Journal of Engineering Science and Technology
Vol. 18, No. 2 (2023) 844 - 861

DETECTION OF DIFFERENT TYPES OF DISTRIBUTED DENIAL OF SERVICE ATTACKS USING MULTIPLE FEATURES OF ENTROPY AND SEQUENTIAL PROBABILITIES RATIO TEST

BASHEER HUSHAM ALI[1,2,*], NASRI SULAIMAN[1], S. A. R. AL-HADDAD[3], RODZIAH ATAN[4], SITI LAILATUL MOHD HASSAN[5]

[1]Dept. of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Malaysia
[2]Dept. of Computer Engineering, Al-Iraqia University, 10054 Baghdad, Iraq
[3]Dept. of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Malaysia
[4]Dept. of Software Engineering and Information Systems, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Malaysia
[5]School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
*Corresponding Author: gs58547@student.upm.edu.my, basheer.husham@aliraqia.edu.iq

Abstract

Distributed Denial of Service (DDoS) is the most dangerous attacks that targeted public servers. It is difficult for victims to detect these kinds of attacks because DDoS attacks can be done remotely and reflected by legal users in the network toward specific victim. The goal of this research is to locate compromised interface and identify different types of DDoS attacks, especially up-to-date kinds of them. Multiple features of Entropy and Sequential Probabilities Ratio Test approach (E-SPRT) was proposed and implemented in order to detect different types of DDoS attacks. CICFlowMeter was used to produce bidirectional network flows and extract 82 of different features from each flow. Multiple features of E-SPRT divide incoming flows into fixed groups that have same number of flows called window size. CICDDoS2019 dataset was chosen in this research because it contains various kinds of recent attacks. The performance of all features of E-SPRT were tested by confusion matrix and compared with other higher-accuracy techniques. Finally, the implemented model with different features detects most up to date DDoS attacks and achieves an accuracy and detection rate almost over 99%.

Keywords: CICDDoS2019, CICFlowMeter, Confusion matrix, DDoS attack, Entropy, Sequential probability ratio test.

1. Introduction

In the recent decades, malicious activities by using distributed denial of service (DDoS) attacks have been increasing [1]. Working online at home because of Covid pandemic during the end of 2019 and the beginning of 2020 leads to increase the rate and volume of DDoS activity attacks in 2020. For instance, protocol-based attacks such as UDP floods and amplification-based attacks have been increased 570 percent in second half of 2020 comparing with same period of 2019. These kinds of attacks are easy to be lunched [2]. They are also difficult to be mitigated by traditional threshold-based mitigation. Machine learning based techniques are good to identify these kinds of malicious activities when the behavior of attacks patterns is similar to the training dataset. However, these techniques fail to identify them when new attacks are coming up toward targeted system [3]. Even though DDoS attacks were available for years, and researchers proposed multiple solutions to face these types of attacks, DDoS attacks are still an important challenge that need to be addressed [4].

DDoS attacks first occurred in 1999 when attackers brought down Minnesota University website for two days. Attackers used tools called Trinoo in order to perform DDoS attacks. After this incident, different famous websites were also crashed due to DDoS attacks such as Yahoo, CNN, eBay, and Amazon. Thirty percent of DDoS traffics were represented as gaming traffics. Finally, the DDoS attacks increased by almost twenty five percent in 2015 [5].

Average of DDoS attacks have growing up recently. For example, the number of packets per second that were used by attackers to conduct DDoS attacks were almost 100 Kpps in 2010 comparing with 652 Mpps in 2020. Another instance, the number of requests per second that were send by intruders were 600 krps in 2010 while it reaches to 6 Mrps in 2020 as shown in [6].

DDoS attacks are type of cyber-attack that targeted services of victim in order to bring it down and prevent legitimate users from getting available services. Hackers in DDoS attacks can make it difficult by recruiting a very large number of infected machines in the network and directed them to target specific server. These compromised devices are called zombies or bots and the main attacker is called botmaster. The botmaster is responsible to run a malicious code via command-and-control server on infected devices in order to direct them to target the victim [7-9].

DDoS detection can be categorized into two main parts which are application-based detection and network-based detection. Application-based detection algorithms focused on identifying attacks by identifying infected flows by user interface layer. However, network-based detection methods focused on monitoring flows in network layer. Monitoring-based detection algorithms can be classified into two groups: signature-based detection and anomalies-based detection. Signature-based detection are these methods that search on previously identified infected flows. However, anomalies are these methods that identified possible infected flows by monitoring abnormal actions in flows. Statistical, machine learning, deep learning, and data mining are all solutions to anomalies algorithms. This research focus on statistical solutions [7].

The main contributions of this paper are summarized as follows:

- Multiple features of Entropy and Sequential probabilities ratio test approach (E-SPRT) that was extracted from flows of CICFlowMeter was proposed and

implemented to detect different types of DDoS attacks and locate compromised interface.

- The performance of all features of E-SPRT were tested by confusion matrix and compared with other higher-accuracy techniques using most up-to-date DDoS attacks such as Network Time Protocol (NTP), Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), Microsoft SQL Server (MSSQL), Network Basic Input/Output System (NETBIOS), Simple Service Discovery Protocol (SSDP), and User Datagram Protocol (UDP).

The rest of the paper organized as the following: literature review discussed in section two and methodology depicted in section three. Results and evaluations are explained in detail in section four. Finally, conclusion mentioned in the last section.

2. Literature Review

Many studies have been done by using statistical based techniques to identify DDoS attacks. Dao et al. [10] proposed a method to identify malicious packets based on behavior of users in the software defined network (SDN). This method was based on statistical calculations such as counting number of incoming packets, determining the minimum number of packets for each session, and finding the average of sessions that frequent users established. Then, they did a comparison among these calculations to take a decision. They also built a small network to test and evaluate their proposed method. Finally, their approach is not effective when attacker send a huge number of packets toward controller.

Piedrahita et al. [11] introduced two solutions to mitigate DDoS attack in SDN. The first one is based on finding the congestion of traffics in computer network. Switches keeps monitoring the number of packets that sending to the controller. When bandwidth of certain switch increased, controller sends command to congested switch to limit bandwidth usage. The second solution is Flowsec which is based on calculation the consumption bandwidth. Switches in computer network sends the flow statistics to the controller, and controller compares its bandwidth usage with certain threshold. When the bandwidth is larger than the threshold, then there is an attack, and controller decreases flow rates. However, they used threshold in order to take decision which leads to increase the uncertainty of results. Durner et al. [12] introduced a solution to detect and prevent DDoS attacks. They extracted a fixed header field from malicious flows under attack and installed these data as rules in the memory of server. When incoming traffics match stored rules, attacks can be detected. However, this method may fail when attacker send new data that do not match with preinstalled rules in the server.

Many methods based on statistical approaches used entropy calculations to detect anomalies. The entropy-based methods are better than other techniques because they include simple calculations and accurate results. Koay et al. [13] introduced an approach to collect multiple features from header of traffics and calculate entropy for these features. They also designed a classifier based on these features and machine learning-based approaches to recognize low from high intensity of DDoS anomalies. They tested and evaluated their method, and they found that their method has higher precision and higher recall.

Moreover, Mousavi and St-Hilaire [14] introduced a method to detect DDoS attacks in SDN based on calculating entropy of destination of IP address. They

divided incoming entropy values into fixed window size based on counting the number of flows or identifying fixed duration of incoming flows and compare these values with certain threshold. If these values are larger than threshold, then attack is detected. However, threshold values are changed based on the environment of the test, and this leads to increase the uncertainty of decision. Nguyen et al. [15] proposed an approach to identify different types of DDoS attacks using entropy-based features. They also used hierarchical temporal memory (HTM) and K Nearest Neighbors (KNN) classifiers in order to mine changing in entropy values. This helps to detect new DDoS attacks. However, their technique works on offline and need to be tested on real time stream. Finally, Liu et al. [16] used fast Fourier transform (FFT) with entropy to detect DDoS attacks (FEDDM). Then, they used FFT with entropy to train neural network and identify DDoS anomalies. However, their approach generated high false positive and miss rate for UDP and UDP-lag data.

In addition, Dong et al. [17] proposed a method to classify flows, detect DDoS attacks, locate compromise switch interface. They first classify flows to either low or normal flows based on the number of packets for each flow. They also determine compromise switch interface and detect DDoS anomalies by using sequential probabilities ratio test (SPRT). However, their method tested on old attacks and need to be tested to detect new DDoS attacks.

Maranhão et al. [18] proposed an approach to detect anomalies against Cyber-Physical Systems (CPSs). They used Higher Order Singular Value Decomposition (HOSVD) to filter out mean values of instances that have same features from the dataset. Then, they fed output of filtration process to the machine learning technique to decide whether traffics are malicious or benign. However, authors did not verify their approach on online data stream. Polat et al. [19] presented an approach to detect DDoS anomalies in SDN network by using machine learning algorithms. They first extracted set of features from SDN under DDoS attacks and in normal situation. They also generated new dataset based on feature selection technique. The feature selection method was used to make the detection model simple and decrease training time. The generated dataset with and without feature selection used to test and train with multiple machine learning algorithm such as Artificial Neural Network (ANN), (KNN), Support Vector Machine (SVM), and Naive Bayes (NB). The authors found that accuracy with feature selection technique is better than without it.

Many other methods used machine learning and deep learning in their detection method. For example, Long and Jinsong [2] implemented a method that has two phases. The first phase used information entropy to fast identify infected traffics. The second phase is based on using machine learning techniques such as stacked sparse autoencoder (SSAE) and support vector machine (SVM). These techniques are used to prove that suspected traffics that were determined in the first phase were compromised. Finally, they verified that their methods had high detection using real time traffics.

Moreover, Liu et al. proposed a method to detect DDoS anomalies using a combination of deep learning technique and information entropy in the SDN environment. The deep learning technique is convolutional neural network (CNN). The first step used to detect suspicious infected traffics, and the second step used to distinguish infected and normal traffics. The detection accuracy for their method was 98.98% as they stated in [1].

Finally, Li [20] introduced a method based on combination of three deep learning techniques to detect DDoS attacks. These techniques are Pearson

Correlation Coefficient, Auto Encoder, and Dense Neural Networks (PCC-AE-DNN). They compared the performance of each method on different kinds of DDoS attacks with other techniques using confusion matrix and CICDDoS2019 dataset. However, their approach is not scalable and not able to detect all kinds of attacks.

3. Methodology

The flowchart of proposed method will be explained in the next subsection. Multi feature of SE-SPRT method has two methods. The first approach is entropy. It will be explained in subsection 3.2. Finally, SPRT approach will be discussed.

3.1. Flowchart

The flowchart of proposed method is illustrated in below Fig. 1.

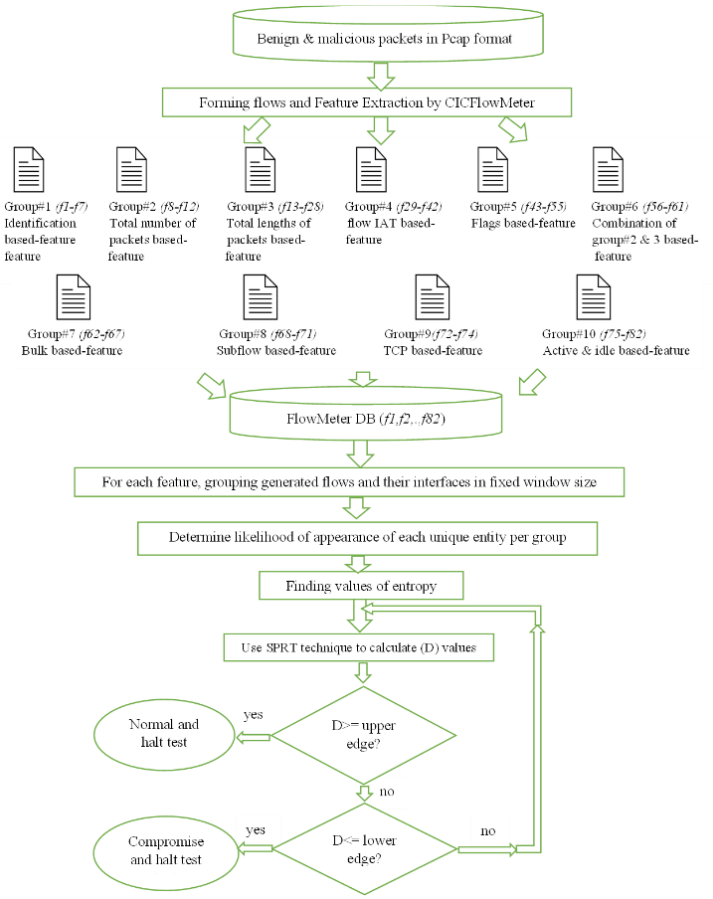


Fig. 1. Flowchart for the multiple feature E-SPRT.

Normal and malicious packets are captured in Pcap format. Each packet has two main parts header and payload. Payload contains data that is going to be transferred from one device to another However, header includes information about how to transfer data. Header of packet contains many fields such as destination/ source MAC address, destination/source IP address, destination/ source port address, flags,

protocol types, etc. Flow can be form by gathering multiple packets that have same characteristics such as protocol type, source IP address, destination IP address, source port address, and destination port address. CICFlowMeter can do that and generate bidirectional flows. The first packet in each flow is either in forward direction which means from source to destination or in backward direction which means from destination to source [21]. CICFlowMeter can also generate multiple features per flow as shown in Fig. 1.

These features were divided into 10 groups. These groups are identification-based feature, the total number of packets-based feature, total length of packet-based feature, flow *IAT* based feature, flags-based feature, combination of groups 2 and 3 based feature, bulk-based feature, subflow-based feature, Tcp-based feature, and active &idle-based feature. The full description of these groups and their feature are illustrated in Table 1.

Table 1. CICFlowMeter group features.

Group#	Group Name	Feature # and Name
Group#1	Identification based feature	Source IP (f1), Destination IP (f2), Source port (f3), Destination port (f4), Protocol (f5), Timestamp (f6), Flow duration (f7)
Group#2	Total number of packets-based feature	Total Fwd Packets (f8), Total Bwd Packets (f9), Flow Packets/s (f10), Fwd Packets/s (f11), Bwd Packets/s (f12)
Group#3	Total length of packets (the payload of app. protocol in bytes) based feature	Total Length of Fwd Packets (f13), Total Length of Bwd Packets (f14), Fwd Packet Length Max (f15), Fwd Packet Length Min (f16), Fwd Packet Length Mean (f17), Fwd Packet Length Std (f18), Bwd Packet Length Max (f19), Bwd Packet Length Min (f20), Bwd Packet Length Mean (f21), Bwd Packet Length Std (f22), Flow Bytes/s (f23), Min Packet Length (f24), Max Packet Length (f25), Packet Length Mean (f26), Packet Length Std (f27), Packet Length Variance (f28)
Group#4	Flow IAT based feature	Flow IAT Mean (f29), Flow IAT Std (f30), Flow IAT Max (f31), Flow IAT Min (f32), Fwd IAT Total (f33), Fwd IAT Mean (f34), Fwd IAT Std (f35), Fwd IAT Max (f36), Fwd IAT Min (f37), Bwd IAT Total (f38), Bwd IAT Mean (f39), Bwd IAT Std (f40), Bwd IAT Max (f41), Bwd IAT Min (f42)
Group#5	Flags features based feature	Fwd PSH flags (f43), Bwd PSH Flags (f44), Fwd URG Flags (f45), Bwd URG Flags (f46), FIN Flag Count (f47), SYN Flag Count (f48), RST Flag Count (f49), PSH Flag Count (f50), ACK Flag Count (f51), URG Flag Count (f52), CWR Flag Count (f53), ECE Flag Count (f54), Up/down ratio (f55)
Group#6	Feature based on combination of Group#2 and 3 (segment and header size) based feature	Average Packet Size (f56), Avg Fwd Segment Size (f57), Avg Bwd Segment Size (f58), Min_seg_size_forward (f59), Fwd Header Length (f60), Bwd Header Length (f61)
Group#7	Bulk based feature	Fwd Bytes/Bulk Avg (f62), Fwd Packet/Bulk Avg (f63), Fwd Bulk Rate Avg (f64), Bwd Bytes/Bulk Avg (f65), Bwd Packet/Bulk Avg (f66), Bwd Bulk Rate Avg (f67)
Group#8	Subflow based feature	Subflow Fwd Packets (f68), Subflow Fwd Bytes (f69), Subflow Bwd Packets (f70), Subflow Bwd Bytes (f71),
Group#9	TCP based features	Fwd Init Win bytes (f72), Bwd Init Win bytes (f73), Fwd Act Data Pkts (f74)
Group#10	Active and Idle based features	Active Min (f75), Active Mean (f76), Active Max (f77), Active Std (f78), Idle Min (f79), Idle Mean (f80), Idle Max (f81), Idle Std (f82)

Finally, for each feature, generated flows and their interfaces are gathered in specified window size. Window size can be form based on counting the number of flows or identifying fixed duration of incoming flows. For each collection of gathered flows, entropy value will be calculated. The output of entropy step will be fed to SPRT detection. Finally, SPRT can then decides whether flows and their interfaces are compromised or not.

3.2. Entropy method

Entropy is used because it can measure randomness of flows. When flows are random, entropy will be high. In the same way, entropy is low when flows are non-random. For each extracted feature (fn), flows and their feature values (x_i) are divided into subset of fixed size based on either the number of flows or time slot. The number of occurrences of feature values (y_i) are calculated for each window size (W) as shown in Eq. (1) [14]:

$$W = \{ (x_1, y_1), (x_2, y_2), (x_i, y_i) \} \quad (1)$$

Likelihood of occurrence of feature values (pi) is calculated by dividing the y_i over the number of flows (num) per window (W).

$$P(\text{i}) = Yi/num \quad (2)$$

Then, entropy (E) can be calculated by the following Eq. (2) [14]:

$$E = \sum_{i=0}^{n} p(i)\, ln\, p\, (i) \quad (3)$$

3.3. SPRT method

SPRT stands on Sequential Probability Ratio Test. It introduced by Wald in 1970, and it is based on mathematical calculation. It is a hypothesis test for multiple samples. These samples are entropies values. SPRT can choose one sample at each time and test the hypothesis. The hypothesis could be either normal (λ_0) or compromise (λ_1). The goal is to locate the compromise (λ_1) ethernet switch that was infected with harm flows and normal (λ_0) ethernet that let normal flows passing through [17].

The detection of SPRT(D_n^e) is going to observe a series of entropies values (T_0,T_1, ...,T_n). The detection is a probability of these values being as compromised divided by the normal values that inject certain ethernet switch (e). The detection equation is shown in Eq. (3) [17].

$$D_n^e = \ln \frac{\text{prob}\ (T_1^e, \ldots.,\ T_n^e\ |\lambda_1\ )}{\text{prob}\ (\ T_1^e,\ \ldots.\ ,\ T_n^e\ |\lambda_0)} \quad (4)$$

where D_n^e is the detection term of multiple observations (n) of certain ethernet switch (e) and T_n^e is series of entropies values observation (n) of certain ethernet (e). Assuming T_n^e as identically independent and distributed. Thus, D_n^e can be rewritten as shown in the following equation [17]:

$$D_n^e = \sum_{\text{n=1}}^{\text{m}} \ln \frac{\text{prob}\ (T_n^e|\lambda_1\ )}{\text{prob}\ (\ T_n^e\ |\lambda_0\ )} \quad (5)$$

Let us consider T_n^e as Bernoulli distribution values, then detection can be written as [17]:

$$\text{prob}\ (\ 0 \le T_n^e \le 0.5 | \lambda_0\ ) = 1 - \text{prob}\ (\ T_n^e > 0.5 | \lambda_0) = \mu_0 \tag{6}$$

$$\text{prob}\ (\ 0 \le T_n^e \le 0.5 | \lambda_1) = 1 - \text{prob}\ (\ T_n^e > 0.5 | \lambda_1) = \mu_1 \tag{7}$$

where value of μ_0 is smaller than value of μ_1 because normal ethernet is less likely to be infected with compromised flows. When T_n^e values are greater than 0.5, the ethernet is more likely to have normal flows. On the other hands, interface has higher probability to be injected with compromised flows when T_n^e close to 0. Thus, detection equation can be written as following [17]:

$$D_n^i = \begin{cases} D_{n-1}^i + \ln \frac{\text{prob}\ (T_n^e | \lambda_1\ )}{\text{prob}\ (\ T_n^e\ | \lambda_0)}, & 0 \le T_n^e \le 0.5 \\ D_{n-1}^i + \ln \frac{\text{prob}\ (T_n^e\ | \lambda_1\ )}{\text{prob}\ (T_n^e | \lambda_0)}, & T_n^e > 0.5 \end{cases} \tag{8}$$

By substituting Eqs. (5) and (6) in Eq. (7), the detection function can be written as shown in Eq. (8) (Where D_0^i=0) [17]:

$$D_n^i = \begin{cases} D_{n-1}^i + \ln \frac{\mu_1}{\mu_0}, & 0 \le T_n^e \le 0.5 \\ D_{n-1}^i + \ln \frac{1-\mu_1}{1-\mu_0}, & T_n^e > 0.5 \end{cases} \tag{9}$$

The detection method produces two kinds of errors which are false positive error (α) and false negative rate (β). False positive error generated when normal ethernet λ_0 is considered as infected ethernet by mistake. However, false negative error generated when compromised ethernet λ_1 is identified falsely as normal. To bypass these two errors, upper bound threshold (A) and lower bound threshold (B) are calculated as shown in the following [17]:

$$\begin{cases} A = \log_2 \frac{\beta}{(1-\alpha)} \\ B = \log_2 \frac{(1-\beta)}{\alpha} \end{cases} \tag{10}$$

For each observation, the detection D_n^i compares its result with upper threshold (A) and lower threshold (B). β set to be between 0.01 and 0.05 while α was 0.08 in order to bound upper and lower threshold. When D_n^i is equal or greater than A, interface and its flows are normal. However, when D_n^i is equal or less than B, interface and its flows are compromise. In both cases, test stop after one condition apply. Finally, test continue by taking another value when these two conditions fail.

4. Results and Discussion

To evaluate the performance of multi-feature E-SPRT detection method, CICDDoS2019 Dataset and confusion matrix were used.

4.1. CICDDoS2019

CICDDoS2019 is the most recent dataset that contains most up-to-date DDoS attacks such as MSSQL, SSDP, DNS, LDAP, NETBIOS, NTP, UDP, UDP-lag, SYN, and TFTP attacks. This dataset was developed by Canadian Institute for Cybersecurity that is located in Canada [21]. In order to mimic the network

properties and real packets, this dataset was designed with two networks which are victim and attack network. Victim network was protected with high level of security. It contains several networks equipment such as computer devices, routers, firewall, servers, and switches [21].

In the attack network, B-Profile method was used to generate these legitimate traffics and represent the behavior of human interactions [22]. However, malicious traffics was generated by third party packages and tools. The generated DDoS attacks were captured for two days. In the first day which is 11th of March, the attack time started from 9:43 till 17:35. In the second day which is 12th of January, the time of DDoS attacks started from 10:35 till 17:15. The number of DDoS attacks that were generated in second day was 10 different types of attacks. CICFlowMeter was used in order to generate flow and extract 82 of network features that were presented and explained in Table 1. The generated features were gathered based on attack type and stored in the csv file format. Each file contains number of infected and normal flows. Completed details of all types of attacks dataset that are available in CICDDoS2019 dataset such as attack times, number of infected flows, and number of normal flows were presented in Table 2.

Table 2. Details of attack types for CICDDoS2019 dataset.

Days	Attack database	Attack times	Infected Flow count in dataset	Normal flow count in dataset
First day (March 11th)	PortScan	9:43 - 9:51	186,960	4,734
	NetBIOS	10:00 - 10:09	3,454,578	1,321
	LDAP	10:21 - 10:30	2,108,110	5,124
	MSSQL	10:33 - 10:42	5,772,992	2,794
	UDP	10:53 - 11:03	3,779,072	3,134
	UDP-Lag	11:14 - 11:24	721,097	4,068
	SYN	11:28 - 17:35	4,284,751	35,790
Second day (January 12th)	NTP	10:35 - 10:45	1,202,642	14,365
	DNS	10:52 - 11:05	5,071,011	3,402
	LDAP	11:22 - 11:32	2,179,930	1,612
	MSSQL	11:36 - 11:45	4,522,492	2,006
	NetBIOS	11:50 - 12:00	4,093,279	1,707
	SSDP	12:27 - 12:37	2,610,611	763
	UDP	12:45 - 13:09	3,134,645	2,157
	UDP-Lag	13:11 - 13:15	366,900	3,705
	SYN	13:29 - 13:34	1,582,289	392
	TFTP	13:35 - 17:15	20,082,580	25,247

As shown above in the table, the number of infected is larger than number of benign samples. Thus, SMOTE (Synthetic Minority Over-Sampling Technique) was used to balance each dataset in order to make number of infected samples equal to number of benign samples. Some features were excluded in this study. For example, identification-based feature that contains source IP, destination IP, source port, destination port, flow ID, and timestamp. These features were excluded to prevent bias and make sure that malicious activities can be identified by attack behaviour. These features can also be changed easily by attacker. Other kinds of features were also excluded such as flag-based feature group. Bulk based feature group (f62 to f67) were also excluded. They removed because their behavior in legitimate and attack cases are the same. For example, their values are all zeros or

(-1) in normal and malicious situations. Therefore, the rest of these feature were kept in this experiment.

4.2. Confusion matrix

Datasets of second day were used to test and evaluate multi-features E-SPRT approach by using confusion matrix metrices. Figure 2(a) shows the sample of confusion matrix. It contains several popular metrics such as True positive (TP), false positive (FP), false negative (FN), true negative (TN), sensitivity or TPR, miss-rate or FNR, TNR or specificity, FPR or probability, positive predictive value (PPV) or precision, negative predictive value (NPV), false discovery rate (FDR), and false omission rate (FOR).

Moreover, Figs. 2(a)-(j) show features of ESPRT that have highest and best confusion matrix results. These features are f14, f19, f20, f21, f37, f38, f39, f41, f42, f52, f58, f61, f71, f72.

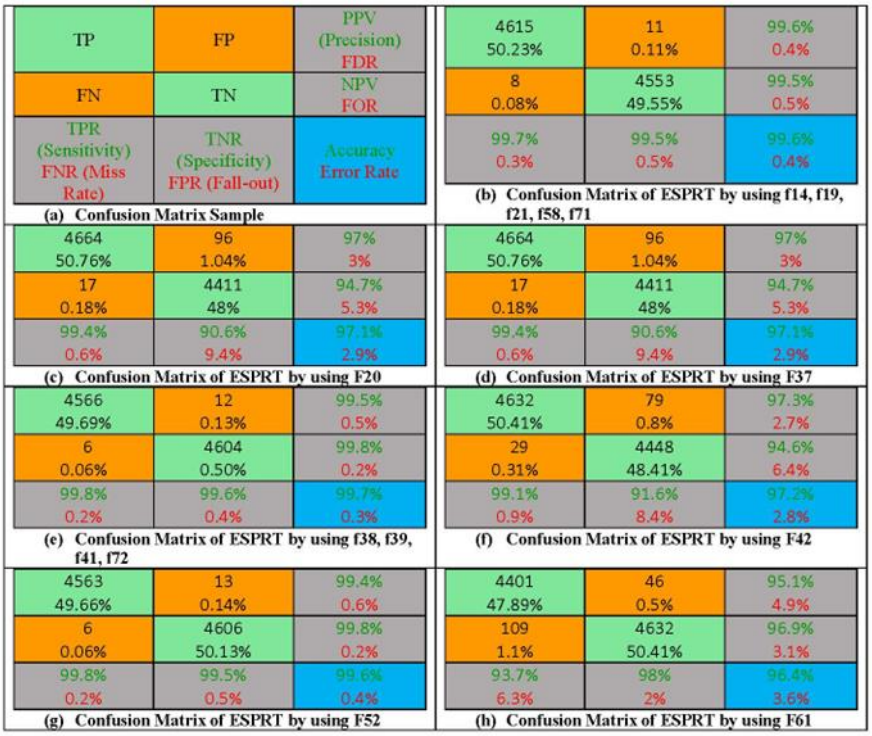
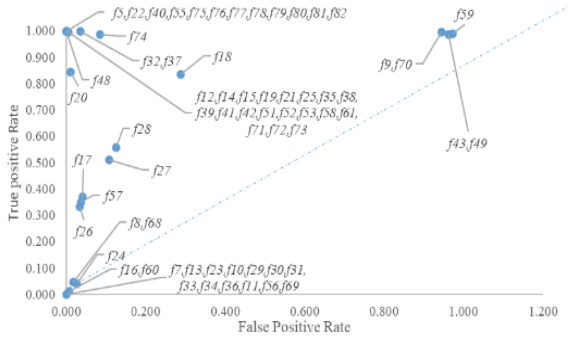
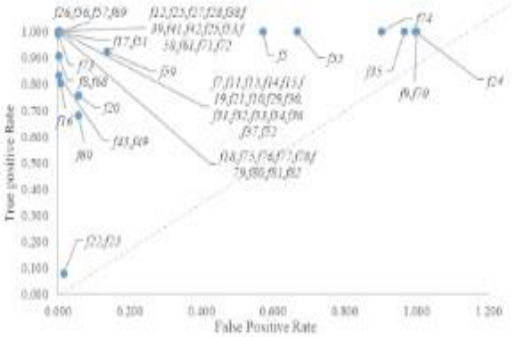


Fig. 2. Confusion matrix of multi-feature ESPRT using CICDDoS2019 dataset.

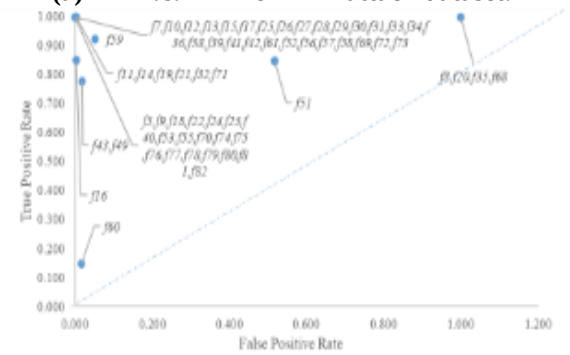
Furthermore, the relationship between sensitivity (TPR) and probability of false alarm (FPR) of all extracted features for all DDoS attacks were tested and shown in Fig. 3. A feature has high detection when it located in the upper left side. However, when a feature located in the lower right side, it has low detection. For example, most features are located in the upper left side for the TFTP attack as shown in Fig. 3 (j). Some features such as f5, f9, f32, f43, f49, f70, f79, f81 and f82 generate high false alarm for TFTP attack. Finally, the performance of all features for other kinds of DDoS can be seen in Figs. 3. (a)-(j).



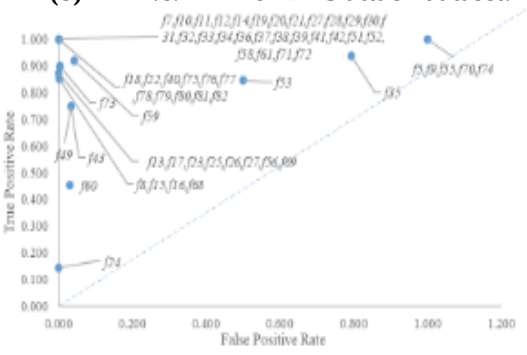
(a) TPR vs. FPR for NTP attack dataset.



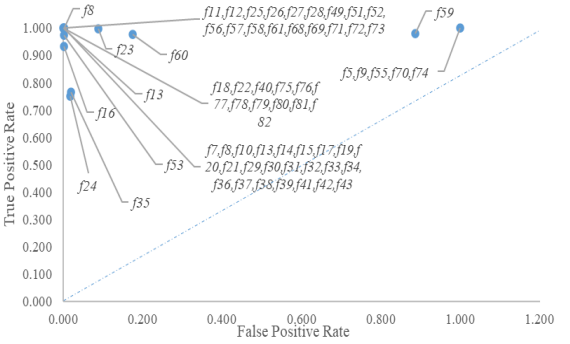
(b) TPR vs. FPR for DNS attack dataset.



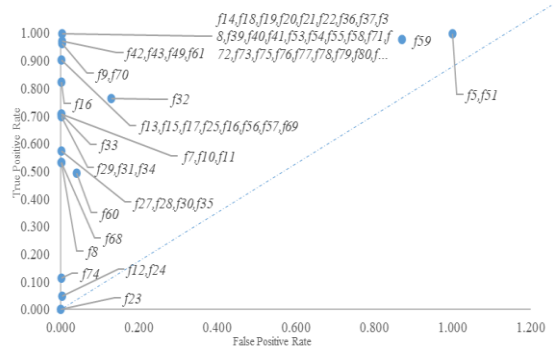
(c) LDAP attack dataset.



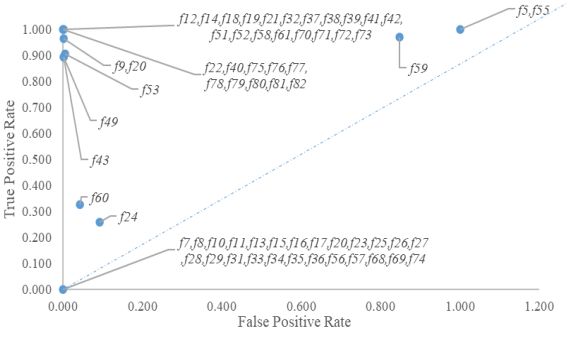
(d) MSSQL attack dataset.



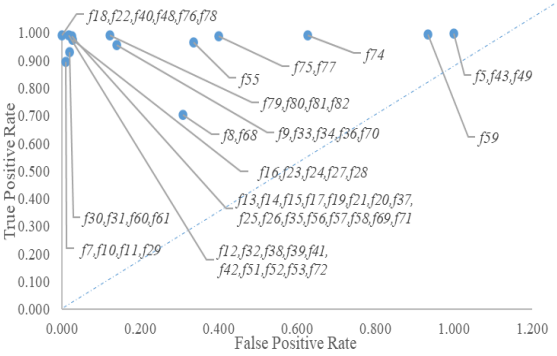
(e) NETBIOS attack dataset.



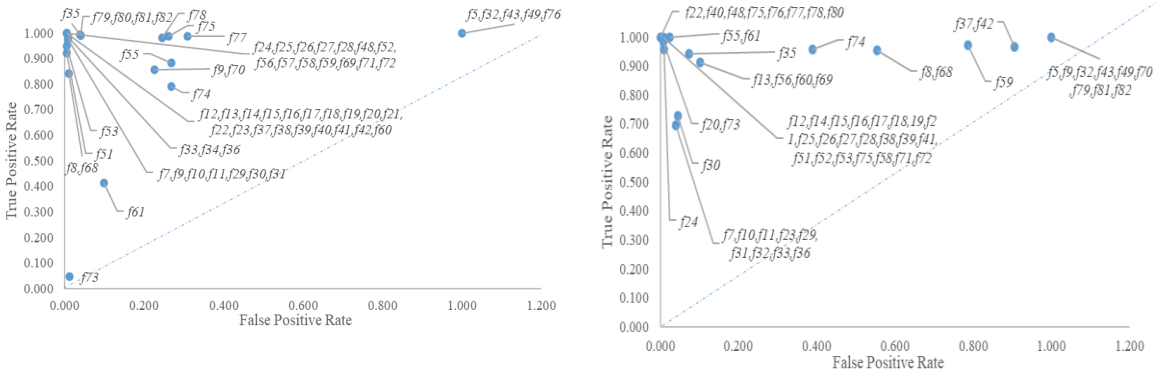
(f) SSDP attack dataset.



(g) UDP attack dataset.



(h) UDP-Lag attack dataset.



(i) SYN attack dataset. **(j) TFTP attack dataset.**

Fig. 3. TPR vs. FPR for all features of ESPRT using different DDoS attacks of CICDDoS2019 dataset.

4.3. Evaluation and parameters

The first parameter used in Multi-features ESPRT detection method is window size. Multi-features ESPRT detection method divides incoming flows into fixed window size. The window size can be determined based either on the number of flows or time frame. The number of incoming flows were used in this experiment. The best number of flows that was used as window size was higher than 300 flows. This window size is better than others such as 75, 100, or 200.

The second parameter used in Multi-features ESPRT detection method was β and α that were used in Eq. (9). The value of false negative β and false positive α should be between 0.01 and 0.05 to have optimum value for A and B and bypass false negative and false positive rate.

4.4. Comparison with other approaches

CICDDoS2019 dataset that contain different DDoS attacks were used to evaluate the performance of multi-feature E-SPRT detection method and compare it's results with other detection approaches. These DDoS attacks that are available in CICDDoS2019 are NTP, DNS, LDAP, MSSQL, NETBIOS, SSDP, UDP, UDP-lag, SYN, and TFTP. The comparison has been done based on some of important metrics of the confusion matrix such as detection rate, probability of false alarm, and accuracy. Some features of ESPRT that detected most of these attacks were chosen in this comparison such as f14, f19, f21, f37, f38, f39, f41, f42, f52, f58, f61, f71, f72.

Detection rate of most of the selected features of Multi-feature ESPRT is better than HTM-KNN [15] and PCC-AE-DNN [20]. These rates fall within the range between 99.39% to 99.99% for most attacks such as NTP, DNS, LDAP, MSSQL, NETBIOS, and UDP. However, detection rate for HTM-KNN [15] method falls within the range of 77% to 91% for these DDoS attacks. Detection rate of PCC-AE-DNN method [20] falls in the range of 56.5% to 99.4 for these attacks.

In addition, the detection rate for selected features of Multi-feature ESPRT is better than HTM-KNN [15] and PCC-AE-DNN in identifying SSDP, UDP-lag, SYN, and TFTP. The rates in identifying these attacks by using Multi-feature

ESPRT fall within range of 95.11% to 99.92% for most features except f61 in identifying SYN attack. However, the detection rate falls within range of 80% to 90% for HTM-KNN [15] and 93.9% to 99.8% for PCC-AE-DNN [20] in identifying these kinds of attacks.

For example, most presented features of ESPRT have 99% or 98% of detection rate of UDP-lag attack except some features such as f61 which generate 92% of detection rate. However, HTM-KNN fails to detect UDP-lag, and PCC-AE-DNN has 93.9 % of detection rate. Finally, Fourier transform and entropy-based DDoS detection [16] was able to detect UDP-lag attack with 99.75% as shown in Table 3.

Another example, detection rate of ESPRT using f61 as input feature was less than 90% in identifying SYN attack, while the rest features were succeeded in identifying of SYN attacks with values above 99% when they were as input for ESPRT. However, detection rate of SYN attacks for PCC-AE-DNN [20] and Fourier transform and entropy-based DDoS detection [16] was more than 99%. Finally, HTM-KNN [15] generated 90% of detection rate as shown in Table 3.

Table 3. Comparison detection rate and FPR metrics of multi-features ESPRT Detection and other approaches using CICDDoS2019 dataset.

Metric	Attack Types	HTM-KNN [15]	PCC-AE-DNN [20]	FEDDM [16]	Multi-features ESPRT						
					f14 ,f19, f21, f58, f71	*f38, f39, f41, f72*	*f42*	*f52*	*f37*	*f61*	*Average of selected features*
Detection Rate	NTP	NA*	0.973	0.9956	0.9949	0.9939	0.9939	0.9939	0.9937	0.9939	0.9940
	DNS	NA	0.686	0.9979	0.9992	0.9993	0.9992	0.9992	0.9974	0.9993	0.9989
	LDAP	0.91	0.565	0.9965	0.9972	0.9986	0.9986	0.9985	0.9985	0.9986	0.9983
	MSSQL	0.77	0.949	0.9997	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993
	NETBIOS	0.87	0.994	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992
	SSDP	0.80	0.982	0.9930	0.9988	0.9988	0.9711	0.9988	0.9988	0.9812	0.9912
	UDP	0.80	0.984	0.9999	0.9990	0.9990	0.9990	0.999	0.9968	0.999	0.9986
	UDP-lag	0	0.939	0.9975	0.9845	0.9909	0.9908	0.9908	0.9843	0.9218	0.9771
	SYN	0.90	0.998	0.9999	0.9978	0.9978	0.9935	0.9978	0.9978	0.4130	0.8996
	TFTP	NA	0.997	0.9947	0.9971	0.9970	0.9519	0.9970	0.9645	0.9970	0.9840
(FPR), probability of false alarm	NTP	NA	NA	0.0021	0.005	0.0060	0.0060	0.0055	0.0359	0.0060	0.0107
	DNS	NA	NA	0.0201	0.0018	0.0012	0.0014	0.0020	0.0060	0.0012	0.0022
	LDAP	NA	NA	0.0098	0.0028	0.0028	0.0028	0.0025	0.0025	0.0028	0.0027
	MSSQL	NA	NA	0.0078	0.0016	0.0016	0.0013	0.0013	0.0013	0.0015	0.0014
	NETBIOS	NA	NA	0.0091	0.0016	0.0016	0.0015	0.0021	0.0024	0.0022	0.0019
	SSDP	NA	NA	0.0222	0.0028	0.0028	0.0031	0.0025	0.0065	0.0030	0.0034
	UDP	NA	NA	0.1576	0.0015	0.0015	0.0031	0.0021	0.0040	0.0015	0.0022
	UDP-lag	NA	NA	0.1123	0.0292	0.0146	0.0190	0.0199	0.0568	0.0703	0.0349
	SYN	NA	NA	0	0.0047	0.0047	0.0048	0.0043	0.0060	0.0988	0.0205
	TFTP	NA	NA	0.132	0.0070	0.0070	0.8781	0.0064	0.9049	0.0241	0.3045

NA: Not Available

Moreover, true positive rate for Fourier transforms and entropy-based DDoS detection method FEDDM [16] lies in the range of 0.993 to 0.999 for these attacks as shown in Table 3. It behaves well in identifying UDP-lag and SYN attacks comparing with Multi-features ESPRT. However, the average detection rate for the selected features of ESPRT is better than FEDDM in identifying DNS, LDAP, and NETBIOS. Finally, the detection rates of Multi-features ESPRT for the rest attacks are almost close or less than FEDDM [16] as shown in the Table 3.

In addition, multi-features ESPRT and Fourier transform entropy-based DDoS detection generate lower value of probability of false alarm for most attack types. For example, the average false alarm for selected features of ESPRT was less than FEDDM [16]. These average of false alarm values for selected features lie in the

range of 0.001 to 0.03 for ESPRT while the range was from 0.007 to 0.157 for FEDDM as shown Table 3.

Finally, HTM-KNN and average of features selected for ESPRT generated high accuracy which is above or equal to 99% for different kinds of DDoS attacks such as NTP, DNS, LDAP, MSSQL, NETBIOS, SSDP, and UDP. However, PCC-AE-DNN [20] produce an accuracy that falls in the range of 68.8% to 99.8% for these attacks type. ESPRT has 97% of accuracy comparing with HTM-KNN and PCC-AE-DNN which have 96% and 93% of accuracy respectively in identifying UDP-lag attack. In addition, HTM-KNN and PCC-AE-DNN has 99% of accuracy in detecting SYN attack while ESPRT has 95%. Furthermore, PCC-AE-DNN has better accuracy which is 99% comparing with ESPRT that has 91% of accuracy as shown in Fig. 4. Finally, execution time in seconds for each feature of proposed detection method for different attacks available in CICDDoS2019 dataset were presented in Fig. 5.

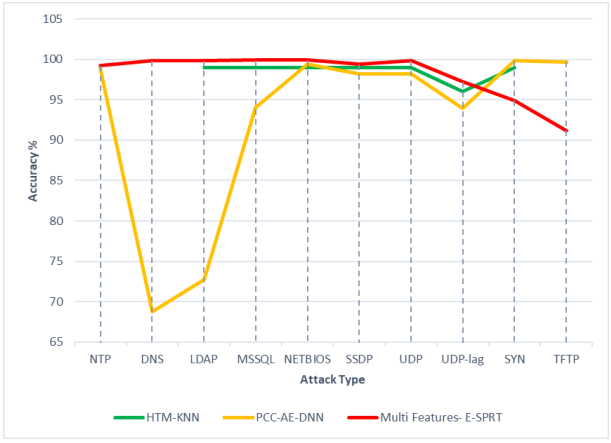


Fig. 4. Comparison accuracy of multi-features ESPRT detection and other approaches using CICDDoS2019 dataset.

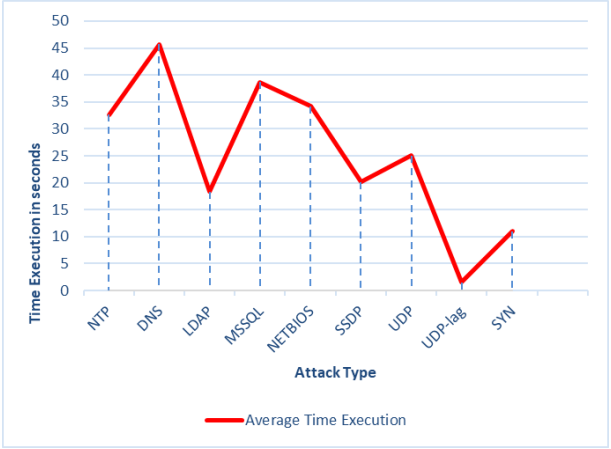


Fig. 5. Average execution time in seconds for multi-features ESPRT detection using different attack types in CICDDoS2019 dataset.

5. Conclusions

Distributed Denial of Service (DDoS) is the most dangerous attacks that targeted public servers. It is very hard to identify DDoS attacks because attackers can hide their identity and use legitimate users as bots to target victim. Identifying different kinds of DDoS attacks and locating malicious interface is the main goal of this research.

In addition, Multiple features were extracted from incoming flows using CICFlowMeter. Then, incoming flows divided to fixed window size that has the same number of flows per feature. Entropy value will be calculated for each group of flows and SPRT applied on entropy vector in order to take decision.

Moreover, CICDDoS2019 and confusion matrix were used in order to evaluate the performance of the detection method. The results were compared with other higher-accuracy techniques. Some features of ESPRT that detected most of these attacks were chosen in this comparison such as f14, f19, f21, f37, f38, f39, f41, f42, f52, f58, f61, f71, f72.

Finally, the implemented model for different features detects most DDoS attacks and achieves a detection rate fall within the range between 99.39% to 99.99% for most attacks such as NTP, DNS, LDAP, MSSQL, NETBIOS, and UDP. In addition, the accuracy of ESPRT was above or equal to 99% for different kinds of DDoS attacks such as NTP, DNS, LDAP, MSSQL, NETBIOS, SSDP, and UDP. Moreover, multi-features ESPRT generated lower value of probability of false alarm for most attack types. For the sake of future work, we will test more features on the proposed approach. Finally, we will also be looking for the best algorithm to do feature selection to choose the best features among features that were generated by CICFlowMeter tool.

Acknowledgement

We would like to thank the staff of the college of engineering of Al-Iraqia university for presenting support to complete this work as shown in an official paper.

Nomenclatures

A	Upper bound threshold
B	Lower bound threshold
D_n^e	The detection of SPRT method
IAT	Time between two packets sent in the flow
Num	Number of flows per window
n	Observations of flows
P	Likelihood of occurrence of feature values
Tn	Series of Entropy values
X	Flows and their feature values
Y	The number of occurrences of feature values

Greek Symbols

α	False positive error
β	False negative error
λ_1	Compromise interface
λ_0	Normal interface
μ_0	The probability of normal interface injected with normal flows

μ_I	The probability of infected interface injected with malicious flows

Abbreviations

ACK	Acknowledgment flag
ANN	Artificial Neural Network
Bwd	Backward
CPSs	Cyber-Physical Systems
CWR	Congestion Window Reduced
DDoS	Distributed Denial of Service
DNS	Domain Name System
F	Feature
E-SPRT	Entropy and Sequential Probability Ratio Test
FDR	False Discovery Rate
FFT	Fast Fourier Transform
FEDDM	Fast Fourier transform with entropy Detection DDoS Method
FIN	Finished flag
FN	False Negative
FOR	False Omission Rate
FP	False positive
Fwd	Forward
HOSVD	Higher Order Singular Value Decomposition
HTM	Hierarchical Temporal Memory
KNN	K Nearest Neighbors
Kpps	Kilo of Packet per Second
Krps	Kilo of Request per Second
LDAP	Lightweight Directory Access Protocol
Mpps	Mega of Request per Second
Mrps	Mega of Packet per Second
MSSQL	Microsoft SQL Server
NB	Naïve Bayes
NETB-IOS	Network Basic Input/Output System
NPV	Negative Predictive Value
NTP	Network Time Protocol
PCC-AE-DNN	Pearson Correlation Coefficient, Auto Encoder, and Dense Neural Networks.
PPV	Positive Predictive Value
PSH	Push flag
RST	reset flag
SDN	Software Defined Network
SMOTE	Synthetic Minority Over-Sampling Technique
SSDP	Simple Service Discovery Protocol
Std	Standard deviation
SVM	Support Vector Machine
SYN	Synchronize
TFTP	Trivial File Transfer Protocol
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol

URG	Urgent flag
W	Windows

References

1. Liu, Y.; Zhi, T.; Shen, M.; Wang, L.; Li, Y.; and Wan, M. (2022). Software-defined DDoS detection with information entropy analysis and optimized deep learning. *Future Generation Computer Systems*, 129, 99-114.
2. Long, Z.; and Jinsong, W. (2022). A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN. *Computers & Security*, 115, 102604, 1-13.
3. Miu, T.; Yeung, R.; Cheung, K.; and Li, D. (2020). DoS threat report 2020 Q2. *Nexusguard*.
4. Chaganti, R.; Bhushan, B.; and Ravi, V. (2022). The role of blockchain in DDoS attacks mitigation: techniques, open challenges, and future directions. *arXiv*, arXiv:2202.03617v1, 1-14.
5. Verma, A.; Arif, M.; and Husain, M.S. (2018). Analysis of DDOS attack detection and prevention in cloud environment: A review. *International Journal of Advanced Research in Computer Science*, 9(Special issue No. 2), 107-115.
6. Menscher, D. (2020). Google cloud. Retrieved March 10, 2022, from https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks.
7. Ali, B.H.; Sulaiman, N.; Al-Haddad, S.A.R.; Atan, R.; Hassan, S.L.M.; and Alghrairi, M. (2021). Identification of distributed denial of services anomalies by using combination of entropy and sequential probabilities ratio test methods. *Sensors*, 21(19), 6453,1-17.
8. Gaur, V.; and Kumar, R. (2022). FSMDAD: Feature selection method for DDOS attack detection. *Proceedings of the 2022 International Conference on Electronics and Renewable Systems* (*ICEARS*), Tuticorin, India, 939-944.
9. Jaafar, A.G.; Ismail, S.A.; Abdullah, M.S.; Kama, N.; Azmi, A.; and Yusop, O.M. (2020). Recent Analysis of forged request headers constituted by HTTP DDoS. *Sensors*, 20(14), 3820, 1-29.
10. Dao, N.-N.; Park, J.; Park, M.; and Cho, S. (2015). A feasible method to combat against DDOS attack in SDN network. *Proceedings of the 2015 International Conference on Information Networking* (*ICOIN*), Cambodia, 309-311.
11. Piedrahita, A.F.M.; Rueda, S.; Mattos, D.M.F.; and Duarte, O.C.M.B. (2015). Flowfence: a denial of service defense system for software defined networking. *Proceedings of the 2015 Global Information Infrastructure and Networking Symposium* (*GIIS*), Guadalajara, Mexico, 1-6.
12. Durner, R.; Lorenz, C.; Wiedemann, M.; and Kellerer, W. (2017). Detecting and mitigating denial of service attacks against the data plane in software defined networks. *Proceedings of the 2017 IEEE Conference on Network Softwarization* (*NetSoft*), Bologna, Italy, 1-6.

13. Koay, A.; Chen, A.; Welch, I.; and Seah, W.K.G. (2018). A new multi classifier system using entropy-based features in DDoS attack detection. *Proceedings of the 2018 International Conference on Information Networking* (*ICOIN*), Chiang Mai, Thailand, 162-167.
14. Mousavi, S. M.; and St-Hilaire, M. (2015). Early detection of DDoS attacks against SDN controllers. *Proceedings of the 2015 International Conference on Computing, Networking and Communications* (*ICNC*), Garden Grove, CA, USA, 77-81.
15. Nguyen, M.; Lai, Y.-K.; and Chang, K.-P. (2021). An entropy-based DDoS attack detection and classification with hierarchical temporal memory. *Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (*APSIPA ASC*), Tokyo, Japan, 1942-1948.
16. Liu, Z.; Hu, C.; and Shan, C. (2021). Riemannian manifold on stream data: Fourier transform and entropy-based DDoS attacks detection method. *Computers & Security*, 109, 102392, 1-15.
17. Dong, P.; Du, X.; Zhang, H.; and Xu, T. (2016). A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. *Proceedings of the 2016 IEEE International Conference on Communications* (*ICC*), Kuala Lumpur, Malaysia, 1-6.
18. Maranhão, J.; Da Costa, J.P.C.L.; de Freitas, E.P.; Javidi, E.; and de Sousa Júnior, R.T. (2020). Error-robust distributed denial of service attack detection based on an average common feature extraction technique. *Sensors*, 20(20), 5845, 1-21.
19. Polat, H.; Polat, O.; and Cetin, A. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, 12(3), 1035, 1-16.
20. Li, J. (2020). *Detection of DDOS attacks based on dense neural networks, autoencoders and Pearson correlation coefficient*. MSc dissertation, Faculty of Computer Science, Dalhousie University, 1-81.
21. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; and Ghorbani, A.A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. *Proceedings of the 2019 International Carnahan Conference on Security Technology* (*ICCST*), Chennai, India, 1-8.
22. Sharafaldin, I.; Gharib, A.; Lashkari, A.H.; and Ghorbani, A.A. (2017). Towards a reliable intrusion detection benchmark dataset. *Journal of Software Networking*, Volume 2017, 1, 177-200.