%	100%	63%
DOS_UDP	82.99%	100%	97%	71%
Data exfiltration	86.55%	66.55%	96%	71%
Keylogging	70.12%	65.62%	95%	71%
OS_Scan	82.20%	68.68%	94%	70%
Service_Scan	69.82%	65.21%	95%	72%

VI. CONCLUSION

This paper has aimed to detect IoT network attacks by using machine learning methods. In this context, the Bot IoT [9] was used as a dataset because of its regular updates, wide attack diversity, and various network protocols. We used CICFlowMeter[25] to extract flow-based features from the raw traffic traces. CICFlowMeter generates 84 network traffic features of the dataset which define the network flow. During the implementation, the importance of weight calculations were made with the Random Forest Regressor algorithm to decide which of the features would be used in the machine learning methods. Two approaches were used when making these calculations. In the first approach, the importance weights were calculated separately for each attack type, and in the second approach, all the attacks were collected in a single group and the importance weights for this group were calculated; i.e., the common properties that were important for all attacks were determined. Finally, seven machine learning algorithms which are widely used and have different qualities were applied to the data. These algorithms and the achieved performance ratios according to F-measure are as follows: F-measure had a value between 0 and 1; Naive Bayes was 0.77; QDA was 0.86; Random Forest was 0.97; ID3 was 0.97; AdaBoost was 0.97; MLP was 0.83; and K Nearest Neighbours was 0.99.

In this research we investigated seven supervised algorithms. As a future work ,it would be interesting to evaluate the performance of some unsupervised algorithms. Furthermore, we applied various machine learning algorithms independently from each other. In the future, we sould like to combine different machine learning algorithms as a multi-layered model to improve the detection performance.

ACKNOWLEDGMENT

The computations for the work presented in this paper were supported by the KAU High Performance Computing Center (Aziz Supercomputer) (http://hpc.kau.edu.sa)

REFERENCES

[1] J. Deogirikar and A. Vidhate, "Security attacks in iot: A survey," *International Conference on I-SMAC (I-SMAC)*, pp. 32–37, 2017.

[2] T. Bodström and T. Hämäläinen, "State of the art literature review on network anomaly detection with deep learning," *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 64–76, 2018.

[3] I. Arnaldo, A. Cuesta-Infante, A. Arun, M. Lam, C. Bassias, and K. Veeramachaneni, "Learning representations for log data in cybersecurity," *International Conference on Cyber Security Cryptography and Machine Learning*, pp. 250–268, 2017.

[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1298, 2017.

[5] B. J. Radford, B. D. Richardson, and S. E. Davis, "Sequence aggregation rules for anomaly detection in computer network traffic," *arXiv preprint arXiv:1805.03735*, 2018.

[6] I. Lambert and M. Glenn, "Security analytics: Using deep learning to detect cyber attacks," 2017.

[7] M. Stevanovic and J. M. Pedersen, "Detecting bots using multi-level traffic analysis." *IJCSA*, vol. 1, no. 1, pp. 182–209, 2016.

[8] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource iot devices: A game-theoretic methodology," *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.

[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[10] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[11] X. Yuan, C. Li, and X. Li, "Deepdefense: identifying ddos attack via deep learning," *IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–8, 2017.

[12] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[13] M. K. Putchala, "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)," 2017.

[14] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, 2019.

[15] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," *arXiv preprint arXiv:1709.04647*, 2017.

[16] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques," *International Conference on Mobile Networks and Management*, pp. 30–44, 2017.

[17] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Rule generation for signature based detection systems of cyber attacks in iot environments," *Bulletin of Networking, Computing, Systems, and Software*, vol. 8, no. 2, pp. 93–97, 2019.

[18] V. H. Bezerra, V. G. T. da Costa, S. B. Junior, R. S. Miani, and B. B. Zarpelao, "One-class classification to detect botnets in iot devices," *Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pp. 43–56, 2018.

[19] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," *International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, 2016.

[20] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for internet of things devices," *IEEE 34th international performance computing and communications conference (IPCCC)*, pp. 1–8, 2015.

[21] F. Y. Yavuz, "Deep learning in cyber security for internet of things," Ph.D. dissertation, 2018.

[22] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," *arXiv preprint arXiv:1905.05137*, 2019.

[23] I. Cvitić, D. Peraković, M. Periša, and M. Botica, "Novel approach for detection of iot generated ddos traffic," *Wireless Networks*, pp. 1–14, 2019.

[24] Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, C. So-In *et al.*, "Averaged dependence estimators for dos attack detection in iot networks," *Future Generation Computer Systems*, vol. 102, pp. 198–209, 2019.

[25] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features." *ICISSP*, pp. 253–262, 2017.

[26] S. Yu, "Study on the internet of things from applications to security issues," *International Conference on Cloud Computing and Security*, pp. 80–89, 2018.

[27] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[28] K. Kostas, "Anomaly detection in networks using machine learning," Ph.D. dissertation, 08 2018.

[29] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International journal of computer science and network security*, vol. 7, no. 12, pp. 258–263, 2007.