Mitigation of DDoS Attack in Cloud Computing Domain by Integrating the DCLB Algorithm with Fuzzy Logic

Amrutha Muralidharan Nair[1]
Research Scholar
Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India

Dr. R Santhosh[2]
Professor
Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India

***Abstract*—Cloud computing would be an easy method to obtain services, resources and applications from any location on the internet. In the future of data generation, it is an unavoidable conclusion. Despite its many attractive properties, the cloud is vulnerable to a variety of attacks. One such well-known attack that emphasizes the availability of amenities is the Distributed Denial of Service (DDoS). A DDoS assault overwhelms the server with massive quantities of regular or intermittent traffic. It compromises with the cloud servers' services and makes it harder to reply to legitimate users of the cloud. A monitoring system with correct resource scaling approach should be created to regulate and monitor the DDoS assault. The network is overwhelmed with excessive traffic of significant resource usage requests during the attack, resulting in the denial of needed services to genuine users. In this research, a unique way to the analyze resources used by the cloud users, lowering of the resources consumed is done when the network is overburdened with excessive traffic, and the dynamic cloud load balancing algorithm DCLB (Dynamic Cloud Load Balancing) is used to balance the overhead towards the server. The core premise is to monitor traffic using the fuzzy logic approach, which employs different traffic parameters in conjunction with various built in measured to recognize the DDoS attack traffic in the network. Finally, the proposed method shows a 93% of average detection rate when compared to the existing model. This method is a unique attempt to comprehend the importance of DDoS mitigation techniques as well as good resource management during an attack and analysis of the.**

Keywords*—DDoS attack; resource scaling; DCLB; fuzzy logic; traffic parameters

I. Introduction

Cloud computing threats are becoming more prevalent regularly, with attack channels and patterns changing. It provides a wide range of services with considerable benefits for corporate organizations, businesses, and individuals transitioning to this environment. Despite its numerous benefits, security is typically the decisive factor for businesses when determining if cloud infrastructure is the best solution for their users. According to the assault report in [1], the most prominent attack is the DDoS attack among the abundance of attacks affecting the great majority of organizations globally. DDoS attacks have fully-fledged in admiration in recent years because of the simplicity with which they may be deployed. DDoS attacks have increased in recent years because of the ease with which they may be launched. According to [2], the scale of a DDoS attack, which was just 8 Gbps in 2004, has already surpassed 800 Gbps in 2016. Among the plentiful recent attacks [3], a few significant assaults have expanded a lot of courtesy in the scholarly community [4]. "Lizard Squad attacked", has affected the Microsoft and Sony cloud-based gaming services, knocking them offline on Christmas Day in 2015. "Rackspace", is a cloud service provider, which was affected by a massive distributed denial of service (DDoS) attack against its services.

Another exceptional assault scenario was for the Amazon EC2 cloud servers to be exposed to yet another incredible DDoS attack. These assaults caused major downtime, commercial losses, and other enduring and short-term effects on the victims' businesses. According to "Verisign iDefense Security Intelligence Services" [4], the cloud and SaaS (Software as a Service) industries have been the most targets of DDoS attacks in recent quarters. The contribution to the paper is listed below:

1) A thorough understanding of DDoS attacks is offered for the reader to gain correct insight and comprehension.

2) A monitoring system to identify DDoS attacks in traffic.

3) An effective Load balancing mechanism for smooth conduction of the cloud services.

As a consequence, the DDoS detection and prevention system [5] is an essential element in the overall growth of an organization's statement since it explains the rules and methods for providing security. The academic community has focused on identifying several forms of DDoS attacks in the cloud, such as ICMP, HTTP, and TCP protocol flooding [6]. Our key discovery is associated with resource scaling, which may become less effective if the conflict in the network is developed during the attack. The following is the objective of this research:

1) Research the DDoS attack and its impact on the cloud server.

2) To create a hybrid approach for mitigating DDoS attacks using DCLB and FUZZY logic.

3) To monitor and assess the algorithm's effectiveness in mitigating DDoS attacks.

This paper is mainly focused on the mitigation of DDoS attacks and reducing resources utilization. The framework used for the mitigation process of a DDoS attack contains a monitoring system and a load balancing method.

The research paper is constructed as follows. Section II depicts a study of DDoS attacks and the various methodologies employed. Section III of the study then discusses methodology and terminology. Section IV describes in detail the technique and algorithm used for network monitoring and load balancing. Section V presents the experimental data, and Section VI covers the discussion. Section VII brings the work to a conclusion.

II. Related Work

Wahab et al. [7] developed a two-pronged strategy that enables the hypervisor to create believable confidence in the virtual machine. In this case, the suggested system employs the Game solution guide hypervisor approach to identify the ideal detection as well as load balancing.

Liang et al., [8] provided detailed research on machine learning algorithms utilized for DDoS attack detection. The ML approaches detected the class imbalance problem, but the results reveal that a single method cannot overwhelm the DDoS assault, hence enhancements to the ML-based strategies are necessary.

Kousar et al. [9] presented a novel mechanism by combining the statistics and machine learning models. To identify the DDoS attempt, the work was implemented in the Apache Spark Framework. In addition, the approach detects the attack using the NSL-KDD cup methodology as the benchmark dataset.

Alsirhani et al. [10] proposed a DDoS detection framework based on the "Gradient Boosting Classification Algorithm (GBT)" and the Apache Spark engine. The traffic volume (dataset) and feature space assist in the creation of a depth decision tree to identify the assault.

Cloud-Traceback technology (CTB) was created to detect and also to mitigate the DDoS assaults in cloud computing by identifying the origin of HTTP and XML-based attacks, Chonka et al. [11]. It also introduces the use of backpropagation in conjunction with a cloud defender, which filters out malicious traffic.

Guo et al. [12] suggested a resource allocation approach for cloud data centers called "dynamic resource allocation" to protect the resources against DDoS attacks. This article made use of idle cloud resources and avoided them by employing quick filtering algorithms.

Bikram et al. [13] identify the features of a DDoS assault and propose a system called a “Snort-based Intrusion Detection System (IDS) tool” for DDoS detection. It also describes a system that would alert the network administrator of every attack on any imaginable resource, as well as the sort of attack. It also temporarily suspends the attacker for the network administrator to devise a backup strategy. The proposed method mitigates the impact of DDoS attacks by detecting them early and altering many parameters that make it simpler to diagnose the problem.

Sathya et al., [14] suggested a new framework entitled "Anti-DDoS" that detects high-rate DDoS assaults. This method uses the “graphical Turing test” and “Authentication model” to prevent the cloud from the attacked. It also uses the count hop filtering technique for detecting the attack, the traffic was controlled using a control list.

Liu et at., [15] proposed a strategy that uses the BIRTH algorithm to detect aberrant traffic flows by employing frequency domain information from the network flow's autocorrelation sequence as a clustering feature.

Sahi et al., [16] demonstrated a methodology for detecting the flooding of DDoS attacks by combining data flow and using a list to blacklist to identify the source IP of attack packets.

Barde et al., [17] A “deception detection” approach were proposed for detecting high-rate DDoS attack traffic in the cloud computing domain.

Navaz et al., [18] provide multi-level detection methodologies for camouflaged small traffic DDoS assaults and employ entropy-based algorithms in combination with anomaly detection systems. The researchers have presented a detection algorithm for flooded DDoS attacks and random DDoS attacks, The method gives acceptable results for DDoS attacks with heavy traffic in a cloud environment, the time required to complete the operation is exceedingly slow. The results in the cloud environment are slower, and the approaches rarely examine the real system that is subjected to a variety of attacks.

Zheng et al., [19] suggested a DDoS attack mitigation architecture that helps to detect and responds to attacks quickly. Furthermore, the author argued that SDN network technology helped in the prevention of DDoS assaults.

Saravanan et al. [20] provided an approach for recognizing and mitigating DDoS attack impact assaults. It employs three screening checks to protect the server against assaults, as well as several limitations to identify the attacks. To repel the attack, it employs two queues.

Scaling the VM's capacity is a critical step in estimating the overall number of requests processed in a particular period. The duration of time necessary to perform the request impacts the number of resources consumed. The scale inside-out strategy enhances capacity while simultaneously scaling internal applications to minimize resource use. Somani et al. [21] pioneered this concept.

This work focuses on DDoS attacks that are primarily aimed at detecting bandwidth and connection flooding. Karan et al., [22] generate a solution by integrating the OpenStack firewall with raw socket programming for monitoring network traffic.

III. Design and Assumptions

In the section, the overall proposed model system design is shown. Each cloud user (CU) will access the cloud resources

effectively. The problem faced by the legitimate user (normal user) is the delay that occurs when the user requests the cloud for particular resources.

As the cloud works on-demand policy [24], the cloud system should be more efficient to provide the resource on time as user requests them. The delay in the network occurs due to some unwanted request that will be placed in the network by the assault(attacker). The attacker floods the network with the unwanted request so that the legitimate user will be halted from the requested resource, such attacks are known as DDoS attack.

So, it is necessary to launch an effective mitigation mechanism to prevent the system from this type of attack. As discussed in the 'related work' section, there are a lot of mechanisms to detect the DDoS attack but the challenge faced is to identify the attacker and to make the system work properly by removing the unwanted request and regulating the relationship between the and cloud user (balancing the workload).

As shown in Fig. 1, the overall architecture is divided into two main parts: verification and detection process and service access in cloud computing. In verification and detection process is again subdivided into two, one is VV model (Virtualization & Verification) and the other is LB model (Load Balancing). Each cloud user is owned by a virtual machine to access the cloud resources.

The request will be monitored by the virtualization and verification model (VV model) based on the analysis detect made of the request characteristic; the VV model is consist of 4 phases of the procedure and each phases process help to the DDoS attack. In each phase different value is computed based on the request characteristics. The computed value is compared with some predefined threshold value by using the concept of fuzzy logic [23].

The VV process applies various analyzing strategies to the incoming request such as in & out statistics, checking the protocol, number of requests coming, and packet analysis. The result obtained from the VV model detects the DDoS attack and forwards the results to the LB model.

In the LB model, the request load is balanced by using a dynamic cloud load balancing algorithm (DCLB), it applies a vertical scaling mechanism on resource allocation to mitigate the attack after the detection process. Usually, each cloud user which owns the virtual machine is allocated a sufficient number of resources such as P, D, M, and NT (processor, disk, memory, and network throughput).

The number of connections at a particular time many varies, so the idea is to scale down the resources when the attack is detected in the network. When the attack is detected in the cloud system the resources are minimized and the virtual machine which provides the same request or random request will be halted for some time, so that the legitimate user will be able to access the system efficiently.

A. Notation and Assumptions

Let $CC = \{cc_1, cc_2, \ldots\ldots\ldots\ldots cc_n\}$ be the finite set of cloud users in the cloud at a particular time. The cloud user (CC)accesses the cloud through the virtual machines. Each CC is owned by a unique virtual machine $VM_i = \{VM_1, VM_2, \ldots\ldots\ldots\ldots VM_n\}$. Each virtual machine VM_i uses some set of resources$\langle P_i, D_i, M_i, NT_i \rangle$ which is represented in vector form.

Along with the cloud user, there is a malicious user known as an attacker which is denoted as AT_i which accesses the cloud as the normal cloud user and captures "n" virtual machine to perform a DDoS attack.

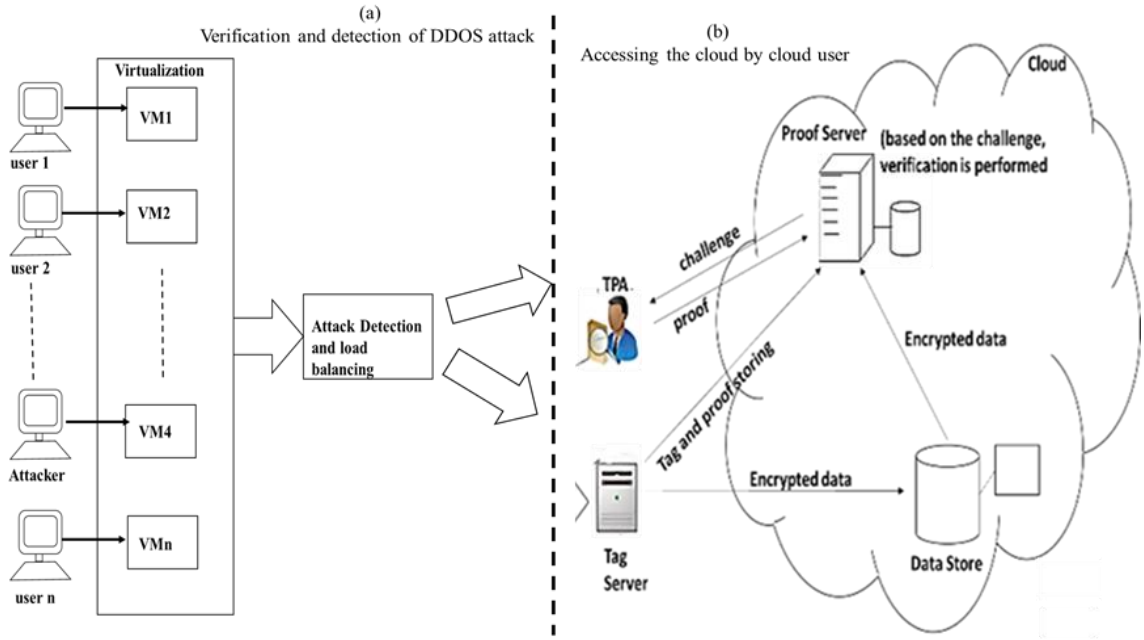

Fig. 1. Overall Proposed System Architecture.

In this proposed system, the request can come from the same source or different sources, so it has been examined that the arrival of several requests from the same source R_{ss} and requests from different resources R_{ds} is less than or equal to the maximum request R_{max} at a particular time.

Also in this system, the number of the incoming packets and the outgoing packet is calculated as n_{rin} and n_{rout}. In the network, there are different protocols from which we can send the request to the cloud such as TCP, UDP, ICMP, etc. In this paradigm, calculating the entropy of protocols that play a substantial role in attack detection, E_p, is critical. In addition, when it comes to protocols, the duration of an IP packet flow is a critical parameter to examine to detect an assault. The flow calculation is denoted as L_{ipflow} in which the average is calculated for all packets from each client virtual machine.

The VV model's output (analysis report) is sent to the LB model, which predicts the manifestation of a DDoS assault, the load balancer checks the result and will reduce the resources allocation by scaling down the Resource Utilization Factor RUF_{nor} to a threshold value which is allocated based on analyzing the intensity of the traffic in the network as RUF_{att}. When RUF_{att} is set instead of releasing the complete resource at a time it will release one by one so that the resource is not affected and legitimate users can also access the resource without any delay.

Virtualization cloud Definition 1: A cloud system consists of a set of cloud users $CC = \{cc_1, cc_2, \dots\dots\dots\dots cc_n\}$ owing a set of virtual machines$VM_i = \{VM_1, VM_2, \dots\dots\dots\dots VM_n\}$. to access a set of resources, provide by the cloud system.

Each virtual machine owned by the user will have a unique session ID S_{id}, so that it will be easy to identify which the virtual machine is loading the network with heavy traffic. The VV model analysis the traffic and remove the same request coming from the single session of the same virtual machine. Notations used in the proposed system are given below in Table I.

TABLE I. NOTATION USED IN THE PROPOSED SYSTEM

Notation	Description
CC_i	A finite set of cloud users
VM_i	A finite set of Virtual Machine owned by the cloud user
P_i	Processor usage
D_i	Disk usage
M_i	Memory Utilization
NT_i	Network Throughput access
AT_i	Represent the attacker from the set of user
R_{ss}	The request coming from the same source
R_{ds}	The Request coming from a different source
R_{max}	A limited number of requests from the source
n_{rin}	Number of the incoming request
n_{out}	Number of outgoing response
E_p	entropy calculation of different protocol
L_{ipflow}	length of ip packet flow
RUF_{nor}	Normal resource utilization
RUF_{att}	The threshold value is set when an attack occur.
S_{id}	Unique session ID for each user in their owned virtual machine

IV. IMPLEMENTATION OF THE PROPOSED

The overall architecture is of the integrated cloud load balancing algorithm and fuzzy logic (i.e. incorporating different parameters of requests coming from the cloud user) along with the dynamic scaling of resources. The proposed model is divided into two models: VV (Virtualization & Verification) model and LB (Load Balancing) model. These two models are used to spot (detect) and mitigate the attack and help to reduce the access to resources and some other services also.

A. VV Model

In this proposed model, each request coming from the user and the attacker has initially undergone to VV model in which the virtual machine details and request verification is done. The VV model verifies the incoming request that comes from each virtual machine. İt also gather the unique session id S_id of each request.

The DDoS attack is attained by utilizing a flooding attack in which the request is following the same pattern concerning the protocol also (Fig. 2). The VV model is subdivided into 3 phases: RAI (Request arrival Inspection), PI (Protocol Inspection), and IPI (IP flow inspection).

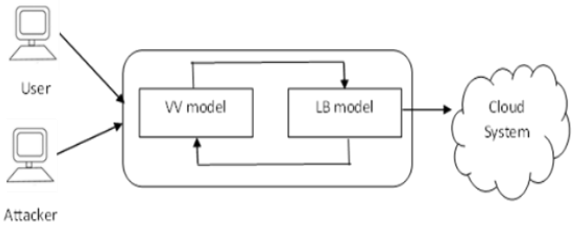

Fig. 2. Overall Implementation Design.

1) RAI: The RAI is the process in which the request arrival is analyzed. Each request comes to the system in the form of a special packet with header fields. The header field help to identify the source and destination address. To identify the chances of a DDoS attack, the source address is considered the critical factor. Here, the packet used is a 32-bit IP packet along with a unique session ID of 4 bits (Fig. 3).

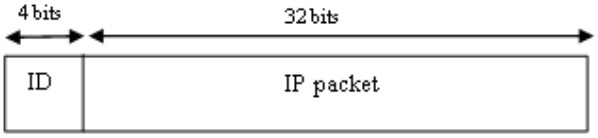

Fig. 3. Packet used for the Accessing the cloud System.

The unique session id is first extracted from each packet. An attack is feasible if the number of requests from the same source consistently exceeds the maximum limit in a certain period. (i.e., $R_{ss} > R_{max}$). İf the number of requests comes from a different source at a particular time and exceeds the maximum limit then there is a chance of an attack to occur (i.e.,$R_{ds} > R_{max}$)Next, we calculate the ratio of the R_{ss} and R_{ds} which are the main parameter to consider in detecting DDoS attacks. The proportion of incoming and outgoing requests over some timestamp is calculated based on the eq (1),

$$R_{io} = \frac{R_{ss}}{R_{ds}} \quad (1)$$

Usually, the propagation should be constant if the traffic is normal. So here if the R_{io} exceeds the threshold value ($i.e \geq 1$) thus it indicates the event of an attack. İt will be balanced if the ratio is less than 1.

2) PI: Essentially, if $< \text{TCP}, \text{UDP}, \text{ICMP} > \Rightarrow < T, U, I > packets$ are employed then, the DDoS attack will be successful [21]. A DDoS attack is indicated by the ratio of these protocols. The formula for calculating the ratio of different methods is displayed in eq (2).

$$R_T, R_U, R_I = \frac{\sum P_T}{\sum Packet\_IP}, \frac{\sum P_U}{\sum Packet\_IP}, \frac{\sum P_I}{\sum Packet\_IP} \quad (2)$$

Then the entropy [25], is calculated for the above-computed value as shown below in eq (3),

$$E_p = \sum^{i \in T,U,I} -P_i \log_2 P_i \quad (3)$$

3) IPI: As the arrival and protocol inspection likewise, another significant criterion is the IP Packet flow in the network. Counting the number of packets that fulfill the same criterion yields the average duration of an IP flow. The requirements include the source and destination addresses, as well as the port number and protocol utilized. A packet with the same source and destination might arrive with a different protocol. As a result, the length of the IP flow is computed using Equation (4).

$$L_{ipflow} = avg\left(\frac{\sum ip\_packets}{\sum ip\_flow}\right) \quad (4)$$

The typical IP flow length is usually between 5 and 10. If the value is close to one, it means an attacking packet was discovered. Here when the attack is detected then the Att_{flag} flag is set to one. Which indicates the load balance of the attack occurrences? The entire operation and algorithm utilized for the VV model are detailed below, and a graphical depiction of the method is depicted in Fig. 4.

Algorithm of VV model

Input :
The ip address of the packet,
initial set threshold values
R_{max}

Output:
Attack Detection or accepting the packet

Procedure VV():

1. Retrieve the session id from the request S_{id_i}

Analysis():

i. if $n_{rin} > 100$ then
halt the all the request from the same session ID S_{id_i}

ii. else
1. Compute the R_{ss} and R_{ds}
2. Calculate the ratio of incoming and outgoing packets
$$R_{io} = \frac{R_{ss}}{R_{ds}}$$
3. If $R_{ss} < R_{max}$ && $R_{ds} < R_{max}$ && $R_{io} \leq 1$
goto step 4
4. else
mark the incoming packet as an attack packet set the $att_{flag} = 1$, and move on to the load balancing process.
5. Calculate $R_T, R_U, R_I = \frac{\sum P_T}{\sum Packet\_IP}, \frac{\sum P_U}{\sum Packet\_IP}, \frac{\sum P_I}{\sum Packet\_IP}$
6. Compute the entropy
for (i$\in < TCP, UDP, ICMP >$)
{
$$E_p = \sum -P_i \log_2 P_i$$
}
7. If $E_p \neq 0$ && deviation (R_T, R_U, R_I)! low
goto step 7
8. else
mark the incoming packet as an attack packet set the $att_{flag} = 1$, and move on to the load balancing process.
9. Calculate IP flow average length $L_{ipflow} = avg\left(\frac{\sum ip\_packets}{\sum ip\_flow}\right)$
10. Calculate the entropy value
for (i$\in < TCP, UDP, ICMP >$)
{
$$E_{ipflow} = \sum -P_i \log_2 P_i$$
}
11. İf $5 < L_{ipflow} < 10$ && $2 < E_{ipflow} < 4$
Accept the packet and forward it to the LB module
12. else
mark the incoming packet an as attack packet, set the $att_{flag} = 1$, and move on to the load balancing process.

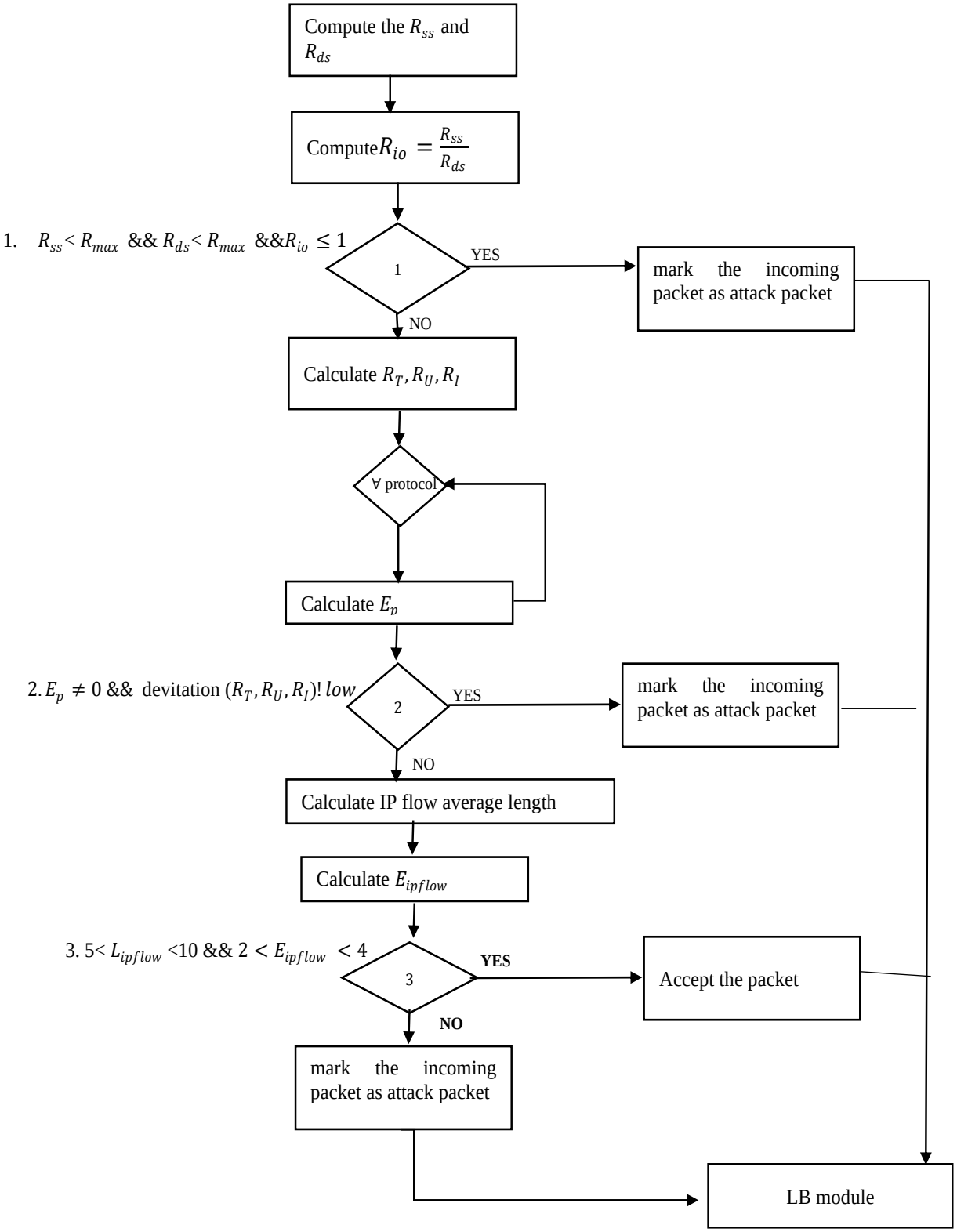

Fig. 4. Flow Chart of VV Model.

B. LB Model

After the VV model, the flow goes to the LB model, in which the resources are balanced based on whether the attack occur or not. The result from the VV model is forwarded to LB however it is using the MAX-MIN load balancing technique. İnitially the services from the cloud will be allocated with favorable resources, if the analysis report shows an occurrence of a DDoS attack, then the MAX-MIN load

balancing is implemented. İn these techniques each request will be provided with the initial resources RUF_{nor} as the occurrence of the DDoS attack is detected in the VV model the resource allocation is minimized toRUF_{Att}. RUF_{Att} is a Resource utilization factor that has a threshold value set, when the attack occurred is detected. But here, this proposed system it is dealing with the dynamic nature. The value of RUF_{Att} will varies based on the intensity of the attack, that is, the resources that are suspended will be released one at a time rather than all at once. Minimizing resources during an assault and returning to normal once the attack has ended will boost the virtual machine's capacity. When there is a high volume of traffic, the initial requests are processed, and subsequent requests are retransmitted. This implies that just the index page is displayed to the user, and all subsequent requests are queued or retransmitted. Assaulters who have released a huge volume of traffic will not wait for the provider to respond. As a result, completing the original request may minimize resource use and enhance virtual machine capacity.

Algorithm of LB model

Input :

Analysis report from the VV model
Att_{flag}value

Output:

Minimize
and maximize the resource usage

Procedure LB():

1. Allocate the resource to the requested services
   a. Compute $RUF_{nor} = \frac{VM_{CAPACITY}}{n_{maxreq}}$
   b. Set the resource factor $RF = RUF_{nor}$
2. do
   a. Call the VV() procedure periodically.
   b. İf $Att_{flag} == 1$
      i. Compute RUF_{att}=$1/2(RUF_{nor})$
      ii. Reduce the $RF = RUF_{att}$
   c. Else
      i. Set the resource factor $RF = RUF_{nor}$
3. Continue the process for all requests.

The procedure of LB Model help us to maintain the resources and keep the system throughput same for all the cloud user CU (Fig. 5). The VV model identifies the high-rate traffic, and set the Att_{flag} to 1 other wise the Att_{flag} will be 0 indicate that the traffic is normal.

In the LB model the RUF_{nor} (Resource Utilization Factor) is calculated for the normal traffic of the cloud user CU. This RUF_{nor}will be allocated to the requested user and when the attack is detected, the resource allocated to the CU user will be 50% of RUF_{nor} (RUF_{att}=$1/2(RUF_{nor})$. So that it helps to maintain the equivalency among the cloud user. All users can access the server and server can process the user request.

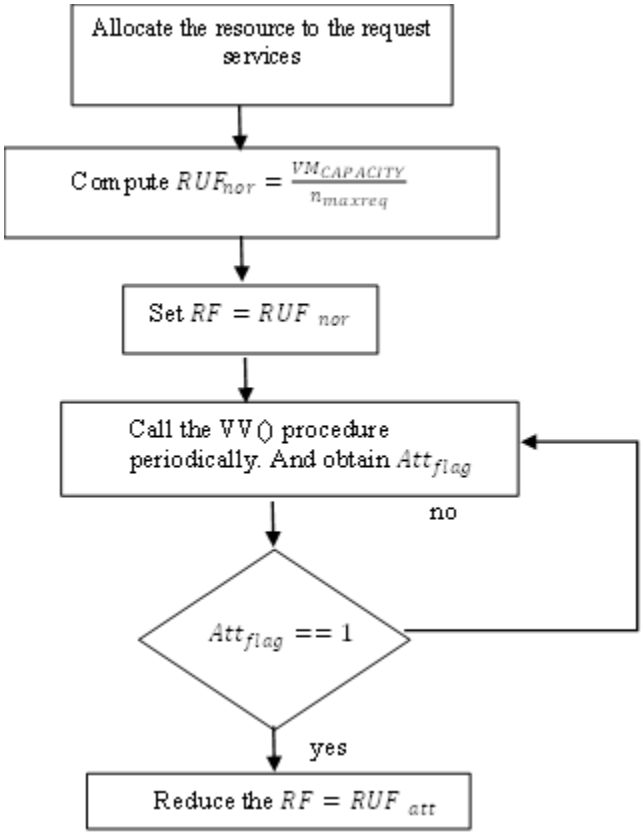

Fig. 5. Working of LB Model.

V. Experimental and Result Analysis

The experiment is carried out by establishing 50 requests to test the availability of cloud service to the requested user even in the event of an attack (including user and attacker requests). At the same timestamp, this request is launched towards the cloud side. The cloud simulator tool is used to mimic the operation of the proposed system and to analyze the system's results. The cloud service taken is the storage of doubled encrypted files and access of the double encrypted file.

A. Computation Cost

The above specified methodology contains operations such as multiplication, division, addition, subtraction and comparison operation with the time t_m, t_d, t_a, t_s *and* t_c. The time taken for the entropy calculation, IP address comparison etc. as calculated for the two phases: The complexity cost of the VV model when compared with the SIO and ANTI-DDoS model is $2nt_d + (n+1)t_s + nt_c$ and for the load balancing the cost is $n(t_d + t_c) + \alpha_d$,where α_d is the delay taken to reduce the resource scale from the normal to minimum when attack is detected. The below Table II show the complexity cost of prosed system with the SIO and ANTI-DDOS schemes.

The proposed model is compared with the SIO and ANTIDDOS model, in which th load balancing is the main factor in the above 3 schemes.

TABLE II. COMPUTATIONAL COST

Complexity Cost	Schemes		
	SIO	**ANTI-DDOS**	**Proposed model**
Verification Cost	$(n^2+1)t_d + 2nt_m$	$n^3t_d + (n+1)^2t_s + 4nt_c$	$2nt_d + (n+1)t_s + nt_c$
Load balancing Cost	$nt_d + n^3t_c - (n+1)^2\alpha_d$	$n(t_d + t_c) + n^2t_s - \alpha_d$	$n(t_d + t_c) + \alpha_d$
Detection cost	$(n+1)(t_d + n^2t_c)$	$nt_d + n^2t_c$	$(n+1)t_c$

The load balancing process complexity in SIO model is $nt_d + n^3t_c - (n+1)^2\alpha_d$ and for ANTIDDOS model is $n(t_d + t_c) + n^2t_s - \alpha_d$, here the complexity is high than the proposed model because in SIO the delay for the resource balancing takes $O(logn^2 + 1)$ and for the ANTIDDOS model the delay for the resource balancing takes $O(n^2)$. All delay is overwhelm in the proposed model in which the load balancing process complexity is $(t_d + t_c) + \alpha_d$, the delay for the resource balancing is $O(1)$.

Also while comparing the verfication cost of SIO and ANNTIDDOS model shows a $O((n+1)^2) and\ O((n-1)^2)$ but the proposed system shows a complexity of $O(2n+1)$. İn the detection process the cost is $O(n+1)$ for the prosed system which is very less complex that the other two schemes.

B. Result Analysis

It is also seen on the TPA side, where auditing requests are tracked. The cloud server processor is an Intel Xeon CPU with 8GB RAM and a 1TB hard disc. The traffic rate is considered to be between 10 and 500 requests per second and does not exceed 500. To analyze the performance, input files of varied sizes are employed.The system's performance is evaluated using three metrics: attack detection time, Rate of Reporting. The above system is contrasted with the AntiDDoS framework [10] and the Scale in-out model [17]. The experiment was carried out by varying the number of DDoS attacks recorded on the deployed cloud server. The detection rate and false-negative rate for a variety of DDoS assaults were compared for the existing methodologies and the proposed model. Table II displays the results. The system's performance is evaluated by comparing the service provided by the cloud during normal and outage periods. The obtained result is shown below in Table III which is compared with existing methods also.

The above table shows the service time of each request without the resource scale down. When the attack occurred in the system the victim server (cloud server) will process the request, the Table IV shows the proposed model taking less time to process the request than the other two.

Table V shows the service time of each request without the resource scale down. When the attack occurred in the system the victim server (cloud server) will process the request, the Table III shows below describe the proposed model taking less time to process the request than the other two. Tables II and III show that the new system works faster than the previous technique. The average value from each table is calculated and it is observed that the proposed system's average value is less than the earlier methodologies. The proposed approach has a detection rate of 93 percent on average. As a result, the suggested technique outperforms the other current methods in terms of detection rate. Also, the proposed system service the request at a high rate than the others, the average service time of the request is 41.3s at the normal rate and 76s during the attack period (see Fig. 6 to 8).

TABLE III. PERFORMANCE COMPARISON OF ATTACK DETECTION

Resources request	SIO	ANTI-DDOS	Proposed model
	AD	**AD**	**AD**
50KB	39.52	37.12	32.12
100 KB	40.87	37.14	33.11
1 MB	42.3	39.45	33.12
2MB	45.23	42.66	40.06
5MB	49.2	43.23	40.31
10 MB	46.3	45.26	39.26

TABLE IV. PERFORMANCE COMPARISON OF SERVIE TIME BEFORE THE ATTACK

Resources request	Service time Before the attack		
	SIO	**ANTI-DDOS**	**Proposed model**
50KB	43.65	33.75	23
100 KB	125.6	41.25	29
1 MB	152.2	45.16	32.13
2MB	187.2	64.14	47.31
5MB	256.7	66.49	49.19
10 MB	358.6	89.69	67.75
Average	**158.9**	**56.7**	**41.3**

TABLE V. PERFORMANCE COMPARISON OF SERVICE TIME AFTER THE ATTACK

Resources request	Service time after the attack		
	SIO	**ANTI-DDOS**	**Proposed model**
50KB	42	87	67
100 KB	141	88	69
1 MB	216	85	65
2MB	405	113	80
5MB	634	129	85
10 MB	707	200	90
Average	**357.5**	**117**	**76**

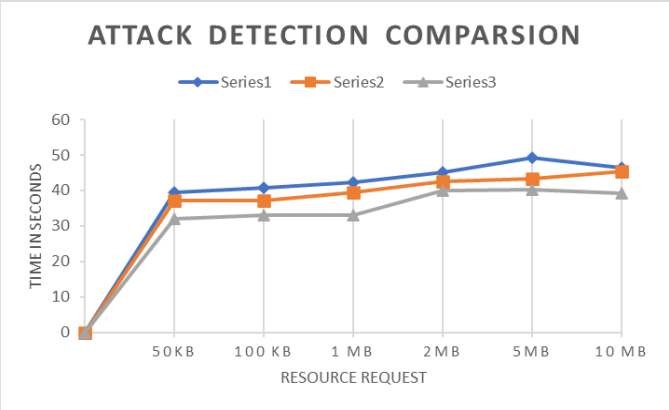

Fig. 6. Performance Comparıson of Attack Detection.

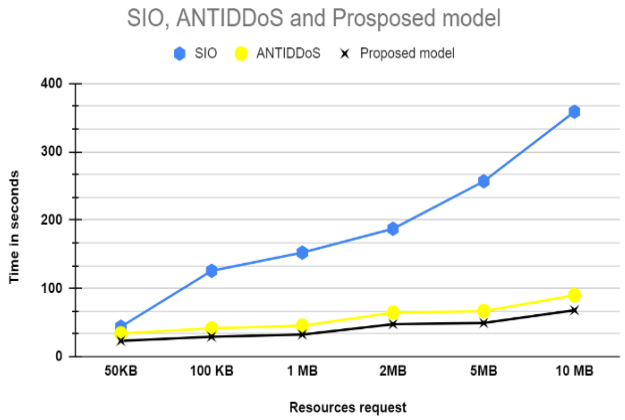

Fig. 7. Performance Comparıson of Service time before the Attack.

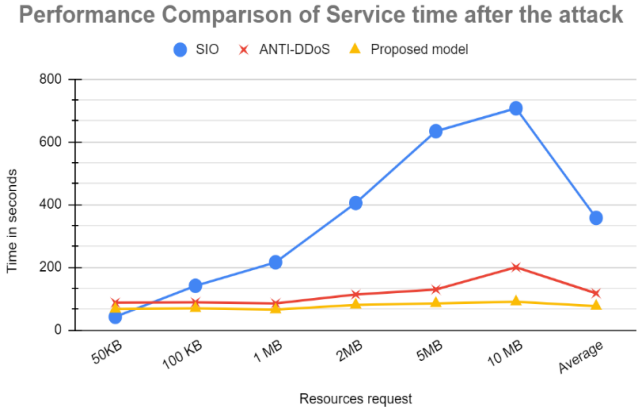

Fig. 8. Performance Comparıson of Service Time before the Attack.

VI. Discussion

This paper proposed a new model IDCLB-FL using the integration of Dynamic Load balancing (Mın-Max load balancing is used) with Fuzzy Logic. The cloud user CU provide request to the cloud server, the cloud server provide the resources required for the user as their demand. The problem discussed in this paper is the high rate Ddos atatck which provide a huge running traffic request along with the legitimate user request. To detect the normal legitimate user request and unwanted attacker request, a proposed model was designed with 2 phase of detection process. The 2 phase of the model is VV model and LB model, one phase is used to deytected the DDoS traffic and other phases are to balance the load of each request. The feature considered in the VV procedure is:

1) The first feature of In and out requestion ratio is calculated as the R_{ss}, R_{ds}, R_{io} value to compare withe condition $R_{ss} < R_{max}$ && $R_{ds} < R_{max}$ && $R_{io} \leq 1$ and based on the analysis, it set the Att_{flag} as 1 or 0.

2) Other feature used in VV procedure is the Entropy calculation of each request based on the protocol used by the traffic request as $E_p \neq 0$ && devitation $(R_T, R_U, R_I)! \, low$ and based on the analysis, it set the Att_{flag} as 1 or 0.

3) The last feature is IP flow along with the entropy value calculated as, $5 < L_{ipflow} < 10$ && $2 < E_{ipflow} < 4$ and based on the analysis, it set the Att_{flag} as 1 or 0.

4) As the value is of Att_{flag} is set to 1 or 0 , the LB model is processed and if Att_{flag} value is 1 then the resource is reduce to be 50% of RUF_{nor} ($RUF_{att} = 1/2(RUF_{nor})$.

5) If Att_{flag} value is 1 then the resource is provided as RUF_{nor}.

C. Limitation of the Proposed Model

The proposed model was test by providing 50 test requests with normal and abnormal traffic pattern. The limitation faced by the system is the time, when detection and verification process take 76s for the 50 requests, but if the number of requests increase the time also increases. Second limitation of the proposed model is that, it is not able to detect the low-rate DDoS attack occurs along with the normal traffic. Third limitation is that, the entropy calculation of each request should lie between the range of 0 and -1, but some time, some normal request also shows the entropy value in this range itself and set the flag as 1 i.e. attack request. So, the proposed model shows a fatal rate of 0.2%.

VII. Conclusion and Future Scope

The cloud server's primary priority is protecting the cloud from numerous threats and vulnerabilities. A distributed denial of service attack is the most frequent vulnerability, which prohibits legitimate users from accessing resources. The analytical report received from the preceding experiment demonstrates an effective methodology for mitigating DDoS attacks. This solution provides an efficient framework for verifying each request and securing cloud server services and resources from being manipulated by an attacker. The proposed model is a combination of verification and load balancing, with the concept of fuzzy logic, which helps to detect the attack easily than the others with an average accuracy of 93%. The model effectively removes the high traffic (request) from a single session id and also verifies the other request by some criteria discussed in the implementation session.

The future work will focus on offering an improved method for identifying low-rate DDoS assaults, which are a concern in cloud systems. Also the improved method should reduce the fatal rate from 0.2% to 0%.

REFERENCES

[1] Arbor Networks, “Worldwide Infrastructure Security Report Volume XI,” 2015.

[2] Arbor Networks Technical Report, “Worldwide Infrastructure Security Report Volume XII,” 2016.

[3] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, “DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions,” ACM Computing Surveys, 2015.

[4] DDoS attacks in Q1 2018, Kaspersky Lab, 2018. [Online]. Available: https://securelist.com/ddos-report-in-q1-2018/85373/. [Accessed: 18-Mar.-2019].

[5] P. Nelson, Cybercriminals Moving into Cloud Big Time, Report Says, http://www.networkworld.com/article/2900125 / malware cybercrime/ criminals-/ moving-into-cloud-big-time says- report.html 2015.

[6] Tara Seals, DDoS Attacks Spike, Targeting Cloud, http://www.infosecurity-magazine.com/news/q1-2015- ddos-attacks-spike/ 2015.

[7] O. A. Wahab, J. Bentahar, H. Otrok and A. Mourad, "Optimal Load Distribution for the Detection of VM-Based DDoS Attacks in the Cloud," in IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 114-129, 1 Jan.-Feb. 2020, doi: 10.1109/TSC.2017.2694426.

[8] X. Liang and T. Znati, "An empirical study of intelligent approaches to DDoS detection in large scale networks," 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 821-827, doi: 10.1109/ICCNC.2019.8685519.

[9] H. Kousar, M. M. Mulla, P. Shettar and N. D. G., "DDoS Attack Detection System using Apache Spark," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9457012.

[10] A. Alsirhani, S. Sampalli and P. Bodorik, "DDoS Detection System: Utilizing Gradient Boosting Algorithm and Apache Spark," 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), 2018, pp. 1-6, doi: 10.1109/CCECE.2018.8447671.

[11] Ashley Chonka, Yang Xiang, Wanlei Zhou, Alessio Bonti, Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks, Journal of Network and Computer Applications,Volume 34, Issue 4, 2011, Pages 1097-1107.

[12] S. Yu, Y. Tian, S. Guo and D. O. Wu, "Can We Beat DDoS Attacks in Clouds?," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245-2254, Sept. 2014, doi: 10.1109/TPDS.2013.181.

[13] Bikram Khadka, Chandana Withana, Abeer Alsadoon, Amr Elchouemi, 2015. Distributed Denial of Service attack on Cloud Detection and Prevention. School of Computing and Mathematics, Charles Sturt University, Sydney, Australia Hewlett Packard. International Conference (pp. 1-5). IEEE.,2015.

[14] A.Saravanan, S.Sathya Bama, Multi-Model Anti-Ddos Framework For Detection And Mitigation Of High Rate Ddos Attacks In The Cloud Environment, International Journal Of Scientific & Technology Research, Volume 9, Issue 03, pp.4503-4511. 2020.

[15] Liu Z G, Yin X C, Lee H J. A new network flow grouping method for preventing periodic shrew DDoS attacks in cloud computing,18th International Conference on Advanced Communication Technology(ICACT), 66-69. 2016.

[16] .Sahi A, Lai D, Li Y.,An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access,6036-6048, 2017.

[17] Jeyanthi N, Barde U, Sravani M. ,Detection of distributed denial of service attacks in cloud computing by identifying spoofed, International Journal of Communication Networks & Distributed Systems, 262-279. 2013.

[18] Navaz A S S, Sangeetha V, Prabhadevi C.,Entropy based anomaly detection system to prevent DDoS attacks in cloud, International Journal of Computer Applications, 42-47, . (2013).

[19] Wang B, Zheng Y, Lou W. ,DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks, 81(C): 308-319, 2015.

[20] Saravanan, A., Bama, S.S., Kadry, S. and Ramasamy, L.K., A new framework to alleviate DDoS vulnerabilities in cloud computing. International Journal of Electrical & Computer Engineering (2088-8708), vol.9,2019.

[21] G. Somani, M. S. Gaur, D. Sanghi, M. Conti and M. Rajarajan, "Scale Inside-Out: Rapid Mitigation of Cloud DDoS Attacks," inIEEE Transactions on Dependable and Secure Computing, vol. 15, no. 6, pp. 959-973, 2017.

[22] Karan B. Virupakshar, Manjunath Asundi, Kishor Channal, Pooja Shettar, Somashekar Patil, D.G. Narayan,” Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-based Private Cloud”, Procedia Computer Science, Volume 167, pp. 2297-2307, 2020.

[23] Singh, P.K.: Three-way fuzzy concept lattice representation using neutrosophic set. Int. J.Mach. Learn. Cybernet. Vol 8,issue 1, pp. 69–79 ,2017.

[24] Tasnuva Mahjabin, Yang Xiao, Guang Sun, Wangdong Jiang,” A survey of distributed denial-of-service attack, prevention, and mitigation techniques”,International Journal of Distributed Sensor Networks,volume 13 ,issue 12,2017.

[25] Lee, T.-H.; He, J.-D.: Entropy-based profiling of network traffic for detection of security attack. In: TENCON 2009-2009 IEEE Region 10 Conference, pp. 1–5, 2009.