Wireless Personal Communications (2021) 117:3487–3503
https://doi.org/10.1007/s11277-021-08280-y

A Profile-Based Novel Framework for Detecting EDoS Attacks in the Cloud Environment

J. Britto Dennis[1] · **M. Shanmuga Priya**[2]

Accepted: 4 February 2021 / Published online: 21 February 2021

Abstract
The future of information technology mainly depends upon cloud computing. Hence security in cloud computing is highly essential for the consumers as well as the service providers of the particular cloud environment. There are many security threats are challenging the current cloud environment. One of the important security threat ever in cloud environment is considered to be the Distributed Denial of Service (DDoS) attack. Where cloud is of greater benefit in terms of providing on-demand services, a certain kind of attack named as Economic Denial of Sustainability (EDoS) occurs in pay per use payment model. Due to the occurrence of this attack the consumers are forced to pay additional amount for the services offered. EDoS attacks are similar to that of DDoS attacks Which is classified as-attacks associated with bandwidth consuming, application targeted attacks and the exhaustion of the connection layer. The main objective of the proposed work is to design a profile-based novel framework for maximizing the detection of various types of EDoS attacks. During this process, the proposed framework consisting Feature Classification (FC) algorithm ensures that false positives and negatives along with bandwidth and memory consumption are highly minimized. The proposed algorithm allows only the limited resources for allocation to the available virtual machines which increases the chances of the detecting the attack and preventing the misuse propagation of resources. The accuracy and efficiency of this approach is proven to be higher with lesser computational complexity when compare to the existing approaches.

Keywords DDoS attacks · EDoS attacks · On-demand services · Cloud computing

✉ J. Britto Dennis
brittodennisj@gmail.com

M. Shanmuga Priya
ssg_priya@mamce.org

[1] Department of Information Technology, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India

[2] Department of Computer Science and Engineering, M.A.M College of Engineering, Trichy, Tamil Nadu, India

1 Introduction

A drastic change has been brought in the field of information technology by cloud computing where it offers the computing resources in the form of services. Based on the requirements of the system, the scalable resources are provided by the cloud based on-demand. Thus, the need for vast number of computer system is greatly reduced. Though the technology seems to be emerging over the years, it is concerned with several security issues [1]. For internet-based applications, Distributed Denial of Service (DDoS) attacks seems to pose a great threat. Botnets are mainly responsible for these types of attacks. In this type of attack, several machines target the same service making it difficult for the service to be provided. As a result of this attack, the server load is drastically increased thereby making the system difficult to access. Machine learning based EDOS attack detection was introduced [2].

There no preventive measure found till date that prevents the DDoS attack to its fullest. Hence, in order to protect cloud against the emerging security threats it is essential to raise the guard and to identify means to detect these attacks as far as possible. As with the DDoS attack, the server is flooded with the incoming packets, thus increasing the traffic at the server end. Because of its distributed nature it finds quite difficult to detect the DDoS attacks. One kind of DDoS attack that is very specific in cloud environment is the Economic Denial of Sustainability (EDoS) attack. Various intrusion detection scheme along with its issues and parameters are discussed [3].

As defined by the National Institute of Science and Technology (NIST), providing on-demand services is considered as the greatest benefit offered by cloud. This implies that the resources can be obtained based on demand whenever required. Payment for these resources can be made based on pay-per-user method. Detection of EDOS attacks in self organizing attacks was proposed [4, 5]. Based on this benefit of cloud computing, EDoS is designed. The implication of this attack results in the customer paying extra amount for the services offered to the cloud service provider. This problem can be prevented by limiting the allocation of the resources. Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment was proposed [6]. The allocation of resources can be controlled by detecting both the DDoS and EDoS attacks which in turn helps in securing the cloud environment. The existing algorithms however focuses on mitigating these kinds of attacks. To be known, there is no such detection mechanism available that is capable of detecting the various kinds of EDoS attacks. Hence, anomaly detection technique is used for detecting both the DDoS and EDoS attacks in the cloud environment. The proposed approach focuses on the idea that the traffic generated in case of normal situation is normal whereas in case of the DDoS and EDoS attacks are not. Hence sample traffic over a period of time and resource usage is obtained form the virtual machine based on which a profile containing the normal and abnormal behaviour of the virtual machine is created. Distributed denial of service attacks detection in cloud computing using extreme learning machine was proposed [7, 8]. The paper contributes towards the following.

- The work concentrates on the development of the Anomaly detection system based on resource usage and behaviour of the virtual machine during network traffic in detecting the DDoS and EDoS attacks.
- An algorithm for detecting HTTP attack and database attack is proposed.

There are two different phases where machine learning algorithms are applied. In phase 1, the metrics are analysed that helps in the detection of attacks. In the phase 2, it involves the detection of attacks where a profile is defined to collect the resources. Though threshold method is considered as a traditional way of detecting the traffic [9], it has its own drawbacks when it comes to the introduction of false positives. The machine learning techniques are used to reap the benefits of the intrusion detection system in particular anomaly-based detection for detecting the EDoS attacks.

In the remainder section of this paper, the Sect. 2 survey about the previous works of several authors, Sect. 3 includes the proposed system architecture and the methodology, Sect. 4 explains the process of profile creation and the results of the same are discussed in the Sect. 5. Section 6 includes the conclusion of the paper and discusses about the future work.

2 Related Works

Since heavy workload is caused by the virtualization technology of the cloud computing on the server, a powerful algorithm is required for the detection of the attacks whose overload capacity is to be very low [7]. To overcome this issue an openstack integrated firewall and raw socket programming was proposed for monitoring traffic in the network [8]. An EDoS-Shield has been proposed to mitigate such attack in cloud which divides the request received from the user into two-categories. One category group the legitimate requests whereas the other category groups the requests generated by the bots. This approach makes use of a verifier that verified the requests received from the client. The results of the verification process involve adding these requests either to the blacklist or toe the white list which corresponds to the legitimate requests as well as the request generated by the bots. The virtual firewall blocks the requests that are present in the blacklist whereas those requests that are present in the whitelist are connected to the cloud services for them to access it whenever required.

On the other hand, an architecture [9] has been proposed to mitigate the web service based EDoS attacks. This work deals with the generation of client puzzle that helps in the identification of the legitimate user. The cloud services can be used by the client upon solving this puzzle. The system information such as the load of the server and the bandwidth helps in determining the severity of the puzzle. Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments had been done.

Another work [10] proposes enhanced mitigation framework for the detection of EDoS attacks. Here a Graphical Turning Test is performed on the very first packet received from the receiver that helps in differentiating the legitimate use from the group of malicious users. Here an intrusion prevention system (IPS) is used in detecting the malwares present inside the packets. The detection of DDOS attacks in software defined networks was discussed.

A framework [11] for e-commerce applications called as EDoS Armor has been proposed for mitigating the attacks in a cost-effective manner. Here, overwhelming of resources can be prevented by limiting the number of users accessing the application. The learning mechanism assigns priority to the user to determine the sharing of resources among the users. When the priority value is high, the user is allocated with a greater number of resources and otherwise when the priority is low. Benchmark-Based Reference

Model for Evaluating Botnet Detection Tools Driven by Traffic-Flow Analytics was proposed.

Another approach [12] has been proposed to mitigate the access of virtual resources in cloud. Here the user request to access the service is controlled by the service provider. These requests are either classified as normal requests or malicious requests. The request from the normal users are provided with higher priority while the requests from the suspicious users are provided with lower priority. The lowest priority ensures that the resources are removed from the list of services offered. Semi-supervised learning based distributed attack detection framework for IoT was proposed.

IP spoofing in EDoS [13] can be identified and mitigated by means of the use of Time to Live field which is present in the IP header. The incoming packets can be classified as either normal packets or malicious one based on the threshold value. Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering was proposed.

Based on the time spent on the web page, the HTTP EDoS attacks can be detected. This time spent on the web page varies between normal and abnormal situation. The mean value of the time spent on the web page is plotted in the form of the graph. These graphs are constantly monitored by the cloud administrator. A new feature selection method was used to improve the document clustering using particle swarm optimization algorithm. However, the proposed approach in this paper can only detect HTTP based EDoS attacks and not any other attacks.

Similar other method to detect the EDoS attacks by using the threshold value is proposed in this paper. A self-adaptable system for DDoS attack prediction based on the metastability theory was proposed [14]. Here an approach has been proposed to monitor the usage of the resources and its impact in the cloud-environment is listed in this paper.

3 Proposed Methodology

The attacks on the virtual server can be classified into three types namely.

- Attacks that are Bandwidth-consuming
- Target specific attacks
- Attacks exhausting the network connection.

3.1 Attacks that are Bandwidth-Consuming

The entire bandwidth concerning the target is consumed in this kind of attack. This is the cause of Denial of Service attack as quick responses is not given to normal requests. Autoscaling intrusion detection and prevention system for cloud was proposed [15]. One such attack is the HTTP attack.

3.2 Target Specific Attacks

Here, the application in the server is targeted by the attack. The virtual machine in the server containing the application is affected by the attacker. In the proposed work, the database is chosen to be the application which is considered as the target of the attack and the work aims at identifying the virtual machine holding the application [16].

3.3 Attacks Exhausting the Network Connection

The protocol features are used in the attacks concerning the connection layer which are directed towards the server. Some of these attacks includes UDP flood attacks, TCP SYN attacks and many more. These attacks are supposed to exhaust the connection layer since these attacks happens at the time the connection is established between the client and the server. It is essential to add the profile features to detect this kind of attack [17].

One another main concept in cloud computing is Isolation. It means that the services are located in individual virtual machines such that databases and the server are placed in different locations. The events and the requests received from the virtual machine can be monitored with the help of the hypervisor [18]. When a request is received from the virtual machine regarding the need for more resources, the monitoring system established monitors the kernel system of the virtual machine and the results are compared with the profiles established. If the results obtained shows no sign of the attacks then the results are allocated. On the other hand, if there are any chances of attacks found then the resources are not allocated. Thus, an algorithm that is generic to all kind of attacks were introduced in this paper. The proposed algorithm produces more reliable results than the previous other algorithms. The proposed algorithm is described in the Fig. 1. For the current research work, the on premise private cloud owned by the ANNA university has been used. The Cloud OS used ESXI. The Compute node configuration is 8 GB Ram, 1 TB hard disk, I3 core processor. 2000 compute nodes are used and these nodes are virtualized into 10,000 VMs.

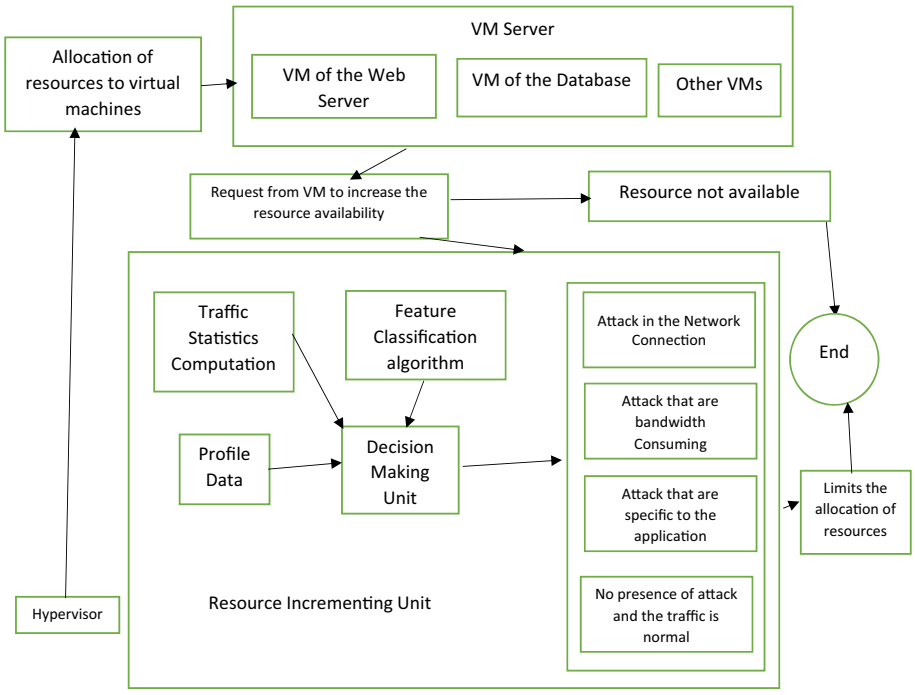

Fig. 1 Proposed architecture model

3.4 Feature Extraction Process

One of the important work is collecting the samples. Based on the collected samples, it is essential to have from hypervisor both normal traffic as well as normal resource pattern. Hence it is created intentionally in the first step. The samples are created in such a way that ample samples are available to handle the traffic at peak hours. Also, these samples are able to handle both heavy traffic as well as light traffic. Any misuse in the resource usage and the traffic can be investigated by creating a threshold from the normal situation described earlier and investigating the behaviour. Detection of economic denial of sustainability (EDoS) threats in self-organizing networks was proposed [19].

3.5 Monitoring of Resources and Sampling

Based on the collected samples, of the resource usage and to investigate them, the traffic for the attacks are created. Upon monitoring the virtual machine for resource usage, it is found that, whenever more resources are requested by the virtual machine, the monitoring and sampling module is evoked. Thus, the module is designed as a part of the proposed architecture. The module is placed inside the hypervisor since all the traffic is bound to be passed through the hypervisor [20].

3.6 Identification of Attacks

The proposed approach insists on the approach that though the packets can be duplicated by the attackers, it is difficult to duplicate the amount of traffic generated when an attack happens. Similarly, when compared to the normal situation the resource usage is different in attack scenarios. It is to be noted that the attack type is specific to the resource used and the traffic as each virtual machine are targeted by different type of attacks [20]. The attacks towards the virtual machine are identified as follows.

- In case of EDoS attack which includes both database attacks and HTTP attacks, the HTTP attacks are targeted by the attackers whereas the database attack produces many database queries for each HTTP request to the virtual machine
- When a connection is established with the virtual machine of the web server the TCP SYN attack occurs.

As a part of the model design, the features of the traffic as well as the resource usage is collected and their respective behaviour is investigated.

The following Fig. 1 shows the architecture diagram for the proposed model. Here when the request is sent from the virtual machine to the decision unit to increase the availability of resources. In the resource incrementing unit the decision making unit then makes decisions based on the feature classification algorithm. In case if the decision unit finds the attacks in either the network connection or the bandwidth or the applications, then the resource allocation is [21]. Else with the help of the hypervisor it allocates the resources to the Virtual Machine server.

3.7 Essential Metrics

The metrics that are essential for designing the model for detecting the attacks are proposed in this section. The metrics are chosen from the existing works of various authors [21]. In the proposed algorithm, the metrics are not prioritized. The results thus obtained are compared with the states identified and the attack type is detected. This attack type is classified as A1, A2 and A3. The essential features that are to be extracted is not limited to the following Table 1.

Once the samples are created from the virtual machine, a behavioural profile is created which is used for the detection of attacks in the detection phase. Since the features are considered for creating a profile for the virtual machine, the proposed machine learning algorithm should check the extracted features before making the decision regarding the normal traffic as well as the traffic when an attack happens. The above-mentioned features (F1 to F18) helps in the classification of the traffic attack. This knowledge about the attacks is later used in the training phase of the machine learning algorithm.

The Augmented Strategy (AS) is used for identifying the suitable category of network traffic for updating the profile. Using this strategy, the features are added to the profile and the data is used for increasing the accuracy of the proposed framework. For each feature corresponding to the arrival of the traffic, the average of the data collected is calculated. Thus, the average of the data collected previously corresponding to the time t is given by *Avg(t-1)* and the corresponding data collected equals to *n-1*.

Avg(t-1) is calculated using the following formula

$$Avg(t-1) = \sum_{i}^{n-1} D_i \tag{1}$$

where D_i represents the previous data present in each feature.

The new average for all the features can be calculated using the Augmented Strategy (AS) as follows

$$Avg(t) = (1-\rho)Avg(t) + \rho D_i \tag{2}$$

here ρ is the weight factor for the data obtained previously and is calculated as follows

Table 1 Traffic-related features

TSW—Time Spent on the Webpage	IIOW—Incoming I/O on the Web server
OIOW—Outgoing I/O on the Web server	IIOD—Incoming I/O on the database
OIOD—Outgoing I/O on the database	CW—CPU usage on the webserver
CD—CPU usage on the database	MeW—Memory usage on the Webserver
Nph—the usage of network bandwidth in the web server per 1 h	Npd—the usage of network bandwidth in the web server per day
Npw—the usage of network bandwidth in the web server per week	Npm—the usage of network bandwidth in the web server per minute
S(SYN)—the SYN packets ratio in the TCP packet	S(ACK)—the SYN packets ratio in the TCP packet
S(SYN + ACK)—both the packets ratio in TCP	Ni—packets incoming ratio in the given second
No—outgoing packets ratio in the given second	NOP—the packet connections that are not fully opened

$$\rho = \frac{2}{n+1} \tag{3}$$

4 Algorithms for Detecting the Attacks

Whenever the demand for the resources increases in the virtual machine increases, the resources are allocated by the virtual machine to the hypervisor. There are chances of migrating the virtual machine to another hypervisor in case if the resources are not sufficient. The framework is used in cases where the request sent from the virtual machine is received by the Traffic Computation part. It is at this moment that the virtual machine features are extracted. The results of the Traffic Computation part is then sent to the verdict section. In this section, the arrival of the traffic is detected by the proposed algorithm and the probability of each attack that may occur is calculated.

The profile data is then used to train the machine learning module. Thus the traffic features are then compared with the profile features in the verdict section and based on the results obtained the traffic is fit to any of the four previously mentioned categories—attacks that are bandwidth-consuming, attacks that exhausts the connection-layer, normal traffic or attack that is specific to an application.

The resource allocation is limited in cases when the above-mentioned attacks are detected. If not, the virtual machine is allocated with the resources. The system sensitivity plays a major role in determining the attack percentage. This in turn result in allocation of resources being limited. Due to the security of the virtual machines the resources are limited. Because the resources used in the cloud environment is heterogeneous resources which in turn have large collection of resources. So the resources are limited to provide the security to the virtual machines.

The metrics that were defined earlier were used in the proposed framework for the detection of attacks. Each attack has different kind of attacks that helps in detecting against each other. The proposed framework is compared against those algorithms that does not consider the metrics in detecting the attacks. Anyway all the metrics are used by each and every attacks as well as by all the methods which comes under the detection of the EDoS attacks. So the metrics are related automatically. These algorithms are named as NSL-KDD, CAIDA and CICDDoS2019 respectively.

The pseudocode for the proposed Feature Classification (FC) algorithm is as follows.

FC algorithm pseudocode

Initialize *the profile data*

Initialize *the profile features*

FOR *each feature*

Read the profile data

Store it in an array S[ ]

End FOR

FOR *each feature*

Initialize *the parameters x, y and z, Sp*

Analyse *the states based on the signs*

Calculate *the sign value*

$X = S(SYN) = S(SYN+ACK) + S(SYN+ACK)$

Check ***IF***

$Y = S(ACK) + S(ACK+SYN) = Sum\ of\ all\ the\ packets\ of\ all\ the\ samples$

$Z = S(ACK) + S(ACK+SYN) \geq Sum\ of\ all\ the\ packets\ of\ all\ the\ samples$

End IF

$S(SYN) = S(SYN+ACK)$

IF $X = Y = 1$ ***THEN***

$S2 = -3$

ELSE

$S2 = -1$

End FOR

The features corresponding to the normal traffic is shown in the following Table 2. The following Table 3 shows the features and its corresponding type of attacks.

4.1 Detection of HTTP Attacks

When the hypervisor receives the request for the resources from the virtual machine, the hypervisor analyses the virtual machine that has sent the request and other virtual machines that are in close proximity with the one that has sent the request. Based on this the following states are analysed. It the virtual machine is found to contain a HTTP request by the hypervisor, then the hypervisor may not allocate any further resources to that particular virtual machine. HTTP attack is found to have occurred when the following Table 4 are noticed.

Table 2 Features corresponding to the normal traffic

Features	Normal traffic
TSW	Matches with the respective profile features
I/O of the webserver in the web server of the virtual machine	The features are similar to the I/O features in the database of the virtual machine
I/O of the database in the web server of the virtual machine	The features are similar to the I/O features in the web server in the virtual machine
The CPU usage of the webserver in the virtual machine	The CPU usage is similar to that of the database in the virtual machine
The CPU usage of the database in the virtual machine	The CPU usage is similar to that of the web server in the virtual machine
Memory usage of the web server	Matches with the other profile features

Table 3 Features corresponding to the attack types

Features	HTTP attack	Database attack	Tcp SYN attacks
TSW	Small	High	High
I/O of the webserver in the web server of the virtual machine	High	High	Low
I/O of the database in the web server of the virtual machine	Low	High	Charge
The CPU usage of the webserver in the virtual machine	Low	High	Low
The CPU usage of the database in the virtual machine	High	Low	High
Memory usage of the web server	Small	Small	Huge

Table 4 Features for detecting HTTP attacks

S1: whenever there is a sudden decrease in the TSW value
S2: whenever the IIOW value increases and IIOD value decreases
S3: when the OIOW value increases and the OIOD decreases
S4: when the CW value increases and CD value decreases
S5: sudden increase in the Nph value
S6: sudden increase in the Npw value
S7: sudden increase in the Npm value

The state parameter for the HTTP attack is given by Sp. If the value of p = 1 THEN it indicates that the sign is met ELSE it indicates that the sign is not met.

4.2 Detection of TCP Syn attack

The algorithm A2 uses a specific set of features to detect the TCP Syn attack. Some of these features that are used for detecting the attack in database is described below in Table 5.

Table 5 Features for detecting the T33CP SYN attacks

S1: Sudden increase in the MW value of the TCP Syn attack
S2: S_SYN, S_ACK and S_SYNACK are interrelated
S3: there is a sudden increase in Ni value of TCP flooding attack
S4: There is a sudden increase in No value of TCP flooding attack
S5: There is a sudden increase in NOP value of TCP flooding attack

The sign value based on its condition can be either 0 or 1. For sign 2 the value may vary between -3 and $+3$ based on the occurrence (Fig. 2). Such occurrences corresponding to the various equations are defined as follows.

E1: S(SYN) = S(SYN + ACK) + S(SYN + ACK)
E2: S(ACK) + S(ACK + SYN) = Sum of all the packets of all the samples
E3: S(ACK) + S(ACK + SYN) > = Sum of all the packets of all the samples
E4: S(SYN) = S(SYN + ACK)

From Fig. 2, the sign 2 value is calculated. The value of sign 2 is assigned -3 if the value of both Eqs. 1 and 2 is 1. In other the value of both the Eqs. 1 and 2 need to be true so that the value of sign 2 is assigned -3. Or else the value of sign 2 is assigned as -1. The attack probability can be detected later based on these values. This negative parameter results in the shifting of the traffic to its normal state which eventually lessens the chances of attack.

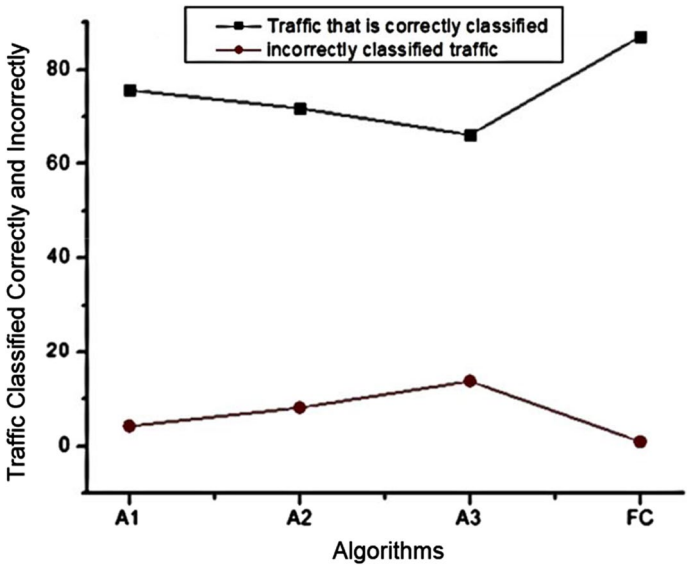

Fig. 2 Evaluation of proposed framework using neural network

4.3 Detection of Database Attack

The Database attack is in a way similar to the HTTP attack, where a hypervisor upon receiving request from the virtual machine, the virtual machine in turn is inspected by the hypervisor when a request is received and also inspects the communication of one virtual machine with the other. If any of the following signs are found in the virtual machine, it indicates that there are chances of occurrence of HTTP and database attacks. Based on their features, these attacks can be identified with the help of A3 algorithm. The following signs shown in Table 6 are made use by the algorithm in detecting the attacks.

4.4 Formula to Detect the Attacks

Once the above signs are identified, the attack state of the virtual machine is investigated. The probability that the attack is detected is calculated using the following formula

$$P(t) = \frac{\sum_{i=1}^{n} S_{ij}}{\sum_{i=1}^{n} S_{ijmax}}$$

Here i = sign number, j = attack type, n = total number of conditions, $P(t)$ = attack probability.

5 Results and Discussion

In the implementation of the prototype, the metrics are extracted and the virtual machine is allocated to the server as well as to the database. The input to the prototype is the metrics of the traffic arrived. The proposed framework is explained in the following section. The abstract view of the calculation is given through configuration of the setup which includes the concept of IPTraffic and Vnstat. The system evaluation is done through the metrics of the algorithms NSL-KDD, CAIDA, CICDDoS2019. Weka tool implements neural network based machine learning. The proposed FC algorithm provides the information of traffic which is classified correctly and incorrectly. Finally the detection rate is compared with the SNORT algorithm.

5.1 Configuration of the Setup

HTTP flooder and LoadRunner is used for performing HTTP attack and database attack respectively. *IPTraffic* is used for identifying the network statistics. The bandwidth usage can be measured with the help of Vnstat. The *Top* command is used for monitoring the usage of the memory and CPU. The results obtained from the detection algorithm implies

Table 6 Features for detecting the database attacks

S1: Sudden increase in the TSW value of the database attack
S2: Sudden decrease in the IIOW value than the IIOD value of the database attack
S3: Sudden decrease in the OIOW value than the OIOD value of the database attack
S4: The CW value is smaller than the CD value

that any of the three attacks are identified then the resources are not allocated to the virtual machine. The proposed system is evaluated based on the metrics and accuracy.

5.2 System Evaluation Using Metrics

The efficiency of the metrics can be evaluated by comparing those metrics against the proposed algorithms A1—NSL-KDD, A2—CAIDA and A3—CICDDoS2019. Based on this the traffic is classified. In order to achieve this, for our proposed framework, a Feature Classification algorithm based on support Vector Machine has been defined. The criteria used for the evaluation of the metrics are percentage of correct traffic classification and the percentage of time. The proposed algorithm is compared against the neural network-based calculation done previously.

5.2.1 Neural Network-Based Machine Learning

The chosen neural network algorithm for traffic classification has been implemented with the help of Weka tool. The neural network is represented using the Multi-Layer perception function of the Weka tool. The training samples (around 65%) containing the 18 features mentioned previously are tested using the neural network algorithm. The remaining 35% of the test samples is later used for testing the same algorithm. The result shows that around 96% of the traffic is correctly classified while the rest is wrongly classified.

5.2.2 Proposed Feature Classification (FC) Algorithm

Like the previous existing works, the proposed algorithm is also tested using WEKA tool. The configuration of the algorithm is similar to that of the neural networks where the 65% of the training sample is tested using the Feature Classification (FC) algorithm and the rest of the samples are tested using the same Feature Classification (FC) algorithm afterwards. The result shows that the proposed algorithm classified the traffic correctly with 99.1% accuracy which is far higher than the other existing algorithms. The following Fig. 2 shows the evaluation of proposed framework using neural networks.

The following Table 7 shows the proposed framework evaluation using neural network. The values shows that the FC algorithm classifies the traffic more accurately than neural networks.

The performance is evaluated by comparing the designed framework based on support vector machine against the three algorithms namely NSL-KDD, CAIDA and CICDDoS2019. The performance evaluation shoes that the performance is far better than the algorithms with which it is compared. Thus, in terms of classifying the traffic,

Table 7 Proposed framework evaluation using Neural networks

Algorithm	Traffic that is correctly classified (in percentage)	Incorrectly classified traffic (in percentage)
NSL-KDD	75.70	4.29
CAIDA	71.82	8.18
CICDDoS2019	66.19	13.81
FC	90	0.94

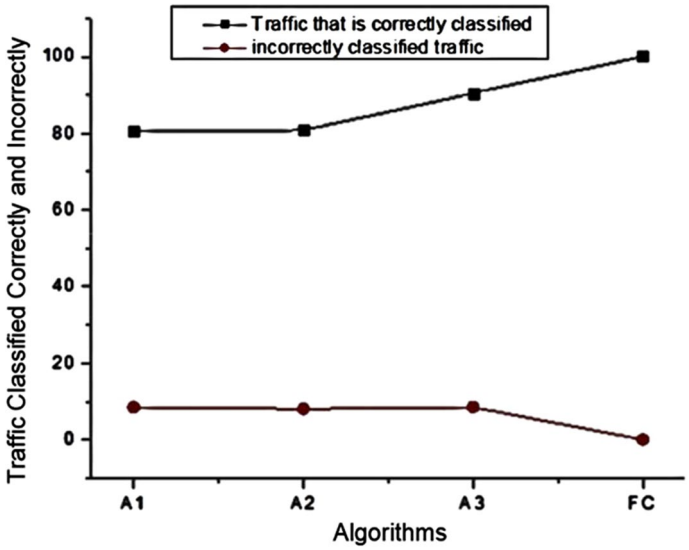

Fig. 3 Framework evaluation using proposed FC algorithm

Table 8 Proposed framework evaluation using FC algorithm

Algorithm	Traffic that is correctly classified (in percentage)	Incorrectly classified traffic (in percentage)
NSL-KDD	80.48	8.52
CAIDA	80.91	8.09
CICDDoS2019	90.48	8.52
FC	100	0

Feature Classification (FC) algorithm is much better and involves less cost in terms of response time. The following Fig. 3 shows the evaluation of the framework using the FC algorithm.

The following Table 8 shows the proposed framework evaluation using FC algorithm. The table values shows that the FC algorithm can correctly classify the traffic more accurately when compared to the other algorithms.

5.3 Evaluation of Accuracy

For the purpose of accuracy evaluation, the detection module is compared against the existing works of other algorithms. In order to accomplish this, an intrusion detection system in open-source named SNORT has been chosen. SNORT is able to detect the attacks correctly up to 70% which is highly reliable and acceptable than the previous works.

The following Fig. 4 shows the detection rates of SNORT and proposed FC algorithm. The graph values show that the detection rate of the proposed FC algorithm is higher than that of the SNORT algorithm.

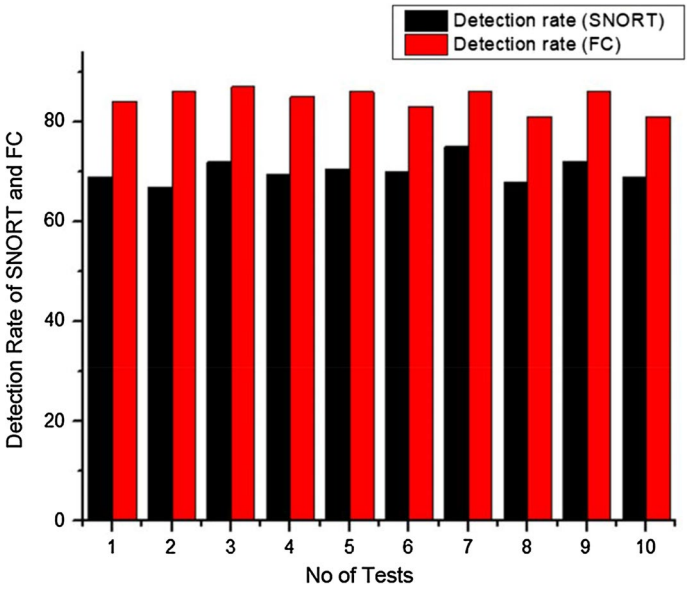

Fig. 4 Comparison of detection rates of SNORT and proposed FC algorithm

5.4 Limitations of the Proposed Algorithm

Though the proposed algorithm is highly reliable and acceptable, it still has its own limitations. In case if both the Database and HTTP attack occurs simultaneously, the traffic seems to be busy and the resource usage is high even under normal circumstances. This may in turn degrade the detection accuracy. Though it is the case, the possibility of occurrence is pretty low.

6 Conclusion and Future Work

One of the biggest concerns in the cloud computing is the security which is the core of discussion of this paper. One of the important types of DDoS attack called the EDoS attack has been discussed in this paper. In order to get rid of those attacks it is essential that these attacks are to be detected as fast as possible. For this, a good framework with 100% reliability is required. Thus, a novel framework based on Feature Classification (FC) algorithm has been proposed in this paper that is capable of detecting various kinds of EDoS attacks. The proposed algorithm is capable of detecting the attacks by correctly classifying the traffic. With this, the process of migration of attacks from the virtual machine to the hypervisor can be prevented. The accuracy of the detection of attacks will be high when a separate framework is used for different types of attacks rather than using the same framework for all the attacks. As said in the proposed framework, the metrics of one attack is involved in detecting the other attack, the framework can be extended for identifying the features of other types of attacks as well. The proposed work can be further extended in the future

to automatically detect the EDoS attacks. Since the paper is not concerned about the real-world network traffic, the issue can be addressed in the future work of this paper.

References

1. Saleh, M. A. & Manaf, A. A. (2015) A novel protective framework for defeating HTTP-based denial of service and distributed denial of service attacks. *The Scientific World Journal*, *2015*, Article ID 238230, 19. https://doi.org/10.1155/2015/238230
2. Abbasi, H., Ezzati-Jivan, N., Bellaiche, M., et al. (2019). Machine learning-based EDoS attack detection technique using execution trace analysis. *Journal of Hardware Systems and Security, 3,* 164–176. https://doi.org/10.1007/s41635-018-0061-2.
3. Shamshirband, S., Fathi, M., Chronopoulos, A. T., Montieri, A., Palumbo, F., & Pescapè, A. (2020). Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues. *Journal of Information Security and Applications*, *55*, 102582. ISSN 221-2126. https://doi.org/10.1016/j.jisa.2020.102582.
4. Monge, M. A. S., Vidal, J. M., & Pérez, G. M. (2019). Detection of economic denial of sustainability (EDoS) threats in self-organizing networks. *Computer Communications*, *145*, 284–308. ISSN 0140-3664. https://doi.org/10.1016/j.comcom.2019.07.002.
5. Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). Hybrid clustering analysis using improved krill herd algorithm. *Applied Intelligence, 48,* 4047–4071. https://doi.org/10.1007/s10489-018-1190-6.
6. Agrawal, N., & Tapaswi, S. (2019). Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges. In *IEEE Communications Surveys & Tutorials*, Vol. 21, no. 4, pp. 3769–3795, Fourthquarter 2019. https://doi.org/10.1109/COMST.2019.2934468.
7. Kushwah, G. S., & Ali, S. T. (2019). Distributed denial of service attacks detection in cloud computing using extreme learning machine. *International Journal of Communication Networks and Distributed Systems (IJCNDS), 23*(3), 328.
8. Virupakshar, K. B., Asundi, M., Channal, K., Shettar, P., Patil, S., & Narayan, D. G. (2020). Distributed Denial of Service (DDoS) attacks detection system for OpenStack-based private cloud. *Procedia Computer Science*, *167*, 2297–2307. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2020.03.282.
9. Dong, S., Abbas, K., & Jain, R. (2019). A survey on Distributed Denial of Service (DDoS) attacks in SDN and cloud computing environments. *IEEE Access, 7,* 80813–80828. https://doi.org/10.1109/ACCESS.2019.2922196.
10. Karan, B. V., Narayan, D. G., & Hiremath, P. S. (2018). Detection of DDoS attacks in software defined networks. In *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, Bengaluru, India, pp. 265–270. https://doi.org/10.1109/CSITSS.2018.8768551.
11. Huancayo Ramos, K. S., Sotelo Monge, M. A., & Maestre Vidal, J. (2020). Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics. *Sensors, 20,* 4501.
12. Rathore, S., & Park, J. H. (2018). Semi-supervised learning based distributed attack detection framework for IoT. *Applied Soft Computing, 72,* 79–89.
13. Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational Science*, *25*, 456–466. ISSN 1877-7503. https://doi.org/10.1016/j.jocs.2017.07.018.
14. Pelloso, M., Vergutz, A., Santos, A., et al. (2018). A self-adaptable system for DDoS attack prediction based on the metastability theory. In *2018 IEEE Global Communications Conf. (GLOBECOM)*, Abu Dhabi, UAE, pp. 1–6.
15. Xing, J., Zhou, H., Shen, J., et al. (2018). AsIDPS: Auto-scaling intrusion detection and prevention system for cloud. In *2018 25th Int. Conf. on Telecommunications (ICT)*, Saint Malo, France, pp. 207–212.
16. Desnoyers, M., & Dagenais, M. (2018). LTTNg: Tracing across execution layers, from the hypervisor to user-space. In *Proceedings of the Ottawa linux symposium.*
17. Alger, L. (2018). *DDoS attackers increasingly abuse public cloud services* (11 September 2018). https://www.devopsonline.co.uk/ddos-attackers-increasingly-abuse-public-cloud-services/.
18. Karakaya, G., Galelli, S., Ahipaşaoğlu, S. D., et al. (2016). Identifying (quasi) equally informative subsets in feature selection problems for classification: A max-relevance min-redundancy approach. *IEEE Transactions on Cybernetics, 46,* 1424–1437.

19. Monge, M. A. S., Vidal, J. M., & Pérez, G. M. (2019). Detection of economic denial of sustainability (EDoS) threats in self-organizing networks. *Computer Communications, 145,* 284–308.
20. Nguyen, T. T. T., & Armitage, G. (2019). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Survey & Tutorials, 10*(4), 56–76.
21. Shon, T., & Moon, J. (2017). A hybrid machine learning approach to network anomaly detection. *Information Sciences, 177*(18), 3799–3821.

Mr. J. Britto Dennis has received his B.E. degree in Computer Science and Engineering from Karunya University, Coimbatore, Tamil Nadu, India in 2008 and M.Tech degree in Computer Science and Engineering from Karunya University, Coimbatore, Tamil Nadu, India in 2010. He is currently pursing Ph.D. in the Department of ICE in Anna University Chennai. Now he is working as Assistant Professor in the Department of Information Technology, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India. He has 10 years of experience in academics and research. His current research areas include cloud computing, Network, Network Security and Machine Learning Techniques. He has published and presented few International journal and conference papers. He is a Life Member of the Indian Society for Technical Education (ISTE), International Association of Engineers (IAENG) and Society of Digital Information and Wireless Communications (SDIWC).

Dr. M. Shanmugapriya has received her B.E. in Electronics and Communication Engineering from Bharathidasan University, Trichirappalli, Tamil Nadu, India in 1996 and completed a Master Degree in Communication Systems from Anna University, Chennai, Tamil Nadu, India in 2005. She has done his Ph.D. in Design and development of miniaturized Antenna for WSN applications from Anna University Chennai, India in 2015. Now she is working as Professor in the Department of Computer Science and Engineering, M.A.M. College of Engineering, Trichirappalli, Tamil Nadu, India. She has 20 years of experience in academics and research. She has published and presented many International journal and conference papers. Her main research interests are design and development of Microstrip Antenna. She is Life Member of the Indian Society for Technical Education (ISTE) and Institute of Engineers (India).