Scientific Research Publishing

Journal of Information Security, 2018, 9, 45-69
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

A Survey of Cloud Computing Detection Techniques against DDoS Attacks

Sabah Alzahrani, Liang Hong

Department of Electrical & Computer Engineering, Tennessee State University, Nashville, TN, USA
Email: salzahr1@my.tnstate.edu, lhong@tnstate.edu

How to cite this paper: Alzahrani, S. and Hong, L. (2018) A Survey of Cloud Computing Detection Techniques against DDoS Attacks. *Journal of Information Security*, **9**, 45-69.
https://doi.org/10.4236/jis.2018.91005

Received: October 26, 2017
Accepted: December 18, 2017
Published: December 21, 2017

Abstract

A Distributed Denial of Service Attack (DDoS) is an attack in which multiple systems compromised by a Trojan are maliciously used to target a single system. The attack leads to the denial of a certain service on the target system. In a DDoS attack, both the target system and the systems used to perform the attack are all victims of the attack. The compromised systems are also called Botnets. These attacks occur on networked systems, among them the cloud computing facet. Scholars have tried coming up with separate mechanisms for detecting and preventing such attacks long before they occur. However, as technology progresses in advancement so do the attack mechanisms. In cloud computing, security issues affect various stakeholders who plan on cloud adoption. DDoS attacks are such serious concerns that require mitigation in the cloud. This paper presents a survey of the various mechanisms, both traditional and modern, that are applied in detecting cloud-based DDoS attacks.

Keywords

DDoS, IDS, Signature, Anomaly, Hybrid, SVM, Neural Network, Cloud, Machine Learning, Big Data

1. Introduction

Internet has led to cloud computing which constitutes three major services namely platform as a service, infrastructure as a service, and software as a service [1]. This increase in data and information storage within the cloud environment has raised cloud security concerns on the safety of data and information. It has also led to distributed attacks such as ICMP flood, the Ping of Death, the slowloris, the SYN flood attack, the UDP flood attack, malformed packet attacks, protocol vulnerability exploitation, and the HTTP flood molest [2] [3]. The choice on any attack type depends on the ease of such exploitation or its mastery by the attacker.

Previous researchers have expounded on how Distributed attacks in the cloud can be detected, prevented and mitigated. These techniques greatly apply two major detection mechanisms of signature or anomalies. They can use one, both, or be intelligent enough to learn new attacks based on set rules. The next section offers a review of various traditional based intrusion detection techniques. Further, it reviews the various classes of cloud computing based detection methods and offers examples. The underlying purpose being to compare the various detection methods and point out the strengths and limitations they pose. Beyond the review, the paper will show how specific techniques by specific scholars were successful or failed in the detection process against DDoS attacks in the cloud. In the analysis, the performance evaluation metrics used in a given technique will be shown. Additionally, the analysis will point out the various data sets and tools used by these techniques. As such, it will be possible to decide which of the techniques is efficient or has potential for future enhancement.

2. Literature Review

Existing techniques utilize different forms of algorithms to detect and determine attack levels within the cloud. HTTP-DoS and XML-Dos attacks are known to lead to exhaustion of resources [4]. Cloud-based intrusion detection techniques are an improved version of traditional intrusion detection system. The first section of this paper discusses various traditional intrusion detection techniques that are as well applied in the cloud. The second section will show cloud-specific intrusion detection techniques.

2.1. Traditional Intrusion Detection Techniques

2.1.1. Signature Based Detection Technique

This detection, also known as misuse technique, compares known information to already captured signatures stored in the database. The technique is only suitable for the detection of known attacks. A common tool used in signature detection technique is the SNORT tool [5]. SNORT is greatly used as it allows its users to set their rules and use those rules in regulating attacks on either the training set or real data set of attack.

In the study conducted by Mazzariello, Canonico, and Bifulco, the authors deployed the network based IDS at separate cloud positions. By considering two scenarios in calculating the performance of the IDS, two results were depicted. First, they inferred that the load on the controller increased, and the IDS detected the likelihood of the attack. Secondly, deploying an IDS close to the virtual machine resulted in the increase of the CPU load [6].

2.1.2. Anomaly Based Detection Technique

These techniques observe the behavior of an event and determine existing anomalies. Shannon-Wiener’s index theory analyzes random data with an aim to unravel existing uncertainty. Reference [7] defines an entropy as the measure of abnormal behavior or randomness. In the separate study, data from a single class proved to contain a lesser entropy unlike statistics from multiple ones.

Headers present in the sampled data are analyzed to determine the IP and ports before computing their entropy. A certain threshold is then constituted to detect a DDoS attack where incase the observed abnormality surpasses a set threshold, the IDS raises alarm alerts [8] [9]. An approach for detecting HTTP based DDoS attacks is proposed by [10]. It entails a five step filter tree approach of cloud defense. These steps include filtering of sensors and Hop Counts, diverging IP frequencies, Double signatures, and puzzle solving [10]. The approach helped in determining anomalies with the various Hop Counts and treating the sources of such anomaly as attack source.

2.1.3. Artificial Neural Network Intrusion Detection Technique

Techniques utilizing ANN to detect intrusions aim at generalizing incomplete data and classifying it as either intrusive or normal. An ANN IDS can either utilize a Multi-Layer Perceptron (MLP), Back propagation (BP), or a Multi-Layer Feed-Forward (MLFF) technique. An approach by Gradiega Ibarra, Ledesma, and Garcia compared the use of self organization map (SOM) to MLP in determining intrusion rates and found that SOM provides high accuracy rates of detection compared to ANN [11].

Cannady utilized a signature-based detection mechanism in a three layer neural network as a means to detect any intrusions. He used a nine network feature vector consisting of the Source port, protocol id, Raw data, destination port, Data Length, source IP address, ICMP code, the type of ICMP, and the destination IP address to determine the intrusions [11].

2.1.4. Genetic Algorithm Intrusion Detection Systems

The use of genetic algorithms in the development of IDS helps in incorporating various network features towards determining best possible parameters for accuracy improvement and result optimization. Gong, Zulkernine, and Abolmaesumi implemented seven network features namely Duration, Protocol, Source IP, Destination IP, Source port, destination port, and attack name in analyzing packets. By using fitness function frameworks that support confidence, the authors were able to detect and determine network intrusions with high accuracy levels.

Reference [11] proposed a solution that combined both genetic algorithms and fuzzy to detect signature and anomaly attacks. Fuzzy logic helps in accounting for quantitative parameters while genetic algorithm determines the best fit parameters that are introduced by the fuzzy logic. This approach proved to solve the best fit problem in Cloud environment. It also showed that since selecting optimal network features as the parameters for intrusion detection increases an IDS accuracy level, the use of Genetic algorithm in developing IDS is effective for Cloud use [11].

2.1.5. Fuzzy Logic Intrusion Detection System

Fuzzy logic provides high flexibility levels to intrusion detection problems. It helps deal with imprecise intrusions. A Fuzzy IDS was proposed by Tillapart, Thumthawatworn, and Santiprabhob to deal with network intrusions such as the Ping of Death, SYN, UDP floods, E-mail Bomb, port scanning, and FTP password guessing. Chavan, Shah, Dave, and Mukherjee implemented both Fuzzy logic and ANN to develop Evolving fuzzy neural network (EFuNN) that applied both unsupervised and supervised learning. Their experiment concluded that the used of EFuNN with fewer inputs produces high accuracy levels than the use of ANN alone [11].

2.1.6. Support Vector Machine (SVM) Intrusion Detection Systems

Techniques utilizing SVM detect intrusions using limited samples of data whose dimensions do not affect the accuracy of the outcome. Comparing SVM to ANN, Chen, Su and Shen determined that rates of false positive were more accurate with SVM since the parameters set with SVM are minimum. A limitation for SVM is that it is only usable to test binary data. Li and Liu proposed and alternate intelligent network intrusion and prevention system that utilized a configurable firewall and a SNORT tool to reduce the rates of alarm and raise the accuracy levels of the intrusion detection system [12].

2.1.7. Hybrid Intrusion Detection Systems

Hybrid IDS combine the advantages of two or more techniques discussed above. A new DDoS detection mechanism was introduced by Krishna and Quadir who implemented an architecture based on the Hidden Markov Model and the double TCP mechanism. Five packets apply the 3-way handshake procedure twice, and a SYN is used to maintain a log [13]. The purpose of the double TCP technique is to ensure that there is an identity match before a connection is completed.

Reference [14] notes that the Markov's model when applied to wireless sensor networks helps in detecting any unusual activity. No connection is left half open as the client cannot reciprocate a matching pattern, and an attack is traceable back to its originator [15]. Vissers proposed the Cloud Trace Back (CTB) approach as a defense mechanism for web services through detection at the edge routers. In a reverse manner, SOA is applied to trace back the exact source of a distributed denial of service attack. A Cloud Traceback Mark (CTM) is placed within the header of a web message. All requests are then passed through the CTB thereby preventing any direct attack. To detect it, the victim client requests for message reconstruction in order to pull out the CTM which helps in retracing the source of the attacking request [16] [17].

Ismail presented the covariance matrix approach to detect flood based denial of service attacks. A statistical method scrutinizes the correlativity aspects of network traffic and evaluates the resulting covariance matrix to the already preset one as exhibited by normal traffic. The covariance approach proved to be very effective and accurate in the Neptune and Smurf attack simulation experiments [16]. A separate variation that utilizes both the covariance approach and entropy based system is proposed by [18] that offers in-depth detection at the host and network levels.

A table illustrating the discussed traditional intrusion detection techniques and as presented in the works of [8] [10] and [11] alongside their advantages and limitations is depicted in Table 1.

2.2. Intrusion Detection Systems (IDS) Used in the Cloud

There exist four main IDS types that are applicable to cloud computing. They are the Host based IDS (HIDS), Network-based IDS (NIDS), Hypervisor based IDS, and Distributed IDS (DIDS). A pictorial representation of the various categories of IDS used in the cloud as illustrated by [6] is shown in the Figure 1.

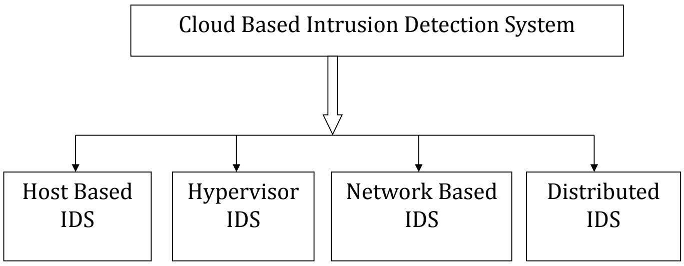


Figure 1. Cloud-based intrusion detection systems.

Table 1. Summary of traditional IDS techniques.

IDS Technique	Advantages	Limitations
Signature-based IDS	1) High accuracies in detecting known attacks 2) Offers low computational costs 3) Easy to track and stop an attack since log files are exhaustive	1) Cannot track down intelligent intrusions. 2) New attacks have to be updated in the database 3) Huge traffic limits the inspection of every packet causing unattended packets to pass through
Anomaly-based IDS	1) Higher the false alarm rate for unknown attacks 2) New threats are easily detectable without updating the database 3) System is self learning. It gradually learns the network and builds profile 4) The more it is used the higher the accuracy level	1) While building profile, a network is left in an unmanaged state hence prone to attack 2) When malicious activities assume the features of normal traffic it is untraceable. 3) Collected behavior and features determine the accuracy of detection
Fuzzy logic IDS	1) Increased flexibility in addressing uncertain problems	1) Offers low accuracy levels compared to ANN
SVM based IDS	1) Correctly classifies intrusions even with limited sample data 2) Ability to handle huge number of features	1) Classifies only distinct features hence the features have to be preprocessed before their application
Genetic algorithm IDS	1) Offers best detection features 2) Has better efficiency	1) Very complex 2) Its usage is of specific pattern as opposed to a general pattern
ANN based IDS	1) Effectively classifies unstructured network packets 2) Classification efficiency achieved by introducing multiple hidden layers	1) Requires a lot of time at the training phase 2) Has lesser flexibility 3) Effective training requires larger data samples
Hybrid Techniques	1) Efficient as it combines multiple techniques to accurately classify rules	1) Its computational costs are high

2.2.1. Network Based Intrusion Detection Systems (NIDS)

These are IDS that detect malicious network activities by monitoring the network traffic. Collected information is the compared to already known attacks before an intrusion is confirmed. This approach is utilizes signature and anomaly techniques to determine both known and unknown network attacks. However, the approach is ineffective as it offers very limited visibility in the host machines and cannot be used to detect intrusion for encrypted network traffic.

Reference [19] proposed a network-based Intrusion Detection System by conducting a turning test for all the IP addresses in the network. It identifies faulty IPs and labels them as blacklist addresses. When an IP requests for the resource, it is checked against the blacklist list. If it exists in the survey, the IP request is dropped. In case the IP address is not faulty, the system checks if the requested resources are available and do not surpass the set threshold. Reference [20] recommended a trilateral trust mechanism for detection and protection against traffic injection attacks. A client always requests for a service through the specified data center hosted by the cloud service provider. Further, the request is routed via a traffic injection rate detector which is preset with the maximum threshold.

A survey by [21] on what security can help detect ARP spoof attacks concluded that by combining XArp 2 tool with an ARP request storm and ARP scanner, ARP spoofing can be greatly managed. Another study analyzed DDoS detection in the multilevel environment whereby a new user freely connects via a router, and the detection algorithm is used to verify the individual as genuine. A register status is stored in CDAP logs [22] During the subsequent access via the router, an entropy is calculated based on data packet size and then compared to already stored range to determine its legitimacy or raise the alarm [23] [24].

Reference [25] recommended a network-based intrusion detection mechanism by combining the rough set theory with the K-nearest neighbor classification technique. Their approach aimed at performing mathematical analysis on connections within a network to determine their categories as either normal, probing, DOS, R2L, or U2R. The analysis further gives the rates of imperfect data that helps in determining the connection.

2.2.2. Host Based Intrusion Detection System (HIDS)

HIDS are deployed at the host machine to monitor and analyze the information collected by the host. They first learn the host's file system, network events, system calls and then observes any modification that may occur at the kernel or file system of the host before raising an alert.

In a cloud environment, HIDS are placed on all VMs, host machines, and hypervisors to monitor and analyze log files, policies of security access, user login information in the bid to detect intrusions. Vieira and Schulter proposed a grid architecture where each node in the cloud has an IDS that interacts with the service offered such as IaaS, storage and IDS services. The IDS service consists of an analyzer and an alert system. Data is captured from and event auditor and the

IDS uses either behavior techniques to detect unknown attacks or knowledge techniques to detect known attacks. When one host detects an attack, the IDS raises an alert and informs other IDS in other hosts. However this approach cannot detect any insider intrusion occurring within the hosts themselves [11].

Reference [15] implemented a network-based IDS against known and unknown attacks. In their model, they used a snort tool and Bayesian classifier. The tool helps in detecting known attacks by comparing them to stored signatures while the classifier tracks any anomalies within the network. When the component of the model determines a possible intrusion, it sends an alert into a common knowledge base to be accessed by the other thereby increasing the rates of intrusion detection [6] [26].

In another approach, a host-based IDS (HIDS) incorporates the external software agent at each cloud server with an aim to increase the resiliency of attacking the VMs without disrupting normal services in the cloud. The agents securely connected to the center of control using virtual LAN. An attack analyzer then decides whether to block or accept the user's request [27]. Reference [28] proposed two way detection techniques that apply the bother tree in packet transmission and augment attack to enforce bottom up detection.

2.2.3. Distributed Intrusion Detection System

Multiple IDS can be combined to save a large network. All IDS collect information and transmit to the central analyzer where centralized analysis takes place. Reference [29] proposed a flexible, scalable and cost effective mechanism for intrusion detection in cloud applications using mobile agents. The mechanisms were meant to help monitor and protect VMs that were outside an organization. The approach was not as effective as it introduced large network loads with increase VMs attached to the mobile agent.

Reference [30] proposed DIDS with various agents for intrusion detection namely the collector agent, the misuse detection agent, the anomaly detection agent, the classifier agent, and the alert agent. Their approach used mobile agent to detect known and unknown attacks and centrally place them in a classifier before raising an alert via the alert agent.

Reference [31] proposed a Cloud service queuing defender (CSQD) technique that aims at protecting the cloud from HTTP and XML forms of DDoS attacks. Using this approach, a server has to be up before a request is processed which is uniquely prefixed with an ID. Reference [32] proposed a VM profiling model aimed at detecting virtual networks attacks by ensuring resilience in the explorations of zombies.

A team led by Lonea proposed a DDoS attack detection technique that uses the Dempster-Shafer theory [33]. In their proposition, the authors set a private cloud consisting of the front-end server and set of three virtual machines (nodes) each with a snort. The IDS set within nodes generate and store alerts in the Mysql database located within the CFU. These alerts are further analyzed and converted into basic probability assignments (bpa) of either true, false, or (true, false). By using the Dempster-Shafer's combination rule to analyze the computed bpa's, the system increases true positive rates and greatly reduces false positive alarm rates [33]. Reference [34] ascertains the Dempster-Shafer Theory by arguing that the use of the centralized database reduces data loss risk and improves the capacity for result analysis and reduces any conflicts.

2.2.4. Hypervisor Based Intrusion Detection System

These are intrusion detection systems running at the hypervisor level. A hypervisor is a platform for running VMs. IDS at hypervisor levels work on virtual networks and allows a user to monitor and analyze all communications occurring within the hypervisor, between the various VMs, and between the VM and the hypervisor. The VM introspection based IDS is an example of a hypervisor intrusion detection system. Research by IBM gives hope to virtual machine introspection approach that creates layered security service levels within a protected VM running on the same machine consisting of guest VMs running in the cloud [11].

Reference [35] proposed a VM introspection based approach that directly observes the hardware state, events, and software states of host machine and offers a robust view of the system. A VM monitor virtualizes the hardware and offers isolation and interposition. This approach helped in lie detection and row socket detection. A table summarizing the strengths and weaknesses of the above cloud based intrusion detection systems is depicted in the Table 2.

3. Analyzing Specific DDoS Detection Techniques

Different scholars have presented specific techniques for detecting distributed denial of service attacks in the cloud. Each technique depicts the metrics used for performance evaluation alongside the datasets and tools.

3.1. Big Data Testbed for Detecting Network Attacks

The detection method presented by [35] simulated network traffic and relied heavily on packet per second passing via a certain route. The technique only

Table 2. Summary of cloud based IDS techniques.

IDS Technique	Strengths	Limitations
Network based IDS	1) Ability to monitor multiple systems at once 2) Their placement is only done on the underlying network	1) Cannot detect intrusions from encrypted network traffic 2) Difficult to detect intrusion in virtual networks 3) Only detects external intrusions
Host based IDS	1) No external hardware required	1) Only monitors attacks on the host it is deployed and set 2) Costly as it is installed on every network host machine
Distributed IDS	1) Has benefits of both NIDS and HIDS as it combines the features of both	1) Central server may become too overloaded and hard to manage 2) High costs of computation and communication
Hypervisor based IDS	1) User is able to examine and explore communication between separate VMs, hypervisors, or between VM and hypervisor	1) Its new and difficult to comprehend

captures HTTP based traffic and avoids other possible network attacks like the UDP and SMTP attacks that may lead to DDOS. This method is meant to detect HTTP GET flood attacks. This application layer attacks never use malformed packets and less consumers of bandwidth compared to other attacks like spoofing. Additionally, they do not generate significant traffic hence they are hard to detect [35]. The approach involved two phases of analyzing a training set of certain normal traffic and then using the parameters as inputs for detecting DDoS attacks using Snort tool

However, there is need to adjust the system in order to allow for detection of dynamic threats. There is need for a self-correction mechanism on already compromised data and a way for detecting already exploitable weaknesses. Introducing aspects of Fuzzy logic or SYSSTAT can help in leveraging the dynamism of the technique in offering proactive defense. Security for big data is an important aspect that needs integration into existing and upcoming cloud based intrusion detection system [35]. In the event that system component such as the memory are compromised, there is need to develop detective mechanisms using reactive defense strategies. This is possible if the system incorporates neural networks and machine learning techniques [36].

3.2. Change-Point Detection Framework in the Cloud

Reference [37] proposed a conceptual cloud DDOS change-point detection mechanism as a means to detecting and preventing DDOS attacks. The technique consists of a change point detection, a packet inter-arrival time (IAT), and a flow based classifier (FBC). The technique is still in its conceptual stage and not practically tested but claims that by reading a packet header to determine its source and destination addresses, it will be possible to determine the packet inter-arrival time of packets from the same source and hence easy to detect any anomalies in packet transmission. A probable demerit with the approach is the possibility of high rates of false negatives and false positives [37].

3.3. Hybrid Intrusion Detection System (H-IDS) for DDoS Attacks

Reference [38] presented a technique combining signature based and anomaly based mechanisms for attack detection. They used two different types of datasets; real data from previous penetration tests done on a commercial bank; and DARPA 2000 dataset. A time analysis was conducted on the DARPA 2000 dataset to offer a priori idea of the detection issue and results presented graphically in **Figure 2**. The performance metrics used included the packet inter-arrival time, the packet size, and the protocol frequencies.

Anomaly detection is provided for by use of the Gaussian Mixture Model (GMM). The detector distinguishes normal traffic from abnormal traffic using data from the extraction phase. The parameters for GMM are estimated using the Expectation Maximization (EM) algorithm and the informatics distance metric method. The EM algorithm helps in determining the probability density

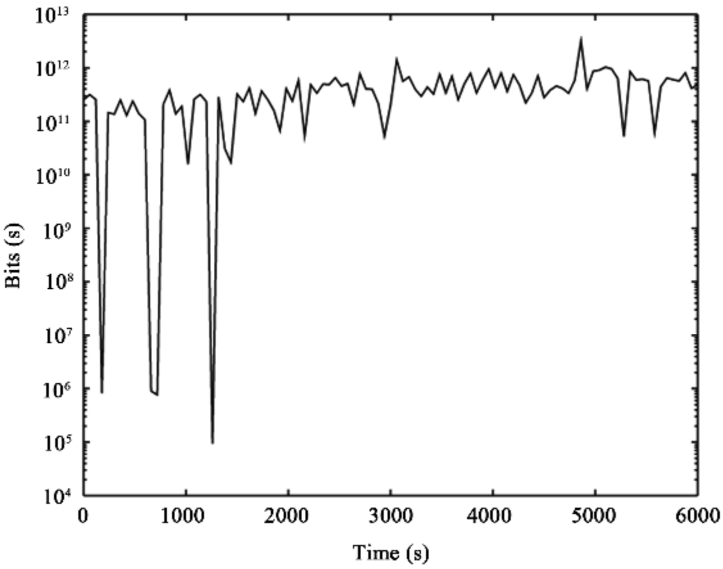


Figure 2. DARPA Analysis of time domain by evaluating density in bits per second (bps) against time in logarithmic scale.[1]

function denoted by $p(\mathbf{x})$. Distance between the parameters is computed and detection determined on that comparison. Using $X=\{x_1,x_2,\cdots,x_n\}$ as a dataset and x_i as a measure of M-dimensional vector, then it a probability density function, $p(\mathbf{x})$ having a finite K component is calculated as below.

$$p(x\,|\,\theta)=\sum_k(\omega_k p_k)(x\,|\,\theta_k)$$

On the other hand, the information distance metric helps in determining the alarm level or mechanism of an attack [38]. The second part of the H-IDS system is the signature-based detective mechanism that uses the SNORT tool to set and modify rules as per the required performance results. A Hybrid Detection Engine (HDE) sets the rules granularity and the SNORT output is denoted as **isAlarm**$_r$ which is calculated based on the number of alerts within a given time frame as is noted with the formula below.

$$\text{isAlarm}_r=\begin{cases}0, & \mathcal{A}(k)=0\\ 1, & \mathcal{A}(k)\geq 0\end{cases}$$

Using the HDE, the authors were able to calculate the attack probability by combining both the anomaly and signature-based detectors. Using the penetration test data, 99% accuracy on True Positive rate (TPR) was attained while DARPA dataset produced a 92.1% accuracy level on TPR [38].

3.4. Hadoop as a Tool for Live DDoS Detection

Reference [39] proposed a live DDoS detection with Hadoop that comprises four stages of Network capturing and Log generation, Log transfer, DDoS detection,

[1] Cepheli, O., Buyukcorak, S. and Kurt, K., G. (2016) Hybrid Intrusion Detection System for DDoS Attacks. *Journal of Electrical and Computer Engineering*, **2016.** Article ID 1075648, 8 pages, Figure 3.

and Result notification. This technique utilizes a web interface with parameterized parameters before capturing the network traffic. A strength with the approach is its ability to detect and analyze live network traffic. The technique proved efficient while analyzing large data sizes unlike in the analysis of small data logs. The approach is as well non-intelligent to handle internal attacks resulting from compromised systems within itself. Introducing fuzzy and machine-learning approaches within the technique can help in tracking dynamic DDoS attacks.

A similar technique is proposed by [40] in which hadoop is used to analyze incoming HTTP, ICMP, UDP, and or TCP packets. The process will involve capturing the packets and generating logs, transferring the logs to HDFS, determining the DDoS attack, and keeping the result. A diagrammatic illustration of the above phases is depicted in Figure 3. Packet capturing is done by Wireshark as it proves to capture huge traffic amounts. Each packet consists of source IP, the packet protocol, some header data, and destination IP. A Traffic Handler is used in the generation of log files. The handler suspends the capturing process of Wireshark upon generating a log. It then transmits the file to the detecting server using a flume as illustrated in Figure 4.

The DDoS detection phase utilized a counter-based algorithm presented in Figure 5. The algorithm uses time interval, threshold and unbalanced ratio as the inputs for the detection. Time acts as a limiting feature to monitor page requests while threshold determines the page request frequency to the server in

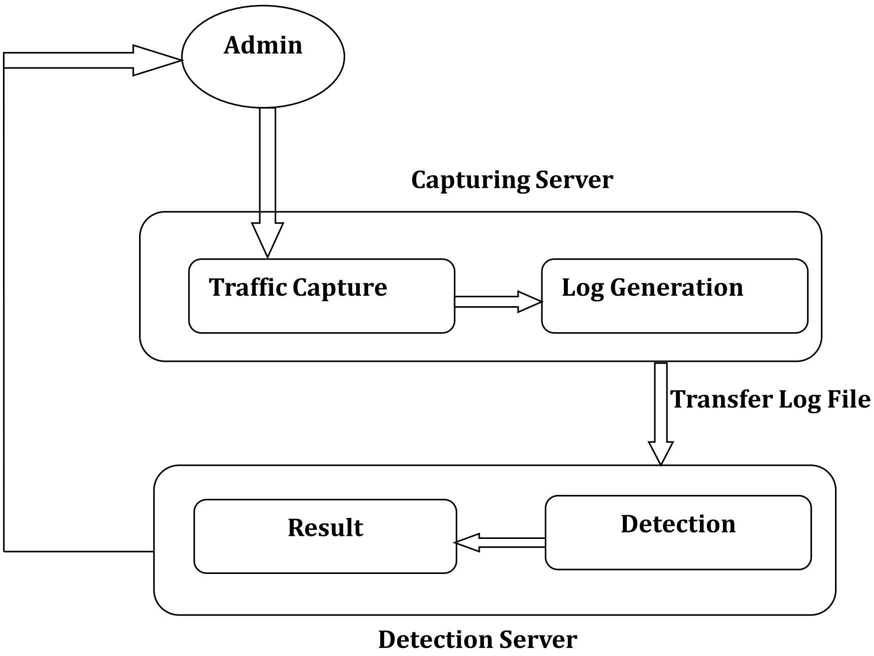


Figure 3. Phases of Hadoop DDoS detection framework.[2]

[2] Korad, S., Kadam, S., Deore, P., Jadhav, M., and Patil, R. (2016) Detection of Distributed Denial of Service Attack with Hadoop on Live Network. *International Journal of Innovative Research in Computer and Communication Engineering*, **4**, 93, Figure 2.

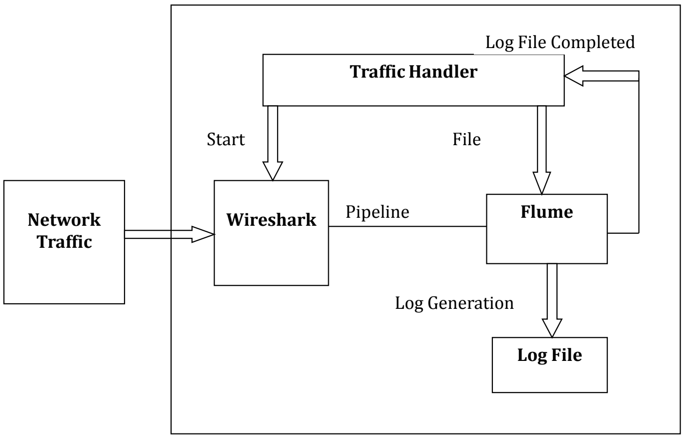


Figure 4. Component for network traffic monitoring and log generation.[3]

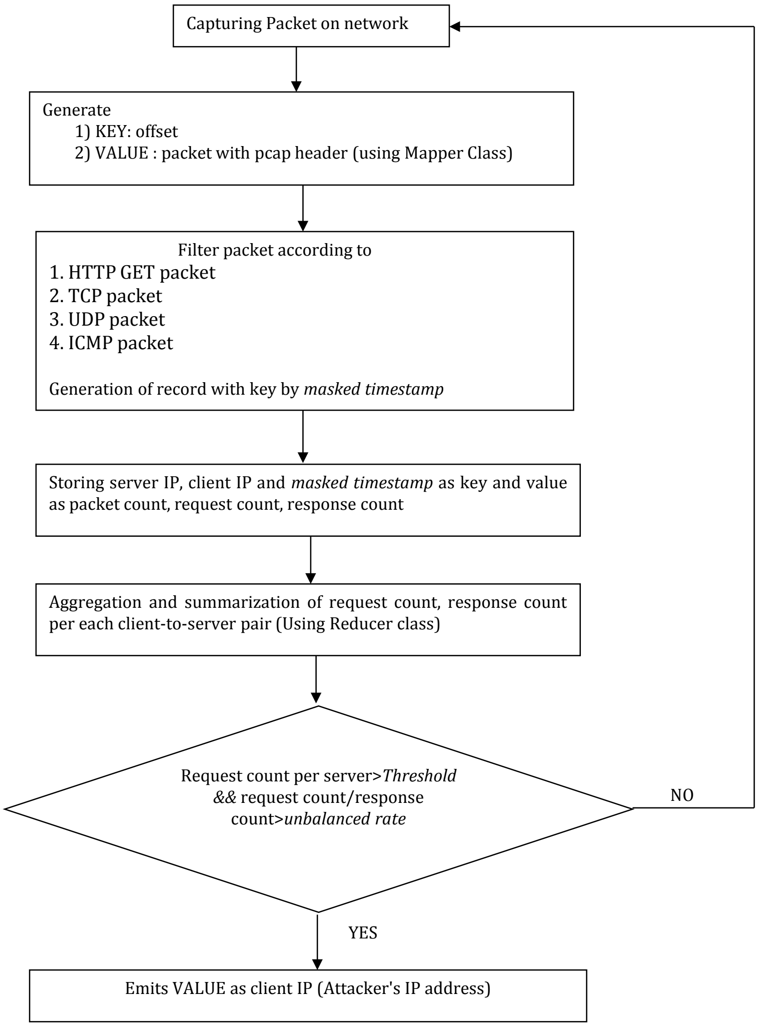

Capturing Packet on network
Generate
1) KEY: offset
2) VALUE : packet with pcap header (using Mapper Class)
Filter packet according to
1. HTTP GET packet
2. TCP packet
3. UDP packet
4. ICMP packet
Generation of record with key by masked timestamp
Storing server IP, client IP and masked timestamp as key and value as packet count, request count, response count
Aggregation and summarization of request count, response count per each client-to-server pair (Using Reducer class)
Request count per server>Threshold && request count/response count>unbalanced rate
NO
YES
Emits VALUE as client IP (Attacker's IP address)

Figure 5. Counter-based DDoS detection algorithm using mapreduce.[4]

[3] Korad, S., Kadam, S., Deore, P., Jadhav, M., and Patil, R. (2016) Detection of Distributed Denial of Service Attack with Hadoop on Live Network, 95, Figure 3.

[4] Korad, S., Kadam, S., Deore, P., Jadhav, M., and Patil, R. (2016) Detection of Distributed Denial of Service Attack with Hadoop on Live Network, 95, Figure 6.

comparison to normal network status. An unbalanced ratio is calculated as the ratio of page request response for a client and its server. An alarm is raised when requests by a client exceeds a threshold [40]. Even though the technique proves to be fast in detection of DDOS attacks and has low complexity of computation, mechanisms for internal attack detection need be introduced. Additionally, the success of its implementation lies in the capability to having beforehand determinacy of threshold value.

3.5. Real-Time Intrusion Detection Using Hadoop and Naive Bayes

Reference [41] proposed an approach for detecting intrusions in real-time by using Hadoop and Naive Bayes classifier. In their approach, the two created a heterogeneous and homogenous clusters for performing the training job. The Snort tool is used to capture packets from the NIC of a firewall and convert them into a binary file. Using Tshark, the system converts the binary data into CSV file which is then converted into UDP stream by a streamgen. A Naive Bayes Classifier present through MapReduce job writes records into an output file which is then read by a java program into disk. The results are graphically presented on a web interface using a D3 render. An architecture of this system is presented in Figure 6. Their approach proved a proof-of-concept technique with 90% success in detecting intrusions through the use of Hadoop and Naives classifier. But then, their results were based on comparison with another technique

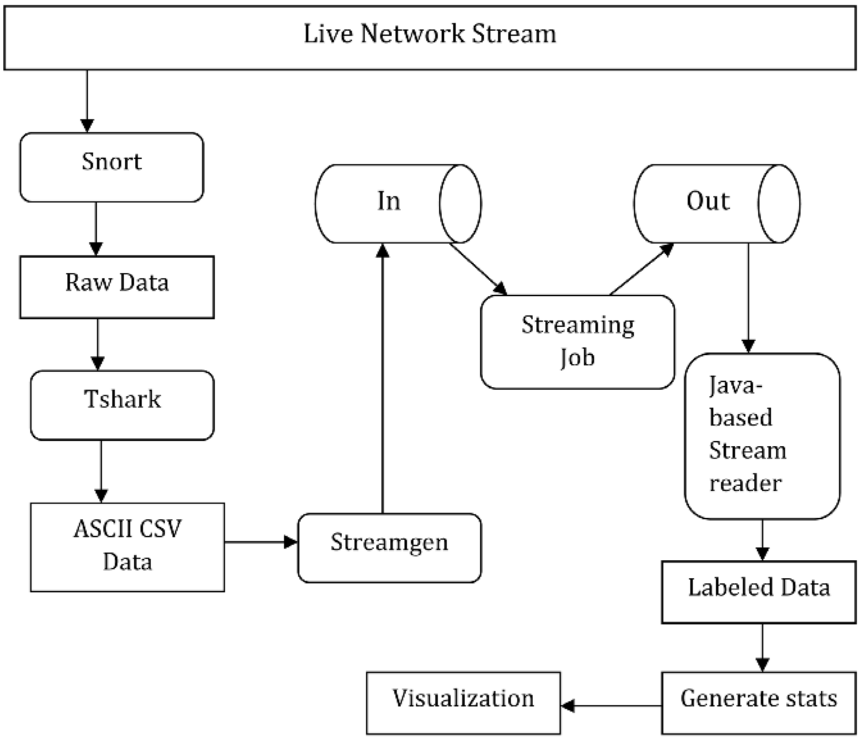


Figure 6. Proposed real-time intrusion through Hadoop and naives bayes.[5]

[5] Veetil, S., and Gao, Q. (2014) Real-time Network Intrusion Detection Using Hadoop-Based Bayesian Classifier. *Emerging Trends in ICT Security*, 288, Figure 1.

which is a small percentage of all available techniques and parameters for analyzing and detecting attacks.

3.6. Botnet Detection Using Big Data Analytics

The work of [42] presented an important approach to combating botnet attacks in a peer-to-peer network. Their approach included three components; a traffic sniffer that captures and preprocesses packets, a feature extraction mechanism for engendering feature sets, and a machine learning techniques provided by Mahout that offers parallel processing in building a random forest based decision tree model. The technique uses dumpcap to sniff into network packets while Tshark extracts fields and sends them to Hadoop based Distributed File System. At feature extraction, an Apache Hive program extract, transforms, and loads the datasets. Using hadoop's HQL language, selection of packet features is extracted using a group by clause based on an algorithm present in MapReduce. Mapping generated key-value pairs that are transmitted to a reducer that groups all values based on given key. This implies that Hadoop's MapReduce framework is dependent on <key, value> pair [42]. Both the input and output are <key, value> pairs as presented in the formula below.

(input) <k1; v1> → map → <k2; v2> → combine → <k2; v2> → reduce → <k3; v3> (output)

The key and value pair is basically the source IP and port and the destination IP and port. This approach utilized the key and value pair mechanism as the great interest was determining problems based on raw data packet flow. By using the Ranker algorithm, the authors were able to determine from the entire feature set for the most influential features. The method measures Information Gain as described in the equation below.

$$\text{Information Gain}\left(\text{Class, Attribute}\right) = H\left(\text{Class}\right) - H\left(\text{Class / Attribute}\right)$$

Capture files from existing Bot attacks such as those of Keliho-Hlux, Conficker, Storm, Zeus, and Waledac were used to train the system's classification module. The datasets were PCAP captures. 90% of the dataset was used as training set while 10% formed the testing set. The classifier validity was tested by comparing results of the predicted against those of the experiments using the Pearson product-moment coefficient derived by the formula below [42].

$$r = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}}$$

A 99.7% accuracy level using Random Forest Algorithm with 10 trees was attained by the classifier as is presented in **Table 3**. A receive-operation (ROC) curve of various classifiers is presented in **Figure 7**. The Random Forest is seen to outperform all other machine learning algorithms like Naïve Bayes and SVM. The presented architecture ensures fault tolerance and dynamically adapts to various network situations [42]. The model can be applied in peer-to-peer security modules of threat detection.

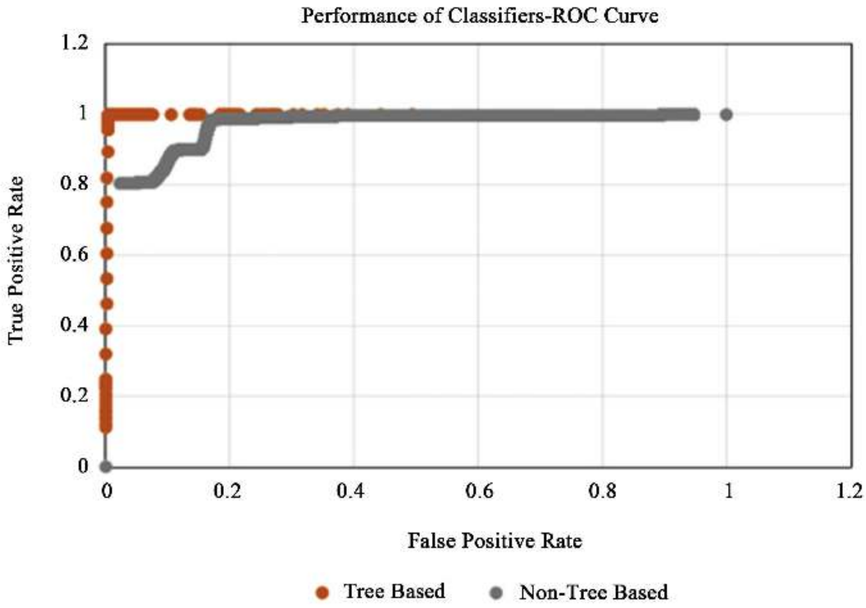


Figure 7. Classifiers' performance comparison.[6]

Table 3. Accuracy Measures of the proposed classifier.

True Positive Rate	False Positive Rate	Precision	Recall	Class
0.998	0.003	0.999	0.998	Malicious
0.997	0.002	0.996	0.997	Non-malicious

3.7. MDRA-Based DDoS Detection Technique

Reference [43] proposes an almost perfect technique for detecting DDoS attacks using Multivariate Dimensionality Reduction Analysis (MDRA). This technique combines the features of Multivariate Correlation Analysis (MCA) and Principal Component Analysis (PCA) with aim to increase detection efficiency, reduce resource consumption and computing complexity, as well as handle large network traffic in Big Data. Even though the technique is still theoretical, its practicality will result in better detection mechanisms and reduced resource consumption. A KDD Cup 1999 dataset is used for verification against the novel algorithm. A flowchart for the novel method is illustrated in Figure 8.

The PCA method helps in obtaining P principal components. Linear combination for the maximum variance forms the first principal component. In the event that the first principal component does not satisfy the total reflection of the original variable, a second linear combination is formed. In their analogy, a sample set X of network traffic having n samples each with a dimension d then the principal components can be illustrated as below.

$X = \{X_1, X_2, \cdots, X_n\}$ and $X_i = (x_{i1}, x_{i2}, \cdots, x_{id}) R^d, i = 1, 2, \cdots, n$. A DDoS attack detection algorithm based on MDRA is shown in Figure 9. Using Precision, FPR, TNR, and DR formulae, this approach helps in DDoS attack detection using MDRA and MCA [43].

[6] Singh, K., Guntuku, S. C., Thakur, A., and Hota, C. (2014) Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests. *Information Sciences*, **278**, 492, Figure 4.

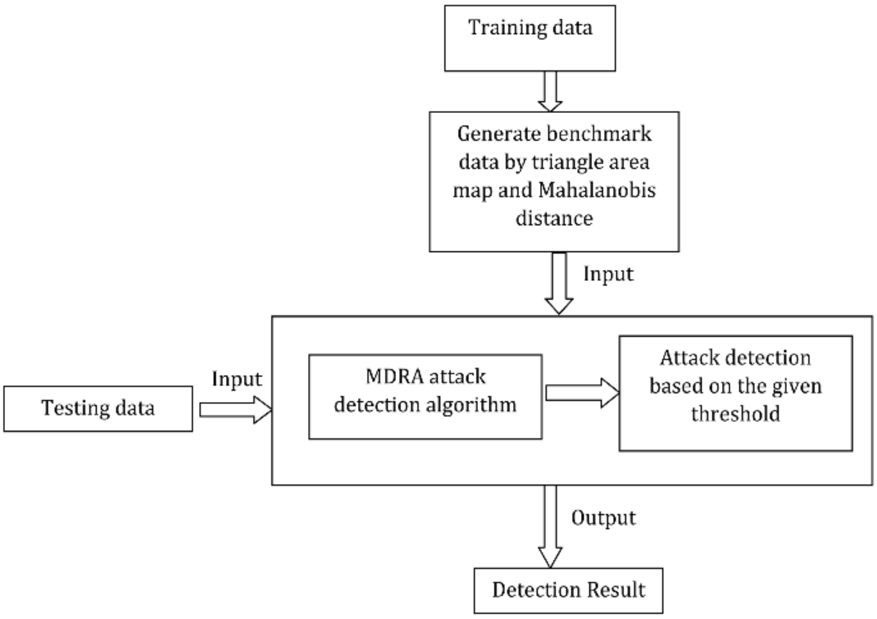
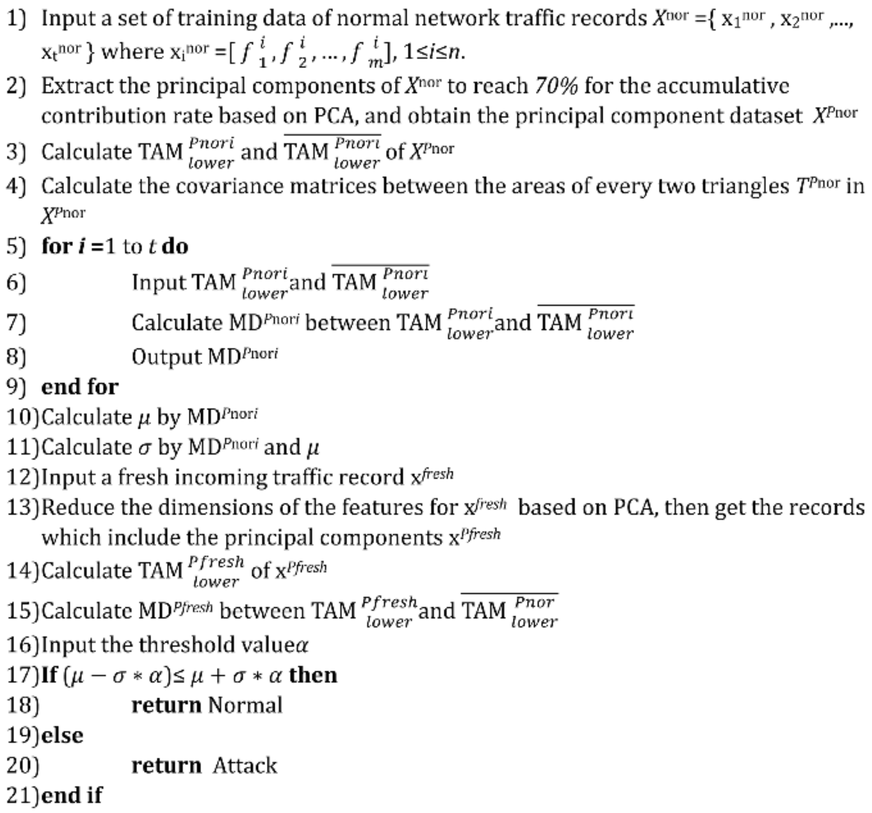

Training data
Generate benchmark data by triangle area map and Mahalanobis distance
Input
Testing data
Input
MDRA attack detection algorithm
Attack detection based on the given threshold
Output
Detection Result

Figure 8. Attack detection flowchart.[7]



```
Input a set of training data of normal network traffic records $X^{nor}=\{x_1^{nor}, x_2^{nor},..., x_t^{nor}\}$ where $x_i^{nor}=[f_1^i, f_2^i, \ldots, f_m^i]$, $1\le i\le n$.
Extract the principal components of $X^{nor}$ to reach $70\%$ for the accumulative contribution rate based on PCA, and obtain the principal component dataset $X^{Pnor}$
Calculate TAM $_{lower}^{Pnori}$ and $\overline{\text{TAM}}\,_{lower}^{Pnori}$ of $X^{Pnor}$
Calculate the covariance matrices between the areas of every two triangles $T^{Pnor}$ in $X^{Pnor}$
for $i$ =1 to $t$ do
    Input TAM $_{lower}^{Pnori}$and $\overline{\text{TAM}}\,_{lower}^{Pnori}$
    Calculate MD$^{Pnori}$ between TAM $_{lower}^{Pnori}$and $\overline{\text{TAM}}\,_{lower}^{Pnori}$
    Output MD$^{Pnori}$
end for
Calculate $\mu$ by MD$^{Pnori}$
Calculate $\sigma$ by MD$^{Pnori}$ and $\mu$
Input a fresh incoming traffic record x$^{fresh}$
Reduce the dimensions of the features for x$^{fresh}$ based on PCA, then get the records which include the principal components x$^{Pfresh}$
Calculate TAM $_{lower}^{Pfresh}$ of x$^{Pfresh}$
Calculate MD$^{Pfresh}$ between TAM $_{lower}^{Pfresh}$and $\overline{\text{TAM}}\,_{lower}^{Pnor}$
Input the threshold value$\alpha$
If $(\mu-\sigma*\alpha)\le\mu+\sigma*\alpha$ then
    return Normal
else
    return Attack
end if
```

Figure 9. MDRA-based DDoS detection algorithm.[8]

[7] Jia, B., Ma, Y., Huang, X., Lin, Z., and Sun, Y. (2016) A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data. *Mathematical Problems in Engineering*, **2016**, 3, Figure 3.

[8] Jia, Ma, Huang, Lin, and Sun, A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data , 4, Algorithm 1.

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

$$\text{TNR} = \text{TN}/(\text{FP}+\text{TN})$$

$$\text{FPR} = \text{FP}/(\text{FP}+\text{TN})$$

$$\text{DR} = \text{TP}/(\text{TP}+\text{FN})$$

where:

1) TP is True Positive and represents attack numbers correctly classified as attacks,

2) FP is False Positive and represents normal record numbers in correctly classified as attacks,

3) TN is True Negative and represents normal record numbers correctly classified as normal records,

4) FN is False Negative and represents attack numbers incorrectly classified normal records.

Using a set between 1 and 3 with an increment of 0.2, Table 4 shows the resulting detection results of TP, TN, FN, and FP. Figure 10 and Figure 11 illustrates the tabulated detection results graphically for precision and TNR respectively. The approach led to high precision rate of almost 100% in True Negative Rate (TNR) with reduced computing time which equated to an eighth of the previous CPU time by MCA method. And even though the process was theoretical in nature, its practicability could alter how DDoS attacks are detected in Big Data environment. It would lead to greater efficacy even with heavy network traffic.

The strengths and limitations of the various specific cloud computing DDoS detection techniques as stipulated in this section are illustrated in Table 5.

Table 4. TP, FP, TN, and FN Results using MDRA and MCA.

α	Indicator on MDRA basis				Indicators on MCA basis			
	TP	FP	TN	FN	TP	FP	TN	TN
$\alpha = 1$	166,299	278	60,315	63,554	223,587	1743	58,850	6266
$\alpha = 1.2$	166,299	249	60,344	63,554	221,873	1469	59,124	7980
$\alpha = 1.4$	166,292	227	60,366	63,561	206,504	1313	59,280	23,349
$\alpha = 1.6$	166,289	217	60,376	63,564	191,190	1214	59,379	38,663
$\alpha = 1.8$	166,289	204	60,389	63,564	190,394	1159	59,434	39,459
$\alpha = 2$	166,289	194	60,399	63,564	190,342	1115	59,478	39,511
$\alpha = 2.2$	166,289	191	60,402	63,564	190,311	1065	59,528	39,542
$\alpha = 2.4$	166,289	188	60,405	63,564	190,277	1027	59,566	39,576
$\alpha = 2.6$	166,282	180	60,413	63,571	190,254	988	59,605	39,599
$\alpha = 2.8$	166,282	176	60,417	63,571	190,230	953	59,640	39,623
$\alpha = 3$	166,282	172	60,421	63,571	190,199	927	59,666	39,654

Table 5. Specific DDoS detection techniques based on author.

Author/ Date	Detection Technique	Performance Evaluation metrics	Datasets	Tools used	Advantages	Disadvantages	Limitations
Csubak, Szucs, Voros, and Kiss, 2016	Big data Testbed for Network Attack detection	Packets per second rate	Simulated network traffic using NS3, Normal traffic data ranging from MBs to GBs	1) Snort 2) NS3 3) Wireshark, 4) Python-dpkt package	1) Using Snort, a user defines their own rules for which network traffic is analyzed against 2) Snort can analyze and log network packets in real time. 3) Big data testbed is capable of handling hundreds of GB network traffic	1) Since the technique checks the already set packet rates threshold, attacks occurring below the set threshold are undetectable	1) The technique has not been applied on large scale rather only tested via simulation
Chen Xu, Mahalingam Ge, Nguyen, Yu, and Lu, 2016	Cloud computing based network monitoring and threat detection system for critical infrastructures	Traffic volume per minute to detect abnormal behavior	Uses real Large traffic data from logs	1) Hadoop 2) Spark 3) Mysql database 4) PHP with AJAX	1) Three-fold solution of network monitoring, threat detection, and system performance 2) Fast data processing by concurrently running Hadoop and Spark 3) Easy for network administrators to detect any abnormal network behaviors	1) Accuracy level relies on collected data samples. 2) Cannot detect dynamic attacks 3) New components require extra monitoring agents	1) Accuracy of the detection greatly relies on collected traffic information 2) The technique is only suitable for analyzing static data
Osanaiye, Choo, and Dlodlo, 2016	Conceptual Cloud DDoS change-point detection framework	Packet inter-arrival time (IAT)	Conceptual network traffic data. No simulation or real data tests done.	1) CUSUM algorithm	1) Easily detects abnormal packet pattern by comparing with normal packet behavior 2) Able to detect DDoS attacks using statistical anomaly 3) IAT feature helps determine the probability of a DDOS attack long before it occurs	1) Abnormally based attacks cannot learn new attack types 2) Leads to a lot of false positives and false negatives and no optimal threshold is set	1) There is no standard mechanism to determine the optimal threshold for determining abnormal traffic
Borisenko, Smirnov, Novikova, and Shorov, 2016	DDOS attack detection in cloud computing using Data Mining Techniques	Incoming network traffic data vectors	Uses Hping to simulate SYN, NTP, and HTTP-based traffic data, source IP and port, destination IP and ports, packets, data bytes length	1) Real Service in Virtual Network Framework (RSVNet) 2) Ansible 3) Siege 3.1.0 4) Hping	1) The technique performs test on real and virtual nodes 2) RSVNet is used to implement and create new protection mechanisms, and attack scenarios 3) Fast data processing and prediction of less than one second 4) This technique can be tailored to independently detect TCP, UDP, and ICMP flood attacks	1) For attack detection, powers have to be set to act as threshold and hence the process is not dynamic in nature 2) Separate attacks require separate classification models	1) The technique has no capacity for complex attacks
Hameed, Ali, and IT Security Labs, June 2015	Live DDOS Detection with Hadoop	File size, number of files before detection, path to save captured file	Real-time Live network traffic	1) HADEC 2) Apache Hadoop	1) Ability to analyze huge volume of DDOS flood attacks in less time	1) Hadoop does not offer parallelism for small log files 2) Capturing consumes over half of the overall detection	1) Using small log files implies reduced number of attackers

Continued

Veetil and Gao, 2013	Real-time Intrusion Detection System by using Hadoop and Naive Bayes Classification	Packets per second, packets per minute	10% KDD intrusion detection dataset, Live network stream packets as training data	1) Snort 2) Tshark 3) D3	1) Increased parallelism due to the Naive Bayes algorithm 2) Using Hadoop-based Naive Bayes algorithm training speed increases implying faster detection rates 3) High detection rate of over 434 network packets per minute	1) This approach compared its performance to a previous approach rather than testing new attacks	1) The technique may not perform well in a distributed environment since its ineffective in a heterogenous cluster
Cepheli, Buyukcorak, and Kurt, 2016	Hybrid Intrusion Detection System (H-IDS) for DDOS attacks	Protocol frequencies, packet sizes, packet inter-arrival times	DARPA 2000 dataset, Real training data from a past penetration test of commercial bank in Turkey	1) Gaussian Mixture Model 2) SNORT	1) Combines the power of anomaly and signature based techniques for a more accurate detection 2) Combining anomaly and rule-based detection reduces detection delays 3) Easily integrates as a module with other IDS	1) Cannot detect complex DDoS attacks 2) Cannot detect attacks internally generated attacks	1) Training data does not reflect real network data implying reduced performance
Singh, Guntuku, Thakur, and Hota, 2014	Using Random Forests for Big Data Analytics in Peer-to-Peer Botnet detection	Packet buffer sizes	CAIDA datasets. 84,030 instances of mixed traffic	1) Hadoop 2) Mahout 3) MapReduce 4) Tshark using Libpcap library	1) Usable for predictive data modeling as Mahout ensures high data accuracy and time efficacy 2) Ease of detecting peer-to-peer attacks due to ability to process high bandwidths in real-time with 30 seconds delay	1) High computational costs due to the use of MapReduce jobs 2) Cannot run with non-distributed classifiers due to the large space required by data and JVM	1) Inability to block traffic from botnets or isolate compromised machines
Korad, Kadam, Deore, Jadhav, and Patil, 2016	Using Hadoop on Live Network to detect DDOS	Packet file sizes and packet pairs	Simulation of Live HTTP GET packet, UDP, TCP, and ICMP packet. Masked timestamp	1) Hadoop 2) Wireshark	1) Ability to handle and analyze petabytes of data with ease 2) Hadoop clustering help in harnessing the processing power of many computer as one 3) Ease of management and paremeter setting through a web interface	1) Cannot be used to detect internal attacks such as from memory corruption 2) High computational costs from combining multiple nodes	1) Ineffective with few nodes due to the high computational costs
Jia, Ma, Huang, Lin, and Sun, 2016	Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data	Precision rate, TNR, memory resource, computing complexity, and time cost		Knowledge Discovery and Data Mining (KDD) Cup 1999 data set for training and testing. The data set is real	1) High precision rates of almost 100% for True Negative Rates (TNR) 2) Reduced CPU computation cost 3) Reduced memory consumption compared to MCA based techniques 4) Network DDoS attacks in real-time	1) The technique only depicts abnormal network traffic after it has been predefined	1) Since the approach is theoretical, it may not be possible to ascertain its effectiveness

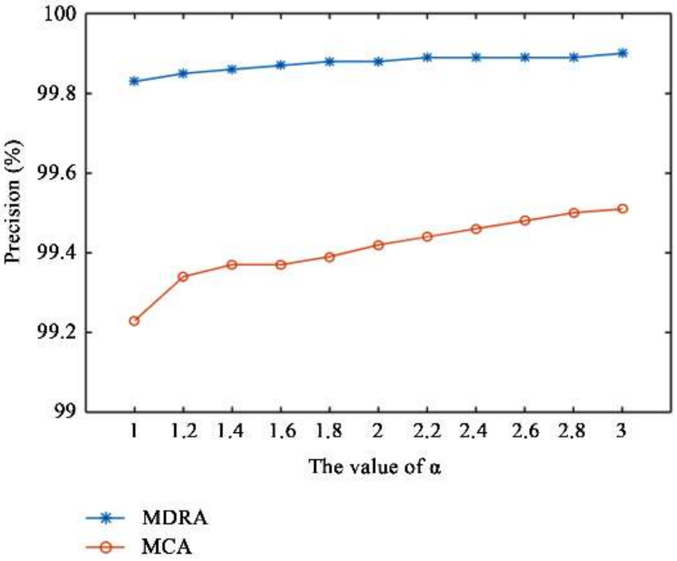


Figure 10. Using precision to compare detection based on MDRA and MCA.[9]

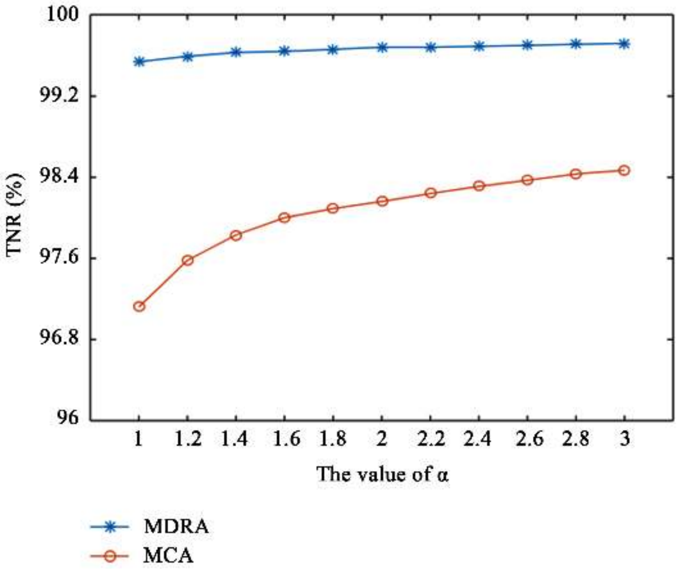


Figure 11. Using TNR to compare detection based on MDRA and MCA.[10]

4. Contrastive Analysis

Each discussed technique possesses its strengths and limitations. Their strengths are based on the need to fill a certain limitation offered by a previous technique. Before a scholar assumes the feasibility of their technique they make comparisons of their methods to those of their predecessors. To study an ideology, a researcher has to consider all the variants and objects making it up and their interrelation [44]. Further, they need to apply objective research to analyze and

[9] Jia, Ma, Huang, Lin, and Sun, A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data, 4, Figure 4.

[10] Jia, Ma, Huang, Lin, and Sun, A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data , 4, Figure 5.

contrast their findings.

With DDoS attacks, contrastive analysis is greatly applied when using training data set to prepare the detection mechanism. For instance, datasets from previously known attacks are used to first test the new method before applying it into real-time situation. For instance, [10] used a DARPA 2000 dataset with already known anomalies so as to test if their technique could detect anomalies compared to other techniques that utilized the same set. Similarly, the same technique used data set from a previous penetration test done on a Turkish commercial bank. The tests results are already known and using the dataset as input is only meant to compare the technique's output to that of the penetration test. Other than mere detection, the use of datasets helps in determining the accuracy levels of the current technique in comparison to previous techniques.

In most instances, the use of contrastive research is successful since it is possible to adjust parameters to fit the required outcome or to alter the expected outcome to a given level. In the technique presented by [42] to combat botnets attack in a peer-to-peer network, training data was pulled from previous Bot attacks. These were the Conficker, Storm Zeus, Waledac, and Keliho-Hlux Bot attacks that then helped in creating a classification mechanism for this technique. The experimental results compared to the already predicted results helped to gauge the efficacy of the technique. The researchers would then alter their parameters to determine the attack outcome on those features.

In other scenarios, attacks are directly launched on hosts and the detection mechanisms deployed to try and detect. This is enabled through the use of rules that define attack behaviors. SNORT is one such tool that has rules defined to detect an attack based on those rules and threshold. Additionally, setting a threshold level helps in detecting traffic anomalies by raising an alarm if traffic goes beyond such level. However, threshold may not be as effective. Attacks such as HTTP GET consume little bandwidth resulting in insignificant network traffic. Using threshold as a measure to such attacks would lead to a lot of false negatives.

5. Conclusions and Future Work

There is need to ensure that data in the cloud is safe from any form of attack. Securing the cloud is hard but inevitable. One among the many feared attacks in the cloud is the Distributed Denial of Service attack. As this paper has expounded, the techniques against DDoS attacks borrow greatly from the already tested traditional techniques. However, no technique has proven to be perfect towards the full detection and prevention of DDoS attacks. In determining the detection or prevention mechanism for a DDoS attack, the motivation behind the attack has to be determined. Reference [45] stipulates seven motivations for DDoS attacks namely; financial and economic gain, slow network performance, ideological belief, revenge, intellectual challenge, cyberwarfare, and service unavailability.

One or multiple motivations can lead to an attack. Future researchers need to develop techniques that not only detect an attack but also intelligently identify the attacker's methods and the traffic rates. As well, the mechanisms should be capable of determining the legitimacy of the source of the attack.

Most of the previously proposed and implemented approaches can further be advanced to ensure an increase in the IDS performance. For instance, instead of concentration on one point for detecting an attack, the approach can work towards having distributed points of attack detection and correction. To increase the detection and inference speed, the approaches can further provide distributed points of attack analysis separate from the attack points but relaying attacks descriptions to a central point. This would ensure that all facets of an attack are determined without negatively affecting performance.

References

[1] Subramaniam, T. and Bethany, D. (2016) Preventing Distributed Denial of Service Attacks in Cloud Environments. *International Journal of Information Technology, Control and Automation*, **6**, 23-32. https://doi.org/10.5121/ijitca.2016.6203

[2] Sivamohan, S., Veeramani, R., Liza, K., Krishnaveni, S. and Jothi, B. (2016) Data Mining Technique for DDoS Attack in Cloud Computing. *International Journal of Computer Technology and Applications*, **9**, 149-156.

[3] Masdari, M. and Marzie, J. (2016) A Survey and Taxonomy of DoS Attacks in Cloud Computing. *Security and Communication Networks*, **2**, 3274-3751. https://doi.org/10.1002/sec.1539

[4] Bonquet, A. and Martine, B. (2017) A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defense in Cloud Computing. *Future Internet*, **9**, 1-9. https://doi.org/10.3390/fi9030043

[5] Kaur, A. and Anupama, K. (2015) A Review on Various Attack Detection Techniques in Cloud Architecture. *International Journal of Advanced Research in Computer Engineering & Technology*, **4**, 3861-3867.

[6] Kene, S.G. and Deepti, P.T. (2015) A Review on Intrusion Detection Techniques for Cloud Computing and Security Challenges. 2*nd International Conference on Electronics and Communication Systems*, Coimbatore, 26-27 Februaty 2015, Vol. 2, 227-231. https://doi.org/10.1109/ECS.2015.7124898

[7] Deshmukh, R.V. and Kailas, K.D. (2015) Understanding DDoS Attack & Its Effect in Cloud Environment. *Procedia Computer Science*, **49**, 202-210. https://doi.org/10.1016/j.procs.2015.04.245

[8] Sattar, I., *et al.* (2015) A Review of Techniques to Detect and Prevent Distributed Denial of Service (DDoS) Attack in Cloud Computing Environment. *International Journal of Computer Applications*, **115**, 23-27. https://doi.org/10.5120/20173-2370

[9] Navaz, S., *et al.* (2013) Entropy Based Anomaly Detection System to Prevent DDoS Attacks in Cloud. *International Journal of Computer Applications*, **15**, 42-47.

[10] Ankita, P. and Fenil, K. (2015) Survey on DDoS Attack Detection and Prevention in Cloud. *International Journal of Engineering Technology, Management, and Applied Sciences*, **3**, 43-47.

[11] Modi, C., Dhiren, P., Bhavesh, B., Avi, P. and Muttukrishnan, R. (2013) A Survey on Security Issues and Solutions at Different Layers of Cloud Computing. *The Journal of Supercomputing*, **63**, 561-592. https://doi.org/10.1007/s11227-012-0831-5

[12] Kacha, C.C., *et al.* (2013) Improved Snort Intrusion Detection System using Modified Pattern Matching Technique. *International Journal of Emerging Technology and Advanced Engineering*, **3**, 81-88.

[13] Parwani, D., *et al.* (2015) Various Techniques of DDoS Attacks Detection and Prevention at Cloud: A Survey. *Oriental Journal of Computer Science & Technology*, **8**, 110-120.

[14] Dewal, P., *et al.* (2016) A Survey of Intrusion Detection Systems and Secure Routing Protocols in Wireless Sensor Networks. *International Journal for Research in Emerging Science and Technology*, **3**, 16-20.

[15] Modi, K. and Abdul, Q. (2014) Detection and Prevention of DDoS Attacks on the Cloud using Double-TCP Mechanism and HMM-Based Architecture. *International Journal of Cloud Computing and Services Science*, **3**, 113-120.

[16] Chawla, I., *et al.* (2015) DDoS Attacks in Cloud and Mitigation Techniques. *International Journal of Innovative Science*, *Engineering & Technology*, **2**, 596-600.

[17] Reddy, S.V., *et al.* (2012) Efficient Detection of Ddos Attacks by Entropy Variation. *IOSR Journal of Computer Engineering*, **7**, 45-67.
https://doi.org/10.9790/0661-0711318

[18] Girma, A., *et al.* (2015) Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment. 12*th International Conference on Information Technology—New Generations*, Las Vegas, 13-15 April 2015, 212-217. https://doi.org/10.1109/ITNG.2015.40

[19] Nitesh, B., *et al.* (2017) Mitigating Distributed Denial of Service Attack in Cloud Computing Environment using Threshold based Technique. *Indian Journal of Science and Technology*, **3**, 1-7.

[20] Iyengar, N. and Gopinath, G. (2015) Trilateral Trust Based Defense Mechanism against DDoS Attacks in Cloud Computing Environment. *Cybernetics and Information Technologies*, **15**, 122. https://doi.org/10.1515/cait-2015-0033

[21] Al-Hemairy, M., *et al.* (2009) Towards More Sophisticated ARP Spoofing Detection/Prevention Systems in LAN Networks. *International Conference on the Current Trends in Information Technology*, Dubai, 15-16 December 2009, 1-6.
https://doi.org/10.1109/CTIT.2009.5423112

[22] Jeyanthi, N. and Chris, M. (2014) A Virtual Firewall Mechanism using Army Nodes to Protect Cloud Infrastructure from DDoS Attacks. *Cybernetics and Information Technologies*, **14**, 71-85. https://doi.org/10.2478/cait-2014-0034

[23] David, J. and Ciza, T. (2015) DDoS Attack Detection using Fast Entropy Approach on Flow-Based Network Traffic. *Procedia Computer Science*, **50**, 30-36.
https://doi.org/10.1016/j.procs.2015.04.007

[24] Singh, N., *et al.* (2015) Comprehensive Study of Various Techniques for Detecting DDoS Attacks in Cloud Environment. *International Journal of Grid and Distributed Computing*, **8**, 119-126. https://doi.org/10.14257/ijgdc.2015.8.3.12

[25] Adetunmbi, A.O., *et al.* (2008) Network Intrusion Detection Based on Rough Set and K-Nearest Neighbor. *International Journal of Computing and ICT Research*, **2**, 60-66.

[26] Gourkhede, M.H. and Peter, T. (2014) Preserving Privacy and Illegal Content Distribution for Cloud Environment. *International Journal of Computing and Technology*, **1**, 124-148.

[27] Gayatri, P., *et al.* (2015) Comprehensive Comparative Study on Intrusion Detection System in Cloud Computing. *International Journal for Research in Applied Science*

& Engineering Technology, **3**, 926-930.

[28] Parwani, D. and Amit, D. (2017) Prevention Mechanisms of DDoS Attacks: A Critical Review. *International Journal of Science, Engineering and Technology*, **5**, 99-112.

[29] Dastjerdi, A.V., *et al.* (2009) Distributed Intrusion Detection in Clouds using Mobile Agents. *3rd International Conference on Advanced Engineering Computing and Applications in Sciences*, Sliema, 11-16 October 2009, 175-180. https://doi.org/10.1109/ADVCOMP.2009.34

[30] Karthi, M.M., *et al.* (2013) Intrusion Detection System for Cloud System using Intelligent Agents. *International Journal Of Engineering And Computer Science*, **2**, 1868-1873.

[31] Sahardi, R.M. and Vahid, G. (2013) New Approach to Mitigate XML-DOS and HTTP-DOS Attacks for Cloud Computing. *International Journal of Computer Applications*, **72**, 27-31. https://doi.org/10.5120/12579-9201

[32] Subapriya, S. and Nathan, R. (2014) DNIDPS: Distributed Network Intrusion Detection and Prevention System. *International Journal of Innovative Science, Engineering & Technology*, **6**, 56-67.

[33] Lonea, A.M., *et al.* (2012) Detecting DDoS Attacks in Cloud Computing Environment. *International Journal of Computers Communications & Control*, **8**, 70. https://doi.org/10.15837/ijccc.2013.1.170

[34] Patel, S. and Fenil, K. (2016) A Review Paper of an Encryption Scheme using Network Coding for Energy Optimization in MANET. *International Conference on Wireless Communications, Signal Processing and Networking*, Chennai, 23-25 March 2016, Vol. 34, 45-67. https://doi.org/10.1109/WiSPNET.2016.7566298

[35] Csubak, D., Szucs, K., Voros, P. and Kiss, A. (2016) Big Data Testbed for Network Attack Detection. *Acta Polytechnica Hungarica*, **13**, 47-57.

[36] Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W. and Lu, C. (2016) A Cloud Computing Based Network Monitoring and Threat Detection System for Critical Infrastructures. *Big Data Research*, **3**, 10-23. https://doi.org/10.1016/j.bdr.2015.11.002

[37] Osanaiye, O., Choo, K.R. and Dlodlo, M. (2016) Distributed Denial of Service (DDoS) Resilience in Cloud: Review and Conceptual Cloud DDoS Mitigation Framework. *Journal of Network and Computer Applications*, **67**, 147-165. https://doi.org/10.1016/j.jnca.2016.01.001

[38] Cepheli, O., Buyukcorak, S. and Kurt, K.G. (2016) Hybrid Intrusion Detection System for DDoS Attacks. *Journal of Electrical and Computer Engineering*, **2016**, Article ID: 1075648. https://doi.org/10.1155/2016/1075648

[39] Hameed, S. and Ali, U. (2016) Efficacy of Live DDoS Detection with Hadoop. *IEEE/IFIP Network Operations and Management Symposium*, Istanbul, 25-29 April 2016. https://arxiv.org/pdf/1506.08953.pdf

[40] Korad, S., Kadam, S., Deore, P., Jadhav, M. and Patil, R. (2016) Detection of Distributed Denial of Service Attack with Hadoop on Live Network. *International Journal of Innovative Research in Computer and Communication Engineering*, **4**, 92-98.

[41] Veetil, S. and Gao, Q. (2014) Real-Time Network Intrusion Detection using Hadoop-Based Bayesian Classifier. In: Akhgar, B. and Arabnia, H.R., Eds., *Emerging Trends in ICT Security*, Elsevier Inc., 281-299. https://doi.org/10.1016/B978-0-12-411474-6.00018-9

[42] Singh, K., Guntuku, S.C., Thakur, A. and Hota, C. (2014) Big Data Analytics

Framework for Peer-to-Peer Botnet Detection using Random Forests. *Information Sciences*, **278**, 488-497. https://doi.org/10.1016/j.ins.2014.03.066

[43] Jia, B., Ma, Y., Huang, X., Lin, Z. and Sun, Y. (2016) A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data. *Mathematical Problems in Engineering*, **2016**, Article ID: 1467051. https://doi.org/10.1155/2016/1467051

[44] Jin, W. and Yu, Z. (2016) The Analysis of Information System Security Issue Based on Economics. *International Conference on Information Engineering and Communications Technology*, Kunming, 21-22 2016. https://doi.org/10.12783/dtetr/iect2016/3801

[45] Prasad, K.M., Reddy, R.A. and Rao, K.V. (2014) DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms—A Survey. *Global Journal of Computer Science and Technology*: *E Network Web & Security*, **14**, 16-32.