The ISC Int'l Journal of Information Security

July 2020, Volume 12, Number 2 (pp. 113–128)
http://www.isecure-journal.org

F-STONE: A Fast Real-Time DDOS Attack Detection Method Using an Improved Historical Memory Management

Mahsa Nooribakhsh [1], and Mahdi Mollamotalebi [1,*]
[1] *Department of Computer, Buinzahra Branch, Islamic Azad University, Buinzahra, Iran*

ARTICLE INFO.

Article history:
Received: 12 January 2019
Revised: 15 November 2019
Accepted: 27 May 2020
Published Online: 31 May 2020

Keywords:
DDoS Detection, Real-Time Detection, Datastream Algorithm, Binary-Mapped Historical-Memory Management, Anomaly Detection, Expected Profile Update Time.

Abstract

Distributed Denial of Service (DDoS) is a common attack in recent years that can deplete the bandwidth of victim nodes by flooding packets. Based on the type and quantity of traffic used for the attack and the exploited vulnerability of the target, DDoS attacks are grouped into three categories as Volumetric attacks, Protocol attacks, and Application attacks. The volumetric attack, which the proposed method attempts to detect it, is the most common type of DDoS attacks. The aim of this paper is to reduce the delay of real-time detection of DDoS attacks utilizing hybrid structures based on data stream algorithms. The proposed data structure (BHM [1]) improves the data storing mechanism presented in the STONE method and consequently reduces the detection time. STONE characterizes regular network traffic of a service by aggregating it into common prefixes of IP addresses, and detecting attacks when the aggregated traffic deviates from the regular one. In BHM, history refers to the output traffic information obtained from each monitoring period to form a reference profile. The reference profile is created by employing historical information and only includes normal traffic information. The delay of DDoS attack detection increases in STONE due to long-time intervals between each monitoring period. The proposed method (F-STONE) has been compared to STONE based on attack detection time, Expected Profile Update Time (EPUT), and rate of attack detection. The evaluation results indicated significant improvements in terms of the EPUT, acceleration of attack detection, and reduction of false positive rate.

1 Introduction

Due to rapid growth of the Internet and its various services, Distributed Denial of Service (DDoS) attacks have become a common threat to Internet services such as the Web, Domain Name Service (DNS), and Internet backbone links in recent years (flood attacks to the bandwidth) [1]. A DDoS attack attempts to prevent or limit the legitimate users using a specific service. This study focused on detecting bandwidth-flood attacks. One of the most common bandwidth-flood attacks is the SYN flood attack [2], which uses the weakness of the TCP/IP three-way handshaking process.

Given that all computers connected to a TCP-based network (such as WEB, FTP, and Mail servers)

* Corresponding author.
Email addresses: `nooribakhsh@buiniau.ac.ir`, `motalebi@qiau.ac.ir`
ISSN: 2008-2045 © 2020 ISC. All rights reserved.

use a three-way handshaking process to connect to one another, they are all at the risk of an SYN flood attack. When a server receives an SYN packet, it construes the packet as a request by a remote node to initiate a TCP connection. The server allocates the required resources to track the TCP state and returns an SYN-ACK packet to the sender node. Then it waits until either the half-open connection completes or the TCP connection times out. In an SYN flood attack, the server receives several SYN packets but never gets any ACK packets to complete the three-way handshake. Thus the serverâĂŹs backlog queue would be exhausted, and all the new incoming SYN requests are dropped. This caused the server as denied to serve correctly.

DDoS detection techniques are often divided into anomaly-based and signature-based categories [3]. The signature-based (or knowledge-based) detection is used to identify security threats (for instance, viruses or incomplete packets) by following specific patterns appearing in separate packets. For detecting attacks, such methods compare network traffic with existing patterns of known attacks and usually have a database of signature/pattern attacks. By examining network traffic, they find similar patterns to those in their database. These methods are only capable of detecting known attacks, and the network managers should always add new attacksâĂŹ pattern to the intrusion detection system. Snort [4] is a sample of the signature-based attack detection system.

Anomaly-based methods create a profile of normal network traffic behavior and then consider any violation or deviation from the normal profile as an abnormal/attack behavior. Such methods are also categorized into two types of offline and real-time detection, depending on the time of the attack detection. This paper intended to accelerate the detection of anomalies in real-time. Regardless of offline or real-time detection, DDoS attacks can be identified in five different categories based on data mining, machine learning, soft computing, statistical analysis, and data stream algorithms [3–5].

The statistical methods observe and analyze the input traffic behavior in different time periods. Soft computing-based methods (e.g., neural networks) attempt to classify inbound packets automatically to detect DDoS attacks; such methods suffer from inaccuracy and uncertainty. Data mining and machine learning-based methods use historical network traffic data. They, along with classification, association rules, and clustering, extract patterns from network traffic to distinguish between normal and abnormal traffics and, subsequently, detect suspicious activities. Datastream algorithms have been provided to analyze large volumes of data as real-time in high-speed networks; they monitor the network traffic stream continuously and make a decision instantly [6–9].

The techniques based on data mining and machine learning [6–8] and soft computing [9] suffer from a high degree of computational complexity, and consequently, their detection delay is relatively high. Another drawback is a high rate of false detection because the behavior of the system may not be completely covered during the learning phase. Moreover, the data stream behavior varies over time, and creating an online retraining course is needed for traffic behavior features. When an attack occurs, the data rate increases rapidly, and machine learning algorithms are susceptible to bottlenecks as a result of the large volume of analytical samples and a large number of features. Also, the delay in detecting an attack in the above methods is high [10]. Statistical analysis methods examine the user/network behavior by measuring definite statistical variables over time. The data stream algorithms are proposed to detect DDoS attacks in high-speed network links; they analyze packets stream with continuous queries to find anomaly traffic patterns associated with DDoS attacks in real-time [11].

The remainder of this paper is organized as follows; in Section 2, related prior methods of DDoS attack detection are reviewed. The detail of the current paperâĂŹs proposed method is presented in Section 3. In Section 4, the results of the implementation are presented and compared to the recent related methods, and Section 5 concludes the paper.

2 Related Work

The purpose of a DDoS attack detection system is to monitor communication traffic and detect an attack as real-time. The results of the attack detection system determine the appropriate defense against the attack [12]. In the following, various data-streaming mechanisms for DDoS detection are reviewed.

Anceaume *et al.* [13] presented an algorithm called Ankle, whose goal is to process large input data and evaluate the Kullback-Leibler divergence [14] in DDoS attacks on distributed large-scale systems. This algorithm is based on the data stream and evaluates the divergence between the two streams. The divergence parameter employed in this algorithm is used to compare current traffic and legitimate traffic. Ankle processes a large amount of data, alternately in real-time. This method has high accuracy in detecting the DDoS attack; however, it is not able to properly assess the divergence between malicious and legitimate streams in low rate traffics.

Gulisano *et al.* [10] presented a hybrid structure using a Stream-based expert system to detect anomalies associated with real-time DDoS attacks for high-speed networks. This method, called STONE, includes items such as aggregating based on the same IP prefix, providing a set of detected traffic features to detect DDoS attacks, and maintaining/updating reference profiles in a database. This structure is designed to reduce the most disruptive traffic while eliminating the least legitimate traffic. Detecting a high-precision DDoS attack as real-time, and mitigation effectiveness attack, are the main advantages of STONE. On the contrary, due to frequent use of database and storage limitations, data are determined in the regulatory period, and its run time is relatively high.

Andrysiak *et al.* [15] developed a method for detecting DDoS attacks based on the modeling of traffic variability using conditional mean and variances in time series. It uses the maximum likelihood function to evaluate the analysis parameters. In order to reduce the prediction errors, the traffic features were summed up, and it detects the anomaly behavior by stochastic predicting and comparing to current network traffic. The results of this method indicate improvements in the detection rate and decreasing the false-positive rate.

Hoque *et al.* [16] used a correlation criterion to detect DDoS attacks. In order to reduce the detection delay, they implemented the algorithm on FPGA. Although the correlation measurement methods are low accurate in detection for real-time network traffic analysis with a small number of features, this method detects attacks with appropriate accuracy. It also calculates the correlation based on the standard deviation of the two entities utilizing their average values. The correlation criterion calculates the absolute distance between two entities and is able to analyze the correlation between the two samples with a very small number of parameters. This improves the real-time detection of DDoS attacks.

Nagy *et al.* [17] presented an FPGA-based DDoS detector that is able to detect the top nine types of DDoS attacks within milliseconds. It receives chunked packets through a switch N:1 mirroring port and triggers ACL rule injection within the switch as a sub-second attack mitigation action. Its parser extracts the header fields from the captured packets; they are used by the flow rate classification engine. Also, some new encapsulations are added to the detector so that if a data field deemed interesting for an attack detection algorithm, it could be parsed easily. Since the detector is a time-variant system, even with identical traffic playbacks, there will be variance on detection times. With regard to using FPGA in this system, the detection time is short significantly; however, it is not flexible easily.

Bista *et al.* [18] proposed a system to detect DDoS attacks using the clustering followed by classification. It uses Heuristics Clustering Algorithm (HCA) for clustering the available data. It also uses NaÃŕve Bayes (NB) classification to detect the attacks based on the features of data packets. The clustering is based on unsupervised learning. For some kind of DDoS attacks that include few normal instances, this system is not able to detect the attack; therefore, a NaÃŕve Bayes classification is also used along with clustering to deal with this problem and improving the accuracy. On the other hand, this system is able to improve accuracy with a low false-positive rate.

Baskar *et al.* [19] provided a time-variant predicate based traffic approximation algorithm for detecting the low rate DDoS attacks. It maintains a set of predicates generated using the access trace available of low rate attacks in some time windows. The algorithm generates the predicates from the malicious access trace and how they can be done at a low level. Then, each received request is classified as genuine or malicious by closure weight computed based on the frequency of the accessed service. This algorithm improves the detection of low rate DDoS attacks; however, it suffers from high overheads.

Koay *et al.* [20] proposed a set of new entropy-based features to increase the accuracy of DDoS attacks detection. The entropy-based features are widely used for DDoS detection, mainly focusing on the use of a small number of features to distinguish attack, and normal traffics. This results in detecting the limited types of DDoS attacks. By the above-proposed features, a multi classifier system is presented based on a set of multiple entropy-based features and machine learning classifiers. This system could increase the generality and accuracy of detecting low-rate and high-rate DDoS attacks with the cost of more delays.

Krasnov *et al.* [21] proposed an approach for network traffic anomaly detection that describes network traffic consisting of generalized coordinates of traffic workload parameters and generalized velocities determined by non-recursive filtering. The information of traffic states is kept in the form of phase portraits that is a two-dimensional frequency distribution of traffic phase coordinates. Wald's sequential analysis and the Bayesian decision theory with some levels of significance are applied. This system is able to detect the real attack flows efficiently; however, it could not keep the performance in terms of delay and workloads.

Ma *et al.* [22] have used entropy chaos analysis and Liaponov's power [23] to detect network anomalies. The entropy describes the network trafficâĂŹs characteristics, but it considers only the calculated values of each packet field independently and ignores the relationship between the fields. In some DDoS attacks such as spoof based ones, the characteristics are more dispersed, and the entropy can detect them. The network traffic is pre-processed by entropy-based methods. Then, a DDoS attack could be detected by analyzing the chaos on the entropy of the IP address for the source and destination at each time. The false-positive and false-negative rates of this method are almost zero though its processing time is relatively long.

Behal *et al.* [24] proposed a set of information theory metrics called φ-Entropy and φ-Divergence metrics for detecting DDoS attacks and flash events. The above metrics are sensitive towards detecting smooth variations in the network traffic and extract more information distance between normal, and attack traffics. This technique computes metrics on the network flows and defines attack or normal traffic on corresponding values of the threshold. It calculates the entropy values based on the number of packets per source IP to highlight the difference in the pattern of incoming requests in different types of network traffic. The algorithm is independent of any specific attack tool and can detect DDoS attacks and flash events.

Fortunati *et al.* [25] provided an algorithm to improve the statistical anomaly detection in the network traffic using covariance criteria to create a profile for common network traffic and detect anomaly in the data flow. A revised version of decision-making rules based on the Chebyshev difference [26] is used to improve the detection. There are decision-making rules that take into account all the information in the covariance matrix. The ROC (Receiver Operating Characteristic) curve is used to evaluate the performance of this algorithm (based on false-positive alarms). This algorithm suffers from latency due to considering all information in the covariance matrix and subsequent high computational complexity.

ÃŰzÃğelik *et al.* [27] presented a DDoS attack detection approach called CUSUM-Entropy, which uses the accumulation algorithm to handle the traffic entropy along with a wavelet pre-filtering. The entropy measures the data disruption. Increasing the entropy of the source IP address during a DDoS attack reduces the entropy of the destination IP address. The wavelet is used to filter the long-term changes of the observed entropy and reduce the false alarms. The signal is reconstructed to filter the entropy data. This method is able to detect DDoS attacks with high precision and low false-positive rates; however, normalizing the entropy results in more delay in attack detection.

Considering the reviewed literature, the DDoS detection methods and their advantages/disadvantages have been summarized in Table 1.

In this paper, a method for real-time detection of network layer DDoS attacks is proposed. The aim of the proposed method is to reduce the attack detection delay and decrease the false positive rates in database attack detection systems. To this aim, an efficient data structure is designed to manage the storage of network traffic features, which significantly reduces the number of operating function calls for access to the database. Because the proposed method has improved the STONE method [10], it is called as F-STONE (Fast STONE).

3 The Proposed Method

In this paper, a data structure for Binary-mapping memory History Management (BHM) is proposed, in which, history refers to the output traffic information obtained from each monitoring period to form a reference profile. The reference profile is created employing historical information (several previous regulatory periods) and only includes normal traffic information. In order to detect anomalies suspected of the DDoS attack, the information in the reference profile is compared to current traffic.

The delay of DDoS attacks increases due to intervals between each monitoring period as much as the functions of the update, refresh, and persist, which will have an adverse effect on performance. In this way, as a result of considering all requests, the response time lasts longer from the servers that provide online services. Therefore, providing an appropriate strategy to reduce the information exchange with the database could have a significant effect on performance improvement. In this research, a binary-mapping memory history management (BHM) method is proposed to reduce the database dependency. It is carried out through the management of the monitoring period information and reducing the frequent calls to the database access functions. In other words, BHM reduces the delay in detecting center modules and increases the rate of attack detection. Figure 1 illustrates the schema of the proposed system.

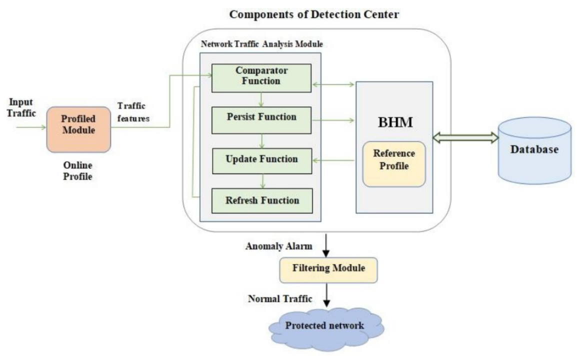

Figure 1. The schema of the proposed DDoS attack detection system

Table 1: The comparison of DDoS detection methods based on the main features

Ref.	Strengths	Weaknesses	Dataset	Remarks	Features Used in Detection
[10]	1. Detects the DDoS attacks in real-time 2. Rapidly provides minimal degradation of legitimate traffic. 3. Mitigating the attacks by filtering out the majority of the suspicious packets	1. Storing the data just in the monitoring period. 2. Dependence on the Database for accessing the data	Datasets derived from real network traffic collected by CAIDA	It presents a framework with expert system functionality that aggregates the network traffic into common prefixes of IP addresses and detects the attacks when the aggregated traffic deviates from the regular one.	1. prefix of the packetsâĂŹ IP address. 2. The average number of packets, bytes and flow duration. 3. Source clusters sending a significant amount of packets or bytes.
[12]	1. Low false positive rate. 2. Low computational overhead	1. Weak detection for attacks with low traffic rates. 2. Long detection time	Traffic data generated by the Hoepelnuke V2 attack software	Detection of DDoS attack based on two t-test samples by comparing the arrival SYN rate and its normal rate	1. Traffic inertia factor. 2. Number of packets received by the routers during a specified time slot
[13]	1. Quickly process a huge amount of data received in the form of a stream. 2. Efficient both in terms of space and time complexities. 3. Requires only a single pass over the data stream	Reducing the accuracy in low rate traffics	ClarkNet: a full Internet access provider for the Metro Baltimore-Washington DC area. NASA: two months of HTTP requests to the NASA Kennedy Space Center server. Saskatchewan traces: seven months of HTTP requests to the WWW server of the University of Saskatchewan, Canada	Estimates the Kullback-Leibler divergence of an observed stream compared with the expected one. Combines sampling techniques and information-theoretic methods.	1. Number of distinct data items in a stream 2. Size of stream 3. Frequency moments of a stream. 4. Set of the most frequent data items in a stream

Table 1: The comparison of DDoS detection methods based on the main features

[15]	Low rate of false-positive detection	Summarization has been made for studied features, but anomaly detection does not necessarily mean a DDoS attack.	Using CAIDA and also SNORT IDS as sensors in real networks and collecting statistics of attacks	Use of mean and conditional variances in time series to model and predict network traffic changes, and detecting unusual behaviors. Moreover estimating the accuracy of prediction	1. The number of in/out TCP/UDP packets. 2. The number of in/out ICMP packets. 3. The number of TCP packets with SYN and ACK flags. 4. Number of in/out TCP packets on port 80
[16]	1. Low computational overhead. 2. High detection throughput. 3. No negative effects on accuracy despite using correlation measurement for network analysis.	1. Having limitations on the algorithms able to be implemented on hardware, and difficulty of making changes on it. 2. Low sensitivity to attacks compared to the completely software methods	CAIDA DDoS 2007, MIT DARPA, and TUIDS	The ability to analyze the correlation between every two samples in traffic with a limited number of characteristics (entropy of the source IP address, index changes of the source IP address, and packet rates)implementation on FPGA to improve the detection speed	1. The entropy of source IP address. 2. Variation index of source IPs. 3. Packet rate
[17]	1. Using FPGA and parallelized modules to speed-up the DDoS attack detection 2. Ability to detect the top nine DDoS attack types. 3. Based on traffic patterns, not message patterns.	1. Having variance on detection times. 2. Weak flexibility because of using the FPGA	Real network traffic including ordinary data center traffic as well as various DDoS traces	The header fields are extracted from the captured packets. Some new encapsulations are added to the detector to parse the packets easier.	1. Detecting imbalances in the ratio of service requests and responses. 2. The statistics of the incoming packets fragmentation states for each endpoint. 3. Fragmentation offsets, the flags, and IP lengths
[18]	1. high accuracy. 2. low false-positive rate	Weak detection for DDoS attacks with few normal instances	CAIDA UCSD DDoS Attack 2007 Dataset and DARPA 2000 Dataset	Heuristic Clustering Algorithm and Naive Bayes Classification are used for classifying theClusters of data into either Normal or Attack instances. Then a labeling scheme is performed for results comparison.	1. Using Heuristics Clustering to cluster the available data. 2. Using NaÃŕve Bayes (NB) classification

Table 1: The comparison of DDoS detection methods based on the main features

[19]	1. Detecting the low rate of DDoS attacks.2. High accuracy	Suffers from high overheads	CAIDA dataset and DARPA dataset	A time-variant predicate based traffic approximationfrom the malicious access	The frequency of the service being accessed
[20]	1. High accuracy. 2. Using a multi classifier system. 3. detecting the low-intensity and high-intensity DDoS attacks	1. detecting limited types of DDoS attacks2. high amount of delay	UCSD DDoS attack 2007 dataset and DARPA 2000	Proposes a multi classifier system based on a set of multiple entropy-based features and machine learning classifiers	1. Source IP Address, 2. Destination IP Address, 3. Protocol, 4. Source Port, 5. Destination Port, 6. Sequence number, 7. Acknowledgment number, 8. Length, 9. Window size.
[21]	1. The ability to detect real attack flows efficiently. 2. Compression of traffic state descriptions. 3. Ability to detect deviations from normal traffic states.	1. It could not keep the performance in terms of delay. 2. High volumes of workloads	DDoS Attacks in Real Traffic Traces from: - normal traffic with web-surfing - anomalous traffic with HTTP flood DDoS attack - anomalous traffic generated by Slowloris DDoS tool All data were collected by NetFlow protocol.	Describes network traffic with phase coordinates consisting of generalized coordinates of traffic workload parameters and generalized velocities determined by non-recursive filtering of generalized coordinates. Applying Wald's sequential analysis and the Bayesian decision theory	1. Packet length 2. Number of packets
[22]	High detection accuracy	Almost long detection time	LLS DDoS dataset from MIT University	Uses of chaos analysis by Liaponov's power to detect anomalies in network traffic	1. Entropies of source and destination IP address 2. Traffic anomalies at a specific time point
[24]	1. Ability to detect very low-rate DDoS attacks 2. Detecting different types of DDoS attacks and flash events 3. Independent of any specific attack tool	1. Low scalability2. weak in high rate traffic conditions	MIT Lincoln, CAIDA, FIFA and synthetically	Uses two metrics (φ-Entropy and φ-Divergence) that are sensitive towards detecting smooth variations in the network traffic Calculates the entropy values based on the number of packets per source IP	1. Source IP 2. Destination IP 3. Protocol and incoming packet rate

Table 1: The comparison of DDoS detection methods based on the main features

[25]	Low rate of false alerts	1. Slow detection due to considering all the information of the covariance matrix in calculations. 2. Detecting an attack only if a sudden change occurred in the traffic rate	KDD CUP 99	Using covariance to create profiles of normal traffic and detecting abnormal behavior in the data flow. Providing a modified version of decision-making rules based on the Chebyshev difference for random vectors	1. Connections with SYN / REJ errors to the same host 2. Connections to the same service on the same host 3. Connections to different services on the same host 4. Connections to different hosts
[27]	High precision of detection	1. Long detection time due to the need for normalization in the entropy. 2. Analysis of only the sourcesâĂŹ IP address.	Condor cluster traffic of Clemon University	Applying the total accumulation algorithm (CUSUM) on the observed traffic entropy along with the wavelet filtering	1. The entropy of source IP addresses 2. Attack start/stop time 3. The number of Attackers

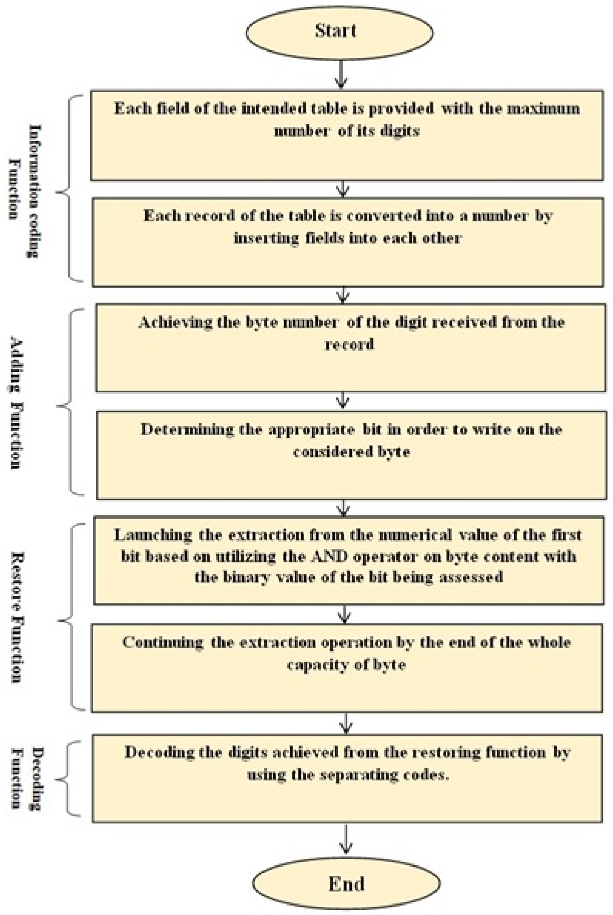
Start
Information coding Function
Each field of the intended table is provided with the maximum number of its digits
Each record of the table is converted into a number by inserting fields into each other
Adding Function
Achieving the byte number of the digit received from the record
Determining the appropriate bit in order to write on the considered byte
Restore Function
Launching the extraction from the numerical value of the first bit based on utilizing the AND operator on byte content with the binary value of the bit being assessed
Continuing the extraction operation by the end of the whole capacity of byte
Decoding Function
Decoding the digits achieved from the restoring function by using the separating codes.
End

Figure 2. The steps of using traffic analysis functions in the proposed DDoS attack detection system

Based on Figure 1, the BHM module is located in the detection center component and reduces the detecting steps. The network traffic is entered into monitored links to check the attack traffic. To detect an attack, a pattern (or profile) of legal traffic is first created. This traffic profile is continuously compared to the current traffic flow, and if a difference is observed, an alert will be issued by the detection control center.

BHM manages the database refers. It removes the repeated updating and refresh tasks on the database and performs them in scheduled times. In this manner, it overcomes the limitation of storing data at the end of the monitoring period and reduces the number of a database refers. Moreover, in contrast to the storing method used in STONE, the storing task of BHM can be performed as parallel to other detection tasks (in different threads). It subsequently speeds up the detection process of the proposed method. Thus, the functionality of the process is independent of the storing tasks, and if the database becomes inaccessible during the storing, BHM schedules the storing to another time. The advantage of BHM is minimizing the waiting times of detection modules for a database connection.

In each module of the detection system, the persist function keeps the required information in the database as the following. The rate monitoring module compares current traffic and historical information to know the rate of resource clusters. The alarm is issued when the distribution of resource clusters outside the expected area is changed. A three-dimensional system (consisting of monitoring, storing, and update functions) is considered for grouping the resource clusters.

In a monitoring period, the considered features of each cluster are kept. When the monitoring period is finished, if no attack is detected, the feature list would be stored in the historical information part of the database. This task is performed by the persisting function. In each monitoring period, the index module records a list of resource cluster probabilities for each group. When the monitoring period is finished and no attack has occurred, this list is recorded in a table named HIDB [2] resides in the database.

This paper employs the structure and processes presented in [10] to form the traffic profile. BHM is mutually linked to the database and manages the number of referrals to the database and storing operations at specified times. This module removes the operations update and refresh in the database, and only performs storing at certain times. Thus, the storing time limit is eliminated at the end of each monitoring period, which results in decreasing the number of database references.

[2] HIDB: History Information Data Base

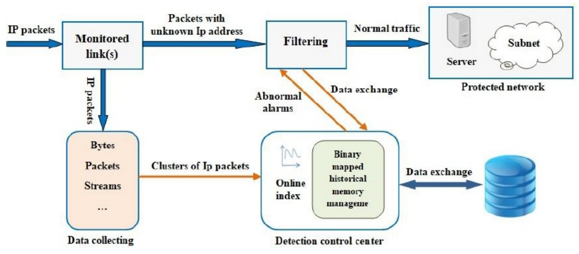

Figure 3. The activity diagram of F-STONE

Moreover, storing operations are performed as parallel to other detection operations (in a separate thread). Storing operations are performed at an appropriate time (e.g., when the server is idle). In this way, the function of the program is not dependent on the storing operations; and if the database is inaccessible for any reason during the store process, the storing will be postponed to another time with no defect in the detection operation.

The network traffic analysis module (including Comparator, Persist, Update, and Refresh functions) compares the received traffic with the reference profile in the database and detects the potential DDoS attack. The task of Refresh function at the beginning of the monitoring period is performed using database information. The Persist function is activated at the end of each monitoring period and stores the traffic features observed during the monitoring period in the database. The Update function specifies how historical information is merged with the latest traffic profile observed to update the reference profile for the next regulatory period. Figure 3 presents the activity diagram of F-STONE.

With the BHM data structure, the connection between the network traffic analyzer and the database is unnecessary. The task of managing the information derived from the traffic analysis functions is borne by BHM, which is designed to store a large volume of numerical information as a binary map for storing operation, recovery, search, integration, and share rapidly. In this structure, if the capacity of each byte is n bits, n can be stored simultaneously in each byte so that instead of saving the binary value equivalent to the number X in bytes, only the bits corresponding to 2^X are valued to 1. Considering 8 bits of capacity for each byte, in BHM structure, 16 numbers can be stored at a time. For instance, numbers 1, 2, 5, 7, 8, 10, 11, 12, and 15 are stored in two bytes in the BHM structure, as shown in Figure 4.

The steps for using traffic analysis functions have been shown in Figure 2. In order to store data, the first

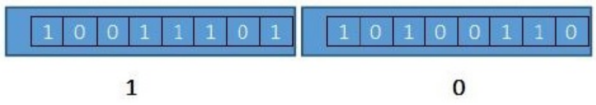

Figure 4. The numbers stored simultaneously in BHM structure

Table 2. A sample of historical data of source clusters in an assumed database

Field1	Field2	Field3
A	B	C
D	EF	GAC
L	K	N
W	UY	DK

data is converted into numeric values and then stored in memory. The operation mode of coding function is that each field of the intended table (to prevent data loss) is provided with the maximum number of its digits. In the next step, each record of the table is converted into a number by inserting fields into each other. Table 2 indicates a sample of data history of resource clusters in the assumed database table.

In order to store the data of Table 2 in the memory, each record is converted to a number by merging the fields (the result is presented in Table 3). The numbers obtained from the encoding function are stored in memory by the redundancy function. To this aim, the corresponding byte should be summed (logical OR operation) with the desired number. In order to search for a number in BHM, the AND operator is utilized. In the search function, it has been determined that the number is located in which bit and byte of the memory. The search function divides the intended number by the total capacity of the byte and gets the number of the intended byte.

In order to simultaneously restore the list of numbers from memory, the recursive function is used in which the scanning and extraction begin from the first bit. The role of the decoding function is to convert the number to the corresponding record. To prevent the data from being lost, it is necessary to allocate space to multiple fields as much as the maximum number of digits after decoding. The BHM data structure provides conditions that allow faster access to data rather than the database. Therefore, data processing is done faster, and consequently, the rate of attack detection is improved.

Similarly to the STONE method, the detection control center in the proposed method consists of three main modules of ratio-monitoring, axes builder, and profile maintainer. The ratio-monitoring module compares the current traffic with information history in order to issue an alarm when an attack occurs. This module has five functions of monitoring, grouping, persist, refresh, and update. The monitoring function keeps the counters for source clusters belonging to each group. The grouping function specifies which group each cluster belongs to. At the end of each monitoring period, the persist function stores the traffic features observed during the monitoring period in the database. The update function specifies how historical information will be merged with the latest traffic attributes observed to update the reference profile for the next monitoring period. The refresh function acts at the beginning of the monitoring period using the database information that determines which expected behavior for the traffic features is monitored.

Table 3. A sample of merging the fields of persist function table to the number by coding function

Field1	Field2	Field3	Merging the field of each record
A	B	C	ABC
D	EF	GAC	DEFGAC
L	K	N	LKN
W	UY	DK	WUYDK

A three-dimensional system is considered for grouping source clusters. The role of the axes builder module is to calculate the source coordinates that update it with historical information over the period of the history. This module has three functions of monitoring, storing, and updating. During a monitoring period, the attributes seen for each source cluster are stored in a list. When the monitoring period is over, if no attack is detected, the above list is stored in the historical data field in the database named HIDB (Historical Information DataBase). In addition to the source cluster information in each group, the weight of each group is also kept. The value of each group in the old and new days determines the weight of each group. The rate of each group must be calculated to determine the weight of each group. The rates of the old groups are reduced by the calculated rate of the old attributes of the source cluster, while the rates of the new groups are increased by the calculated rate of the new attributes of the source cluster. If no attack occurs at the end of the monitoring period, the group rates are stored in the HIDB, and then, the weight of each group is updated.

The axes builder module is responsible for computing and maintaining the origin of the three-dimensional system used to partition source clusters into different groups. The monitoring function is responsible for maintaining counters based on source clusters belonging to each group. The group function determines what clusters fall into each group. The persist function stores the information obtained at the end of each monitoring period. The update function integrates the obtained data with the previous data in the database and finally. The refresh function uses the information in the database as the reference profile information for the next period. Also, the profile maintainer module is responsible for maintaining the information blocks used to filter malicious packets.

4 Results and Evaluation

In this section, the proposed method is evaluated by comparing its experimental results with the similar and recent method STONE [10], in terms of the time it takes to update the referral profile information Expected Profile Update Time (EPUT), the detection time of the attack, the true detection rate, and the false-positive rate. The algorithms were implemented using C# under a hardware platform with a Core i5 2.53 GHz processor and 8GB of RAM. It should be mentioned that hardware-based methods, such as the method presented in [17], are able to gain better processing performance and subsequently faster detection of attacks. The reason is the independence of compile or software limitations of packets monitoring. On the other hand, the hardware-based detection methods (e.g., FPGA) are not as flexible as the software-based ones. With regard to our proposed method that is software-based, its experimental results are compared to similar software-based methods STONE.

The first part of the proposed method pertains to the implementation of the STONE structure and its modifications, and the second part pertains to the design of the BHM data structure, components, and operators required. For evaluation, four factors as EPUT, attack detection time, attack detection rate, and false-positive rate are used. EPUT is the update time of the reference profile, the number of which indicates the times the system updates the reference profile to detect the attack. The false-positive rate is the number of warnings the system detects as the attack, while the traffic is normal. The correct detection of the attack is the number of warnings that the system correctly detects. Moreover, the time of attack detection is the time interval between the arrival of traffic to the network and the moment of anomaly detection in seconds.

Some datasets are used to conduct experiments. The first dataset comprised 1000 random samples generated with the source IP addresses, destination IP addresses, number of packets, number of bytes in the packets, and packet lifetime based on the values of start and end times of each packet. To complete the evaluation in the first experiment, two standard datasets of the network traffic dataset [28] were used. The first dataset, LBL-Conn7 [29], includes TCP Ethernet network traffic between the DMZ Lawrence Berkeley Lab (LBL) in California and the Internet over the course of thirty days from 1,645 hosts, which is a total of three-quarters of a million records. LBL records include the attributes timestamp, duration, protocol, the number of bytes sent by the originator, the number of bytes sent by the responder, local host, remote host, location, and the flag. Less than 15 SYN/FIN/RST packets in one million packets are discarded, and timestamps are measured in microseconds.

The second dataset, NASA-HTTP, derived from the HTTP requests of NASA Kennedy Space Center in Florida [30], includes host features, timestamps, requests, HTTP reply codes, and bytes in the reply. The above information is intended to be used in the proposed method in a format including the Host IP address, number of packets, number of bytes, the start time of sending the packet, and the end time of sending the packet, and presented as input tuples to the simulation process. The information of two consecutive weeks of data (461,612, 3 requests from NASA's Space Center web server in Florida) was used in the first experiment.

In subsequent experiments, the dataset KDD-CUP99 [31] is used to detect attack times. In this dataset, four groups were injected into a normal network traffic attack, one of which is a Denial of Service attack. In the present paper, 10% of KDD-CUP99 was used that includes 494,021 records, of which 97,277 (19.69%) were normal records, and 39,458 (79.24%) were DDoS records. In this dataset, each connection was defined with 41 fields along with a label that specifies a normal connection type or an attack. In order to evaluate the proposed method, three experiments were planned. The first experiment was performed to evaluate the time needed to update the reference profile information (EPUT), the second experiment was based on the true detection rate and the false positive, and the third experiment was provided to evaluate the time of detection of DDoS attacks.

The experiments focus on attacks that can be modeled as connection request flooding (e.g., TCP SYN flooding) and bandwidth flooding (e.g., UDP or ICMP flooding). The targets can be a single (or multiple) servers. SYN-flood is done by flooding a vast amount of SYN packets and thus forcing the victim to maintain many sessions. In addition, a bandwidth-flooding attack, a big volume of different types of traffic (TCP packets, UDP packets, ICMP packets, etc.) is sent to congest the target network link.

The method STONE compares the input traffic be-

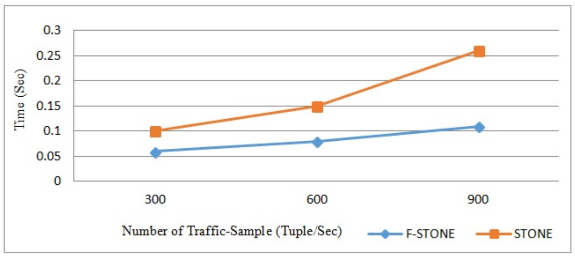

Figure 5. Comparing the EPUT in F-STONE and STONE methods using Traffic-Sample dataset

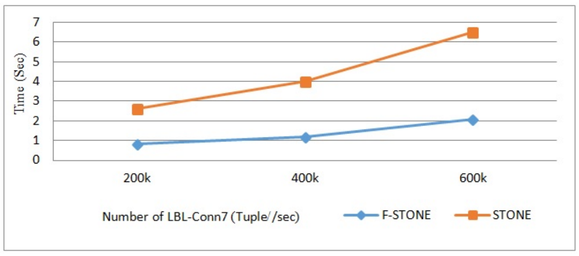

Figure 6. Comparing the EPUT in F-STONE and STONE methods using LBL-CONN7 dataset

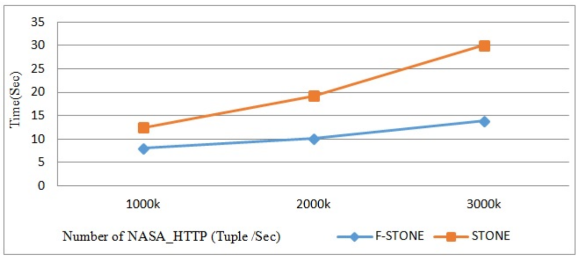

Figure 7. Comparing the EPUT in F-STONE and STONE methods using NASA dataset

havior with the reference index (expected profile) for detecting anomalies, which requires several databases refers and profile updates (refreshing it during each monitoring period). Nevertheless, this duplicate operation is eliminated in F-STONE, and it is expected that the update time for the reference profile information decreases during the monitoring period. Figures 5 to 7 compare the EPUT value in F-STONE and STONE methods, using the Traffic-Sample data, LBL-Conn7, and NASA datasets, respectively.

As shown in Figures 5 to 7, the EPUT values obtained in the experiments with all three selected datasets for F-STONE is less than the STONE method. In addition, by increasing the incoming tuples, the EPUT of the STONE method increases compared to F-STONE. The dependent variable (time) under the influence of the independent variable (tuples) is gradually increasing. The reason for this is an increase in the number of refers to the database, which in F-STONE, the use of BHM is increased instead of increasing the number of referrals to the database, and due to the removal of the waiting time,

Table 4. The results of Sensitivity, False Positive Rate, Specificity, and True Positive Rate for STONE and F-STONE methods

Comparing Factor	STONE	F-STONE
Sensitivity	0.9936	0.9949
False Positive Rate	0.00185	0.00107
Specificity	0.9981	0.9989
True Positive Rate	0.9954	0.9973

updating and refreshing are not time-consuming for each tuple.

In order to evaluate the accuracy of the attack detection, four parameters are calculated as Sensitivity, False Positive Rate, Specificity, and True Positive Rate using dataset KDD-CUP99-10%, and presented in Table 3. The formulas for calculating the above parameters are as follows:

Sensitivity= TP/(TP+FN)

False Positive Rate = FP/(FP+TN)

Specificity = TN/(TN+FP)

True Positive Rate= TP/(TP+FP)

Where, TP represents the number of attacks correctly detected by the system; FN is the number of connections that were attacked, but the system wrongly detected them as normal; FP is the number of connections that were normal, but the system wrongly detected them as attack traffic, and TN is the number of normal connections that are detected correctly as normal by the system.

Moreover, the False Negative Rate for F-STONE and STONE is calculated as FN/ (FN+TP), and the values are obtained as 0.0050 and 0.0063, respectively. The results shown in Table 4 indicate that the detection rate of F-STONE is approximately the same as STONE. The reason is that changing the way the database access in F-STONE does not affect the quality of the attack detection. Keeping the true positive rate and the false negative rate as almost unchanged while reducing the time of attack detection is among the advantages of F-STONE. Figure 8 compares the attack detection time in F-STONE and STONE methods.

As shown in Figure 8, the attack in F-STONE is detected faster than STONE method, which is due to the use of BHM in F-STONE. The results indicate that with the increase of tuples to 400000, the detection time in STONE has increased dramatically because in each monitoring period, it has to get data from the reference database profile. In each database referring, the time is spent as much as three functions of the update, refresh, and persist. On the other hand,

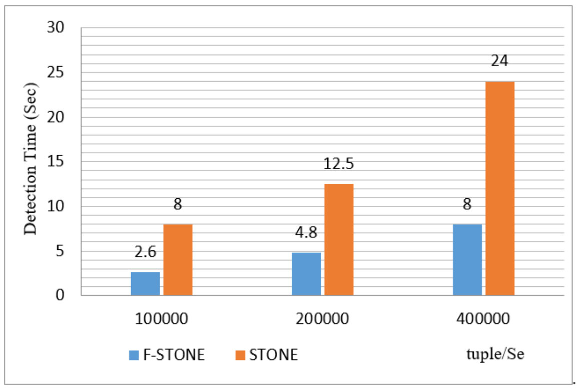

Figure 8. Comparing the attack detection time in F-STONE and STONE methods

in F-STONE, all the information required to detect an attack is available using the BHM data structure, and the storing operations can be done in appropriate times.

5 Conclusion

A DDoS attack aims to deny the legal users getting their required service. In the last decade, DDoS attacks have increased dramatically in terms of the number of attacks, methods, used protocols, and attack platforms. One of the most common DDoS attacks is the SYN bandwidth attack that was dealt with in this research. One of the methods used to detect DDoS bandwidth attacks is the anomaly-based strategy. Anomaly detection systems first create an index of normal network traffic behavior, and then they consider any violation or deviation from the normal index as abnormal and invasive behavior. Real-time attack detection methods, alongside the dataflow-based computational algorithms, aim at processing large amounts of data in high-rate network links as new methods for detecting DDoS flood attacks in recent years. By detecting a real-time attack, service providers can prevent serious damage to the victim and reduce the rate of attack traffic on the path to reach the victim by resorting to an appropriate response mechanism.

In this paper, a method is proposed to accelerate the detection of DDoS bandwidth attacks in real-time. An efficient data structure called BHM is proposed to reduce the exchange of information with the database, and consequently, reduce the time of attack detection. Data in this structure is stored at the minimum space possible so that in each byte, the number of bits can be stored simultaneously. All monitoring information is stored in memory so that there is no need to refer to the database at each monitoring period. In this way, the update and refresh operations are deleted, and at the adjustable times, storing operations are performed in the database.

To evaluate F-STONE, the standard LBL-CONN7, the NASA, and the KDD-CUP99 datasets are used, and the results obtained from the experiments are compared with the similar and recent method STONE [10] in terms of the time required to update the reference profile information (EPUT), attack detection time, and detection rate. The results indicate that reducing the refers to the database, as well as reducing the number of functions of the update, refresh and persist in each monitoring period in F-STONE, lead to a reduction in the time required for updating the reference profile information during the monitoring period, and attack detection time. Furthermore, F-STONE was able to keep the true positive rate and false-negative rate of detections as almost unchanged while reducing the attack detection time.

References

[1] An Wang, Aziz Mohaisen, Wentao Chang, and Songqing Chen. Capturing ddos attack dynamics behind the scenes. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 205–215. Springer, 2015.

[2] Samad S Kolahi, Amro A Alghalbi, Abdulmohsen F Alotaibi, Saarim S Ahmed, and Divyesh Lad. Performance comparison of defense mechanisms against tcp syn flood ddos attack. In *2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 143–147. IEEE, 2014.

[3] Monowar H Bhuyan, Hirak Jyoti Kashyap, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Detecting distributed denial of service attacks: methods, tools and future directions. *The Computer Journal*, 57(4):537–556, 2013.

[4] Martin Roesch et al. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, 1999.

[5] Richa Srivastava and Vineet Richhariya. Survey of current network intrusion detection techniques. *Journal of Information Engineering and Applications*, 3(6):27–33, 2013.

[6] Bayu Adhi Tama and Kyung-Hyune Rhee. Data mining techniques in dos/ddos attack detection: A literature review. *Information (Japan)*, 18(8):3739, 2015.

[7] Amey Kulkarni, Youngok Pino, Matthew French, and Tinoosh Mohsenin. Real-time anomaly detection framework for many-core router through machine-learning techniques. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(1):10, 2016.

[8] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for

cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2015.

[9] P Arun Raj Kumar and S Selvakumar. Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, 34(11):1328–1341, 2011.

[10] Vincenzo Gulisano, Mar Callau-Zori, Zhang Fu, Ricardo Jiménez-Peris, Marina Papatriantafilou, and Marta Patiño-Martínez. Stone: A streaming ddos defense framework. *Expert Systems with Applications*, 42(24):9620–9633, 2015.

[11] Hao Huang and Shiva Prasad Kasiviswanathan. Streaming anomaly detection using randomized matrix sketching. *Proceedings of the VLDB Endowment*, 9(3):192–203, 2015.

[12] Chin-Ling Chen. A new detection method for distributed denial-of-service attack traffic based on statistical test. *J. UCS*, 15(2):488–504, 2009.

[13] Emmanuelle Anceaume and Yann Busnel. A distributed information divergence estimation over data streams. *IEEE Transactions on Parallel and Distributed Systems*, 25(2):478–487, 2013.

[14] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[15] Tomasz Andrysiak and Łukasz Saganowski. Ddos attacks detection by means of statistical models. In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, pages 797–806. Springer, 2016.

[16] Nazrul Hoque, Hirak Kashyap, and DK Bhattacharyya. Real-time ddos attack detection using fpga. *Computer Communications*, 110:48–58, 2017.

[17] Balázs Nagy, Péter Orosz, Tamás Tóthfalusi, László Kovács, and Pál Varga. Detecting ddos attacks within milliseconds by using fpga-based hardware acceleration. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–4. IEEE, 2018.

[18] Sharmila Bista, Roshan Chitrakar, et al. Ddos attack detection using heuristics clustering algorithm and naïve bayes classification. *Journal of Information Security*, 9(01):33, 2017.

[19] M Baskar, T Gnanasekaran, and J Frank Vijay. Time variant predicate based traffic approximation algorithm for efficient low rate ddos attack detection. 2018.

[20] Abigail Koay, Aaron Chen, Ian Welch, and Winston KG Seah. A new multi classifier system using entropy-based features in ddos attack detection. In *2018 International Conference on Information Networking (ICOIN)*, pages 162–167. IEEE, 2018.

[21] Andrey Evgenievich Krasnov, Evgeniy Nikolaevich Nadezhdin, Vladimir Sergeevich Galayev, Evgenia Andreevna Zykova, Dmitrii Nikolaevich NikolâĂŹskii, and Dmitrii Sergeevich Repin. Ddos attack detection based on network traffic phase coordinates analysis. *International Journal of Applied Engineering Research*, 13(8):5647–5654, 2018.

[22] Xinlei Ma and Yonghong Chen. Ddos detection method based on chaos analysis of network traffic entropy. *IEEE Communications Letters*, 18(1):114–117, 2013.

[23] Xubin Zeng, R Eykholt, and RA Pielke. Estimating the lyapunov-exponent spectrum from short time series of low precision. *Physical Review Letters*, 66(25):3229, 1991.

[24] Sunny Behal and Krishan Kumar. Detection of ddos attacks and flash events using novel information theory metrics. *Computer Networks*, 116:96–110, 2017.

[25] Stefano Fortunati, Fulvio Gini, Maria S Greco, Alfonso Farina, Antonio Graziano, and Sofia Giompapa. An improvement of the state-of-the-art covariance-based methods for statistical anomaly detection algorithms. *Signal, Image and Video Processing*, 10(4):687–694, 2016.

[26] Paul L Butzer and François Jongmans. Pl chebyshev (1821–1894) and his contacts with western european scientists. *Historia mathematica*, 16(1):46–68, 1989.

[27] İlker Özçelik and Richard R Brooks. Cusum-entropy: an efficient method for ddos attack detection. In *2016 4th International Istanbul Smart Grid Congress and Fair (ICSG)*, pages 1–5. IEEE, 2016.

[28] lbl dataset. lbl dataset. http://ita.ee.lbl.gov/html/traces.html, 2017. [Online; accessed 2017/23/9].

[29] lbl conn dataset. lbl conn dataset. http://ita.ee.lbl.gov/html/contrib/LBL-CONN-7.html, 2017. [Online; accessed 2017/23/9].

[30] NASA dataset. NASA dataset. http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html, 2017. [Online; accessed 2017/23/9].

[31] KDD-CUP dataset. KDD-CUP dataset. http://www.ll.mit.edu/mission/communications/ist/index.html, 2017. [Online; accessed 2017/23/9].

Mahsa Nooribakhsh Mahsa Nooribakhsh received a B.Sc. degree in computer engineering from Islamic Azad University of Naragh, Iran, in 2010, and an M.Sc. degree in computer engineering from Islamic Azad

University of Buinzahra, Iran, in 2015. She is currently a lecturer and researcher in the Islamic Azad University of Buinzahra, Iran. Her research interests are network security, cloud computing, recommended systems, project management, and data mining.

Mahdi Mollamotalebi Mahdi MollaMotalebi received the B.Sc. degree in computer engineering from Islamic Azad University of Qazvin (QIAU), Iran, in 1999; M.Sc. degree in computer engineering from Islamic Azad University of Arak, Iran, in 2004; and Ph.D. degree in computer science from Universiti Teknologi Malaysia (UTM), Malaysia, in 2013. He is currently a senior lecturer and researcher in the Islamic Azad University of Buinzahra, Iran. His research interests are computer network management, computer security, Internet protocols, Grid resource discovery, Cloud computing, Web search engine, and smart home.