Hittite Journal of Science and Engineering, 2020, 9 (2) 73-82
ISSN NUMBER: 2148-4171
DOI: 10.17350/HJSE19030000257

Development and Evaluation of Ensemble Learning Models for Detection of Distributed Denial-of-Service Attacks in Internet of Things

Yildiran Yilmaz[1] **Selim Buyrukoglu**[2]
[1]Recep Tayyip Erdogan University, Computer Engineering, Rize, Turkey
[2]Cankiri Karatekin University, Computer Engineering, Cankiri, Turkey

ABSTRACT

Internet of Things that process tremendous confidential data have difficulty performing traditional security algorithms, thus their security is at risk. The security tasks to be added to these devices should be able to operate without disturbing the smooth operation of the system so that the availability of the system will not be impaired. While various attack detection systems can detect attacks with high accuracy rates, it is often impossible to integrate them into Internet of Things devices. Therefore, in this work, the new Distributed Denial-of-Service (DDoS) detection models using feature selection and learning algorithms jointly are proposed to detect DDoS attacks, which are the most common type encountered by Internet of Things networks. Additionally, this study evaluates the memory consumption of single-based, bagging, and boosting algorithms on the client-side which has scarce resources. Not only the evaluation of memory consumption but also development of ensemble learning models refer to the novel part of this study. The data set consisting of 79 features in total created for the detection of DDoS attacks was minimized by selecting the two most significant features. Evaluation results confirm that the DDoS attack can be detected with high accuracy and less memory usage by the base models compared to complex learning methods such as bagging and boosting models. As a result, the findings demonstrate the feasibility of the base models, for the Internet of Things DDoS detection task, due to their application performance.

Article History:
Received: 2021/08/17
Accepted: 2022/04/22
Online: 2022/06/30

***Correspondence to**: Yildiran Yilmaz,*
Recep Tayyip Erdogan University,
Computer Engineering, Rize, Turkey
E-Mail: yildiran.yilmaz@erdogan.edu.tr
Phone: +90 464 223 75 18 (1242)

Keywords:
DDoS detection; Base-Learner algorithms; Bagging; Boosting; IoT devices.

INTRODUCTION

The Distributed Denial-of-Service (DDOS) attack, which is a type of cyber-attack based on preventing a device or network resources from being accessed by temporarily or indefinitely disrupting the services of a host connected to the Internet, causes enormous economic losses [1]. For example, the DDOS attacker sends excessive requests to the target web server in IoT and prevents this website from working correctly by exceeding its capacity to process multiple requests. Producing a real-time attack detection system with a low cost in terms of computational burden remains one of the foremost challenges.

Internet of Things (IoT) devices pose a greater risk than other computing devices in public networks because firmware updates and maintenance are not accomplished on most IoT devices after deployment [2]. For instance, in an IoT network, an attacker initiates a DDoS attack on DHCP clients (which is called DHCP starvation attacks) so that genuine clients cannot obtain their IP addresses and their availability will be compromised. The attacker allocates all the IP addresses in the IP pool to himself with malicious DHCP discover messages and finally, there will be no IP left in the pool for real clients. However, traditional DDoS and Host-based intrusion detection systems require a lot of communicational and computational power on a device to be able to run smoothly [3].

Fortunately, detection of DDOS attacks on both client and server is possible by exploiting transaction rates data on the client and latency data on the server-side [4]. This detection can be achieved using machine learning algorithms along with both data such as transac-

tion rates and latency. By using machine learning methods independently on both sides, and setting them up to work concurrently, DDOS attacks can be detected in traditional networks. However, in IoT networks that contain resource-constrained devices, attack detection using machine learning methods is a difficult task to establish due to the scarce computation and communication resources. While establishing an attack detection method on such IoT networks, it should be considered that the client can perform the detection method without compromising the system availability. When the client is unable to operate the detection method due to the overflow, the client will be unavailable. Therefore, this paper analyses robust DDOS detection methods that use fewer resources e.g. RAM and ROM memory. The proposed study employs base (Logistic Regression, Support Vector Machine, Naive Bayes, Artificial Neural Network), bagging (Random Forest), and boosting (Adaboost) ensemble methods after the information gain (IG) based feature selection techniques jointly. The feature selection method selects the most significant attributes from the dataset CIC-DDoS2019 [4] such that the 2 most significant features are selected out of 79 features for the proposed method. Then the base, bagging, and boosting methods classify the communication traffic into benign(innocuous) and DDoS attack traffic. Bagging and boosting are popular ensemble approaches used for classification and regression in different fields [5]. In the bagging ensemble approach, weak learners are built independently from each other. Then, the output of the weak learners is completed by applying any aggregation technique to complete the classification or regression process. In contrast to bagging, weak learners are built sequentially in boosting approach, and then observations are weighted because some of the observations are used in the new sets.

In the literature, DDOS detection methods can be classified into two groups. The first group exploits matching patterns and the second group practices detection depending on anomalies. The first group compares the attack characteristics stored in the library with the input stream properties. In cases where there is a match, the stream is considered as part of the DDoS attack traffic [6,7]. The biggest disadvantage of this method is that it cannot provide any security in case of an attack-type missing in the library [8]. The second group detection method can detect malicious traffic containing DDOS anomalies using machine learning techniques [9]. However, if these methods are installed in a resource-limited controller without proper collection of network traffic, it may cause overloads within the control and data units. As an example, among the anomaly detection techniques, floating-point-based detection operations can be given [10]. Such overloading also leads to the unavailability of the IoT device and false positives. There are also several solutions to prevent DDoS attacks by using matching patterns and machine learning methods such as SVM [11], NN [12], Naive Bayes [13], Random Forest [14], and Deep Neural Network-DNN [34]. In a different study, Naive Bayes, Random Forest, Multilayer Perceptron (MLP), and J48 machine learning algorithms were employed for the DDoS attack detection [35]. Moreover, light gradient boosting machine learning algorithm was used for the detection of DDoS attacks [36]. A similar study with [35] was proposed for DDoS attack detection employing k-Nearest Neighbor (kNN), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM) algorithms [37]. These solutions are mostly aimed at increasing the accuracy of DDOS detection, but they do not focus on providing DDOS detection solutions for specifically resource-constrained Internet of Things, taking into account resource usage.

Even if the machine learning models have been employed for the detection of DDOS attacks in IoT, to the best of our knowledge, there is an apparent lack of literature in the appropriate assessment of the cost of implementing base, bagging, and boosting models in IoT devices. The absence of the before-mentioned analysis is an important limitation to the broader adoption of these models in IoT. To this end, this paper proposes the following contributions:

1. New DDOS detection models combining base, bagging, and boosting algorithms with (IG) feature selection for resource-constrained IoT networks, separately.

2. Evaluation of memory consumption of base, bagging, and boosting algorithms on the client-side which has scarce resources.

3. Detection of DDOS attacks with 99.5% accuracy by considering an analysis of resource consumption in an IoT device.

The remainder of this paper is as follows. The next section declares the proposed approach for the DDOS detection method and gives information about the employed feature selection method and base, bagging, and boosting methods. The following section includes the results and the evaluation of the implemented DDoS detection methods. The final section concludes the paper.

THE PROPOSED APPROACH FOR DETECTION OF DDOS ATTACKS IN IOT

This section provides background information about the implemented base, bagging, and boosting algorithms in this paper for the purpose of comparison for the detection of DDOS attacks in IoT. The proposed solution combines the information gain-based feature selection method with base, bagging, and boosting classification

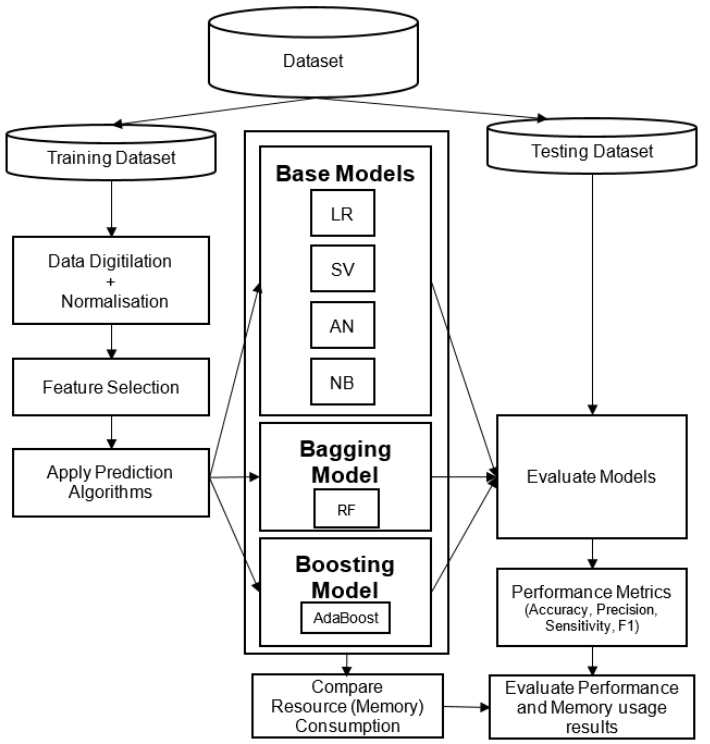

Figure 1. The diagram of the proposed approach.

algorithms. Fig. 1 shows the diagram of the proposed approach for the detection of DDOS attacks in IoT.

Dataset Description

Several studies towards attack classification have been performed on various valid and current data sets, such as DARPA, KDD'99, DEFCON, CDX, KYOTO, TWENTE, UMASS, ISCX [4, 15]. The CIC-DDoS2019 [4] data set shared by the Canadian Cyber Security Institute, prepared in a suitable test environment, was created by remedying the shortcomings in the previously used data sets. This data set was produced in a one-day training and one-day testing process, taking into account the communication traffic that includes malicious and benign behaviour.

These data shared by the Canadian Cyber Security Institute are public for researchers who want to work in the field of cybersecurity [4]. This dataset was created based on the behaviour of 25 users using HTTP, HTTPS, FTP, SSH and e-mail protocols by simulating realistic background network traffic.

Data Preprocessing

This work implements preprocessing on the data set by applying data digitisation and normalization processes before feature selection and classification. This is because most machine learning models can only work with numerical values for training and testing purposes. Therefore, it is required to convert all non-numeric values to numerical values by performing data preprocessing. In literature [16], there are two methods of performing data digitisation. The first method (i.e. single-hot coding) assigns a different binary vector to each type of nominal property. The second method lists the values alphabetically for each nominal property. The nominal values listed are then converted to numeric values by assigning specific values to each variable. This paper employs the latter method because it offers the following advantages compared to (the single-hot coding) method. The latter method does not increase the number of features as each nominal feature is represented by a value. In contrast, the first method increases the number of features because each nominal feature is represented by a binary vector whose length depends on the number of nominal property values.

Consequently, in the case of using the latter method, the architecture of the models will be more compact than using the first method because inputs for the model will be less. Therefore, the latter method not only lessens the training and testing time but also reduces memory consumption. Min-Max transformation was used to normalise the values in the data set and the data were scaled linearly in the range of [0,1].

Table 1. The selected features for DDOS attack type.

Feature	Description
Init Forward Win Bytes	The number of bytes sent in the forward direction in the first frame.
Packet Length Var	Length variance of a packet.

$$z_i = (x_i - min) / (max - min) \quad (1)$$

where x_i is a numerical value of a feature, *Min* and *Max* are the minimum and maximum values of each numerical feature respectively.

Data engineering is necessary for digital transformation including data digitisation and normalization processes. The IoT networks generate unimaginable volumes of data and this makes data complicated. In order for all network data to make sense, the quality, availability and security of this data must be ensured by data engineering. This is the reason to see the role of data engineering grow in importance. Therefore, the data preprocessing part is to process the data in such a way that machine learning algorithms described in the following section can extract value from it.

Base Models

This section describes the phases of the lightweight DDOS detection method and other base detection models. There are two main phases after the data preprocessing stage which include data digitisation and normalization. Two main phases such as feature selection and DDOS detection are explained as follows.

Phase 1: Feature Selection

The feature selection phase was performed after the data preprocessing operations, as an additional operation to increase the performance of the learning algorithms. The information gain algorithm [17] which is a feature selection algorithm developed by Quinlan, was used in this phase. This algorithm is used to select the test attribute in each node created with decision trees. Information gain is one of the Entropy-based methods used to estimate losses when the data set is separated by attributes [18]. Entropy is a value between 0 and 1 representing the uncertainty of the system. If the entropy value is closed to 1, this means that the system contains more information. The entropy values are calculated separately for each feature in the data set. Information gain gives the value of representing the entire data set only with selected features. At the stage of feature selection with the information gain method, the attribute variables that are incomplete or insufficient in defining the system are removed from the data set and the remaining attribute variables are used to train the machine learning algorithm [18]. Within all datasets including normal and abnormal behaviours, only 2 features out of 79 were selected as a result of the feature selection as they give better results. The selected features and their descriptions for the DDOS attack type are given in Table 1.

Phase 2: DDoS Detection

In the DDoS detection phase, the base model along with bagging and boosting models which are ensemble approaches are used to classify the communication traffic into benign(innocuous) and DDoS attack traffic.

Logistic Regression (LR)

In the detection phase, Logistic Regression (LR) which is a statistical method used to analyse the data set with two independent features selected in the first phase was used for the classification of DDOS attacks. As logistic regression is used in the modelling of numerical variables that give binary results, in this study, it is used to determine whether the network traffic is normal or there is a DDoS attack in the network. LR algorithm is a multivariate statistical analysis method that is preferred in terms of allowing simple regression model creation without the need for assumptions compared to other algorithms [19]. The LR classification algorithm is formulated with an equation as follows [20].

$$P = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}} \quad (2)$$

Due to its high accuracy and low memory requirements, the LR algorithm has been adopted and applied as a machine learning algorithm suitable for the data set and the classification problem. In this study, Artificial Neural Networks, Support Vector Machines, Naive Bayes, and Random Forest classification algorithms are also used for comparison.

Table 2. Acronyms List.

Variable	Description
β_0	Fixed Value
$\beta_1, \beta_2, ..., \beta_k$	Regression coefficient for independent variables
$X_1, X_2, ..., X_k$	Independent variables
k	Number of independent variables
e	Euler's number

Support Vector Machine (SVM)

Support vector machine algorithm is based on statistical learning theory and was developed by Vapnik for the solution of prediction and classification problems[21]. The purpose of SVM is to obtain the best hyperplane to separate the classes from each other in k-dimensional space(k is the number of attributes). In other words, it aims to maximise the distance between support vectors belonging to the classes [22]. The hyperplane that can make the most appropriate separation by maximising the limit is called the optimum hyperplane that precisely divides the data points. The data points determining the border width are called support vectors. SVM is broadly utilised in classification problems as in [23] where authors endeavour to detect the DDoS attack traffic in Software-Defined Ad-Hoc Networks.

In this study, the DDoS classification was performed based on the optimum hyper line dividing the data points. It is also called the decision boundary which classifies points that fall on one side as a group, and points that fall on the other side as another group. As a consequence of the attempts, the kernel function with the highest performance was determined as polynomial and the regression loss epsilon was 0.20 and numerical tolerance was 0.001.

Naive Bayes (NB)

Naive Bayes classifiers use Bayes' theorem, which shows the relationship between conditional probabilities and marginal probabilities within the probability distribution for a random variable [24]. Although it is known as a lazy learning algorithm, it works considerably successfully on unstable data sets. The working logic of the algorithm is as follows. It calculates the probability of each state for any element and makes a classification according to the one with the highest probability value. It can also work very successfully on small data sets [25]. For example, the author in [26] created a network protection method based on a Naive Bayes classifier that detects the DDoS attack traffic to ensure cloud security.

In the Naive Bayes method, the classification was performed with a fundamental assumption between predictors such that each feature gives an identical and independent contribution to the result. Therefore, it gives better results than other classifiers when there is less training data.

Artificial Neural Network (ANN)

ANN is a mathematical model consisting of many neurons that are connected to each other in a weighted manner and is a technology developed by completely sampling the human brain [27]. It consists of several components described as follows. Inputs are data coming to neurons. Weights show the influence and effect of the independent variable arriving in the artificial neuron. The addition function is a function that calculates the net input of an artificial neuron after adding up the inputs multiplied by the weights. The activation Function takes the net input produced by the addition function and produces a result depending on the function of the activation function. Finally, the output represents the value generated by the activation function.

As this study is a classification problem, the classification was performed based on the output value collected from output neurons. Therefore, as a consequence of the attempts, the number of neurons in hidden layers with the best performance was determined as 100 and the activation function was ReLu and the number of features was determined as 2.

Bagging Model

Random Forest (RF)

The Random Forests (RF) or random decision forests algorithm is an ensemble learning method that aims to increase classification accuracy by generating more than one decision tree during the classification process. The decision trees created individually assemble to form a decision forest. The RF classifier consists of a combination of tree classifiers in which each classifier is created using a random vector sampled independently of the input vector. Each tree gives a unit vote for the most popular feature to classify an input vector [28]. Tree models grow to maximum depth on new data using a combination of features. Therefore, the Random forest algorithm performs well in large data sets, unlike the naive Bayes algorithm. It produces more accurate results than Support Vector Machines for many data sets [29]. For example, it works well in datasets that contain categorical variables with a large number of variables and class labels, as well as on datasets with an uneven distribution and missing data [30].

Because this study is a classification problem, the classification was performed based on the superiority of the predictions collected from the trees, whereas, unlike regression problems, the decision is made by averaging the results. Therefore, as a consequence of the attempts, the number of trees with the best performance was determined as 10 and the number of features, and the number of attributes granted at each split was determined as 2.

Boosting Model

AdaBoost

AdaBoost is an efficient ensemble learning model for the purpose of classification [32]. It aims to combine the output of the weak learners for obtaining better performance compared to the weak learners into a weighted sum. In the classification process of this model, a sequential path is followed in which weak learners are adjusted in favour of cases that were misclassified by former classifiers.

In this paper, various estimator numbers (25, 50, 100, 250) were used to obtain the optimum result in the building of the AdaBoost model for the detection of DDOS attacks in IoT. The optimum result was obtained with 50 estimators.

RESULTS AND DISCUSSION

This section provides performance results of the base, bagging and boosting models as well as their memory consumption estimations on the IoT device called RPI-Pico. Subsequently, those results are discussed in this section.

Performance Metrics

This paper focuses on finding the best features of the DDOS attack using the current data set CSE-CIC-IDS2019 and reducing the training and test time costs of the data sets by expressing the samples in the data set with fewer features. After the data sets with selected features were created, Machine Learning Methods were applied to these sets. It is aimed to evaluate the performance of classification algorithms for the data set of the DDOS attack type. As regards the evaluation of different machine learning methods, this paper uses performance metrics to decide the most appropriate model amongst classification models. As DDOS detection is a classification problem, the robust evaluation metrics used to assess the classification methods in this paper are Area under the ROC Curve (AUC), Classification Accuracy (CA), F1 score, Precision, Sensitivity. The performance metrics taken into account in this paper are described as follows.

$$CA = (TN + TP) / (TN + TP + FN + FP) \quad (3)$$

$$\Pr ecision = TP / (TP + FP) \quad (4)$$

$$Sensitivity = TP / (TP + FN)$$

$$F1 = 2 * (Precision * Sensitivity) / (Precision + Sensitivity) \quad (5)$$

where CA is classification accuracy and TP, FP presents the outcomes of true-positive, false-positive and TN, FN signifies the outcomes of the true-negative, false-negative respectively.

Other performance metrics for the purpose of comparison base, bagging and boosting models are avg. RAM and ROM consumptions. Those metrics are measured by using the MSP430-size tool on the Raspberry PI Pico [31].

Performance Comparison of Base, Bagging and Boosting Learning Models

Table 3 presents the performance results of the base, bagging and boosting. Random Forest (Bagging) algorithm achieved to provide the best results of AUC, CA, F1, Precision and Recall. On the other hand, Logistic Regression (base model) provided the worst result for the detection of DDOS attacks in IoT. However, in general, the results of the models presented in Table 3 have nearly similar results for AUC, CA, F1, Precision and Recall, even if the Random Forest (bagging) provided the best performance for detection of DDOS attacks in IoT.

As seen in Table 3, bagging and boosting ensemble learning algorithms have been carried out for DDoS attack detection. However, the use of bagging and boosting ensemble learning algorithms are not common in the detection of DDoS attack detection. Even if some of the studies have used bagging or boost ensemble algorithms for the DDoS detection, most of the studies employed single based learners such as SVM, Naive Bayes, Multilayer Perceptron (MLP), k-Nearest Neighbor (kNN) [11-12, 34-35, 37]. Moreover, bagging and boosting ensemble learning algorithms are not compared for the detection of DDoS attacks in a

Table 3. Performance results of the base, bagging and boosting algorithms.

Model	Algorithm	AUC	CA	F1	Precision	Recall
Base	Logistic Regression	0,983	0,972	0,968	0,975	0,962
	SVM	0,993	0,980	0,980	0,991	0,980
	Naive Bayes	0,994	0,984	0,979	0,977	0,984
	Neural Network	0,995	0,986	0,986	0,986	0,986
Bagging	Random Forest	0,997	0,990	0,991	0,994	0,990
Boosting	AdaBoost	0,985	0,975	0,970	0,987	0,975

Table 4. The memory consumption results of the base, bagging and boosting models on the RPI pico device.

Model	ML Algorithm	text	data	bss	dec	hex
Base	LR	4237	304	4	4545	11c1
	NB	11063	312	4	11379	2c73
	SVM	12473	428	728	13629	353d
	ANN	15586	348	32808	48742	be66
Bagging	RF	32328	2824	20	35172	8964
Boosting	Adaboost	38618	964	0	39582	9a9e

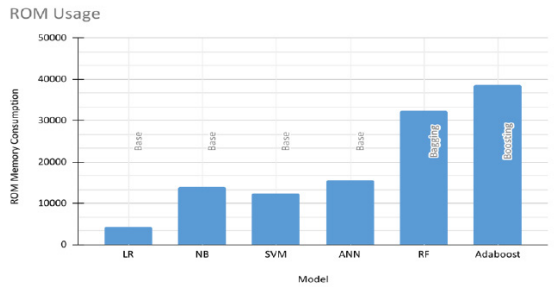

Figure 2. The ROM memory usage of the base, bagging and boosting classification algorithms in IoT device RPI Pico.

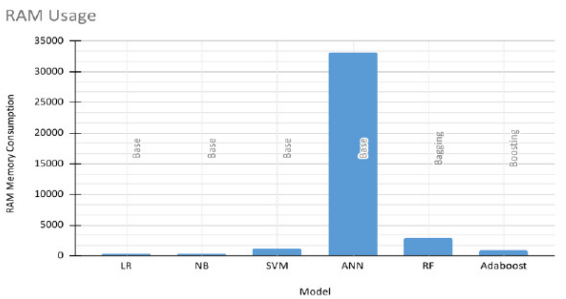

Figure 3. The RAM usage of the base, bagging and boosting classification algorithms in IoT device RPI Pico.

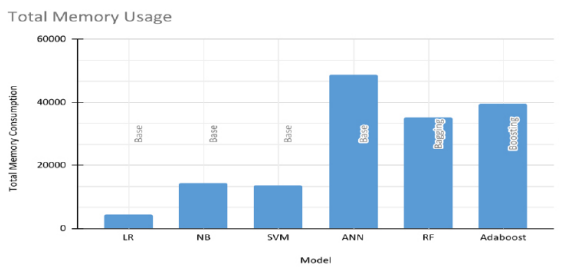

Figure 4. The total memory usage of the base, bagging and boosting classification algorithms in IoT device RPI Pico.

study. For instance, light gradient boosting machine learning algorithm was used for the detection of DDoS attacks [36]. On the other hand, the presented studies in the literature have not focused on the resource consumption of the single-based, bagging and boosting models. Next section gives information about the resource consumption of the employed algorithms in this study.

Resource Consumption Comparison of the Base, Bagging and Boosting Models

Table 4 shows the memory consumption results of the base, bagging and boosting models on the RPI pico device. The text section in the table indicates the Read-Only Memory (ROM) usage in bytes which is the size of the code segment. The data and bss section indicate the Random Access Memory (RAM) partitions holding variables in RAM.

The bagging model improves the accuracy by reducing overfitting in decision trees and is adaptable to both regression and classification challenges and runs efficiently on continuous and categorical continuous data and ignores missing values found in data [32]. The boosting model increases the accuracy by merging weak learning methods and by using multiple classifiers. However, those algorithms also have several drawbacks [33]. They demand considerable computational capability moreover memory resources e.g. RF builds a great deal of trees to combine its outputs. On the other hand, for NN implementation on the device, the memory usage of the models for two layers with 8 hidden neurons and four layers with 100 hidden neurons exemplify the best and worst instances. Therefore the memory usage for NN depends on the number of layers and neurons. The number of neurons in hidden layers with the best performance was determined as 100 in this work. The detection of DDOS attack is achieved by the Logistic Regression method consuming less memory than heavy computational ML algorithms as can be seen in Table 4. This method meets the following requirements for IoT networks, smooth implementation on IoT devices, less memory consumption in the resource-constrained IoT devices such as clients, and detection of DDOS attacks with high accuracy equivalent to heavy computational ML algorithms.

Based on the results illustrated in Fig. 4, Logistic Regression, Naive Bayes and Support Vector Machine can be classified as relatively lightweight machine learning algorithms. For comparison and upper bound calculation, more complex learning algorithms, such as random forest and a feed-forward neural network, were additionally applied

Table 5. The selected features for DDOS attack type.

Model	Algorithm	DPCR
Base	Logistic Regression	983000
	SVM	993016
	Naive Bayes	994170
	Neural Network	995372
Bagging	Random Forest	996506
Boosting	AdaBoost	985000

in this work. LR model trained for DDOS classification is 4545 bytes in size and achieves an accuracy of 97.2%. NB model trained for DDOS classification is 11379 bytes in size and achieves an accuracy of 98.4%. SVM model for DDOS classification is 13629 bytes in size and achieves an accuracy of 98%. RF model for DDOS classification is 35172 bytes in size and achieves an accuracy of 99%. NN model for DDOS classification is 48742 bytes in size and achieves an accuracy of 98.6%. Based on the results shown in Fig. 2, the algorithms that require the least ROM usage and the most ROM usage are LR and AdaBoost, respectively. Besides, based on the results shown in Fig. 3, the least RAM usage and the most RAM usage are required by LR and ANN, respectively. Based on the total memory consumption results, the algorithms that require the least memory usage and the most memory usage in total are LR and ANN, respectively.

In terms of the DDoS detection accuracy results, no matter how high these accuracy rates may seem, these accuracy rates still need to be improved by considering the smooth implementation of these algorithms. The reason for this and the discussion of these accuracy rates will be in the following section.

DISCUSSION

The packet capture rate of DDoS packets in an outgoing network of 1 million packets per second is calculated as follows.

$$DPCR = \frac{Accuracy\ Rate}{100} * 1000000 \quad (6)$$

DPCR stands for DDoS Packet Capture Rate. DDoS traffic capture rates on the network for each detection model are estimated based on this formula. Therefore, the rate of unsuccessful DDoS traffic capture on the network for each detection model can be estimated by (1 million - DPCR). The average DDoS packet capture rates based on detection accuracies of the three models are shown in Table 5.

Based on the results shown in Table 5, the number of unsuccessful DDoS traffic capture per second on the network for the base (LR, SVM, NB, NN), bagging and boosting models are 17000, 6983, 5829, 4627, 3493, 15000 respectively. This means that 15000 packets out of 1 million DDOS packets per second will be included in the traffic without being recognised by e.g. the boosting model.

Fig. 5 illustrates the number of DDoS attack packets during ten seconds that the bagging and boosting models are unable to capture on the network. As can be seen in Fig. 5, there are enormous undetected DDoS packets. For example, the boosting model cannot detect 75 thousand attack packets in 5 seconds and this number doubles in ten seconds and becomes 150 thousand. Therefore, it is still essential for machine learning models to increase their capture rate of DDoS attack packets by improving their accuracy so that services and hosts in the network with heavy network traffic and running for long periods of time can be accessible. For especially IoT networks, it is very important that DDoS detection models not only increase their accuracy but also be smoothly applicable. This is because there are a great deal of resource-constrained service providers and hosts in IoT networks. Therefore, this study evaluates and compares different machine learning models for DDoS detection along with the assessment of the memory cost of implementing base, bagging, and boosting models in IoT devices. Consequently, such analysis in this study will contribute to the broader adoption of these models, especially in the IoT ecosystems occupied with resource-constrained IoT devices.

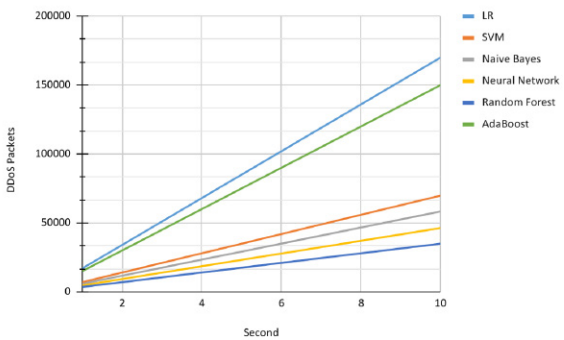

Figure 5. The number of DDoS packets that base, bagging, and boost models fail to capture on the network during ten seconds.

CONCLUSION

This paper uses the base, bagging and boosting models that detect DDOS attacks and evaluates their performance on an IoT device named RPI-Pico. The accuracy results and memory usage of the aforementioned models created with machine learning algorithms were evaluated. This paper employs the CSE-CIC-IDS2019 data set prepared in the appropriate test environment considering the deficiencies in the current and old data sets. It is ensured that the size of the data sets employed was reduced by obtaining the best features for the DDOS attack type. In addition, it has been observed that the base model achieved high accuracy results by using less memory resources. The base methods were compared with other classification models such as bagging and boosting applied to the feature selected data set. Future work can be expanded in order to detect more cyber threads in an IoT environment by performing future Selection for any cyber-attack in a comprehensive dataset including DOS detection, BOT detection, brute-force detection, and intrusion detection.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

AUTHOR CONTRIBUTION

Public data were used in this study. The rest of all sections including conceptualisation, methodology, software, analysis, writing, review and editing were equally organised and performed by Yıldıran Yılmaz and Selim Buyrukoğlu.

References

1. Salim, M. M., Rathore, S., & Park, J. H. Distributed denial of service attacks and its defenses in IoT: a survey. The Journal of Supercomputing, 76(7), 5320-5363, 2020.
2. Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84, 2017.
3. Vishwakarma, R., & Jain, A. K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommunication systems, 73(1), 3-25, 2020.
4. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST), 1-8. IEEE, 2019.
5. Sutton, C. D. Classification and regression trees, bagging, and boosting. Handbook of statistics, 24, 303-329, 2005.
6. Dang-Van, T and Truong-Thu, H. A Multi-Criteria based Software Defined Networking System Architecture for DDoS-Attack Mitigation. REV J. Electron. Commun., vol. 6, no. 3, pp. 50–60, 2017, doi: 10.21553/rev-jec.123.
7. Al-Duwairi, B., Al-Kahla, W., AlRefai, M. A., Abdelqader, Y., Rawash, A., and Fahmawi, R. SIEM-based detection and mitigation of IoT-botnet DDoS attacks. Int. J. Electr. Comput. Eng., vol. 10, no. 2, pp. 2182–2191, 2020, doi: 10.11591/ijece.v10i2.pp2182-2191.
8. Mubarakali, A., Srinivasan, K., Mukhalid, R., Jaganathan, S. C. B., and Marina, N. Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems. Comput. Intell., vol. 36, no. 4, pp. 1580–1592, 2020, doi:10.1111/coin.12293.
9. Dong P, Du X, Zhang, H., and Xu, T. Adetectionmethod for a novel DDoS attack against SDN controllers by vast new low-traffic flows. IEEE International Conference on Communications (ICC); May 22-27, 1-6, 2016.
10. Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. International Conference on Computing, Networking and Communications (ICNC); February 16-19, 2015.
11. Li, J. IOT security analysis of BDT-SVM multi-classification algorithm. International Journal of Computers and Applications, 1-10, 2020.
12. Ma, L., Chai, Y., Cui, L., Ma, D., Fu, Y., & Xiao, A. A deep learning-based DDoS detection framework for Internet of Things. In ICC IEEE International Conference on Communications (ICC), 1-6, IEEE, 2020.
13. Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., & Sakurai, K. Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors, 20(16), 4372, 2020.
14. Karthik, M. G., & Krishnan, M. M. Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks. Journal of Ambient Intelligence and Humanized Computing, 1-11, 2021.
15. Agarwal, M., Biswas, S., & Nandi, S. Detection of de-authentication dos attacks in wi-fi networks: A machine learning approach. In 2015 IEEE International Conference on Systems, Man, and Cybernetics, 246-251, 2015.
16. Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. Big data preprocessing: enabling smart data. Springer Nature, 2020.
17. S. Lei. A Feature Selection Method Based on Information Gain and Genetic Algorithm. International Conference on Computer Science and Electronics Engineering, 355-358, 2012, doi: 10.1109/ICCSEE.2012.97
18. Omuya, E. O., Okeyo, G. O., & Kimwele, M. W. Feature Selection for Classification using Principal Component Analysis and Information Gain. Expert Systems with Applications, 174, 2021.
19. Ahmed, A., Jalal, A., & Kim, K. A novel statistical method for scene classification based on multi-object categorization and logistic regression. Sensors, 20(14), 3871, 2020.
20. Alasmary, H., Khormali, A., Anwar, A., Park, J., Choi, J., Abusnaina, A., & Mohaisen, A. Analyzing and detecting emerging internet of things malware: A graph-based approach. IEEE Internet of Things Journal, 6(5), 8977-8988, 2019.
21. Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification, 207-235, 2016, Springer, Boston, MA.
22. Gomez, F. R., Rajapakse, A. D., Annakkage, U. D., & Fernando, I. T. Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements. IEEE Transactions on Power Systems, 26(3), 1474-1483, 2010.
23. Sahoo, K. S., Tripathy, B. K., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M., & Burgos, D. An evolutionary SVM model

for DDOS attack detection in software defined networks. IEEE Access, 8, 132502-132513, 2020.
24. Berrar, D. Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands, 403-412, 2018.
25. Islam, M. J., Wu, Q. J., Ahmadi, M., & Sid-Ahmed, M. A. Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers. International Conference on Convergence Information Technology (ICCIT 2007), 1541-1546, 2007, IEEE.
26. Subramanian, E. K., & Tamilselvan, L. A focus on future cloud: machine learning-based cloud security. Service Oriented Computing and Applications, 13(3), 237-249, 2019.
27. Anthony, M., & Bartlett, P. L. Neural network learning: Theoretical foundations, Cambridge University Press, 2009.
28. Breiman, L. Random forests, UC Berkeley TR567, 1999.
29. Friedman, J. H., & Hall, P. On bagging and nonlinear estimation. Journal of statistical planning and inference, 137(3), 669-683, 2007.
30. Kang, H., & Kim, H. Household appliance classification using lower odd-numbered harmonics and the bagging decision tree. IEEE Access, 8, 55937-55952, 2020.
31. Raspberry Pi (Trading) Ltd. [Accessed by 16 May 2020] https://datasheets.raspberrypi.org/pico/pico-datasheet.pdf.
32. Chang, V., Li, T., & Zeng, Z. Towards an improved Adaboost algorithmic method for computational financial analysis. Journal of Parallel and Distributed Computing, 134, 219-232, 2019.
33. Kotsiantis, S. B. Bagging and boosting variants for handling classifications problems: a survey. The Knowledge Engineering Review, 29(1), 78-100, 2014.
34. Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 169, 114520.
35. Saini, P. S., Behal, S., & Bhatia, S. (2020, March). Detection of DDoS attacks using machine learning algorithms. In 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 16-21). IEEE.
36. Marvi, M., Arfeen, A., & Uddin, R. (2021). A generalized machine learning-based model for the detection of DDoS attacks. International Journal of Network Management, 31(6), e2152.
37. Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., & Kocaoğlu, R. (2021). Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. Electronics, 10(11), 1227.