Dağıtılmış Hizmet Reddi Saldırılarını Algılamak için bir Metodoloji

Araştırma Makalesi/Research Article

Ömer ASLAN

Software Engineering Department, Bandırma Onyedi Eylül University, Balıkesir, Turkey
omer.aslan.bisoft@gmail.com

(Geliş/Received:29.09.2021; Kabul/Accepted:18.02.2022)
DOI: 10.17671/gazibtd.1002178

Özet**—**Dağıtılmış hizmet reddi (Distributed denial of service- DDoS) saldırıları, sistemin kullanılabilirliğini hedef alarak normal kullanıcıların sisteme erişimini engelleyen en yıkıcı siber saldırılardandır. DDoS saldırılarından sadece bilgisayarlar değil, aynı zamanda çok sayıda akıllı telefon ve Nesnelerin İnterneti (IoT) cihazları da etkilenmektedir. DDoS saldırılarını etkili bir şekilde durduran veya önleyen iyi bilinen bir sistem yoktur. Düşük hesaplama yükü ile yüksek doğrulukta etkili bir DDoS tespit sistemi tasarlamak hala çok zorlu bir iştir. Bu makalede, DDoS saldırı türlerini tespit etmek ve sınıflandırmak için kullanılan bir yöntem önerilmiştir. Metodolojimiz üç bölümden oluşmaktadır: veri ön işleme, özellik seçimi ve sınıflandırma. Öncelikle modelimize uygun olmayan bazı özellikleri elimine etmek için veri ön işleme yapılmıştır. İkinci olarak, en önemli özellikler Bilgi Kazanımı, Kazanç Oranı, Korelasyon Katsayısı ve Relief algoritmaları kullanılarak seçilmiştir. Öznitelik sayısı 87'den 20'ye düşürülmüştür. Son olarak, çeşitli makine öğrenmesi algoritmaları kullanılarak normal ağ trafiği DDoS saldırılarından ayrıştırılmıştır. Ayrıca, DDoS saldırı türlerine göre de sınıflandırma yapılmıştır. Önerilen yöntem, CIC-DDoS2019 veri seti üzerinde test edilmiştir. Deneysel sonuçlar, önerilen yöntemin literatürdeki öncü yöntemlere göre daha iyi performans gösterdiğini doğrulamıştır.

Anahtar Kelimeler**—**siber saldırılar, DDoS tespiti, DDoS saldırılarının sınıflandırılması, özellik seçimi, makine öğrenmesi

A Methodology to Detect Distributed Denial of Service Attacks

Abstract**—**Distributed denial of service (DDoS) attacks is one of the most destructive cyber attacks which target the availability of the system when legitimate users try to access the system. Not only computers, but also the growing number of smartphones as well as Internet of Things (IoT) devices are affected by DDoS attacks. There is no well-known system which effectively stops or prevents DDoS attacks. Designing an effective DDoS detector with high accuracy with low computational overhead is still a very challenging task. In this paper, a methodology, which is used to detect and classify the types of DDoS attacks, is proposed. Our methodology is divided into three parts: pre-processing, feature selection, and classification. First, pre-processing is performed to eliminate some features which are not suitable for our model. Second, most significant features are selected by using Information Gain, Gain Ratio, Correlation Coefficient, and Relief. We declined the number of features from 87 to 20. Finally, various classifiers are used to detect DDoS attacks from the bening ones. The proposed methodology is performed on the CIC-DDoS2019 dataset. The experimental results show that the proposed methodology performed pretty well when it is compared to leading methods in the literature.

Keywords**—** cyber attacks**,** DDoS attacks, DDoS detection, DDoS attacks classification, feature selection

1. INTRODUCTION

Recently, the number, severity of attacks, and sophistication of cyber attacks are increasing [1] rapidly. It is impossible to detect and prevent well prepared cyber attacks. One of the most destructive cyber attacks is

DDoS. It is challenging to stop DDoS attacks [2, 3] because of its nature. DDoS attacks disrupt the normal internet traffic by sending several requests to the victim machine in a short period of time. Due to excessive requests, the victim machine is unable to accept the legitimate users' requests. In DDoS attacks, attack agents are distributed along the globe from different sources which makes the detection and prevention processes more difficult.

There are different forms of DDoS attacks including Exploitation-based and Reflection-based attacks [2], which can be classified further such as TCP-Based Attacks, TCP/UDP-Based Attacks, and UDP-Based Attacks. In each class, the type of protocol used and the purpose of the attacks are different. Email Flooding, SYN Flooding, Ping of Death, and Reflection attack are well-known forms of DDoS attacks.

In this paper, we suggested a methodology to separate the DDoS attacks from the normal network traffic. We further classified the attacks into the different groups based on the different forms of DDoS attacks. We first performed a pre-processing stage to eliminate the inappropriate features for our model. Then, we applied a feature selection stage to eliminate redundant, less important, and irrelevant features from the dataset. After that, we performed the classification stages to distinguish DDoS as well as types of DDoS attacks.

The proposed methodology greatly reduced the number of features. It was reduced from 87 features to 20. Our model is pretty fast when detecting attacks and classifying types of attacks. We obtained 99.9% accuracy when detecting DDoS attacks from the normal network traffic, which is quite high when compared to state-of-the-art studies in the literature.

The remainder of the paper is structured as the following. In section 2, types of DDoS attacks, feature selection process, and literature review on DDoS attack detection methods are summarized. In section 3, materials and methods are explained. In section 4, experiment results and evaluation are summarized. In section 5, a conclusion is presented.

2. LITERATURE REVIEW

This section is divided into three parts. In the first part, types of DDoS attack have been discussed. In the second part, the need for feature selection, feature selection approaches and techniques are explained. In the last part, the-state-of-the-art methods, which are detecting DDoS attacks, are reviewed based on the main idea and used methods.

2.1. Types of DoS/DDoS Attack

In denial of service (DoS) attacks, attackers send several requests to a targeted machine in a short period of time. After a certain time, the targeted machine cannot respond to normal users requests because of overwhelming network traffic. In denial of service, attackers aim to prevent normal users from accessing the system by threatening the availability of access to information. If an attack is distributed along the globe, we call the attack DDoS (Distributed denial of service). During the DDoS attack, several systems around the world can attack the victim system. General view of DDoS attack can be seen in Figure 1. It is almost impossible to prevent DoS and DDoS attacks. To effectively detect attacks and filter the packets, load balancers may be used to decrease or stop the DDoS attacks.

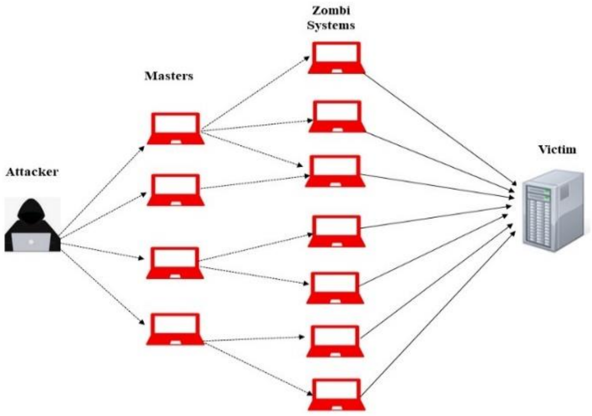

Figure 1. General view of DDoS attacks

DoS and DDoS attacks can be classified in various forms based on the protocols that are used as well as the distribution of the attack. We can categorize the DDoS attacks into three main classes, namely: TCP-Based Attacks, TCP/UDP-Based Attacks, and UDP-Based Attacks (Figure 2). In each class, the type of protocol that has been used and the purpose of the attacks are different. Email Flooding, SYN Flooding, Ping of Death, and Reflection attack are well-known forms of DDoS attacks. In SYN Flooding DDoS attack, the attacker sends several packets with a fake (bogus) source address, the target responds with SYN/ACK, but the response goes nowhere. After a certain period of time, the target cannot handle more requests and become unavailable. In Ping of Death on the other hand, an attacker attempts to crash the victim system by sending oversized packets using a ping command. The oversized packets cause Buffer overflow which crash the system.

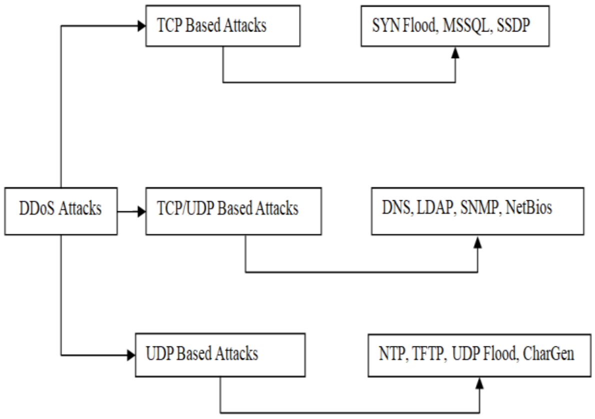

Figure 2. Types of DDoS attacks

2.2. Feature Creation and Selection Processes

Features are generated from raw data manually or automatically. Even though it can be said that manual feature extraction can generate more meaningful features, it is very slow and requires man powers. On the other hand, automatic feature creation techniques are more effective when dealing with large volumes of data. After features are generated, the feature selection process takes place. The feature selection process is the finding the subset of the features which represents the dataset more effectively [4]. Feature selection is very crucial because generally datasets have redundant, irrelevant, and unrelated features. Eliminating unrelated and redundant features improve model performances, decrease overfitting, and reduce false classification. There are different approaches to select features, namely: Filter, Wrapper, and Embedded. In the Filter approach, mainly heuristic search is used to choose the most important features by looking at the general characteristics of the data. Selection is performed one time before the classification stage. The Wrapper approach on the other hand combines the feature selection and classification phases. Subsets of features are generated and each time classification takes place, those features are used sequentially. In Embedded approach, the feature selection approach and the classifiers work together, yet the features are selected in the learning process. Based on the features, distribution of these approaches can be utilized during the feature selection.

There are various techniques that can be used during the feature selection process. Some of them are listed as the following: Information Gain, Gain Ratio, Chi-square Test, Correlation Coefficient, Fisher's Score, and Relief.

2.2.1. Information Gain

Information gain comes from Shannon theory, uses entropy and selects the features with the highest information gain which is minimizing the information required for the classifier [5].

2.2.2. Gain Ratio

To prevent biased selection of the information gain technique, the measurement of the gain ratio has been proposed. The property with the maximum gain ratio is recursively selected, which measures the information according to the classification obtained with the same partitioning [5]. The gain ratio is not resistant to unstable partitions and can therefore create unstable trees.

2.2.3. Chi-square Test

It is a technique that measures how a model compares to actual observed data [6]. Chi-square technique is used for categorical properties in the dataset. It is calculated between each property and the target class which selects the desired number of properties with the optimal Chi-square scores. Chi-square is calculated as:

$$\chi 2 = \sum_i \left(\frac{(O_{i-}E_i)}{E_i}\right)^2 \qquad (1)$$

$\chi 2$ is a Chi-Square value, O_i is an observed frequency, and E_i is an expected frequency, respectively.

2.2.4. Correlation and dependence measures

Correlation is the measurement of the connection among the variables. By using the correlation the more suitable variables which are greatly correlated with the target can be selected [6]. There are various correlation distance (CD) algorithms including pearson CD, spearman CD, kendall CD, and eisen cosine CD. The calculation of each correlation distance can be seen as follows:

Spearman correlation distance =

$$1 - \frac{6\sum d_i^{\mathbf{2}}}{\mathrm{n}\ (n^2-1)} \qquad (2)$$

Kendall correlation distance =

$$1 - \frac{n_c - n_d}{\left(\frac{1}{2}\right) n\ (n-1)} \qquad (3)$$

Eisen cosine correlation distance =

$$1 - \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |x_i - y_i|}{\sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (x_i)^{\mathbf{2}}\ \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (y_i)^{\mathbf{2}}}} \qquad (4)$$

In the formula, x and y represent different variables, n shows the total number of samples, d_i represents difference between paired variable ranks, n_c shows the number of compatible variables, and n_d represents the number of incompatible variables.

2.2.5. Fisher Score

Fisher's Score finds the ranks of the variables based upon the calculated fisher's score sort descending order [7]. Afterwards, features are selected based on the rank score.

$$S_i = \sum n_j\ (M_{ij} - M_i)^2 / \sum\ n_j * (P_{ij})^2 \qquad (5)$$

where M_{ij} mean and P_{ij} variance for related feature, *n* shows the total number of the samples in the dataset, n_j is the *j* class and M_i is the mean of the *i* feature.

2.2.6. Relief

It computes a feature score for every property and selects the top score features from the dataset [8]. When the feature scores are calculated, the feature value disparities between closest neighbor instance couples are taken into account. If a property value disparity is seen in the neighboring instance couple with the same class ('near-hit'), the feature score declines; if with the different class values ('near-miss), the feature score increases.

2.3 State-of-the-Art Studies on DDoS Detection and Prevention

There are various methods which are mentioned in the literature that detect network intrusions. However, there are not many studies that specifically detect DoS and DDoS attacks. Some of the recent studies which separate DDoS attacks from the normal network traffic are given at the below. The literature studies are reviewed based on proposed method, main idea, applied datasets, and obtained performances.

Zhang *et al.* reviewed the artificial intelligence methods to recognize DDoS attacks [9]. To identify the attacks, machine learning algorithms such as ANN (Artificial Neural Network), RF (Random Forest), NB (Naive Bayes), and SVM (Support Vector Machine) were examined. As stated in the paper that mostly ML algorithms including ANN with hadoop, NB, and SVM had been used for DDoS detection. Recently other algorithms like deep learning have become the trend when detecting DDoS attacks as well.

Doshi *et al.* proposed IoT specific network behaviors for consumer IoT devices when detecting DDoS attacks [10]. They divided features into two classes: stateless and stateful. Stateless features derived from the flow independent packets. The stateless features are lightweight. Packet size, inter-packet interval and protocol are examples of stateless features. Stateful features on the other hand, can capture changes in the network traffic. Stateful features are generated by dividing network traffic into streams and evolving network behaviors are generated. Bandwidth and IP destination address cardinality, which measure the regular time intervals between packets and limited number of end points, are examples of stateful features. After feature generation and selection process are completed, the classification is performed. As stated in the paper that RF with Gini impurity score, Neural Network (4-layer fully-connected feedforward), K-nearest neighbors “KDTree”, Decision tree using Gini impurity score, and SVM with linear kernel used for separating DDoS attacks from the normal traffic. According to the test result, the proposed method performed well on the collected consumer IoT devices.

A DDoS attack identification and mitigation framework is proposed in [11]. Initially, they explained a framework for SD-IoT (Software-defined Internet of Things) based upon the SDx (Software-defined anything) paradigm. The suggested framework comprises a controller pool, SD-IoT switches and IoT devices. To detect and mitigate the DDoS attacks on IoT devices, the cosine similarity of the vectors of the packet in message rate at boundary Software-defined Internet of Things switch ports is used. The proposed framework tested on two computers. One computer used the controller and the other one used to simulate the network topologies for IoT. The experimental results presented that the proposed framework could detect and mitigate the DDoS attacks efficiently.

A smart DDoS detection system for IoT devices proposed in [12]. The system is designed by using SDN (Software Defined Network) and tested on three datasets including CICIDS2017, CIC-DoS, and a customized dataset which contains DDoS attacks. The suggested system consists of three parts: OpenFlow Switch, SDN Controller, and Detection Module. Traffic generated from the internet of things devices is sampled by OpenFlow Switch. Collected data analyzed by using signature based machine learning algorithms which detect DDoS attack patterns. The ML algorithms are used including RF, LR (Logistic Regression), and XGB (Extreme Gradient Boost). As stated in the paper that the proposed approach could effectively separate DDoS attacks from the normal network traffic.

Li et al. proposed a real-time recognition system for DDoS attacks [13]. The presented detection scheme comprises of three section:

1. A sliding time window to speed up entropy calculation,
2. A directional filter to notice early recognition,
3. A quintile deviation control in order to optimize the recognition results.

Firstly, the network traffic was monitored by a traffic processor. Then, the network traffic was pre-processed and forwards into the entropy calculator. Secondly, the calculated entropy values were sent to the detection module. Finally, the detection module matched the entropy values against the given risk models to decide whether there were intrusions or not. The proposed scheme was tested on DARPA intrusion detection 1999 evaluation dataset, DARPA 2009 DDoS dataset, UNB-CIC-DDoS2019 dataset, as well as generated dataset. The experiment test results presented that the suggested scheme efficiently detects DDoS attacks.

Doriguzzi-Corin *et al.* presented a deep learning-based DDoS detection method [14]. The suggested method used CNNs (Convolutional Neural Networks) to separate attack traffic from the normal ones. The proposed CNN architecture consists of 4 layers: input, CNN, Max pooling, and classification. The proposed approach contributions can be listed as follows:

1. The proposed CNN-based detection method is Lightweight and can specify the attack's behaviors with low processing overhead,
2. The proposed method used pre-defined time windows for the feature preprocessing stage,
3. The proposed method utilized activation analysis to specify most significant features for DDoS detection,
4. The proposed method was proved to be using less computing resources.

The presented method was tested on three datasets including ISCX2012, CIC2017 and CSECIC2018. According to the authors, the proposed method *TPR* (true positive rate) and *FPR* (false positive rate) are measured as 99.5%, 1.79% for the ISCX2012 dataset, respectively which is quite satisfactory when compared with pioneer methods in the literature.

Asad *et al.* explained a deep learning-based method, which is called Deepdetect, to detect DDoS attacks [15]. The proposed method was detecting application layer DDoS attacks by using a neural network with feedforward backpropagation architecture. The proposed system contains input, hidden and output layers. The system used 7 hidden layers to recognize attack patterns in the network traffic. The suggested system was tested on CIDCIDS 2017 dataset. According to the paper, the suggested method obtained a 98% accuracy rate for different forms of DDoS attacks.

Wei *et al.* proposed a new deep learning method to separate DDoS attacks from the normal network traffic [16]. The proposed hybrid method is called AE-MLP (Autoencoder) (Multi-layer Perceptron). The AE part identifies the most significant features automatically. On the other hand, the MLP part takes the selected features as an input, and classifies the DDoS attacks based on the attack types. The suggested technique was tested on the CICDDoS2019 dataset. According to the results, the suggested method precision, recall, and accuracy are measured as 97.91%, 98.48%, and 98.34%, respectively.

Recently, classical machine learning (ML) as well as deep learning (DL) techniques have become popular to detect attacks in computer networks. Some of these studies are reviewed based upon the main point, used techniques, and measured performances. DL techniques can be evaluated superior to ML techniques due to auto feature extraction, can be applicable for large datasets, and decrease feature space rigorously. However, DL techniques has some disadvantages which needs to be mentioned:

1. DL cannot specify the meaningful features all the time,
2. Missing domain experts knowledge,
3. No well educated data scientists to control the implementation,
4. Using several hidden layers take a lot of time, and increasing the number of hidden layers not always enhances the performance,
5. Crafted inputs (evasion attacks) can easily deceive the deep learning techniques.

Besides, deep learning by itself is not enough to solve most security problems. Because of these deficiencies, we used ML-based techniques to identify DDoS attacks. Our proposed method performed necessary contributions in each stage to improve the detection and classification accuracy.

3. MATERIALS AND METHODS

This section explains materials and proposed methodology. The artirecture of the suggested methodology can be seen in Figure 3. This section is divided into four parts: data collection, feature selection, classification, and performance evaluation. We download DDoS datasets online and perform preprocessing stages to prepare the data for our model. After the preprocessing stage is completed, we perform the feature selection process. In this stage, we select the most significant subset of features to increase the detection rate (*DR*) and accuracy while decreasing the false positive rate (*FPR*) and false negative rate (*FNR*). We use Information Gain, Gain Ratio, Correlation Coefficient, and Relief algorithms to select the most significant features.

When the feature selection stage is completed, we perform a classification process to distinguish DDoS from the normal network traffic. In the classification and learning process, well-known ML classifiers such as C4.5 (J48), RF, DS (Decision Stump), KNN (K-Nearest Neighbors), AdaBoost (Adaptive Boosting), and BN are used. The proposed methodology was implemented by using Python scripting language on Windows 10 Pro. The tested system has 8GB RAM with Intel (R) Core(TM) i5-3470 CPU 3.2GHZ. For the preprocessing stage, the proposed algorithms are performed on Python. For feature selection stages, and classification phases Python scripting language as well as Weka tool is used.

3.1. Data Collection

DDoS evaluation dataset (CIC-DDoS2019) [2] has been used for training and testing purposes. Six files out of eleven in the CIC-DDoS2019 are used. The name of the files are DrDos_NTP.csv, DrDoS_NetBIOS.csv DrDoS_LDAP.csv,DrDos_DNS.csv,DrDoS_MSSQL.csv, DrDoS_SSDP.csv. There are 87 features which represent the network traffic data in those files. There are hundreds of thousands instances in each file, but we used 4039 instances to separate DDoS from the benign one. In addition, several instances are used to classify the types of DDoS attacks. Types of DDoS are classified based upon the protocol that is used, namely: DNS, NTP, MSSQL LDAP, NetBions, and SSDP.

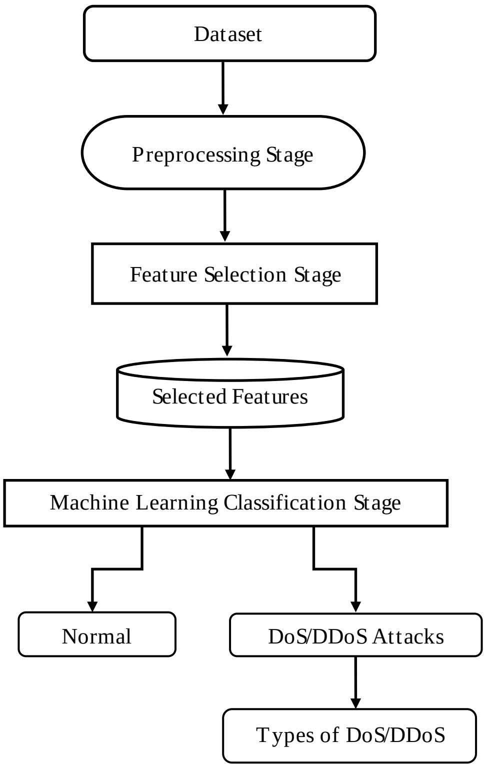

Figure 3. General view of proposed methodology

3.2. Pre-processing and Feature Selection Processes

In order to make data appropriate for our proposed methodology, we perform a pre-processing stage to convert or eliminate some features. In this stage we eliminate 5 features. After the preprocessing stage is completed, the feature selection process takes place. To select the best sub-set of features, we applied Information Gain, Gain Ratio, Correlation Coefficient, and Relief algorithms for six files of CIC-DDoS2019 dataset. We use a filter approach to select the features. In other words, first the sub-sets are selected, and then classification is performed.

3.2.1. Information Gain

We use Information gain to calculate the entropy and select the features with the highest information gain which is minimizing the information required for the classifier. We calculate the information gain as follows.

$$Gain\ (A) = Information\ (D) - Information_A(D) \qquad (6)$$

$$Information(D) = -\sum_{i=1}^{m} P_i\ log_2 P_i \qquad (7)$$

$$Information_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Information(D_j) \qquad (8)$$

Information(D) shows the average amount of information needed to identify the class labels on the dataset, *information_A(D)* shows the amount of information needed after each partitioning during classification for features in the dataset, and *Gain(A)* represents how much information can be gain when split by using feature *A*, respectively. This shows that if the feature with the highest information gain is chosen as the separation property, appropriate classification will be made with less information in the next selections. The information gain criterion does not work well in datasets with several class labels as it tends to select features with high values.

3.2.2. Gain Ratio

Gain Ratio selects the features with the maximum gain ratio, which measures the information according to the classification obtained with the same partitioning. We used the following formula to measure the Gain Ratio.

$$Gain\ Ratio\ (A) = Gain\ (A)/SplitInformation_A(D) \qquad (9)$$

$$Gain\ (A) = Information(D) - Information_A(D) \qquad (10)$$

$$SplitInformation_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} log_2 \left(\frac{|D_j|}{|D|}\right) \qquad (11)$$

$Gain(A)$ represents the information can be gain when branching use the feature A, and $SplitInformation_A(D)$ presents the intrinsic information that evaluates the entropy of the subdataset.

3.2.3. Correlation Coefficient

It measures the relationship among the variables. By using the correlation the more significant variables, which are highly connected with the target, are selected. We used Pearson correlation coefficient when selecting the features. We used the following formulas to calculate it:

$$\text{Pearson correlation} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \qquad (12)$$

Pearson correlation distance =

$$1 - \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \qquad (13)$$

In the formula, x and y are the different variables, $\bar{x}$ ve $\bar{y}$ are the mean of the variables, n is the total number of samples.

3.2.4. Relief

We compute a feature score for each property and select the top score features from the dataset. We utilized the feature value differences between closest neighbor instance couples when selecting most significant features. We updated the weight of the features as:

$$W_i = W_i - (x_i - nearHit_i)^2 + (x_i - nearMiss_i)^2 \qquad (14)$$

where W and x_i show weight vector and feature vector, respectively, while $nearHit_i$ and $nearMiss_i$ represent the nearest the same-class instance and nearest different-class instance, respectively.

By applying the feature selection algorithms, we decreased the number of features from 83 to 20. Until 15 features, the detection performances have not declined, but after 15 features the performances of the machine learning has declined. This shows that there is a threshold value for each dataset, which cannot decrease the number of the features further.

3.3 Classification and Learning Processes

For classification, well-known ML classifiers including J48, RF, DS, KNN, AdaBoost, and BN are used. Selected features are given to those classifiers for training and testing. The explanation of each classifier given as follows:

3.3.1. C4.5

The C4.5 algorithm is an improvement of the ID3 algorithm and was proposed by Quinlan in 1993. C4.5, which is a statistical classifier, works according to the depth priority technique and works by constantly arranging the data at each node to reach the best branching criterion [17]. It uses the Gain Ratio method to evaluate the partition attribute and operates on both continuous and categorical data. By using advanced tree pruning methods, it both makes the tree smaller and reduces the misclassification errors by reducing the noise in the data. The advantages of this algorithm are that the tree created is easy to implement, easy to understand, operate with both categorical and continuous data, eliminate noisy and forgotten data, and use pruning effectively. This classifier works well on our proposed methodology.

3.3.2. Random Forest (RF)

RF, which is a combination of tree estimators, consists of many trees and makes classification using features that every tree is sampled independently [18]. This classifier can be used for both regression and classification purposes. When solving classification problems, the output of the algorithm appears as a class membership that associates a set of independent predictive values with the matching category present in the dependent variable [18]. This logic-based classifier produces results with high accuracy, detects outliers and anomalies in data. It produces satisfactory results in datasets with low variance and many related features. It can also give an estimation of the significant features for the classification stage.

3.3.3. Decision Stump (DS)

DS is a machine learning algorithm which is comprise of a one internal node decision tree [19]. It makes a prediction with single input feature which also can be also called 1-rules tree. The DS tree can classify unknown samples efficiently.

3.3.4. K-Nearest Neighbors (KNN)

It uses sample-based learning [5] which can be used for classification as well as regression. It generates satisfying results in the absence of prior knowledge about the data distribution [20]. Generally, smaller k values return better results. We used $k=1$ for our training and test case.

3.3.5. Adaptive Boosting (AdaBoost)

Adaboost combines several classifiers to enhance the performance of ML algorithms [21]. The goal in Adaboost is to set the weights of learners and make accurate predictions of unusual observations for each iteration and minimize the training error.

3.3.6. Bayesian Network (BN)

BN a statistical model classifier, generally returns fast and effective results, but it is not practical to implement for datasets with many features [22]. Since we apply BN after the feature selection stage, the performance of the BN on the datasets will be satisfactory.

3.4 Performance Evaluation

To measure the performance, we calculate the *DR*, *FPR*, precision, accuracy and *f*-measure by using the confusion matrix (Table 1). During the training and testing, 10-fold cross-validation and holdout (70% and 30% split) procedures are applied. Initially, when fewer samples are used, the cross-validation generated more satisfactory results than holdout. However, when more samples are used, the performance of the holdout method increases as well. The six classifiers' performance improved after the feature selection process. We used 20 features out of 87 for training and testing. Until 15 features, the performance is increased, but performing classification with less then 15 features decreases the detection performances. This is because we believe that there is a threshold value for each dataset which shows the minimum number of features for classification. If the number of features is less than the specified threshold, the performance is affected in a negative way.

Table 1. Confusion Matrix

Predicted Class			
		Yes	No
Actual Class	Yes	True Positive	False Negative
	No	False Positive	True Negative

$$Detection\ Rate = Recall = TP/\ (TP+FN) \tag{15}$$

$$Precision = TP/\ (TP + FP) \tag{16}$$

$$False\ Positive\ Rate = FP/\ (FP+TN) \tag{17}$$

$$F\text{-}Measure = (2 * Precision * Recall)/\ (Precision + Recall) \tag{18}$$

$$Accuracy = TP+TN/\ (TP+TN+FP+FN) \tag{19}$$

4. RESULTS AND DISCUSSION

This part explains the experimental test results and evaluates the suggested methodology performances. The performances of the classifiers results before the feature selection process takes place have not been given in the paper because the performances were lower. The performances of various classifiers, which separated DDoS attacks from the benign ones, can be seen in table

2, table 3, table 4, and table 5 when different feature selection techniques (Information Gain, Gain Ratio, Correlation Coefficient, and Relief) are used. For instance, when Information Gain is used as a feature selection and J48 algorithm is used for classification, performance measures as 99%, 0.2% and 99.6% for *DR*, *FPR* and accuracy, respectively. When Information Gain is used as a feature selection and RF algorithm is used for classification, performance measures as 99.4%, 0.1% and 99.8% for *DR*, *FPR*, and accuracy, respectively. Similar performance improvement can be seen for other classifiers as well.

Table 2. Proposed methodology performances when Information Gain measure used for feature reduction on selected ML algorithms

Classifiers	DR (%)	FPR (%)	F-Measure (%)	Accuracy (%)
J48	99	0.2	98.8	99.6
RF	99.4	0.1	99.4	99.8
DS	94.9	13.5	71	87.8
KNN	96.9	0.3	97.7	99.2
AdaBoost	92	0.2	95.3	98.5
BN	96	0.1	97.6	99.2

Performance improvement can be seen in table 3, table 4, and table 5 when Gain Ratio, Correlation Coefficient, and Relief feature selection method are used. However, some feature selection measures are better than others. To illustrate, J48, RF, DS, and AdaBoost classifiers performed well no matter which feature selection method is used. On the other hand, KNN and BN classifiers' performances are affected based on the feature selection process. For example, when Information Gain is used for feature selection, the *DR* for KNN measured as 96.9%. However, when the Correlation Coefficient is used as a feature selection, the *DR* for KNN is measured as 99.8%. This presents that one feature selection method can be better than another for different classifiers.

Table 3. Proposed methodology performances when Gain Ratio measure used for feature reduction on selected ML algorithms

Classifiers	DR (%)	FPR (%)	F-Measure (%)	Accuracy (%)
J48	98.9	0.4	98.5	99.52
RF	100	0.1	99.8	99.9
DS	94.5	13.7	71.3	87.6
KNN	99.7	0	99.7	99.9
AdaBoost	93.7	0.4	95.7	98.6
BN	97.2	0.1	98.3	99.4

Table 4. Proposed methodology performances when Correlation Coefficient used for feature reduction on selected ML algorithms

Classifiers	DR (%)	FPR (%)	F-Measure (%)	Accuracy (%)
J48	98.2	0.2	98.5	99.5
RF	100	0.1	99.6	99.8
DS	94.5	13.7	71.3	87.6
KNN	99.8	0.1	99.5	99.8
AdaBoost	93.4	0.5	95.3	98.5
BN	95.3	4.8	86.7	95.2

The selected properties and the order of the feature sets can be seen in table 6. It can be said that each feature selection method chooses a different subset among the dataset in different order, but a few features can be the same for other feature selection methods. We could decrease the number of features from 87 to 15. Similar performances are obtained when 15 features are used, but 15 features are threshold values for this dataset. If we continue to decline the number of features, the performances are affected negatively. Thus, we stop the decrease in the number of features after 15.

Table 5. Proposed methodology performances when Relief algorithm used for feature reduction on selected ML algorithms

Classifiers	DR (%)	FPR (%)	F-Measure (%)	Accuracy (%)
J48	99.6	0.4	98.9	99.6
RF	99.6	0.1	99.5	99.8
DS	94.5	13.7	71.3	87.6
KNN	99.6	0.1	99.6	99.8
AdaBoost	94.2	0.6	95.4	98.5
BN	97.4	0.3	98	99.3

The further classification is performed after separating DDoS attacks from the benign ones. The results of classifying the DDoS attacks among themselves can be seen in table 7. The DDoS attacks are divided into six different classes including DrDoS_NTP, DrDoS _SSDP, DrDoS_MSSQL, DrDoS_LDAP, DrDoS_DNS, and DrDoS_NetBios. We could classify the types of attack with more than 90% *DR* except DrDoS_MSSQL attack. The system overall accuracy was measured as 94.41%. The classification *DR* can be improved further, if we perform the feature engineering process more effectively to specify the more related features for protocol types.

Table 6. Selected 20 features for each feature selection algorithm

Feature Selection Techniques	The Order of Selected Features
Information Gain	Init_Win_bytes_forward,Init_Win_bytes_backward,Fwd_Header_Length,Destination_Port,Bwd_Header_Length, Bwd_Packets, Source_Port, min_seg_size_forward, Flow_IAT_Max,Flow_IAT_Mean,Flow_Duration,Flow_Packets, Max_Packet_Length,Flow_IAT_Std,Fwd_Packets,Bwd_Packet_Length_Max,Total_Length_of_Bwd_Packets,Subflow_Bwd_Bytes,Packet_Length_Variance,Packet_Length_Std
Gain Ratio	min_seg_size_forward,Init_Win_bytes_forward,Init_Win_bytes_backward,Fwd_Header_Length,Fwd_Header_Length,Destination_Port,Bwd_Packet_Length_Max,Subflow_Bwd_Bytes,Total_Length_of_Bwd_Packets,Max_Packet_Length,Packet_Length_Variance, Packet_Length_Std, Inbound, Bwd_Packets, Flow_Bytes, SourcePort, Bwd_Header_Length,Bwd_Packet_Length_Min, Flow_IAT_Max, Min_Packet_Length,
Correlation Coefficient	Flow_Bytes, Flow_Packets, min_seg_size_forward, Inbound, Protocol, Source_Port, CWE_Flag_Count, Bwd_Packet_Length_Min, URG_Flag_Count, Unnamed, ACK_Flag_Count,Init_Win_bytes_forward,Fwd_PSH_Flags,Min_Packet_Length,Fwd_IAT_Total,Bwd_IAT_Total, FlowDuration, Fwd_Packets, Fwd_IAT_Min, Idle_Max
Relief	Flow_Bytes,Flow_Packets,Inbound,min_seg_size_forward, SourcePort, ACK_Flag_Count, Destination_Port, Protocol,Fwd_Packets,Init_Win_bytes_forward,CWE_Flag_Count,Unnamed,RST_Flag_Count,Fwd_PSH_Flags,Min_Packet_Length,Init_Win_bytes_backward,FlowDuration,Fwd_IAT_Total,Bwd_IAT_Total,Fwd_Packet_Length_Max

Table 7. Classification of Dos-DDoS Attacks

Protocol Types	Types of Attacks	DR (%)	Overall Accuracy
Any	BENIGN	98.7	94.41
UDP	DrDoS_NTP	90.3	
TCP	DrDoS_SSDP	90.6	
TCP	DrDoS_MSSQL	80.6	
UDP	DrDoS_LDAP	97.7	
TCP & UDP	DrDoS_DNS	98.9	
TCP & UDP	DrDoS_NetBios	99.7	

The comparison of suggested methodology against state-of-the-art methods is presented in table 8. In table 8, related studies are compared based upon the used method or feature representation, and obtained performances. As it can be seen that ML and DL-based DDoS detection methods are mostly used in the literature. Even though DL-based techniques are preferable and got popular recently, in some cases ML-based detectors generate better results. According to the comparison table 8, the proposed method performance measured highest by 99.9% accuracy while other methods' performance were lower. For instance, the study of Doshi *et al.*, Silveria *et al.*, Doriguizzi-corin *et al.*, Asad *et al.*, and *Wei et al.* performances measured as 99% accuracy, 96% *DR*, 99.5% *TPR*, 98% accuracy, 98.34% accuracy, respectively which is lower than our results. Besides, the complexity of proposed algorithms, and running time of the algorithms are higher in these some of the state-of-the-art methods. Furthermore, our methodology reduces the feature space drastically without reducing the model performance. In some cases malware is used to launch DDoS and Botnet attacks [23, 24]. Thus, in the future study we aim to analyze the malware families which launch the DDoS attacks as well. Besides, for more advanced security, sophisticated network infrastructure is needed [25] which reduces the network protocols vulnerabilities.

Table 8. Performance comparison of proposed method versus leading methods in the literature

Paper	Year	Used Method/Feature Representation	Success
Doshi *et al.* [10]	2018	IoT specific network behaviors from the flow independent packets	99% accuracy
Yin *et al.* [11]	2018	Cosine similarity to measure the similarity between the feature vectors	-
Silveira *et al.* [12]	2020	Machine learning techniques to detect attacks in the network traffic	96% *DR*
Doriguizzi-corin *et al.* [14]	2020	Lightweight convolutional neural networks to classify network traffic	99.5% *TPR*
Asad *et al.* [15]	2020	Neural network with feedforward backpropagation architecture to classify packets	98% accuracy
Wei *et al.* [16]	2021	Hybrid deep learning method to classify network traffic	98.34% accuracy
Proposed Method	**2022**	**ML techniques to detect and classify DDoS attacks in the network traffic**	**99.9% accuracy**

5. CONCLUSION

DDoS attacks target the availability of the system when legitimate users try to access the system. Because of the excessive attackers' request, legitimate users cannot access the system. When big companies' servers are unavailable for some time, these companies may lose a large amount of money as well as prestige. There is no well-recognized system or framework which effectively stops or prevents DDoS attacks. Thus, there is an urgent need to detect and stop DDoS attacks effectively.

In this study, a methodology is suggested to detect and classify the different forms of DDoS attacks. Our methodology is splitted into three sections: pre-processing, feature selection, and classification stages. Initially, the pre-processing stage is performed to eliminate some features which are not suitable for our classification model. Later, most relevant properties are chosen by using Information Gain, Gain Ratio, Correlation Coefficient, and Relief algorithms. We decreased the number of features from 87 to 20. Finally, different classifiers are used to separate DDoS attacks from the normal traffic. The proposed methodology is tested on the CIC-DDoS2019 dataset. The experimental results presented that when Gain Ratio is chosen for feature selection and RF is used as a classifier, the accuracy is measured best which is 99.9%. Similar but lower accuracy rates are obtained when different feature selection methods and classifiers are selected. We further classify the types of DDoS attacks including DrDoS_NTP, DrDoS _SSDP, DrDoS _MSSQL, DrDoS_LDAP, DrDoS_DNS, and DrDoS_NetBios. In the future, we aim to apply methodology on real network traffic. In addition, we would like to propose a new feature selection and classification method as well.

REFERENCES

[1] Ö. Aslan and R. Samet, "Mitigating cyber security attacks by being aware of vulnerabilities and bugs", **2017 International Conference on Cyberworlds (CW)**, IEEE, 2017.

[2] İnternet: DDoS Evaluation Dataset (CIC-DDoS2019), https://www.unb.ca/cic/datasets/ddos-2019.html, 15.09.2021.

[3] S.N. Shiaeles, V. Katos, A.S. Karakos and B.K. Papadopoulos, "Real time DDoS detection using fuzzy estimators", *computers & security 31*.6 (2012): 782-790, 2012.

[4] M. Ozkan-Okay, R. Samet and Ö. Aslan, "A new feature selection approach and classification technique for current intrusion detection system", **IEEE 6th International Conference On Computer Science and Engineering (UBMK),** 2021.

[5] J. Han, P. Jian, and K. Micheline, "Data mining: concepts and techniques", *Elsevier,* 2011.

[6] İnternet: A. Gupta, "Feature Selection Techniques in Machine Learning", https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/, 1.1.2022.

[7] D. Aksu, S. Üstebay, M.A. Aydin and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm", **International symposium on computer and information sciences**, Springer, Cham, 2018.

[8] T.H. Phyu and N.N Oo, "Performance comparison of feature selection methods", *MATEC web of conferences*, EDP Sciences, 42, 2016.

[9] B. Zhang, T. Zhang and Z. Yu, "DDoS detection and prevention based on artificial intelligence techniques", **3rd IEEE International Conference on Computer and Communications (ICCC)**, 2017.

[10] R. Doshi, N. Apthorpe and N. Feamster, "Machine learning ddos detection for consumer internet of things devices", IEEE Security and Privacy Workshops (SPW), 2018.

[11] D. Yin, L. Zhang and K. Yang, "A DDoS attack detection and mitigation with software-defined Internet of Things framework", *IEEE Access* 6 (2018): 24694-24705.

[12] F. A. F. Silveira, F. Lima-Filho, F. S. D. Silva, A. D. M. B. Junior and L. F. Silveira, "Smart detection-IoT: A DDoS sensor system for Internet of Things", **International Conference on Systems, Signals and Image Processing (IWSSIP)**, IEEE, 2020.

[13] J. Li, M. Liu, Z. Xue, X. Fan and X. He, "Rtvd: A real-time volumetric detection scheme for ddos in the internet of things," *IEEE Access* 8 (2020): 36191-36201.

[14] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection", *IEEE Transactions on Network and Service Management*, 17(2), 876-889, 2020.

[15] M. Asad, M. Asim, T. Javed, M.O. Beg, H. Mujtaba and S. Abbas, "Deepdetect: detection of distributed denial of service attacks using deep learning", *The Computer Journal*, 63(7), 983-994, 2020.

[16] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu and S. Camtepe, "Ae-mlp: A hybrid deep learning approach for ddos detection and classification", *IEEE Access*, 9, 146810-146821, 2021.

[17] B. Gupta, A. Rawat, A. Jain, A. Arora and N. Dhami, "Analysis of various decision tree algorithms for classification in data mining", *Int. J. Comput. Appl*, 163(8); 15-19, 2017.

[18] L. Breiman, "Random forests", *Machine learning* 45(1); 5-32, 2001.

[19] S.K. Sankaralingam, N.S Nagarajan and A.S. Narmadha, "Energy aware decision stump linear programming boosting node classification based data aggregation in WSN", *Computer Communications,* 155, 133-142, 2020.

[20] O. Kaynar, H. Arslan, Y. Görmez and Y.E. IŞIK, "Makine öğrenmesi ve öznitelik seçim yöntemleriyle saldırı tespiti", *Bilişim Teknolojileri Dergisi*, 11(2), 175-185, 2018.

[21] A. H. Wahla, L. Chen, Y. Wang, R. Chen and F. Wu, "Automatic wireless signal classification in multimedia Internet of Things: An adaptive boosting enabled approach", *IEEE Access*, 7,160334-160344, 2019.

[22] Ö. Aslan and R. Samet and Ö. Ö. Tanrıöver, "Using a Subtractive Center Behavioral Model to Detect Malware", *Security and Communication Networks*, 2020.

[23] E. Masum and R. Samet, "Mobil BOTNET İle DDOS Saldırısı", *Bilişim Teknolojileri Dergisi*, 11(2), 111-121, 2018.

[24] Ö. Aslan and S. Refik, "Investigation of possibilities to detect malware using existing tools", **2017 *IEEE/ACS 14th International Conference on Computer Systems and Applications* (AICCSA)**, 2017.

[25] R. Chaganti, D. Gupta and N. Vemprala, "Intelligent network layer for cyber-physical systems security", *International Journal of Smart Security Technologies (IJSST)*, 8(2), 42-58, 2021.