Multi-Model Anti-DDos Framework For Detection And Mitigation Of High Rate Ddos Attacks In The Cloud Environment

A.Saravanan, S.Sathyaa Bama

Abstract — Cloud computing has become the most inevitable concept in the field of computer science. It renders various services by providing resources such as data storage, software, applications and even more for the cloud users over the internet. The most significant weakness is making the resources unavailable for legitimate users by flooding the network with the attack packets. High rate distributed denial of service (DDoS) attacks are the most common threat for the cloud server in which a large number of attack packets are sent than normal legitimate packets. This paper presents the multi-model based Anti-DDoS framework to mitigate the high rate DDoS attacks for the cloud environment. The method makes use of the graphical turing test model and authentication model for preventing the DDoS attacks along with count based filtering model and statistical based filtering model for detecting the attacks with the support of access control lists. From the experimental analysis, it is found that the proposed model has an attack detection rate of 91% and 8% of the dropout rate which is better than the existing techniques.

Index Terms — Highrate DDoS Attack, DDoS Detection and Mitigation, Count-based Filtering, Statistical based Filtering, Virtual Firewall, Authentication.

1 INTRODUCTION

OWING to the advantage of cloud computing, plenty of users are influenced by its theories. In the 1960s, the basic idea of sharing the computer system by two are more people has been initiated by DARPA (Defense Advanced Research Projects Agency), an agency that is responsible for developing the technologies for military use. Later, during the 1970s, virtualization began to emerge which is the key factor of cloud computing. However, in the late 1990s, the cloud became a very prevalent concept as the users had a superior understanding of the services provided by the cloud. Due to the growth of the internet, cloud computing has now become the obvious technology as it carries significant services over the internet. The main idea behind cloud computing is to pay as per the use. The cloud user can rent the services data storage, computing power or software as an alternative over buying the required software and hardware. The clouds can be either public or private. The main advantage of using cloud computing is that the vendor is responsible for all the administrative activities including maintenance and troubleshooting. For the users of the cloud, it is simple to use and easy to complete the intended task.

The public clouds can be either used for personal use or business use. The services provided by the cloud can be of three types such as infrastructure as a service (IaaS), software as a service (Saas), platform as a service (PaaS). Similarly, private clouds are created by business organizations for their personal use. The customers approach cloud because of the few different characteristics such as on demand service, pay as per the use and elasticity. An increase in the usage of the cloud also increases vulnerabilities such as privacy issues, legal issues, resource availability issues and even more [1]. Among them, the unavailability of resources is the major concern in the cloud environment. The unavailability of service can be successfully achieved by implementing a DoS attack which aims at sending the unwanted floods of traffic to the targeted server as a legitimate user [2]. Another variant of the DoS attack is the distributed denial of service attack which uses the same idea but employs a group of systems termed as bots to flood the unwanted request to the target server. This is the most highly implemented attack in a cloud environment [3]. The DDoS attacks can be either low rate or high rate attacks in which low rate attacks are implemented by intermittently transmitting burst attack packets over short periods of time continuously or transmitting packets attack packets at a constant low-rate [4], whereas, for high rate attacks, the attacker transmits the high rate packets repeatedly over a period of time to make the resources unavailable [5].

In this work, a multi-model based anti-DDoS framework for detection and mitigation of high rate DDoS attacks in the cloud environment has been presented. The model makes use of two components such as virtual firewall and virtual verifier node. The verifier node does all the verification for the packets forwarded by the virtual firewall. The verifier initially verifies whether the requests are sent by the bots or the normal user using the graphical turing test. Then the user and the device through which the requests are sent are examined using an authentication model. These two methods are used as a preventive measure for DDoS attacks. Then as the next step, the count based filtering model and statistical based filtering are employed by the verifier for detecting the attacks. On detecting the attacks, the verifier nodes update the suspicious list, white list, and blacklist, through which the subsequent packets are either forwarded to the cloud, dropped the packets or forwarded to the verifier node for further verification. As a result, the DDoS attacks are detected for protecting the resources from the attackers. The rest of the paper is presented as follows. Section 2 discusses the works from the literature related to the proposed model. Section 3 presents the overall framework of the proposed multi-model based anti-DDoS model. The attack prevention steps and the attack
Detection steps of the proposed model are elaborated in section 4 and section 5 respectively. Section 6 discusses the experimental analysis and results obtained for the proposed model. Finally, the conclusion of the paper is presented.

2 RELATED WORKS

Several techniques have been suggested in the literature for detecting and mitigating DDoS attacks on cloud infrastructure. A typical scenario of DDoS attack in the cloud environment and the common defense mechanism are presented by Saravanan et al., [6]. The defense mechanism has three main categories as attack prevention, attack detection, and attack discovery. The authors proposed a context that includes prevention and detection process. The author suggested three methods for graphical turing tests such as VISUALCOM, IMGCOM, and AD-IMGCOM for preventing the DDoS attacks among which AD-IMGCOM is considered to be the most complex mechanism. In AD-IMGCOM, the users are given the shuffled parts of the image with an anomaly and the users are demanded to arrange the complete picture and identify the anomaly part [7]. Similarly, the method has been enhanced by including the count based filtering for detecting the attacks [8]. The most widely used count based filtering is providing the limit to the number of requests that originate from the same source or different sources. Based on these behaviors, the subsequent packets originating from the same host are dropped and are not processed by which the attacks are controlled [9]. The authors Jeyanthi and Mogankumar [10] uses a similar method to limit the number of requests arriving from the same host and if it exceeds the given threshold value, the IP address is stored in the blacklist and the corresponding packets are dropped. Generally, the count based filtering techniques include computation of hop count as a significant parameter in detecting the attack packets. Most of the method uses the IP2HC table for comparing the results. On receiving the packets, the destination computes the hop count. The hop count is also retrieved from the IP2HC table stored in the server and is compared with the computed hop count. If there is a match, the packet is considered as the legitimate packet or else it is detected as spoofed packet [11]. On the other hand, hop-count diversity is utilized by Al-Haidari et al., [12]. The authors employ an access control list that contains the computed TTL value for accepting and dropping the packets. If the TTL value does not match or when the IP address is not in the white list, the verification process will be carried out that contains a graphical turing test [13]. Hop count based filter with cloud defender is utilized by the authors Karnwal et al., [14]. However, the method is utilized only for detecting some specific DDoS attacks such as HTTP and XML based DDoS. Several other filtering methods have been introduced with the aim of mitigating the attacks. Pushback based approach by utilizing the resource regulation was suggested by Ioannidis & Bellovin [15]. However, the method consumes more computational power and turnaround time due to which it is considered to be less effective. To protect the servers from attackers, a simple distance based computation has been suggested by Chapade et al., [16]. Similarly, distance-based defense system for DDoS attacks has been proposed by You [17] with three major components such as detection, traceback, and traffic control. However, these methods does not utilize the resources of the server properly. For filtering the DDoS attacks, a data mining technique called neural network has been implemented [18]. The main drawback of this method is that the method provides less accurate results based on inherent settings. To increase the resource utilization and the functionality of the system, a method has been suggested that uses three filters to ensure the authenticated access to the targeted system. The method practices user authentication and device authentication along with the filter to limit the services to the user [19]. Statistical based analysis has been suggested in the literature to effectively detect the DDoS attack. A machine learning technique that incorporates the statistical analysis for monitoring the online traffic has been introduced and the method is termed as stream processing based big data framework [20]. However, as the method uses only the statistical analysis about the incoming and outgoing packets, the results obtained are not accurate. Based on the knowledge gained from the literature, the proposed method uses several models for mitigating the high rate DDoS attack, a serious challenge in the cloud environment.

3 PROPOSED MULTI-MODEL BASED ANTI-DDOS FRAMEWORK

The overall architecture of the proposed multi-model based anti-DDOS framework is shown in Fig. 1. The proposed model has two main components.

The first component is the virtual firewall (VF) and the second is the virtual verifier nodes (VVN) which is similar to the idea proposed by Sqalli et al., [13]. Though the base for the proposed model is the EDoS-Shield mechanism, the proposed model also has several layers of defense to protect the cloud from the DDoS attack using various models such as graphical turing test, count based filtering model, statistical based filtering model. The virtual firewall stores two access control lists such as a blacklist and a white list consisting of IP addresses of the source system. It acts as the first layer of defense as it verifies the source IP address with that of the one on the blacklist and white list. Additionally, the suspicious list is also maintained by VVN. These lists are updated by the verifier nodes based on the verification made by them. The
components of the framework are described in the below subsection.

3.1 Virtual Firewall (VF)
A virtual firewall is a device particularly designed for the virtual environment with the intention of filtering and observing network traffic. It can be implemented as software on the virtual machines to examine the input packets. The set of security policies are implemented as a program in which each packet will be verified to ensure secure communication. In this proposed model, the virtual firewall is implemented for routing and filtering packets. It maintains two lists such as a white list and blacklist for making judgments about the incoming packets that aim at acquiring service from the cloud originating from the outside user. Obviously, the white list contains a set of authenticated source IP addresses whereas the blacklist contains a list of unauthenticated source IP addresses. For all the incoming packets, the virtual firewall compares the source IP address of the incoming packet with the set of source IP addresses in the blacklist and if there is a match, the firewall will drop the request initiating from that particular source IP address. On the other hand, if the source IP address of the incoming packet is found in the white list, then the firewall forwards the request towards the requested service towards the cloud. If the IP address is not available in either the black or white list, then the request is forwarded to the VVN for further analysis. Virtual firewalls are generally employed frequently in the virtual environment as it is less expensive when compared to the other services. It has so many advantages such as portability, easy to maintain and quick and fast upgradation. It is mainly practiced to protect the cloud along with the data and its service from unauthenticated users.

3.2 Virtual Verifier Node (VVN)
The second significant component is the virtual verifier node which is responsible for processing all the defense mechanisms concerning the packets sent by the legitimate user inside the cloud infrastructure. Though it is represented as a singular node, the actual implementation involves a group of virtual nodes deployed in the cloud environment. It is considered to be the heart of the security mechanism as it involves several layers of defense. On accepting the incoming packets, the virtual firewall forwards the request to the virtual verifier node which is responsible to examine users of the corresponding packets employing graphical turing tests [21]. This acts as the second layer of defense. The graphical turing test is the most widely applied method for authentication with the purpose of identifying the bots from the legitimate user [22]. On identifying the normal user (not the bot), as the next step, the verifier node sequentially verifies the authentication of the user. This is considered to be the third layer of defense. These two steps act as a prevention strategy against DDoS attacks and are happening at the application layer. As the next layer of defense, the proposed verifier applies attack detection steps such as count based filtering and statistical based analysis. In count based filtering, the thresholds are employed to detect the attacks initiated towards the cloud. The method uses several parameters such as the number of request connections from the same source, number of request connections from various new different sources, hop counts and even more. This acts as the fourth layer of defense. In the statistical based analysis, the ratio of incoming and outgoing packets and flow of packets are analysed to detect the initiated DDoS attacks. This is the fifth layer of defense. The verifier node is responsible for attack detection and is processed at the network layer. Apart from these steps, the verifier node is also an authority for updating the access control lists periodically which helps to improve the performance of the first layer of defense at the virtual firewall.

On identifying the source IP address as the legitimate user, the virtual verifier node sends the packet towards the cloud server. At this point, the source IP address is updated in the white list and thus all the successive requests originating from this source IP address are forwarded by the firewall towards the cloud server for the requested service. If the IP address is not available on the white list, then it is verified at the blacklist. On seeing the match, the firewall will drop the packets and all the subsequent requests from the specific IP address are also dropped. If the IP address is not found in the white list or blacklist, the requests are forwarded to the verifier node. The verifier node checks whether the IP address is available on the suspicious list. If there is a match, it proceeds with further measures to verify the legitimate request. Once the request packets pass all the verification, it is then forwarded to the cloud for the requested service. On identifying the packet as a doubtful at any of the above measures, the IP address will be updated on the suspicious list along with the number of requests arrived from the particular host and the packets are dropped. If the count of the request reaches the particular threshold, then the source IP address is updated in the blacklist due to which all the successive requests originating from this source IP address will be dropped by the virtual firewall in the near future. The responsibilities of the verifier node are consolidated and are shown in Fig. 2.

4 DDOS ATTACK PREVENTION

4.1 Graphical Turing Test
The botnet is a set of connected devices mostly infected by the malware and is controlled as a group which is the biggest threat for the cloud environment. It is most widely used for attempting the distributed denial of service attacks. For preventing the bots from successful DDoS attacks, several preventive methods have been proposed in the literature [23].

The methods can be grouped under five main categories such as challenge-response system [7], crypto puzzles [24], hiding
the servers or resources [10], restricting the access [25] and limiting the resources [26]. Owing to the advantages of simplicity and easy implementation, the challenge-response system has been used in the proposed work. However, one of the most common prevention strategies that are implemented in any network-based service is the graphic turing test and is particularly termed as CAPTCHA [27]. The graphic turing test includes test puzzles, graphical test and crypto-puzzles [28], [29] with an aim of detecting the bots by providing the challenges to the user that are illegible even for the intelligent machine and thus bots cannot produce the appropriate response. On verifying the source IP address of the user with the access control list, the firewall forwards the request to the verifier node. Upon receiving the request, the verifier node verifies the user by sending the image puzzle as a challenge to the requested system. On providing the correct response for the given challenge, the user will be allowed for the next level defense verification. On the other hand, even after the few trials, if the user failed the graphical test, the request packet will be dropped and the IP address will be added to the suspicious list and on reaching the given threshold, it is updated in the blacklist. In this work, testing strategy Image Completion with Anomaly Detection (AD-IMGCOP) suggested by the authors Saravanan et al., [7] is utilized in which the parts of the image and an anomaly part along with the blank boxes are given as a challenge to the user. The user has to identify the anomaly and drag and drop the parts in the respective boxes to make a complete image.

4.2 Authentication Model

Once the user is identified as a normal user and not the bot by employing the graphical turing test, the next step is to authenticate the user as a legitimate. In this authentication phase, the user is verified through their profile created at the time of registration. The authentication is made for the existing user and if the new user arrives, the login credentials such as user name, password, and other personal details along with their mobile number will be stored in the user_profile database. Additionally, information concerning the device and browser utilized by the user at the time of registration is stored in the device_profile database with the unique identifier of the user. During the login phase, the user enters the username and password which is then compared with the one present in the user_profile table. If a match occurs, the device and the browser currently used by the user is then compared with one present in the device_profile. Again if a match occurs, a simple text puzzle created from the user profile will be given to the user. If the user passes the test, the request is moved to the next level or otherwise, the request will be blocked and the source IP address will be updated in the suspicious list. On the other hand, if the device and browser doesn’t match with the one in the device_profile, then the One Time Password (OTP) will be sent to the user’s registered mobile number (RMN). The user is then allowed to enter the OTP and if a match occurs, as discussed earlier, the text puzzle generated from the user profile will be given as the challenge. On the other hand, if the OTP entered by the user does not match with the one sent to the RMN, the request will be discarded and the source IP address will be added to the suspicious list and on reaching the given threshold, it is updated in the blacklist. The process of user authentication is shown as the workflow diagram in Figure 3 for a better understanding.

Here the text puzzle can be generated commonly based on the mandatory details provided by the user at the time of registration. Some of the text puzzles that can be sent as a challenge for the user can be ‘Enter the last two digits of your pincode’, ‘Enter the fourth digit of your pincode’, ‘Enter your first and last number of your RMN’ and so on. The questions can be framed in common based on the mandatory fields of the user profile. This method will consume less work as it eliminates the process of generating specific questions to the specific user. The flowchart for the user authentication phase is shown in Fig. 3.

5 DDOS ATTACK PREVENTION

5.1 Count based Filtering

This model generally employs threshold based filtering techniques for detecting the attacks and to secure the cloud environment. This method is also used to prevent the cloud from DDoS vulnerability. The most common measures utilized in count based filtering are the number of requests, number of connections and hop count, etc., [9]. The computed TTL value of the received packet is also verified with the IP2HC table. If there is a match, then the packets are moved to the next verification process or else the IP packets are considered to be spoofed and is stored in the suspicious list along with the counter indicating the number of suspicious request from the same host. The number of requests from the same host, the number of new requests from different hosts and the number of packets with SYN flag set as ON is also verified for the particular timestamp [6]. If it reaches the threshold value, the packets are dropped and are updated in the suspicious list. Once the counter of the IP address in the suspicious list reaches the particular threshold, the details are deleted from the suspicious list and are added to the blacklist. Along with the IP address, the suspicious list consists of several other details such as TTL value, number of request from the source IP address for each timestamp including start and end time for future study.
5.2 Statistical based Filtering

Several parameters are available in the literature for analysing the incoming and outgoing IP packets statistically [30]. The attacker uses several zombie systems to enable the attack on the particular system thereby consuming the system resources with an aim of denying the request send by the legitimate user. Most commonly, the adverts employ several flooding attacks based on UDP, ICMP and TCP packets. Thus by analysing the incoming and the outgoing packets, the DDoS flooding attacks can be detected and dropped. The parameters used for the proposed research are ratios of various protocols such as TCP, UDP and ICMP, the ratio of incoming and outgoing packets, and the entropy values of the protocols [19], [31]. The parameters and the formula to compute the values are presented below.

**Ratio of Incoming and Outgoing Packets:**
The ratio of incoming packets towards outgoing packets is found to be the most effective as the attackers generally engage with the more number of incoming packets than outgoing packets generated by the targeted system. The sudden increase in the ratio indicates the occurrence of attack packets. The formula to compute the ratio of incoming and outgoing packets is given (1).

\[ \text{Ratio}_{\text{in/out}} = \frac{\sum \text{Incoming Packets}}{\sum \text{Outgoing Packets}} \]  

(1)

**Ratio of various Protocols:**
Basically, the DDoS attack will be successful by utilizing the TCP (T), UDP (U) and ICMP (I) packets. The ratio of these protocols specifies the hint about the existence of a DDoS attack. The formula to compute the ratio of various protocols shown in (2).

\[ \text{Ratio}_{T,U,I} = \frac{\sum P_i \cdot \text{Packets}}{\sum \text{TP.Packets}} \]  

(2)

Here, i represents various protocols such as TCP, UDP and ICMP packets. The proportion depicts the presence of the attack.

**Entropy of Protocol (Ep)**
Not only the ratio of the protocols, but the entropy of protocols also provides a better result in detecting the DDoS attacks. In the case of normal traffic, the ratios of the various protocol packets are constant and thus the entropy will be a non-zero constant. However, in the case of an attack, the entropy values will be 0. The formula to compute the entropy values is shown (3).

\[ \text{Entropy}_{Ep} = \sum -p_i \log_2 p_i \]  

(3)

Here, i represents various protocols such as TCP, UDP and ICMP packets.

Based on the above statistical measures the proposed method works in an efficient way. The detailed architecture of the method is presented in Fig. 4.

![Flowchart of the User Authentication Process](image)

Fig. 3. Flowchart of the User Authentication Process.

![Detailed Architecture of the Proposed Model](image)

Fig. 4. Detailed Architecture of the Proposed Model.

The cloud server contains several services and is represented as service 1, service 2, ..service n. The components present in the architecture are already explained. The process of a virtual firewall is represented as algorithm pseudocode in Fig. 5. Only fo the first time, the requests are verified using a virtual verifier node and for the subsequent requests, the packets are verified using a white list and blacklist.

**Algorithm : Pseudocode for VF_Event**

Input : IP address of Packet p, Whitelist w_list, Blacklist b_list

Output: Decision about the Packet as Accept, Drop, Forward_VNN.
The experimental analysis has been made by simulating the cloud environment. A cloud server is installed in a system and a set of 10 client systems have been installed for sending legitimate requests to the server in the process of evaluating the performance of the proposed model in detecting the DDoS attacks. For creating real traffic, the DDOSIM tool has been employed. The aim of this DDoS simulator which is written in C++ is to generate the distributed denial of service attacks in the laboratory environment against a target server. It mimics the various zombies with a random IP address to generate TCP connections based attacks, TCP flood attacks and even more especially in Linux environment. Various commands can be used to create the attack such as "./ddosim -d 192.168.1.2 -p 110 -c 0 -i eth0" to create TCP connections [32]. Once the attacks are stimulated, the packets are to be accessed and the header information has to be retrieved for which JPCap tool is employed [6]. The legitimate traffic along with the stimulated DDoS attacks have been forwarded to the model implemented in the cloud server. For the evaluation, the threshold value is computed for the total number of the requests based on which the threshold value for the number of requests from the same host, different host and the number of packets with SYN flag set are computed. The threshold for request packets can be computed as \((B/N+1)\) where, \(B\) is the allocated bandwidth and \(N\) is the total number of hosts connected to the server at a particular time.

### 6 EXPERIMENTAL AND RESULT ANALYSIS

The experimental analysis has been made by simulating the cloud environment. A cloud server is installed in a system and a set of 10 client systems have been installed for sending legitimate requests to the server in the process of evaluating the performance of the proposed model in detecting the DDoS attacks. For creating real traffic, the DDOSIM tool has been employed. The aim of this DDoS simulator which is written in C++ is to generate the distributed denial of service attacks in the laboratory environment against a target server. It mimics the various zombies with a random IP address to generate TCP connections based attacks, TCP flood attacks and even more especially in Linux environment. Various commands can be used to create the attack such as "./ddosim -d 192.168.1.2 -p 110 -c 0 -i eth0" to create TCP connections [32]. Once the attacks are stimulated, the packets are to be accessed and the header information has to be retrieved for which JPCap tool is employed [6]. The legitimate traffic along with the stimulated DDoS attacks have been forwarded to the model implemented in the cloud server. For the evaluation, the threshold value is computed for the total number of the requests based on which the threshold value for the number of requests from the same host, different host and the number of packets with SYN flag set are computed. The threshold for request packets can be computed as \((B/N+1)\) where, \(B\) is the allocated bandwidth and \(N\) is the total number of hosts connected to the server at a particular time.
particular point of time [33]. The threshold can be computed for each time window say one minute. The threshold value for the number of requests from the same host $t_{ssr}$ can be computed as the one third of the total limit of the request which can be depicted as $(B/N+1)/3$. On the other hand, the threshold value for the number of new requests from the different hosts $t_{dsr}$ can be computed as the two third of the total limit of the request which can be depicted as $2(B/N+1)/3$. The threshold value for the number of requests with SYN set ON can be computed as one half of the total limit of the request which can be depicted as $(B/N+1)/2$. The performance of the model has been analysed using various measures such as detection rate, false negative rate, and dropout rate. The performance of the proposed model is compared with several methods existing in literature such as pushback [15], distance based [17], C2DF [18] and DDoS Detection [6] and the results have been examined. The analysis has been carried out by varying the number of DDoS attacks detected towards the deployed cloud server. The detection rate and false negative rate for the various number of DDoS attacks for the existing methods and the proposed model have been compared. The values are presented in Table 1.

![Fig. 7. Algorithm Pseudocode for Virtual Verifier Node.](image1)

![Fig. 8. Algorithm Pseudocode for Virtual Verifier Node.](image2)

Additionally, the dropout rate for the proposed and existing systems are compared by varying the number of requests. The dropout rate is the ratio of the number of packets dropped towards the total number of packets. The comparative analysis of the dropout rate by varying the number of attacks from 50 to 500 is shown in Table 2. Table 2 represents the number of attacks along with a number of attacks dropped out (AD) and dropout rate (DoR) for the proposed and the existing methods. The values presented in Table 2 is presented as graph in Fig. 7 and Fig. 8.

### Table 1: Performance Comparison on Various Models

<table>
<thead>
<tr>
<th># of Requests</th>
<th>Pushback</th>
<th>Distance based</th>
<th>C2DF</th>
<th>DDoS Detection</th>
<th>Proposed Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#AD</td>
<td>DR</td>
<td>FNR</td>
<td>#AD</td>
<td>DR</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>0.60</td>
<td>0.40</td>
<td>7</td>
<td>0.70</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>0.70</td>
<td>0.30</td>
<td>16</td>
<td>0.80</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>0.67</td>
<td>0.33</td>
<td>25</td>
<td>0.83</td>
</tr>
<tr>
<td>40</td>
<td>29</td>
<td>0.73</td>
<td>0.28</td>
<td>34</td>
<td>0.85</td>
</tr>
<tr>
<td>50</td>
<td>39</td>
<td>0.78</td>
<td>0.22</td>
<td>42</td>
<td>0.84</td>
</tr>
<tr>
<td>60</td>
<td>45</td>
<td>0.75</td>
<td>0.25</td>
<td>48</td>
<td>0.80</td>
</tr>
<tr>
<td>70</td>
<td>52</td>
<td>0.74</td>
<td>0.26</td>
<td>57</td>
<td>0.81</td>
</tr>
<tr>
<td>80</td>
<td>61</td>
<td>0.76</td>
<td>0.24</td>
<td>68</td>
<td>0.85</td>
</tr>
<tr>
<td>90</td>
<td>71</td>
<td>0.79</td>
<td>0.21</td>
<td>71</td>
<td>0.79</td>
</tr>
<tr>
<td>100</td>
<td>65</td>
<td>0.65</td>
<td>0.35</td>
<td>75</td>
<td>0.75</td>
</tr>
<tr>
<td>Avg</td>
<td>0.72</td>
<td>0.28</td>
<td>0.80</td>
<td>0.80</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 1 presents the number of attacks, number of attacks detected (AD), detection rate (DR) and false negative rate (FNR) for all the methods. The number of attacks given as an input for the deployed model is varied from 10 to 100 for which the number of attacks detected, detection rate and false negative rate are presented for all the proposed and existing methods in Table 1. From the analysis, the pushback method has an average detection rate of 72%, distance based method has an average detection rate of 80%, C2DF has an average detection rate of 80%, DDoS detection model has an average detection rate of 90% and the proposed model has an average detection rate of 91%. Thus, the proposed method has a better detection rate than the other existing methods. The values of detection rate and false negative rate presented in Table 1 are presented as a graph in Fig. 7 and Fig. 8.

### Table 2: Performance Comparison on Various Models

<table>
<thead>
<tr>
<th># of Attacks</th>
<th>Pushback</th>
<th>Distance based</th>
<th>C2DF</th>
<th>DDoS Detection</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>27</td>
<td>0.54</td>
<td>17</td>
<td>0.34</td>
<td>9</td>
</tr>
<tr>
<td>100</td>
<td>49</td>
<td>0.49</td>
<td>29</td>
<td>0.29</td>
<td>15</td>
</tr>
<tr>
<td>150</td>
<td>75</td>
<td>0.50</td>
<td>61</td>
<td>0.41</td>
<td>26</td>
</tr>
<tr>
<td>200</td>
<td>89</td>
<td>0.45</td>
<td>59</td>
<td>0.30</td>
<td>35</td>
</tr>
</tbody>
</table>
From the experimental analysis, it is found that the average dropout rate for the pushback method is 45%, for distance based method is 33%, for C2DF method the value is 19%, for DDoS Detection method the value is 10% whereas for the proposed model the average dropout value is 8% which is minimum than other methods. It is clear from the comparison that the proposed method has less dropout value than the existing methods. From the results obtained, it is clear that the model produces better results than many of the existing algorithms used for the analysis. The method delivers a robust multi-model based anti-DDoS mitigation framework that employs various models such as graphical turing test, authentication, count based filtering and statistical based filtering in verifying the packets. Also, the overhead of verifying the packets is highly reduced as the verification takes place only for the first time and for the successive requests, the details are verified with the access list. The future work focuses on providing an enhanced solution for detecting the low rate DDoS attacks which is a challenge among cloud environments.

7 CONCLUSION

Securing the cloud from various threats and vulnerabilities is the major concern for cloud providers. Distributed denial of service attack is the most common vulnerability as it makes the resources unavailable to legitimate users. Multi-model based anti-DDOS framework has been proposed to prevent and detect the high rate DDoS attacks in the cloud environment. The analysis of results obtained for the experimental analysis made from the simulation of a network shows that the proposed method provides better performance in preventing and detecting the attacks. The method provides a strong mitigation framework as it uses several models in verifying the packets. Additionally, the information about the doubtful packets is initially stored in the suspicious list and later it is moved to the white and blacklists. Also, the overhead of verifying the packets is highly reduced as the verification takes place only for the first time and for the successive requests, the details are verified with the access list. The future work focuses on providing an enhanced solution for detecting the low rate DDoS attacks which is a challenge among cloud environments.

REFERENCES


