Advance Access publication on 28 March 2013
doi:10.1093/comjnl/bxt031

Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions

MONOWAR H. BHUYAN[1], H. J. KASHYAP[1], D. K. BHATTACHARYYA[1,*] AND J. K. KALITA[2]

[1]*Department of Computer Science and Engineering, Tezpur University, Napaam, Tezpur 784028, Assam, India*
[2]*Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO 80933-7150, USA*
**Corresponding author: dkb@tezu.ernet.in*

Distributed denial of service (DDoS) attack is a coordinated attack, generally performed on a massive scale on the availability of services of a target system or network resources. Owing to the continuous evolution of new attacks and ever-increasing number of vulnerable hosts on the Internet, many DDoS attack detection or prevention mechanisms have been proposed. In this paper, we present a comprehensive survey of DDoS attacks, detection methods and tools used in wired networks. The paper also highlights open issues, research challenges and possible solutions in this area.

Keywords: DDoS; denial of service; agents; handler; network security

Received 19 July 2012; revised 29 December 2012
Handling editor: George Loukas

1. INTRODUCTION

The minimal processing and best-effort forwarding of any packet, malicious or not, was the prime concern when the Internet was designed. This architecture creates an unregulated network path, which can be exploited by any cyber attacker motivated by revenge, prestige, politics or money. Denial-of-service (DoS) attacks exploit this to target critical Web services [1–5]. This type of attack is intended to make a computer resource unavailable to its legitimate users.

DoS attack programs have been around for many years. Old single source attacks are now countered easily by many defense mechanisms and the source of these attacks can be easily rebuffed or shut down with improved tracking capabilities. However, with the astounding growth of the Internet during the last decade, an increasingly large number of vulnerable systems are now available to attackers. Attackers can now employ a large number of these vulnerable hosts to launch an attack instead of using a single server, an approach which is not very effective and detected easily. A distributed DoS (DDoS) attack [1, 6] is a large-scale, coordinated attack on the availability of services of a victim system or network resources, launched indirectly through many compromised computers on the Internet.

The first well-documented DDoS attack appears to have occurred in August 1999, when a DDoS tool called Trinoo was deployed in at least 227 systems, to flood a single University of Minnesota computer, which was knocked down for more than 2 days.[1] The first large-scale DDoS attack took place on February 2000 (see footnote 1). On February 7, Yahoo! was the victim of a DDoS attack during which its Internet portal was inaccessible for 3 h. On February 8, Amazon, Buy.com, cable news network and eBay were all hit by DDoS attacks that caused them to either stop functioning completely or slowed them down significantly (see footnote 1).

DDoS attack networks follow two types of architectures: the agent–handler architecture and the Internet Relay Chat (IRC)-based architecture as discussed by Specht and Lee [7]. The agent–handler architecture for DDoS attacks is composed of clients, handlers and agents. The attacker communicates with the rest of the DDoS attack system at the client systems. The handlers are often software packages located throughout the Internet that are used by the client to communicate with the agents. Instances of the agent software are placed in the

[1] http://www.garykessler.net/library/ddos.html.

compromised systems that finally carry out the attack. The owners and users of the agent systems are generally unaware of the situation. In the IRC-based DDoS attack architecture, an IRC communication channel is used to connect the client(s) to the agents. IRC ports can be used for sending commands to the agents. This makes DDoS command packets more untraceable. Moreover, it is easier for an attacker to hide his presence in an IRC channel as such channels tend to have large volumes of traffic. A recent attacking tool by Anonymous based on the IRC protocol is Low Orbit Ion Cannon (LOIC) [8]. It includes three primary methods of attacks for transmission control protocol (TCP), user datagram protocol (UDP), hypertext transfer protocol (HTTP) and is found in two versions: *binary* and *web-based*. It allows clients to connect remotely via the IRC protocol and to be a part of a system of compromised hosts. The bigger the size of compromised hosts, the more powerful the attack is. Between these two architectures, the agent–handler architecture is commonly found in use in the literature.

Along with the evolution of new DDoS attack tools, many DDoS defense mechanisms have also been proposed. These approaches are of three types depending on their locality of deployment: source-end approach, victim-end approach and in-network approach. Detecting any DDoS attack at the victim end is easy, but often not useful after legitimate clients have been denied access. Source-end detection is a very challenging task. Detection approaches used include statistical, soft computing, clustering, knowledge-based and classifiers. These approaches can also be classified as supervised or unsupervised [9].

Statistical techniques fit a statistical model to the given data and then apply a statistical inference test on an unseen instance to determine if it belongs to this model. In knowledge-based methods, predefined rules or patterns of attack are checked against connection events to test their legitimacy. Soft computing techniques apply problem-solving technologies such as fuzzy logic, probabilistic reasoning, neural networks and genetic algorithms. Clustering is a data mining technique, which is also known as unsupervised classification. It does not need to be trained with a training dataset and the strength of clustering lies within the algorithm itself. Hence, it is very popular. Classifiers such as support vector machine (SVM) and hidden Markov model are also used in many detection approaches. Detailed discussion of these approaches can be found in [9].

In the past few years, several significant surveys have been published [1–3, 6, 7, 10, 11] to highlight the architectures and methods developed for network defense mechanisms, attack taxonomies, attack launching mechanisms and their pros and cons. However, our survey differs significantly from them in the following ways.

(i) We present an attack taxonomy based on [10]. However, in our taxonomy there are seven distinct possibilities in which an intruder can attempt to launch DDoS attacks. Unlike [10], we include a detailed discussion of various DDoS defense mechanisms and their architectures and methods under the broad categories of statistical, knowledge-based, soft computing, data mining and other machine learning methods. We also include a list of practical issues and research challenges, which is not available in [10].

(ii) Like [1], we report and analyze a large number of defense mechanisms, architectures, methods, tools and solutions countering DoS of attacks. However, unlike [1], we include recent defense solutions and tools and discuss latest DDoS attack strategies. Also, unlike [1], we attempt to provide a possible solution to counter the attacks in the context of latest DDoS attack scenarios.

(iii) Unlike [3], our survey is focused on DDoS attack detection methods, tools and research directions. In [3], a major portion is dedicated to DoS research solutions only and that too for a period upto 2009. Also, unlike [3], we present several research issues and open challenges from a more practical point of view, considering the latest DDoS attack scenarios.

(iv) Like [6], we present possible solutions for DDoS attacks in detail and with a practical viewpoint. Also, we broadly categorize the detection methods as statistical, knowledge-based, soft computing, data mining and other machine learning. We also include tools related to DDoS attacks.

(v) Only a brief overview of DDoS taxonomies, tools and countermeasures is given in [7] without any possible solutions for DDoS attacks. In contrast, we present a list of methods, tools, possible solutions and future research directions for DDoS attacks in detail.

(vi) In [2], the authors present several anomaly detection techniques w.r.t. diverse domains but our work is mainly focused on DDoS attack detection architectures, methods and tools.

Our survey begins in Section 2 with the introduction of DDoS attacks and generic architectures of DDoS defense mechanisms classified with their locality of deployment. Section 3 discusses various methods for DDoS attack detection. Different strategies to evaluate the performance of DDoS attack detection methods are described in Section 4. The challenges faced by DDoS defenders are reported in Section 5 followed by concluding remarks in Section 7.

2. DDoS ATTACKS AND THEIR ARCHITECTURES

As stated in [12], a DDoS attack can be defined as an attack which uses a large number of computers to launch a coordinated DoS attack against a single machine or multiple victim machines. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS attack significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms. Approximate attack

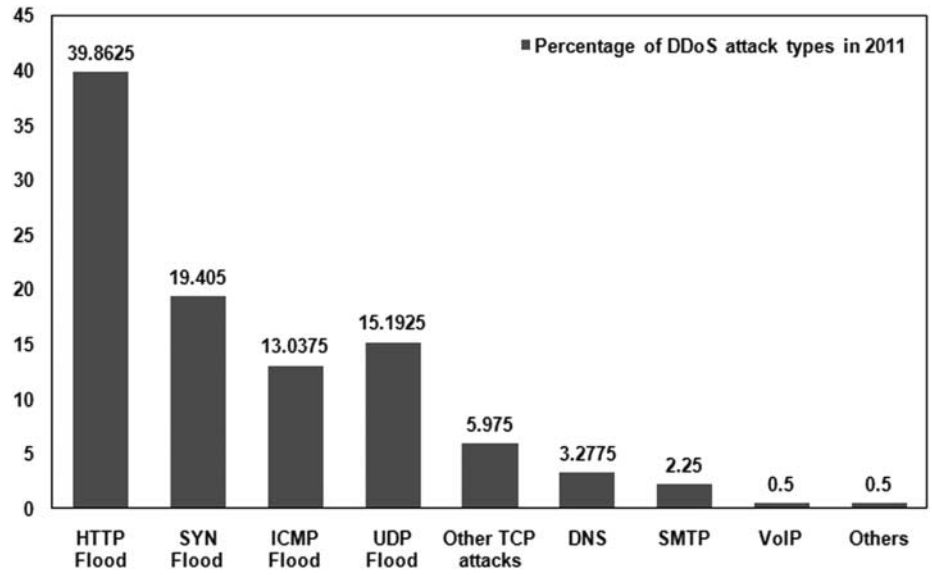

FIGURE 1. DDoS attacks statistics by type [11] for 2011.

statistics for DDoS [11] for the year 2011 are shown in Fig. 1. A DDoS attacker is considered more intelligent than a DoS attacker. It is distinguished from other attacks by its ability to deploy its weapons in a 'distributed' way over the Internet and to aggregate these forces to create lethal traffic. Rather than breaking the victim's defense system for fun or to show prowess, a DDoS attack aims to cause damage on a victim either for personal reasons, material gain or for popularity. A taxonomy of DDoS attacks based on [10] is given in Fig. 2. We see in the taxonomy that intruders attempt to launch DDoS attacks based on exploitation of various means (shown in the left column) and their resultant effects can be observed at various levels or magnitudes.

DDoS attacks mainly take advantage of the architecture of the Internet and this is what makes them powerful. While designing the Internet, the prime concern was to provide for functionality, not security. As a result, many security issues have been raised, which are exploited by attackers. Some of the issues are given below.

(i) Internet security is highly interdependent. No matter how secure a victim's system may be, whether or not this system will become a DDoS victim depends on the rest of the global Internet [13, 14].
(ii) Internet resources are limited. Every Internet host has limited resources that sooner or later can be exhausted by a sufficiently large number of users.
(iii) Many against a few: If the resources of the attackers are greater than the resources of the victims, the success of the attack is almost definite.
(iv) Intelligence and resources are not collocated. Most intelligence needed for service guarantees is located at end hosts. At the same time high bandwidth pathways needed for large throughput are situated in the intermediate network. Such abundant resources present in unwitting parts of the network are exploited by the attacker to launch a successful flooding attack.

By degree of automation	Manual	
	Semiautomatic	Direct
		Indirect
	Automatic	
By exploited vulnerability	Flood attack	Application level flood
		Network level flood
	Amplification attack	Smurf attack
		Fraggle attack
	Protocol exploit attacks	
	Malformed packet attack	
By attack network used	Attacks based on an agent handler network	
	IRC Botnet based	
	Peer to peer network based	
By attack rate dynamics	Continuous	
	Variable	Fluctuating
		Increasing
By victim type	Host	
	Resource	
	Network	
	Application	
By impact	Disruptive	Recoverable
		Phlashing
	Degrading	
By agent set	Constant agent set attacks	
	Variable set attacks	

FIGURE 2. A taxonomy of DDoS attacks [10].

(v) The handlers or the masters, which are compromised hosts with special programs running on them, are capable of controlling multiple agents.
(vi) The attack daemon agents or zombie hosts are compromised hosts that are running a special program each and are responsible for generating a stream of packets toward the intended victim. These machines are commonly external to the victim's own network to disable an efficient response from the victim, and external to the network of the attacker to forswear liability if the attack is traced back.

2.1. DDoS strategy

A DDoS attack is composed of several elements as shown in Figs 3 and 4.

There are several steps in launching a DDoS attack. These are shown in Fig. 5.

(1) *Selection of agents*. The attacker chooses the agents that will perform the attack. Based on the nature of vulnerabilities present, some machines are compromised to be used as agents. Attackers victimize these machines, which have abundant resources, so that a powerful attack stream can be generated. In early years, the attackers attempted to acquire control of these machines manually. However, with the development of advanced security tool(s), it has become easier to identify these machines automatically and instantly.

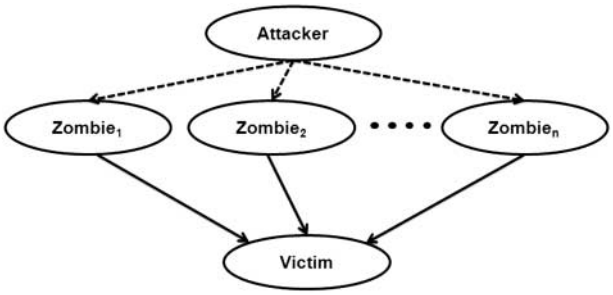

FIGURE 3. *Direct DDoS attack*: send control traffic directly to the zombies to attack the victim host.

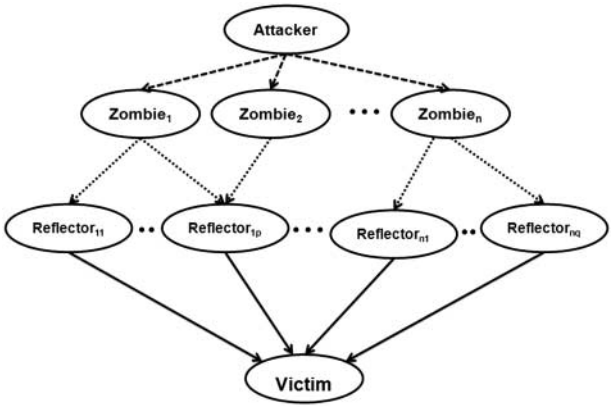

FIGURE 4. *Indirect DDoS attack*: send control traffic indirectly to the zombies to compromise the target host. Reflectors are non-compromised systems that exclusively send replies to a request.

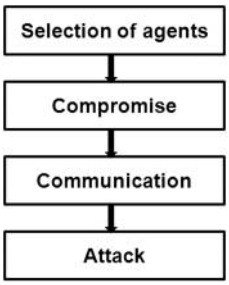

FIGURE 5. Steps to perform a DDoS attack.

(2) *Compromise*. The attacker exploits security holes and vulnerabilities of the agent machines and plants the attack code. Not only that, the attacker also takes necessary steps to protect the planted code from identification and deactivation. As per the direct DDoS attack strategy, shown in Fig. 3, the compromised nodes, i.e. zombies between the attacker and victim are recruited unwitting accomplice hosts from a large number of unprotected hosts connected through the Internet in high bandwidth. On the other hand, the DDoS attack strategy shown in Fig. 4 is more complex due to inclusion of intermediate layer(s) between the zombies and victim(s). It further complicates the traceback mostly due to (i) complexity in untangling the traceback information (partial) with reference to multiple sources, and/or (ii) having to connect a large number of routers or servers. Self-propagating tools such as the Ramen worm [15] and Code Red [16] automate this phase. Unless a sophisticated defense mechanism is used, it is usually difficult for the users and owners of the agent systems to realize that they have become a part of a DDoS attack system. Another important feature of such an agent system is that the agent programs are very cost effective both in terms of memory and bandwidth. Hence, they affect the performance of the system minimally.
(3) *Communication*. The attacker communicates with any number of handlers to identify which agents are up and running, when to schedule attacks or when to upgrade agents. Such communications among the attackers and handlers can be via various protocols, such as internet control message protocol (ICMP), TCP or UDP. Based on the configuration of the attack network, agents can communicate with a single handler or multiple handlers.
(4) *Attack*. The attacker initiates the attack. The victim, the duration of the attack as well as special features of the attack such as the type, length, time-to-live and port numbers can be adjusted. If there are substantial

variations in the properties of attack packets, it is beneficial to the attacker, since it complicates detection.

In the past decade, attackers and agents have started using a multiuser, online chatting system known as IRC [17]. This is because IRC chat networks allow users to create public, private and secret channels. An IRC-based DDoS attack network is similar to the agent–handler DDoS attack model except that instead of using a handler program installed on a network server, an IRC server tracks the addresses of connected agents and handlers and facilitates communication among them. The discovery of a single participant leads to the discovery of the communication channel, but other participants' identities remain protected. Such multiuser online chatting systems or IRC systems have several other significant advantages for launching DDoS attacks. Among the three important benefits are: they afford a high degree of anonymity, they are difficult to detect and they provide a strong, guaranteed delivery system. Furthermore, the attacker no longer needs to maintain a list of agents, since he can simply log on to the IRC server and see a list of all available agents [13]. IRC channels receive communications from the agent software regarding the status of the agents (i.e. up or down) and participate in notifying the attackers regarding the status of the agents. In an IRC-based DDoS attack, the agents are often referred to as 'Zombie Bots' or 'Bots'.

2.2. Generic architecture of DDoS attack defense mechanisms

Based on the locality of deployment, DDoS defense schemes can be divided into three classes: victim-end, source-end and intermediate router defense mechanisms. All of these approaches have their own advantages and disadvantages. We discuss them one by one.

2.2.1. *Victim-end defense mechanism*

Victim-end detection approaches are generally employed in the routers of victim networks, i.e. networks providing critical Web services. A generic architecture of such schemes is shown in Fig. 6. Here the detection engine is used to detect intrusion either online or offline, using either misuse-based intrusion detection or anomaly-based intrusion detection. The reference data stores information about known intrusion signatures or profiles of normal behavior. This information is updated by the processing elements as new knowledge about the observed behavior becomes available. The security manager often updates the stored intrusion signatures and also checks for other critical events such as false alarms. The processing element frequently stores intermediate results in the configuration data.

Detecting DDoS attacks in victim routers is relatively easy because of the high rate of resource consumption. It is also the most practically applicable type of defense scheme as Web servers providing critical services always try to secure their resources for legitimate users. But the problem with these approaches is that, during DDoS attacks, victim resources, e.g. network bandwidth, often gets overwhelmed and these approaches cannot stop the flow beyond victim routers. Another important disadvantage is that these approaches detect the attack only after it reaches the victim and detecting an attack when legitimate clients have already been denied is not useful.

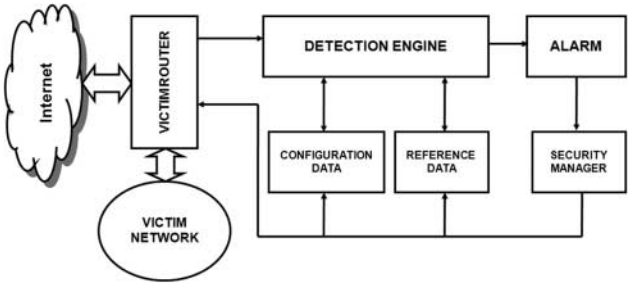

FIGURE 6. Generic architecture for victim-end DDoS defense mechanism.

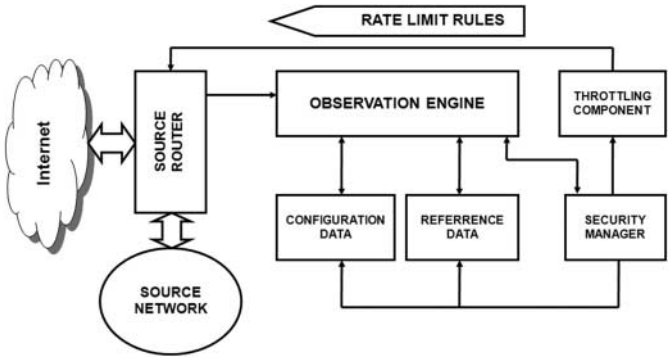

FIGURE 7. Generic architecture for source-end based DDoS defense mechanism.

2.2.2. *Source-end defense mechanism*

A generic architecture of source-end preventive schemes is shown in Fig. 7. This architecture is similar to the victim-end detection architecture. Here, a throttling component is added to impose a rate limit on outgoing connections. The observation engine compares both incoming and outgoing traffic statistics with some predefined normal profiles.

Detecting and stopping a DDoS attack at the source is the best possible defense. It prevents the possibility of flooding not only on the victim side, but also in the whole intermediate network. The main difficulty with this approach is that detecting DDoS attacks at the source end is not easy. This is because in these attacks, sources are widely distributed and a single source behaves almost similarly as in normal traffic. Another problem is the difficulty of deploying system at the source end.

2.2.3. *Intermediate network defense mechanism*

The intermediate network defense scheme balances the trade-offs between detection accuracy and attack bandwidth

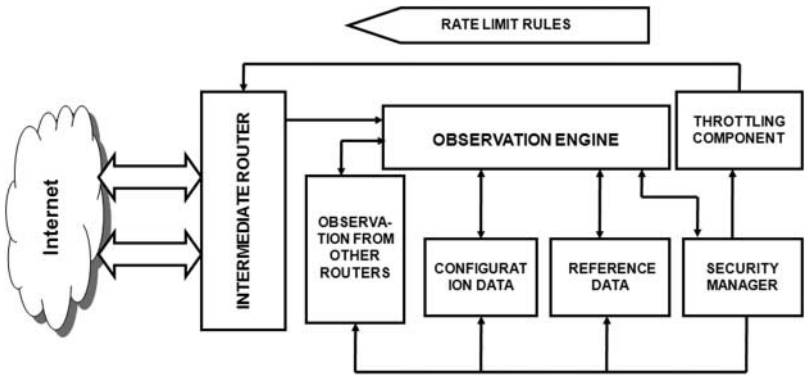

FIGURE 8. Generic architecture for intermediate network-based DDoS defense mechanism.

consumption, the main issues in source-end and victim-end detection approaches. Figure 8 shows a generic architecture of the intermediate network defense scheme, one that can be employed in any network router. Such a scheme is generally collaborative in nature and the routers share their observations with other routers. Like a source-end scheme, these schemes also impose rate limits on connections passing by the router after comparing with stored normal profiles.

Detection and traceback of attack sources are easy in this approach due to collaborative operation. Routers can form an overlay mesh to share their observations [18]. The main difficulty with this approach is deployability. To achieve full detection accuracy, all routers on the Internet will have to employ this detection scheme, because unavailability of this scheme in only a few routers may cause failure of the detection and traceback process. Obviously, full practical implementation of this scheme is extremely difficult by reconfiguring all the routers on the Internet.

3. EXISTING METHODS FOR DDoS ATTACK DETECTION

In this section, we present a summary of existing literature on DDoS attack detection methods. These methods are based on the architectures discussed above, namely, victim-end, source-end and in-network. We discuss these schemes without considering their practical deployability in real networks. Recent trends show that soft computing approaches have been used heavily for DDoS attack detection. Ensembles of classifiers have also performed satisfactorily with high detection rates (DRs). We classify methods for DDoS attack detection into four major classes as shown in Fig. 9.

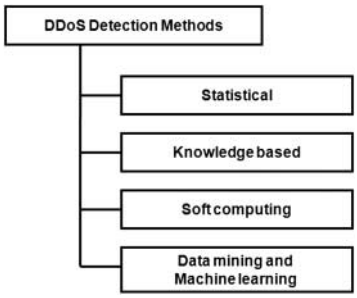

FIGURE 9. Classification of DDoS attack detection methods.

3.1. Statistical methods

Statistical properties of normal and attack patterns can be exploited for detection of DDoS attacks. Generally, a statistical model for normal traffic is fitted and then a statistical inference test is applied to determine if a new instance belongs to this model. Instances that do not conform to the learnt model, based on the applied test statistics, are classified as anomalies. Chen *et al.* [19] develop a distributed change point (DCP) detection architecture using change aggregation trees (CATs). The non-parametric cumulative sum (CUSUM) approach was adapted to describe the distribution of prechange or postchange network traffic. When a DDoS flooding attack is being launched, the cumulative deviation is noticeably higher than random fluctuations. The CAT mechanism is designed to work at the router level to detect abrupt changes in traffic flows. The domain server uses the traffic change patterns detected at attack-transit routers to construct the CATs, which represent the attack flow pattern. A very well-known DDoS defense scheme called D-WARD is presented in [20]. D-WARD identifies an attack based on continuous monitoring of bidirectional traffic flows between the network and the rest of the Internet and by periodic deviation analysis with the normal flow patterns. Mismatched flows are rate limited in proportion to their aggressiveness. D-WARD not only offers a good DR but also reduces DDoS attack traffic significantly. It uses a predefined model for normal traffic to detect anomalies in the two-way traffic statistics for each peer. If it identifies a DDoS attack, it imposes a rate limit on the suspicious outgoing flow for the peer. Next, D-WARD observes the traffic for either confirmation of the attack or refutation. If confirmed, D-WARD further controls the rate limit. However, if

refuted, it gradually allows increased traffic rate. Saifullah [21] proposes a defense mechanism based on a distributed algorithm that performs weight-fair throttling at upstream routers. The throttling is weight-fair because the traffic destined for the server is controlled (increased or decreased) by leaky buckets at the routers based on the number of users connected, directly or through other routers, to each router. In the beginning of the algorithm, the survival capacity is underestimated by the routers so as to protect the server from any sudden initial attack. The rate is updated (increased or decreased), based on the server's feedback sent to its child routers and eventually propagated downward to all routers, in the subsequent rounds of the algorithm with a view to converging the total server load to the tolerable capacity range.

Chen [22] presents a new detection method for DDoS attack traffic based on the two-sample t-test. It first obtains statistics for the normal SYN arrival rate (SAR) and confirms that it follows the normal distribution. The method identifies an attack by computing (a) the difference between incoming SAR and normal SAR and (b) the difference between the number of SYN and ACK packets. Unlike most previous DDoS defense schemes that only deal with either flooding or meek attack, the proposal uses two statistical tests to identify malicious traffic. It first compares the differences between the overall means of the incoming traffic arrival rate and the normal traffic arrival rate by the two-sample *t-test*. If the difference is significant, it concludes that the traffic may include flooding attack packets. However, the low-rate attack traffic may pass the arrival rate test and make the backlog queue full. The approach then compares the two groups that contain different numbers of SYN and ACK packets by the two-sample *t-test*. If there is a significant difference, it recognizes that the attack traffic is mixed into the current traffic. Zhang *et al.* [23] propose a prediction method for the available service rate of a protected server by applying the Auto Regressive Integrated Moving Average (ARIMA) model. They use available service rates to qualify the server's availability to detect DDoS attacks. Their prediction method divides server resources into CPU time, memory utilization and networking buffer. Based on the prediction, they use abnormal detection technology to analyze the consumption of server resources to predict whether the server is under a DDoS attack.

Akella *et al.* [24] explore key challenges in helping an Internet service provider (ISP) network detect attacks on itself or attacks on external sites which use the ISP network. They propose a detection mechanism where each router detects traffic anomalies using profiles of normal traffic constructed using stream sampling algorithms. Initial results show that it is possible to: (1) profile normal traffic reasonably accurately; (2) identify anomalies with low false positive and false negative rates (locally, at the router) and (3) be cost effective in terms of memory consumption and per packet computation. In addition, ISP routers exchange information with each other to increase confidence in their detection decisions. A router gathers responses from all other routers regarding suspicions and based on them decides whether a traffic aggregate is an attack or is normal. The initial results show that individual router profiles capture key characteristics of the traffic effectively and identify anomalies with low false positive and false negative rates. Peng *et al.* [25] describe a novel approach to detect bandwidth attacks by monitoring the arrival rate of new source internet protocol (IP) addresses. The detection scheme is based on an advanced non-parametric change detection scheme, CUSUM. Cheng *et al.* [26] propose the IP Flow Feature Value algorithm based on the essential features of DDoS attacks, such as abrupt traffic change, flow dissymmetry, distributed source IP addresses and concentrated target IP addresses. Using a linear prediction technique, a simple and efficient auto regressive moving average prediction model is established for normal network flow. Then a DDoS attack detection scheme based on anomaly detection techniques and a linear prediction model (DDAP) is used. Udhayan and Hamsapriya [27] present a statistical segregation method (SSM), which samples the flow in consecutive intervals and compares the samples against the attack state condition and sorts them with the mean as the parameter. Then correlation analysis is performed to segregate attack flows from legitimate flows. The authors compare the SSM against various other methods and identify a blend of segregation methods for alleviating false detections.

In [28], the authors introduce a generic DoS detection scheme based on maximum likelihood criterion with random neural networks (RNNs). The method initially selects a set of traffic features in the offline mode to obtain pdf estimates and to evaluate the likelihood ratios. During decision making, it measures the features of incoming traffic and attempts to decide according to each feature. Finally, it obtains an overall decision using both feedforward and recurrent architectures of the RNN. A brief summary of these methods is given in Table 1.

3.2. Soft computing methods

Learning paradigms, such as neural networks, radial basis functions (RBFs) and genetic algorithms, are increasingly used in DDoS attack detection because of their ability to classify intelligently and automatically. Soft computing is a general term for describing a set of optimization and processing techniques that are tolerant of imprecision and uncertainty. Jalili *et al.* [29] introduce a DDoS attack detection system called SPUNNID based on a statistical preprocessor and unsupervised artificial neural nets (ANNs). They use statistical preprocessing to extract features from the traffic, and an unsupervised neural net to analyze and classify traffic patterns as either a DDoS attack or normal.

Karimazad and Faraahi [30] propose an anomaly-based DDoS detection method based on features of attack packets, analyzing them using RBF neural networks. The method can be applied to edge routers of victim networks. Vectors with seven features are used to activate an RBF neural network at each time window. The RBF neural network is applied to

TABLE 1. Statistical DDoS attack detection methods.

Reference	Objective	Deployment	Mode of working	Trainable	Remarks
Mirkoviac *et al.* [20]	Attack prevention	Source side	Centralized	Yes	Detects DDoS attacks at the source end autonomously and stops attacks from the source network using statistical traffic modeling
Akella *et al.* [24]	Attack detection	Source and victim side	Distributed	No	Detects traffic anomalies in router using stream sampling algorithms based on profiles constructed from normal traffic
Peng *et al.* [25]	Detecting bandwidth attacks	Victim side	Centralized	Yes	Uses sequential non-parametric change point detection method to improve the accuracy
Chen *et al.* [19]	Attack detection and attack source identification	Between source and destination network	Distributed	No	Automatically performs traceback during the detection of suspicious traffic flows
Oke and Loukas [28]	Attack detection	Victim side	Centralized	Yes	Employs a wide variety of attack-specific input features that capture both the instantaneous behavior and the long-term statistical properties of the traffic during detection
Saifullah [21]	Attack prevention	Between source and destination network	Distributed	No	Protects Internet server from DDoS attacks using distributed weight-fair throttling at the upstream routers
Chen [22]	Attack detection	Victim side	Centralized	Yes	Detects DDoS attacks based on two-sample t-test by incorporating the statistics of the SAR
Zhang *et al.* [23]	Attack detection	Victim side	Centralized	Yes	Uses an ARIMA model for protecting servers from DDoS attacks
Cheng *et al.* [26]	Attack detection	Victim side	Centralized	Yes	Exploits four flow features: burst in the traffic volume, asymmetry of the flow, distributed source IP addresses and concentrated destination IP address while detecting DDoS attacks
Udhayan and Hamsapriya [27]	Minimize false alarm	Victim side	Centralized	Yes	Uses an SSM for detecting DDoS attacks based on sampling of flow in consecutive time interval

classify data to normal and attack categories. If the incoming traffic is recognized as attack traffic, the source IP addresses of the attack packets are sent to the Filtering Module and the Attack Alarm Module for further actions. Otherwise, if the traffic is normal, it is sent to the destination. RBF neural network training can be performed as an offline process but it is used in real time to detect attacks faster. Gavrilis and Dermatas [31] also present a detector for DDoS attacks in public networks based on statistical features estimated in short-time window analysis of incoming data packets. A small number of statistical descriptors are used to describe the behavior of the DDoS attacks. An accurate classification is achieved using RBF neural

networks. Wu *et al.* [32] propose to detect DDoS attacks using decision trees and gray relational analysis. The detection of the attack from the normal situation is viewed as a classification problem. They use 15 attributes, which not only monitor the incoming/outgoing packet/byte rate, but also compile the TCP, SYN and ACK flag rates, to describe the traffic flow pattern. The decision tree technique is applied to develop a classifier to detect abnormal traffic flow. They also use a novel traffic pattern matching procedure to identify traffic flow similar to the attack flow and to trace back the origin of an attack based on this similarity. Nguyen and Choi [33] develop a method for proactive detection of DDoS attacks by classifying the network status. They break a DDoS attack into phases and select features based on an investigation of DDoS attacks. Finally, they apply the k-nearest neighbor (KNN) method to classify the network status in each phase of the DDoS attack. A method presented in [34] detects DDoS attacks based on a fuzzy estimator using mean packet inter-arrival times. It detects the suspected host and traces the IP address to drop packets within 3-s detection windows.

Lately, ensembles of classifiers have been used for DDoS attack detection. The use of an ensemble reduces the bias of existing individual classifiers. An ensemble of classifiers has been used by Kumar and Selvakumar [35] for this purpose where a resilient back propagation (RBP) neural network is chosen as the base classifier. The main focus of this paper is to improve the performance of the base classifier. The proposed classification algorithm, RBPBoost combines the output of the ensemble of classifier outputs and Neyman Pearson cost minimization strategy [36], for final classification decision. Table 2 presents a brief summary of the soft computing methods presented in this section.

3.3. Knowledge-based methods

In knowledge-based approaches, network events are checked against predefined rules or patterns of attack. In these approaches, general representations of known attacks are formulated to identify actual occurrences of attacks. Examples of knowledge-based approaches include expert systems, signature analysis, self-organizing maps and state transition analysis. Gil and Poletto [37] introduce a heuristic along with a data structure called MUltiLevel Tree for Online Packet Statistics (MULTOPS) that monitors certain traffic characteristics which can be used by network devices such as routers to detect and eliminate DDoS attacks. MULTOPS is a tree of nodes that contains packet rate statistics for subnet prefixes at different aggregation levels. Expansion and contraction of the tree occurs within a prespecified memory size. A network device using MULTOPS detects ongoing bandwidth attacks by the presence of a significant and disproportional difference between packet rates going to and coming from the victim or the attacker. Depending on their setup and their location on the network, MULTOPS-equipped routers or network monitors may fail to detect a bandwidth attack that is mounted by attackers that randomizes IP source addresses on malicious packets. MULTOPS fails to detect attacks that deploy a large number of proportional flows to cripple a victim.

TABLE 2. Soft computing-based DDoS attack detection methods.

Reference	Objective	Deployment	Mode of working	Trainable	Remarks
Jalili *et al.* [29]	Attack detection	Victim side	Centralized	Yes	Uses statistical preprocessor and unsupervised neural network classifier for DDoS attack detection
Gavrilis and Dermatas [31]	Attack detection	Victim side	Centralized	Yes	Detects DDoS attacks based on statistical features estimated in short time intervals in public network using RBF neural network
Nguyen and Choi [33]	Attack detection	Intermediate network	Centralized	Yes	Detects only known attacks using the k-nearest neighbor based technique
Wu *et al.* [32]	Attack detection and traceback	Victim side	Distributed	Yes	Uses a decision tree and traceback to the attacker location using traffic flow pattern matching
Karimazad and Faraahi [30]	Attack detection	Victim side	Centralized	Yes	Uses RBF neural networks and gets a low FAR
Kumar and Selvakumar [35]	Attack detection	Victim side	Centralized	Yes	RBPBoost combines an ensemble of classifier outputs and Neyman Pearson cost minimization strategy for a final classification decision during DDoS attack detection and gets a high DR

对合法列表中的客户提供服务，对非合法列表中的客户进行验证测试，通过测试的客户加入合法列表中。

Thomas *et al.* [38] present an approach to DDoS defense called NetBouncer and claim it to be a practical approach with high performance. Their approach relies on distinguishing legitimate and illegitimate use and ensuring that resources are made available only for legitimate use. NetBouncer allows traffic to flow with reference to a long list of proven legitimate clients. If packets are received from a client (source) not on the legitimate list, a NetBouncer device proceeds to administer a variety of legitimacy tests to challenge the client to prove its legitimacy. If a client can pass these tests, it is added to the legitimacy list and subsequent packets from the client are accepted until a certain legitimacy window expires.

Wang *et al.* [39] present a formal and methodical way of modeling DDoS attacks using an augmented attack tree (AAT), and discuss an AAT-based attack detection algorithm. This model explicitly captures the particular subtle incidents triggered by a DDoS attack and the corresponding state transitions from the view of the network traffic transmission on the primary victim server. Two major contributions of this paper are: (1) an AAT-based DDoS model (ADDoSAT), developed to assess the potential threat from malicious packets on the primary victim server and to facilitate the detection of such attacks; (2) an AAT-based bottom-up detection algorithm proposed to detect all kinds of attacks based on AAT modeling. Compared with the conventional attack tree modeling method, AAT is advanced because it provides additional information, especially about the state transition process. As a result, it overcomes the shortcomings of CAT modeling. There is currently no established AAT-based bottom-up procedure for detecting network intrusions. Limwiwatkul and Rungsawang [40] propose to discover DDoS attack signatures by analyzing the TCP/IP packet header against well-defined rules and conditions, and distinguishing the difference between normal and abnormal traffic. The authors mainly focus on ICMP, TCP and UDP flooding attacks.

Zhang and Parashar [41] propose a distributed approach to defend against DDoS attacks by coordinating across the Internet. Unlike traditional IDS, it detects and stops DDoS attacks within the intermediate network. In the proposed approach, DDoS defence systems are deployed in the network to detect DDoS attacks independently. A gossip-based communication mechanism is used to exchange information about network attacks between these independent detection nodes to aggregate information about the overall network attacks. Using the aggregated information, individual defence nodes obtain approximate information about global network attacks and can stop them more effectively and accurately. For faster and reliable dissemination of attack information, the network grows as a peer-to-peer overlay network on top of the Internet. Previously proposed approaches rely on monitoring the volume of traffic that is received by the victim. Most such approaches are incapable of differentiating a DDoS attack from a flash crowd. Lu *et al.* [42] describe a perimeter-based anti-DDoS system, in which the traffic is analyzed only at the edge routers of an ISP network. The anti-DDoS system consists of two major components: (1) temporal correlation-based feature extraction and (2) spatial correlation-based detection. The scheme can accurately detect DDoS attacks and identify attack packets without modifying existing IP forwarding mechanisms at routers. A brief summary of these knowledge-based methods is given in Table 3.

3.4. Other data mining and machine learning methods

An effective defense system to protect network servers, network routers and client hosts from becoming handlers, zombies and victims of DDoS flood attacks is presented in [43]. The NetShield system protects any IP-based public network on the Internet. It uses preventive and deterrent controls to remove system vulnerabilities on target machines. Adaptation techniques are used to launch protocol anomaly detection and provide corrective intrusion responses. The NetShield system enforces dynamic security policies. NetShield is especially tailored for protecting network resources against DDoS flood attacks. Chen *et al.* [44] present a comprehensive framework for DDoS attack detection known as DDoS Container. It uses a network-based detection method to overcome complex and evasive types of DDoS attacks. It works in an *inline* mode to inspect and manipulate ongoing traffic in real time. By continuous monitoring of both DDoS attacks and legitimate applications, DDoS Container covers stateful inspection on data streams and correlates events among different sessions. It proactively terminates the session when it detects an attack. Lee *et al.* [45] propose a method for proactive detection of DDoS attacks by exploiting an architecture consisting of a selection of handlers and agents that communicate, compromise and attack. The method performs cluster analysis. The authors experiment with the DARPA 2000 Intrusion Detection Scenario Specific Dataset to evaluate the method. The results show that each phase of the attack scenario is partitioned well and can detect precursors of a DDoS attack as well as the attack itself. Sekar *et al.* [46] investigate the design space for in-network DDoS detection and propose a triggered, multistage approach that addresses both scalability and accuracy. Their contribution is the design and implementation of large-scale automated DDoS detection System. The system makes effective use of the data (such as NetFlow and simple network management protocol feeds from routers) readily available to an ISP. Rahmani *et al.* [47] discuss a joint entropy analysis of multiple traffic distributions for DDoS attack detection. They observe that the time series of IP-flow numbers and aggregate traffic sizes are strongly statistically dependent. The occurrence of an attack affects this dependence and causes a rupture in the time series for joint entropy values. Experimental results show that this method could lead to more accurate and effective DDoS detection.

A low-rate DDoS (LDDoS) attack has significant ability to conceal its traffic because of its similarity with normal traffic. Xiang *et al.* [48] propose two new information metrics: (i) generalized entropy metric and (ii) information distance metric, to detect LDDoS attacks. They identify the attack by measuring

TABLE 3. Knowledge-based DDoS attack detection methods.

Reference	Objective	Deployment	Mode of working	Trainable	Remarks
Gil and Poletto [37]	Attack prevention	Between source and destination network	Centralized	No	Each network devices maintains a data structure known as MULTOPS. Fails to detect attacks that deploy a large number of DDoS attack flows using a large number of agents, IP spoofing attacks
Thomas *et al.* [38]	Attack detection	Victim side	Centralized	No	NetBouncer differentiate DDoS traffic from a flash crowd using inline packet processing based on network processor technology
Limwiwatkul and Rungsawang [40]	Attack detection	Victim side	Distributed	Yes	Attack signature model is built using TCP packet header information for DDoS attack detection
Zhang and Parashar [41]	Proactive	Intermediate network	Distributed	Yes	A gossip-based scheme uses to get global information about DDoS attacks by information sharing
Lu *et al.* [42]	Attack detection	Edge router	Distributed	Yes	Exploits spatial and temporal correlation of DDoS attack traffic records for detecting anomalous packets
Wang *et al.* [39]	Attack detection	Victim side	Centralized	No	Uses an AAT model for the detection of DDoS attacks and also can detect other attacks

the distance between legitimate traffic and attack traffic. The generalized entropy metric is more effective than the traditional Shannon metric [49]. In addition, the information distance metric outperforms the popular Kullback–Leibler divergence approach. Francois *et al.* [50] present a method called *FireCol* based on information theory for early detection of flooding DDoS attacks. FireCol is composed of an intrusion prevention system (IPS) located at the ISP level. The IPSs form virtual protection rings around the hosts to defend and collaborate by exchanging selected traffic information. The approach reported in [51] analyzes DDoS and flash crowd characteristics and provides an effective way to distinguish between the two in VoIP networks. The authors validate the method by simulation. A wavelet transformation and probability theory-based network anomaly detection approach is proposed in [52]. The approach is able to identify known as well as unknown attacks. Zhong and Yue [53] present a DDoS attack detection model that extracts a network traffic model and a network packet protocol status model and sets the threshold for the detection model. Captured network traffic values are clustered based on the k-means clustering algorithm to build initial threshold values for network traffic. All captured packets are used to build the packet protocol status model using the Apriori [54] and fuzzy c-means (FCM) [55] algorithms. Whenever the current network traffic is over the threshold value, the network packet protocol status is checked to detect abnormal packets. If there are no abnormal packets, the current network traffic is clustered again by the k-means module to build a new threshold value model.

A two-stage automated system is proposed in [56] to detect DoS attacks in network traffic. It combines the traditional change point detection approach with a novel one based on continuous wavelet transforms [57]. The authors test the system using a set of publicly available attack-free traffic traces superimposed with anomaly profiles. In [58], Li and Lee present a systematic wavelet-based method for DDoS attack detection. They use energy distribution based on wavelet analysis to detect DDoS attack traffic. Energy distribution over time has limited variation if the traffic keeps its behavior over time. Gupta *et al.* [59] use ANN to estimate the number of zombies in a DDoS attack. They use sample data to train a feed-forward neural network generated using the NS-2 network simulator. The generalization capacity of the trained network is promising and the network is able to predict the number of zoombies involved in a DDoS attack with test error. A port-to-port specific traffic in a router, called IF flow is introduced in [60]. An important feature of IF is that it can amplify the attack to normal traffic ratio. An recursive least square filter is used to predict IF flows. Next, a statistical method using a residual filtered process

is used to detect anomalies. Finally, the authors applied the method to three types of traffic: IF flows, input links and output links within a router, and compare the anomaly detection results using receiver operating characteristics curves. Results show that IF flows are more powerful than input links and output links for DDoS attack detection. Cheng *et al.* [61] propose the IP Address Interaction (IAI) Feature algorithm considering interactions among addresses, abrupt traffic changes, many-to-one asymmetries among addresses, distributed source IP addresses and concentrated target addresses [61]. The IAI algorithm is designed to describe the essential characteristics of network flow states. Furthermore, an SVM classifier, which is trained by an IAI time series from normal flow and attack flow, is applied to classify the state of current network flows and identify the DDoS attacks. Experimental results show that the IAI-based detection scheme can distinguish between normal flows and abnormal flows with DDoS attacks effectively, and help identify fast and accurate attack flows when the attacking traffic is hidden among a relatively large volume of normal flows or close to the attacking sources. In addition, it has higher detection and lower false alarm rates (FARs) compared with competing techniques.

The method presented in [62] can identify flooding attacks in real time and also can assess the intensity of the attackers based on fuzzy reasoning. The process consists of two stages: (i) statistical analysis of the network traffic time series using discrete wavelet transform and Schwarz information criterion to find the change point of the Hurst parameters resulting from a DDoS flood attack, and then (ii) identification and assessment of the intensity of the DDoS attack adaptively based on an intelligent fuzzy reasoning mechanism. Test results by ns2-based simulation with various network traffic characteristics and attack intensities demonstrate that the method could detect DDoS flood attack timely, effectively and intelligently. Zhang et al. [63] present a Congestion Participation Rate (CPR)-based approach to detect LDDoS attacks using flow level network traffic. A flow with higher CPR value leads to LDDoS and consequent dropping of the packets. The authors evaluate the mechanism using ns2 simulation, testbed experiments and Internet traffic trace and claims so that the method can detect LDDoS flows effectively. Another protocol specific feature-based DDoS attack detection mechanism is introduced in [5]. It identifies a most relevant subset of features using correlation and can detect DDoS attacks with high detection accuracy.

In [64], a mathematical model is presented to provide gross evaluation of the benefits of DDoS defence based on dropping of attack traffic. Simulation results and testbed experiments are used to validate the model. In the same work, the authors also consider an autonomic defence mechanism based on the cognitive packet network (CPN) protocol and establish it to be capable of tracing back flows coming into a node automatically. Ghanea-Hercock *et al.* [65] provide a survey of the techniques within the Hyperion project. They also suggest an overall system architecture to improve the situational awareness of field commanders by providing an option to fuse and compose information services in real time. In [66], Gelenbe describes an approach to develop self-aware networks to provide end users the option to explore the state of the network to find the best ways to meet their communication needs. In [67], a model is introduced for searching by N agents in an unbounded random environment. The model allows for the loss or destruction of searchers and finite lifetime. A summarized presentation of these methods in this category is given in Table 4.

3.5. Discussion

Exact comparison of DDoS attack detection schemes is not feasible because some papers do not specify their results clearly, whereas others evaluate their schemes using different datasets or in different testing conditions. A comparison (as shown in Table 5) establishes that most papers do not consider all the issues that are pertinent. For example, the DCP detection method [19] performs well for TCP SYN attacks, but its performance degrades for UDP attacks with large packet sizes. D-WARD [20] fails to detect pulsing attacks, especially when the inactive period is large. For NetBouncer [38], the legitimacy tests may not be exhaustive and certain illegitimate clients may also pass the test. In addition, Netbouncer is overwhelmed by flash crowds. Moreover, the delay introduced by the test affects new legitimate clients. The DDoS detection system based on K-means and FCM clustering [53] performs well for unknown attacks, but the trade-off between detection accuracy and speed is high. Detection using RBF neural networks and statistical features [31] performs well for known attacks, but no dynamic modification can be performed easily for unknown attacks.

4. PERFORMANCE EVALUATION

Performance evaluation is important for any DDoS attack defense system. Performance evaluation is highly dependent on (i) the approach, (ii) deployment status and (iii) whether there is facility for dynamic updation of profiles. While designing a DDoS attack defense scheme, these parameters should be taken into consideration. In this section, we primarily discuss the datasets that have been used for evaluating performance of detection methods. We also briefly introduce DDoS tools.

4.1. DDoS tools

There are many tools available to launch DDoS attacks in the literature [44, 68]. The architectures are almost always the same. Some are made by the attackers with slightly modifying others. Table 6 presents some of the tools with brief descriptions.

4.2. Datasets

After a new or enhanced detection mechanism is developed, it needs to be validated with proper datasets before deployment in

TABLE 4. Data mining and machine learning-based DDoS attack detection methods.

Reference	Objective	Deployment	Mode of working	Trainable	Remarks
Hwang *et al.* [43]	Attack prevention	Victim side	Centralized	Yes	Protects network servers, routers and clients from DDoS attacks using the protocol anomaly detection technique
Li and Lee [58]	Attack detection	Victim end	Centralized	No	An energy distribution-based wavelet analysis technique for the detection of DDoS traffic
Sekar *et al.* [46]	Attack detection	Source side	Distributed	Yes	A triggered multistage approach for both scalability and accuracy for DDoS attack detection
Gelenbe and Loukas [64]	Attack defense using packet dropping	Victim end	Centralized	Yes	Detects an attack by tracing back flows automatically
Lee *et al.* [45]	Attack detection	Source side	Centralized	Yes	Detects a DDoS attack proactively based on cluster analysis with agent–handler architecture
Rahmani *et al.* [47]	Attack detection	Victim side	Distributed	No	A joint entropy analysis of multiple traffic distributions for DDoS attack detection
Li and Li [52]	Detection of DDoS attacks automatically	Victim end	Centralized	No	A DDoS attack detection model based on wavelet transformation and probability theory
Dainotti *et al.* [56]	Detection of DoS attack anomalies	Victim end	Centralized	Yes	Detects attacks correctly using a combination of traditional change point detection and continuous wavelet transformation
Zhong and Yue [53]	Attack detection	Victim side	Centralized	Yes	Uses fuzzy c-means clustering and Apriori techniques to build a model and detect unknown DDoS attacks
Xia *et al.* [62]	Detects a flood attack and its intensity	Victim end	Centralized	Yes	A method to detect a DDoS flooding attack using fuzzy logic
Xiang *et al.* [48]	Detects low rate flooding attacks	Victim end	Centralized	No	Detects LDDoS flooding attacks using new information metrics effectively
Gupta *et al.* [59]	Number of zombies identification	Victim end	Distributed	Yes	Uses an ANN to estimate the number of zombies in a DDoS attack
Francois *et al.* [50]	DDoS flooding attack detection	Source end	Distributed	No	A complete DDoS flooding attack detection technique. Also supports incremental deployment in real network
Jeyanthi and Iyengar [51]	Distinguishing DDoS attacks from Flash crowds	Victim end	Centralized	No	Detects DDoS attacks using entropy-based analysis

a real life network. This is because evaluation in real traffic is extremely difficult or not possible because of the non-availability of capturing tools from a high-speed network. The proper choice of network intrusion datasets plays a vital role during evaluation of any computer attack detection method. First, the dataset should contain correctly captured real network traffic. Secondly, it should be unbiased. Only datasets fulfilling these two requirements should be considered for evaluation of

TABLE 5. A general comparison of DDoS attack detection methods.

Scheme	Approach	Architecture/ method	R/N	Scalability	Unknown attack detection	Dynamic signature updation	DR with dataset used	FAR with dataset used
DCD approach [19]	Statistical	DCP/CAT	R	Yes	Yes	No	98% (flooding attacks)	<1% (flooding attacks)
Weighted fair model [21]	Statistical	Weight fair throttling	R	Yes	Yes	No		
SPUNNID model [29]	Statistical	SPUNNID system/unsupervised neural network	R	Yes	Yes	Yes	94.9% (flooding attacks)	5% (flooding attacks)
D-WARD system [20]	Knowledge-based	Self regulating reverse feedback system/rate limited	R	No	Yes	No	(flooding attacks)	0.5% (flooding attacks)
MULTOPS system [37]	Knowledge-based	Multilevel tree-based method	N	Yes	Yes	No	(IP spoofing attacks)	
NetBouncer model [38]	Knowledge-based	Packet filtering method	R	Yes	Yes	No	(real-time data)	
Decision tree model [32]	Knowledge-based	Decision tree-based method	R	Yes	No	No	98% (flooding attacks)	2.4% (flooding attacks)
Clustering model [53]	Statistical	Data mining-based method	N	Yes	No	No	98.65% (real-time)	1.12% (real-time)
RBF neural net model [30]	Soft computing	RBF system/RBF	R	No	No	No	98.2% (UCLA data)	0.01% (UCLA data)
RBF neural net model [31]	Soft computing	RBF-based method	R	Yes	No	No	98–100% (real-time)	0% (real-time)
T-test model [22]	Statistical	*t*-test-based method	R	No	No	No	98–100% (flooding attacks)	5–7% (flooding attacks)
ARIMA-based model [23]	Statistical	Prediction-based method	R	No	No	Yes	(ns2 simulation)	1–3% (ns2 simulation)
Attack tree model [39]	Knowledge-based	AATBD system/Tree-based	R	No	No	No	(flooding attacks)	
Signature discovery approach [40]	Knowledge-based	Traffic statistics-based method	R	No	No	No	(flooding attacks)	
Profile-based approach [24]	Statistical	Stream sampling-based method	R	No	No	No	(IP spoofing attacks)	2% (IP spoofing attacks)
Cooperative model [41]	Knowledge-based	RL-DDoS system/Gossip-based scheme	R	No	No	No	(Emulab simulation)	7–12% (Emulab simulation)
Sequential nonparametric change point method [25]	Knowledge-based	Non-parametric CPD method	R	Yes	Yes	No	90–100% (Aucland traces)	
Perimeter-based anti-DDoS system [42]	Knowledge-based	Spatial correlation-based method	R	Yes	No	No	93.0% (IP spoofing attacks)	0.05% (IP spoofing attacks)
KNN classifier approach [33]	Statistical	Nearest neighbor-based method	R	No	No	No	91.88% (DARPA 2000 DDoS data)	8.11% (DARPA 2000 DDoS data)
Change point detect by fuzzy logic [62]	Soft computing	Fuzzy logic-based method	R	No	No	No	(ns2 simulation)	
Linear prediction model [26]	Statistical	Linear prediction-based method	R	Yes	No	No	96.1% (DDoS flow data, LLDoS 2.0.2)	0.8% (DDoS flow data, LLDoS 2.0.2)
SSM method [27]	Statistical	Statistical segregation-based method	R	No	No	No	(CAIDA data)	1.2% (CAIDA data)
Ensemble of neural net model [35]	Soft computing	RBPBoost system/ensemble of neural net-based classifiers	N	Yes	Yes	Yes	99.4% (DARPA 2000 DDoS data)	3.7% (DARPA 2000 DDoS data)
Autonomic mathematical model [64]	Machine learning	CPN-based method	R	Yes	No	No	(ns2 simulation)	

Column 4 in the table represents if the method can be used real time (R) or non-real time (N). The DR and FAR are given in column 8 and column 9, respectively.

a new scheme. Otherwise a scheme that performs well for a fixed dataset may not perform the same when deployed in a real network. The following are the types of datasets used for evaluating DDoS attack detection methods.

(i) *Benchmark datasets*: Benchmark datasets are prepared using an enterprise network and with proper labels such as normal or attack. Only a few benchmark intrusion datasets are publicly available but they are not for DDoS attacks. The KDDcup99 intrusion dataset is the most popular intrusion dataset publicly available but it is not suitable for DDoS attacks.

(ii) *Simulated datasets*: Another alternative for evaluating DDoS attack detection methods is to simulate the

TABLE 6. DDoS tools and description.

Name and ref.	Description	Protocol	Attack
Trinoo [69, 70]	• Widely used by the attackers as well as research community • A bandwidth depletion attack tool, used to launch coordinated UDP flood attacks against one or many IP addresses • Fixed-size UDP packets are sent to the victim machine's random ports • Does not spoof source addresses • Implements UDP flood attacks against the target victim	UDP	UDP flood
Tribe flood network (TFN) [71]	• Able to wage both bandwidth depletion and resource depletion attacks • Uses a command line interface to communicate between the attacker and the control master program • Offers no encryption between agents and handlers or between handlers and the attacker • Allows TCP SYN and ICMP flood as well as smurf attacks	UDP, ICMP, TCP	TCP SYN flood, ICMP flood, smurf
TFN2K [72]	• Developed using the TFN DDoS attack tool • Adds encrypted messaging among all of the attack components [73] • Communications between real attacker and control master program are encrypted using a key-based CAST-256 algorithm [74] • Conducts covert exercises to hide itself from intrusion detection systems • Can forge packets that appear to come from neighboring machines • Provides other options such as TARGA and MIX attack [75]	TCP, UDP, ICMP	smurf, SYN flood, UDP flood, ICMP flood
Stacheldraht [76]	• Based on early versions of TFN and eliminates some of its weak points by combining features of Trinoo • Performs updates on the agents automatically • Provides a secure telnet connection via symmetric key encryption among the attackers and handlers • Communicates through TCP and ICMP packets	TCP, UDP, ICMP	TCP SYN flood, UDP flood, ICMP echo request flood
mstream [77]	• Uses spoofed TCP packets with the ACK flag set to attack the target • A simple point-to-point TCP ACK flooding tool to overwhelm the tables used by fast routing routines in switches • Communications are not encrypted, and performed through TCP/UDP packets; zombie is connected via telnet by master • Target gets hit by ACK packets and sends TCP RST to non-existent IP addresses • Routers return 'ICMP unreachable' causing more bandwidth starvation • Possesses very limited control features and can spoof by randomizing all 32 bits of the source IP address	TCP, UDP	TCP ACK flood
Shaft [78]	• A successor of Trinoo • Uses UDP communication between handlers and agents • Shaft provides UDP/ICMP/TCP flooding attack options; it randomizes source IP address and source port in packets • The size of packets remains fixed during the attack • Able to switch the handler's IP address and port in real time during the attack • Able to switch control master servers and ports in real time, hence making detection by intrusion detection tools difficult	TCP, UDP, ICMP	TCP/UDP/ICMP flood
Trinity v3 [79]	• Various TCP floods are used by randomizing all 32-bits of the source IP address, such as TCP fragment floods, TCP established floods, TCP RST packet floods and TCP random flag packet floods • Generates TCP flood packets with random control flags set to provide a wider set of TCP-based attacks	TCP, UDP	TCP fragment floods, TCP RST packet floods, TCP random flag packet floods, TCP established floods

(*continued*)

TABLE 6. Continued

Name and ref.	Description	Protocol	Attack
Knight [80]	• A very lightweight yet powerful IRC-based attack tool • Provides SYN attacks, UDP Flood attacks and an urgent pointer flooder [81] • Designed to run on Windows operating systems and has features such as an automatic updater via http or ftp, a checksum generator and more • Uses Trojan horse program called Back Orifice for installation in the target host	TCP, UDP	UDP, TCP flood, SYN and PUSH+ACH flood
LOIC [8]	• A powerful anonymous attacking tool via IRC • Operates in three methods of attack: TCP, UDP and HTTP • Exists in two versions: binary version and web-based version	TCP, UDP, HTTP	UDP, TCP, HTTP flood

environment using available tools such as ns2,[2] Qualnet[3] and OMNeT++.[4] These simulators generally generate traffic statistics randomly and sometimes use statistical distributions. But it is difficult to ensure that the generated traffic correlates well with real network traffic.

(iii) *Private datasets*: The best approach for testing any intrusion detection system or DDoS attack detection method is to create a real network testbed with a large number of host and network components. Then one can generate normal as well as attack traffic and capture them using standard tools, used for high-speed networks. After capture, one preprocesses the raw data and extracts features in a distributed manner and finally correlates each feature to create a complete dataset. After necessary labeling, it should be useful to test any intrusion detection system.

5. OPEN ISSUES AND CHALLENGES

Many methods for DDoS detection have been reported in the literature, but only a few of them have been applied in a real network environment and work effectively. Designing and implementing an ideal and practical DDoS defense system is really difficult. The main challenges that any DDoS defense scheme must overcome to become ideally usable are presented below.

(i) In a DDoS attack, attackers try to make a service unavailable to its legitimate clients and launch the attack using a large number of zombies distributed in different networks. It takes only a few seconds to exhaust the bandwidth and other resources of the victim. A faster detection scheme usually consumes higher processing power and, at the same time, it adversely affects detection accuracy. An offline classification scheme generally ensures good performance as classification can be performed in a more sophisticated manner in comparison with real-time detection. But accurately detecting all attacks after interrupting services to legitimate clients has no use. So, emphasis should be given more to speed over accuracy of detection for a practical DDoS defense mechanism.

(ii) Real-time detection of an LDDoS attack with high detection accuracy and low false alarm is a challenging task, since such traffic follows the normal traffic distribution.

(iii) A DDoS defense mechanism cannot simply be judged based on its performance with a standard fixed dataset containing normal and a few attack packets. It must be scalable to real networks for actual deployment. In a real application, a defense scheme should be able to capture, process and classify incoming packets in real time. It cannot just expect to store captured packets and analyze them later, obviously increasing the time lag between the occurrence of an event and its subsequent detection. Particularly, in the case of DDoS attacks, the problem would become more acute, as all resources will be exhausted. So, all phases of detection should run in a parallel manner in real time. Generally, real-time DDoS detection systems are expected to be scalable for use in high-speed real networks.

(iv) With the continuing progress in Internet technologies, attackers are developing and launching new attacks with greater sophistication day by day. So, detection of new attacks is a challenge for any defense system. In supervised learning, the classifier needs to be trained with a proper minimum training dataset, not biased toward any attack. This means that it cannot detect unknown attacks, particularly those with behavior very different from existing attack classes. Unsupervised approaches are appropriate for detecting unknown attacks. However, these methods are usually not suitable from real-time performance. Hence, developing a combined approach based on both supervised and unsupervised approaches with the capability of detecting

[2] http://www.isi.edu/nsnam/ns/.
[3] http://www.qualnet.ca.
[4] http://www.omnetpp.org/.

both known and unknown attacks real time or near real time is of utmost necessity.

(v) Accurate segregation of high-rate DDoS attack traffic from normal flash crowds with minimum resource consumption or low FAR in real-time or near real-time is a challenging task.

(vi) Transparency to existing Internet infrastructure is very important in terms of deployment. It should not be allowed to slow down the processing of normal packets from legitimate clients, which itself is DoS. Moreover, the defense system itself should not be vulnerable to attacks, creating unavoidable breakdown of service. Most importantly, a DDoS defense scheme should be deployable in real networks.

(vii) High-speed traffic analysis for detecting DDoS attacks is a challenging task. A defense scheme can get overwhelmed in real networks, particularly at the time of a DDoS attack. So, the defense scheme should be able to handle the crowd and still function properly. A defense scheme capable of real-time detection should perform well with high-speed traffic. Offline schemes suffer in high-speed traffic because of the overhead created by processing delay, which can slow down detection speed further or can cause total breakdown in extreme cases.

(viii) Real-time updation of network statistics and fast identification of randomized spoofed IP addresses are challenges.

(ix) In DDoS attacks with a large number of agents, attack behavior often conforms well with normal behavior. In such a situation, for a DDoS defense mechanism aiming to provide a near real-time solution may have to be based on an incremental clustering algorithm to segregate the attack from normal traffic. This requires an appropriate proximity measure that works sensibly, quickly and reliably.

(x) The detection method should be dependent on a minimum number of input parameters if not independent of parameters and should also be based on a minimum number of traffic parameters or features.

6. POSSIBLE SOLUTIONS AGAINST DDOS ATTACKS

DDoS defense schemes are of three types based on their deployment: *source-end*, *victim-end* and *intermediate* router defense mechanisms. Among these three, most researchers are in favor of using the victim-end approach. However, a common disadvantage of this scheme is the consumption of a huge amount of resources to provide a fast detection response. In the latest DDoS attacks scenarios, once the attackers gain access, they can increase the attack intensity instantly, and after acquiring a majority of available resources, they can launch attacks without spoofing IPs and they may extend activities to complex database query transactions also. Generally, segregation of such transactions from the legitimate queries is a difficult task.

Current DDoS attacking tools are capable of launching attacks in different modes [20] such as *increasing rate, constant rate, pulsing* (attack rate oscillates between *maximum* to 0) and *gradual pulsing* (e.g. attack rate achieves a *maximum* in 20 *s* and reduces to 0 in 10 *s*). These various forms of attacks may involve single as well as multiple attack sources, that too using single as well as multiple source addresses. Defenders are interested in providing a semi-automatic solution with an objective of achieving reduced false alarm with minimum resource consumption. A solution we propose at a high level is based on an ensemble approach working in a distributed framework using appropriate fusion techniques for decision making. Protocol-specific feature extraction in a distributed environment involving multiple sensors and detecting DDoS attacks based on a class-specific subset of features at the individual level will provide the base for the underlying layer of the proposed solution. Finally, the individual decisions can be combined using an appropriate combination rule at the upper layer at any of the participating nodes in the distributed framework for the final inference. Once the attack occurrence is confirmed, the next task is to limit the attack rate instantly in a selective manner without affecting service to legitimate users, so that subsequent damage can be minimized immediately. However, for significant reduction of the FAR, involvement of human experts is often useful. Especially, in the case of LDDoS attacks, an appropriate diagnosis of false alarms with the help of human analysts, and then providing feedback to the detecting sensors, are very essential. The three major advantages of our solution are: (i) it helps achieve an unbiased and high DR at reduced false alarms, (ii) it minimizes resource consumption by sharing the computation cost and (iii) it achieves a scalable, real-time detection performance.

7. CONCLUSION

While developing a DDoS defense scheme, the issues discussed in this paper need to be deliberated and considered with due seriousness. In this paper, we have presented an overview of DDoS attacks, detection schemes and finally research issues and challenges. In addition, we provide a comparison among current detection methods. Practically designing and implementing a DDoS defense is very difficult. The comparison of the existing detection mechanisms shows that most schemes are not capable of fulfilling all the requirements for real-time network defense. Different performance parameters need to be balanced against each other delicately and appropriately. A possible solution to counter DDoS attacks is also briefly outlined in this paper.

ACKNOWLEDGEMENTS

The authors are thankful to the funding agencies.

FUNDING

This work is supported by the Department of Information Technology and the Council of Scientific & Industrial Research (CSIR), Government of India.

REFERENCES

[1] Peng, T., Leckie, C. and Ramamohanarao, K. (2007) Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput. Surv.*, **39**, 3:1–3:42.

[2] Chandola, V., Banerjee, A. and Kumar, V. (2009) Anomaly detection: a survey. *ACM Comput. Surv.*, **41**, 15:1–15:58.

[3] Loukas, G. and Öke, G. (2010) Protection against denial of service attacks: a survey. *Comput. J.*, **53**, 1020–1037.

[4] Bhuyan, M.H., Bhattacharyya, D.K. and Kalita, J.K. (2011) Surveying port scans and their detection methodologies. *Comput. J.*, **54**, 1565–1581.

[5] Kashyap, H.J. and Bhattacharyya, D.K. (2012) A DDoS Attack Detection Mechanism Based on Protocol Specific Traffic Features. *Proc. 2nd Int. Conf. Computational Science, Engineering and Information Technology*, Coimbatore, India, October 26–28, pp. 194–200. ACM.

[6] Lin, S. and Chiueh, T.C. (2006) A Survey on Solutions to Distributed Denial of Service Attacks. Technical Report TR201. Department of Computer Science, State University of New York, Stony Brook. http://www.ecsl.cs.sunysb.edu/tr/TR201.pdf.

[7] Specht, S.M. and Lee, R.B. (2004) Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. *Proc. ISCA 17th Int. Conf. Parallel and Distributed Computing Systems*, San Francisco, CA, USA, September 15–17, pp. 543–550. ISCA.

[8] Batishchev, A.M. (2004) *LOIC (Low Orbit Ion Cannon)*. http://sourceforge.net/projects/loic/.

[9] Gogoi, P., Bhattacharyya, D.K., Borah, B. and Kalita, J.K. (2011) A survey of outlier detection methods in network anomaly identification. *Comput. J.*, **54**, 570–588.

[10] Mirkovic, J. and Reiher, P. (2004) A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comput. Commun. Rev.*, **34**, 39–53.

[11] Kaspersky (2012) *Kaspersky Internet Security & Anti-virus. Russian Federation*. http://www.kaspersky.com/.

[12] Stein, L.D. and Stewart, J.N. (2002) *The World Wide Websecurity FAQ, Version 3.1.2*. Cold Spring Harbor, NY. http://www.w3.org/Security/Faq.

[13] Houle, K.J. and Weaver, G.M. (2001) Trends in Denial of Service Attack Technology. Technical Report v1.0. CERT and CERT coordination center, Carnegie Mellon University, Pittsburgh, PA.

[14] Wong, T.Y., Law, K.T., Lui, J.C.S. and Wong, M.H. (2006) An efficient distributed algorithm to identify and traceback DDoS traffic. *Comput. J.*, **49**, 418–442.

[15] Collins, J.R. (2000) *RAMEN—A Linux Worm*. SANS Institute, Maryland, USA. http://www.giac.org/paper/gsec/505/ramen-linux-worm/101193.

[16] CERT (2001) *CERT coordination center, CERT advisory CA-2001-19 'code red' worm exploiting buffer overflow in IIS indexing service DLL*. Carnegie Mellon Software Engineering Institute, Pittsburgh, USA. http://www.cert.org/adviso-ries/CA-2001-19.html.

[17] Loon, R.V. and Lo, J. (2004) *An IRC tutorial*. http://www.irchelp.org/irchelp/irctutorial.html.

[18] Yu, S., Zhou, W., Doss, R. and Jia, W. (2011) Traceback of DDoS attacks using entropy variations. *IEEE Trans. Parallel Distrib. Syst.*, **22**, 412–425.

[19] Chen, Y., Hwang, K. and Ku, W.S. (2006) Distributed Change-Point Detection of DDoS Attacks Over Multiple Network Domains. *Proc. IEEE Int. Symp. Collaborative Technologies and Systems*, Las Vegas, NV, USA, May 14–17, pp. 543–550. IEEE CS.

[20] Mirkoviac, J., Prier, G. and Reiher, P. (2002) Attacking DDoS at the Source. *Proc. 10th IEEE Int. Conf. Network Protocols*, Paris, France, November 12–15, pp. 1092–1648. IEEE CS.

[21] Saifullah, A.M. (2009) Defending Against Distributed Denial-of-Service Attacks with Weight-Fair Router Throttling. Technical Report 2009-7. Computer Science and Engineering, Washington University, St. Louis, USA.

[22] Chen, C.L. (2009) A new detection method for distributed denial-of-service attack traffic based on statistical test. *J. Univers. Comput. Sci.*, **15**, 488–504.

[23] Zhang, G., Jiang, S., Wei, G. and Guan, Q. (2009) A Prediction-Based Detection Algorithm Against Distributed Denial-of-Service Attacks. *Proc. Int. Conf. Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, Leipzig, Germany, June 21–24, pp. 106–110. ACM.

[24] Akella, A., Bharambe, A., Reiter, M. and Seshan, S. (2003) Detecting DDoS Attacks on ISP Networks. *Proc. Workshop on Management and Processing of Data Streams*, San Diego, CA, USA, June 8, pp. 1–2. ACM.

[25] Peng, T., Leckie, C. and Ramamohanarao, K. (2004) Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring. *Proc. 3rd Int. IFIP-TC6 Networking Conf.*, Athens, Greece, May 9–14, pp. 771–782. Springer.

[26] Cheng, J., Yin, J., Wu, C., Zhang, B. and Li, Y. (2009) DDoS Attack Detection Method Based on Linear Prediction Model. *Proc. 5th Int. Conf. Emerging Intelligent Computing Technology and Applications*, Ulsan, South Korea, September 16–19, pp. 1004–1013. Springer.

[27] Udhayan, J. and Hamsapriya, T. (2011) Statistical segregation method to minimize the false detections during DDoS attacks. *Int. J. Netw. Secur.*, **13**, 152–160.

[28] Öke, G. and Loukas, G. (2007) A denial of service detector based on maximum likelihood detection and the random neural network. *Comput. J.*, **50**, 717–727.

[29] Jalili, R., Imani-Mehr, F., Amini, M. and Shahriari, H.R. (2005) Detection of Distributed Denial of Service Attacks Using Statistical Pre-Processor and Unsupervised Neural Networks. *Proc. Int. Conf. Information Security Practice and Experience*, Singapore, April 11–14, pp. 192–203. Springer.

[30] Karimazad, R. and Faraahi, A. (2011) An Anomaly-Based Method for DDoS Attacks Detection Using RBF Neural Networks. *Proc. Int. Conf. Network and Electronics Engineering*, Singapore, pp. 44–48. IACSIT Press.

[31] Gavrilis, D. and Dermatas, E. (2005) Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Comput. Netw. ISDN Syst.*, **48**, 235–245.

[32] Wu, Y.C., Tseng, H.R., Yang, W. and Jan, R.H. (2011) DDoS detection and traceback with decision tree and grey relational analysis. *Int. J. Ad Hoc Ubiquit. Comput.*, **7**, 121–136.

[33] Nguyen, H.-V. and Choi, Y. (2010) Proactive detection of DDoS attacks utilizing k-NN classifier in an Anti-DDoS framework. *Int. J. Electr. Comput. Syst. Eng.*, **4**, 247–252.

[34] Shiaeles, S.N., Katos, V., Karakos, A.S. and Papadopoulos, B.K. (2012) Real time DDoS detection using fuzzy estimators. *Comput. Secur.*, **31**, 782–790.

[35] Kumar, P.A.R. and Selvakumar, S. (2011) Distributed denial of service attack detection using an ensemble of neural classifier. *Comput. Commun.*, **34**, 1328–1341.

[36] Scott, C. and Nowak, R. (2005) A Neyman–Pearson approach to statistical learning. *IEEE Trans. Inf. Theory*, **51**, 3806–3819.

[37] Gil, T.M. and Poletto, M. (2001) MULTOPS: A Data-Structure for Bandwidth Attack Detection. *Proc. 10th Conf. USENIX Security Symp.*, Vol. 10, Berkeley, CA, USA, August 13–17. 3. USENIX Association Berkeley.

[38] Thomas, R., Mark, B., Johnson, T. and Croall, J. (2003) NetBouncer: Client-Legitimacy-Based High-Performance DDoS Filtering. *Proc. 3rd DARPA Information Survivability Conf. and Exposition*, Washington, DC, USA, April 22-24, pp. 111–113. IEEE CS, USA.

[39] Wang, J., Phan, R.C.W., Whitley, J.N. and Parish, D.J. (2010) Augmented Attack Tree Modeling of Distributed Denial of Services and Tree Based Attack Detection Method. *Proc. 10th IEEE Int. Conf. Computer and Information Technology*, Bradford, UK, June 29–July 1, pp. 1009–1014. IEEE CS.

[40] Limwiwatkul, L. and Rungsawang, A. (2004) Distributed Denial of Service Detection Using TCP/IP Header and Traffic Measurement Analysis. *Proc. IEEE Int. Symp. Communications and Information Technology*, Sapporo, Japan, October 26–29, pp. 605–610. IEEE CS.

[41] Zhang, G. and Parashar, M. (2006) Cooperative defence against DDoS attacks. *J. Res. Pract. Inf. Technol.*, **38**, 1–14.

[42] Lu, K., Wu, D., Fan, J., Todorovic, S. and Nucci, A. (2007) Robust and efficient detection of DDoS attacks for large-scale internet. *Comput. Netw.*, **51**, 5036–5056.

[43] Hwang, K., Dave, P. and Tanachaiwiwat, S. (2003) NetShield: Protocol Anomaly Detection with Datamining Against DDoS Attacks. *Proc. 6th Int. Symp. Recent Advances in Intrusion Detection*, Pittsburgh, PA, USA, September 8–10, pp. 8–10. Springer.

[44] Chen, Z., Chen, Z. and Delis, A. (2007) An inline detection and prevention framework for distributed denial of service attacks. *Comput. J.*, **50**, 7–40.

[45] Lee, K., Kim, J., Kwon, K.H., Han, Y. and Kim, S. (2008) DDoS attack detection method using cluster analysis. *Expert Syst. Appl.*, **34**, 1659–1665.

[46] Sekar, V., Duffield, N., Spatscheck, O., van der Merwe, J. and Zhang, H. (2006) LADS: Large-Scale Automated DDoS Detection System. *Proc. Annual Conf. USENIX Annual Technical Conf.*, Boston, MA, USA, May 30–June 3, pp. 16–29. USENIX Association.

[47] Rahmani, H., Sahli, N. and Kammoun, F. (2009) Joint Entropy Analysis Model for DDoS Attack Detection. *Proc. 5th Int. Conf. Information Assurance and Security*, Vol. 02, Xian, China, August 18–20, pp. 267–271. IEEE CS.

[48] Xiang, Y., Li, K. and Zhou, W. (2011) Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE Trans. Inf. Forensics Sec.*, **6**, 426–437.

[49] Shannon, C.E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**, 397–423.

[50] Francois, J., Aib, I. and Boutaba, R. (2012) FireCol: a collaborative protection network for the detection of flooding DDoS attacks. *IEEE/ACM Trans. Netw.*, **20**, 1828–1841.

[51] Jeyanthi, N. and Iyengar, N.C.S.N. (2012) An entropy based approach to detect and distinguish DDoS attacks from flash crowds in VoIP networks. *Int. J. Netw. Secur.*, **14**, 257–269.

[52] Li, M. and Li, M. (2009) A New Approach for Detecting DDoS Attacks Based on Wavelet Analysis. *Proc. 2nd Int. Congress on Image and Signal Processing*, Tianjin, China, October 17–19, pp. 1–5. IEEE.

[53] Zhong, R. and Yue, G. (2010) DDoS Detection System Based on Data Mining. *Proc. 2nd Int. Symp. Networking and Network Security*, Jinggangshan, China, April 2–4, pp. 062–065. Academy Publisher.

[54] Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *Proc. 20th Int. Conf. Very Large Data Bases*, Santiago de Chile, Chile, September 12–15, pp. 487–499. Morgan Kaufmann.

[55] Dunn, J.C. (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.*, **3**, 32–57.

[56] Dainotti, A., Pescapé, A. and Ventre, G. (2009) A cascade architecture for DoS attacks detection based on the wavelet transform. *J. Comput. Secur.*, **17**, 945–968.

[57] Haar, A. (1910) Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, **69**, 331–371.

[58] Li, L. and Lee, G. (2003) DDoS Attack Detection and Wavelets. *Proc. 12th Int. Conf. Computer Communications and Networks*, Dallas, TX, USA, October 20–22, pp. 421–427. IEEE.

[59] Gupta, B.B., Joshi, R.C. and Misra, M. (2012) ANN based scheme to predict number of zombies in DDoS attack. *Int. J. Netw. Secur.*, **14**, 36–45.

[60] Yan, R., Zheng, Q., Niu, G. and Gao, S. (2008) A New Way to Detect DDoS Attacks within Single Router. *Proc. 11th IEEE Singapore Int. Conf. Communication Systems*, Guangzhou, China, November 19–21, pp. 1192–1196. IEEE CS.

[61] Cheng, J., Yin, J., Liu, Y., Cai, Z. and Wu, C. (2009) DDoS Attack Detection Using IP Address Feature Interaction. *Proc. 1st Int. Conf. Intelligent Networking and Collaborative Systems*, Barcelona, Spain, November 4–6, pp. 113–118. IEEE CS.

[62] Xia, Z., Lu, S., Li, J. and Tang, J. (2010) Enhancing DDoS flood attack detection via intelligent fuzzy logic. *Informatica (Slovenia)*, **34**, 497–507.

[63] Zhang, C., Cai, Z., Chen, W., Luo, X. and Yin, J. (2012) Flow level detection and filtering of low-rate DDoS. *Comput. Netw.*, **56**, 3417–3431.

[64] Gelenbe, E. and Loukas, G. (2007) A self-aware approach to denial of service defence. *Comput. Netw.*, **51**, 1299–1314.

[65] Ghanea-Hercock, R.A., Gelenbe, E., Jennings, N.R., Smith, O., Allsopp, D.N., Healing, A., Duman, H., Sparks, S., Karunatillake, N.C. and Vytelingum, P. (2007) Hyperion—next-generation battlespace information services. *Comput. J.*, **50**, 632–645.

[66] Gelenbe, E. (2009) Steps toward self-aware networks. *Commun. ACM*, **52**, 66–75.

[67] Gelenbe, E. (2011) Search in unknown random environments. *Phys. Rev.*, **82**, 061112.

[68] Douligeris, C. and Mitrokotsa, A. (2004) DDoS attacks and defense mechanisms: classification and state-of-the-art. *Comput. Netw.*, **44**, 643–666.

[69] Criscuolo, P.J. (2000) Distributed Denial of Service Trinoo, Tribe Flood Network, Tribe Flood Network 2000, and Stacheld-raht CIAC-2319, Department of Energy Computer Incident Advisory (CIAC). Rev. 1 UCRL-ID-136939. Lawrence Livermore National Laboratory, Livermore, CA. http://ftp.se.kde.org/pub/security/csir/ciac/ciacdocs/ciac2319.txt.

[70] Dittrich, D. (1999) The DoS Project's 'Trinoo' Distributed Denial of Service Attack Tool. Technical Report. University of Washington, Seatlle, USA. http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt.

[71] Dittrich, D. (1999) The Tribe Flood Network Distributed Denial of Service Attack Tool. Technical Report. University of Washington, Seatlle, USA. http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt.

[72] Barlow, J. and Thrower, W. (2000) *TFN2K—an analysis*. http://packetstormsecurity.org/files/10135/TFN2k_Analysis-1.3.txt.html. AXENT Security Team.

[73] CERT (1999) CERT Coordination Center, Center Advisory CA-1999-17 Denial of Service Tools. Technical Report CA-1999-17. Carnegie Mellon University, Pittsburgh, USA. http://www.cert.org/advisories/CA-1999-17.html.

[74] Adams, C. and Gilchrist, J. (1999) The CAST-256 Encryption Algorithm. Technical Report RFC 2612. Ohio State University, Cleveland, USA. http://www.cis.ohio-state.edu/htbin/rfc/rfc2612.html.

[75] Bellovin, S. (2000) The ICMP Traceback Message. Technical Report. Network Working Group, Internet Draft, Canada. http://www.research.att.com/ smb/papers/draft-bellovin-itrace-00.txt.

[76] Dittrich, D. (1999) The 'Stacheldraht' Distributed Denial of Service Attack Tool. Technical Report. University of Washington, Seatlle, USA. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt.

[77] Dittrich, D., Weaver, G., Dietrich, S. and Long, N. (2000) The 'mstream' distributed Denial of Service Attack Tool. Technical Report. University of Washington, Seatlle, USA. http://staff.washington.edu/dittrich/misc/mstream.analysis.txt.

[78] Dietrich, S., Long, N. and Dittrich, D. (2000) Analyzing Distributed Denial of Service Tools: The Shaft Case. *Proc. 14th USENIX Conf. System Administration*, New Orleans, LA, USA, December 3–8, pp. 329–340. USENIX Association.

[79] Hancock, B. (2000) Trinity v3, a DDoS tool, hits the streets. *Comput. Secur.*, **19**, 574.

[80] King, B.B. and Morda, D. (2001) CERT Coordination Center, CERT Advisory CA-2001-20 Continuing Threats to Home Users. Technical Report CA-2001-20. Carnegie Mellon Software Engineering Institute, Pittsburgh, USA. http://www.cert.org/advisories/CA-2001-20.html.

[81] Bysin (2001) Knight.c sourcecode, packetstormsecurity.nl. http://packetstormsecurity.nl/distributed/knight.c.