

International Journal of Information and Computer Security

ISSN online: 1744-1773 - ISSN print: 1744-1765

<https://www.inderscience.com/ijics>

A novel deviation-based detection mechanism for DDoS attacks in the cloud environment

A. Somasundaram, S. Devaraju

DOI: [10.1504/IJICS.2025.10071967](https://doi.org/10.1504/IJICS.2025.10071967)

Article History:

Received:	13 December 2022
Accepted:	17 October 2024
Published online:	05 September 2025

A novel deviation-based detection mechanism for DDoS attacks in the cloud environment

A. Somasundaram*

Department of Computer Applications,
Sri Krishna Arts and Science College,
Coimbatore, India
Email: somasundaram.a@gmail.com
*Corresponding author

S. Devaraju

School of Computing Science and Engineering (SCSE),
VIT Bhopal University,
Bhopal, India
Email: devamcet@gmail.com

Abstract: Distributed denial-of-service attacks are prevalent vulnerabilities in cloud computing, causing disruption to legitimate users. Despite existing detection methods, reliability and accuracy need improvement. A systematic approach is urgently needed for both spoofing and non-spoofing attacks. To distinguish attacks from legitimate network traffic, this paper proposes a deviation-based detection mechanism based on software-defined networks. The model has two significant phases such as knowledge acquisition and deviation-based detection. The model makes use of the variance of a discrete probability distribution on the network features that are used to collect the knowledge base. For the known flow, the deviation between the traffic and the knowledge base is evaluated to determine the attack traffic. The rule-based detection mechanism is proposed for detecting attacks in the unknown flow. The proposed model, analysed through experimental analysis, demonstrated an average detection rate of 98% and an execution time of 0.72 seconds, outperforming its competitors.

Keywords: DDoS attack; cloud environment; attack detection; variance; discrete probability distribution; traffic representatives.

Reference to this paper should be made as follows: Somasundaram, A. and Devaraju, S. (2025) 'A novel deviation-based detection mechanism for DDoS attacks in the cloud environment', *Int. J. Information and Computer Security*, Vol. 28, No. 1, pp.51–72.

Biographical notes: A. Somasundaram obtained his Bachelor's degree in Computer Science in 2001, a Master's degree in Computer Applications in 2004, and a PhD in Computer Science in 2022 from Bharathiar University, Coimbatore. He has over 20 years of teaching experience and is currently serving as an Assistant Professor in the Department of Computer Applications at Sri Krishna Arts and Science College, Coimbatore, India. He has contributed a significant number of articles to international journals, conferences, edited books, and book chapters. His research interests include network security, intrusion detection, computer networking, human-computer interaction, and cloud security, with a particular focus on denial-of-service attacks.

S. Devaraju received his MCA and MPhil degrees from Periyar University, Salem, and an MBA in Human Resource Management from Madurai Kamaraj University, Madurai. He was awarded a PhD in Computer Applications by Anna University, Chennai, in May 2017. He has over 24 years of experience and is currently serving as a Senior Assistant Professor in the School of Computing Science, Engineering and Artificial Intelligence (SCAI) at VIT Bhopal University, Madhya Pradesh. He has published more than 9 patents, 15 book chapters, and 70 papers in international journals and conference proceedings. He also serves as a reviewer for several reputed journals and conferences. His research interests include network security, intrusion detection, soft computing, and wireless communication.

1 Introduction

Cloud computing is an emerging technology in today's business world. Clients who use cloud computing can access resources, which means they do not need specialised hardware or software to carry out their complex activities. In terms of information technology requirements, cloud computing has emerged as the most practical and innovative architecture for businesses worldwide. Instead of using local computers for daily operations such as data storage, information processing, hosting, and the usage of applications, people frequently use remotely hosted servers over the internet. Though the cloud is popular due to its various merits, its popularity also makes it a well-known target for attackers and hackers due to its widely adopted technology and ease of access (Khorshed et al., 2012). During the past ten years, there have been several initiatives on cloud defence to safeguard cloud data and its resources against attackers.

However, a distributed denial-of-service (DDoS) attack has become the most common vulnerability in the last decade. The DDoS attack has affected numerous servers in a dispersed network, which has become increasingly common. Though the motivations for DDoS attacks vary from one case to another, the main goal is to block the server's activity by repetitively forwarding too many packets (Mirkovic and Reiher, 2004). Thus, DDoS attacks happen when several hacked systems flood one or more web servers with illegitimate requests in a distributed environment. In a DDoS flooding attack, the attacker commandeers a small number of hosts scattered around the network, also referred to as 'zombie hosts' or 'bots', and insists on transmitting a large number of attack packets that seem legitimate (Somani et al., 2017).

Thus, the main goal of a DDoS attack is to block genuine people from accessing an application or network service. Once the server is crashed with more requests, it takes longer to restore a server after a successful DDoS attack (Mousavi and St-Hilaire, 2015). To distinguish DDoS attacks from normal network traffic, DDoS detection techniques use statistical and machine learning algorithms over network data. The detection methods also help to carry out effective attack mitigation. DDoS detection must be effective in promptly blocking the attacks, for which two performance criteria must be met:

- 1 detection speed
- 2 detection accuracy.

Therefore, while creating an effective DDoS defence, these performance criteria are to be considered seamlessly (Zhijun et al., 2020).

Detecting DDoS attacks effectively is difficult for several reasons (Deshmukh and Devadkar, 2015). These include:

- 1 the fact that it is difficult to identify the characteristics of the attack traffic
- 2 the lack of coordination between coherent network nodes
- 3 the increase in the complexity of the attack detection tool, which lowers its usage
- 4 the difficulty in identifying the attack's origin due to the widespread use of address fraud
- 5 the time duration of the attack.

When selecting a solution for DDoS attacks, many things are to be taken into account. The primary considerations for choosing a defence solution are as follows (Wang et al., 2015):

- **Functionality:** Regardless of how strong the attack is, the solution must be effective to lessen its impact.
- **Transparent:** The solution must be easy to apply, which implies that it should not require any modifications to the existing network infrastructure.
- **Lightweight:** Most importantly, the solution does not increase the complexity of the procedure.
- **Precise:** The solution must ensure that it produces fewer false positives and that genuine traffic cannot be dropped.

Though there are several DDoS protection systems available in the literature, they are unable to provide an accurate solution because of their shortcomings, such as a low detection rate, high computational complexity, and high false positives and false negatives. Thus, to address these issues, the deviation-based DDoS detection mechanism (DDDM) is proposed, which aims at detecting various attacks from known and unknown flows. For known flows, the generalised variance of a discrete probability distribution and a distance metric are used to accurately identify the attacks. In order to identify the attacks from the unknown flow, the method implements rule-based analysis. Thus, the proposed work is designed with the following goals:

- 1 to detect flow-based attacks
- 2 to detect DDoS attacks with increased accuracy
- 3 to use deviation-based analysis to detect attacks from known flows
- 4 to use rule-based detection for unknown flows.

The results of the experimental analysis performed indicate that the proposed method has increased the detection rate and decreased false positives and false negatives more than other competitors.

The remainder of the paper is organised as follows: The literature study on DDoS attacks and various notable detection mechanisms is inspected in Section 2. The

suggested methodology for identifying DDoS attacks from other normal traffic is described in Section 3. The experimental results and interpretation are described in Section 4, and the conclusion and discussion of the potential scope and necessary improvements to the underlying research are covered in Section 5.

2 Related work

The rise in security concerns and the increased harm caused by DDoS attacks have motivated the development of various forms of attack detection techniques. These methods differ according to the purpose of the detection and the set of rules required for the service. The majority of these methods concentrate on identifying network traffic abnormalities. Upon a quick identification of a DDoS attack, the packets should be discarded at the source to lessen the impact of attack traffic on the cloud environment (Mondal et al., 2017; Alzahrani and Hong, 2017). Determining attack packets from real network packets is therefore essential for identifying DDoS attacks.

A flow-based mechanism is a common and simple approach to detecting flood-based attacks. A model was proposed that performs statistical computation on incoming packets based on the ports and protocols and count-based analysis with the source and destination addresses (Saravanan and Bama, 2020). However, few authors apply feature selection techniques for selecting significant features from the network packets that help to distinguish attack packets from the real ones. Once the features are selected, the classifiers are used to detect attacks (Devaraju and Ramakrishnan, 2011). A wavelet-based feature selection model was proposed to extract features using the wavelet transform and then apply semi-supervised learning to classify the attack packets. Though the model seems suitable for classifying attacks, its complexity is quite high compared to other models (Bhaya and Manaa, 2014). A clustering-based DDoS attack detection technique that makes use of rules generated from the centroid was suggested. Despite the use of the model in real-time applications, it suffers from high false alarm rates due to increasing false positive rates (FPRs) (Srihari and Anitha, 2014).

A novel location separation technique was suggested to detect DDOS attacks. It works by separating the identifiers from the locators, and these identifiers are further used for node replacement. It is not perfectly suited to detect DDoS attacks since it was originally suggested to solve scalability issues on the internet (Luo et al., 2013). Another model that uses the variance of entropy and feature representative generation to detect DDoS attacks was proposed (Sindia and Dhas, 2017). Similarly, Devaraju and Ramakrishnan (2015, 2019) proposed an entropy-based feature selection method that employs association rule mining to distinguish between normal and intrusive packets. However, as the number of rules generated grows, the algorithm's performance suffers. This method was enhanced by utilising fuzzy control language and a fuzzy rule-based layered classifier on top of entropy-based feature selection in detecting intrusions (Ramakrishnan and Devaraju, 2016; Devaraju and Ramakrishnan, 2020). Unfortunately, the models suffer from high complexity.

In general, attack detection models can be divided into two types: signature-based detection and anomaly-based detection. In the signature-based model, the attacks are identified based on predefined or stored attack patterns (Dimolianis et al., 2021). The limitation of such signature-based detection methods was their inability to identify attacks that did not follow recognised attack patterns. Contrarily, anomaly-based

detection distinguishes the attack packets by comparing the packets with the normal packets set as a baseline (Almseidin et al., 2021). Thus, in the case of anomaly-based detection techniques, the lower layers tend to have more misclassification as a result of the high number of false positives produced (Rawashdeh et al., 2018).

Software-defined networks (SDN) are widely used for investigating and detecting DDoS attacks. A unique technique was proposed to make use of a deep learning algorithm to model attack behaviour based on information gathered from the SDN controller (Ye et al., 2018). The SVM classification algorithm was used to analyse multidimensional data and map low-dimensional, nonlinearly separable data into high-dimensional feature space to make it linearly separable. It was reported that this model performs the classification of attacks with improved accuracy.

Several machine learning methods were analysed for spotting and thwarting DDoS attacks in an SDN network, including J48, random forest (RF), support vector machine (SVM), and K-nearest neighbours (K-NN) (Rahman et al., 2019). The experimental findings demonstrate that J48 outperforms other machine learning assessment strategies. Similarly, a model SDCC was presented that employs bandwidth detection and data flow detection by applying the confidence-based filtering (CBF) method to detect DDoS in an SDN network (He et al., 2018).

Saied et al. (2016) insisted that, for detecting DDoS attacks, it is highly necessary to differentiate packets based on the protocols used. Though it is difficult and complex to differentiate the packets based on the protocols used, the author employed a novel ANN algorithm to train the model for distinguishing the packets by analysing the characteristics of the protocols such as TCP and UDP (Saied et al., 2016). A DDoS attack detection system that works based on a valid source and destination IP address database was investigated (Wang et al., 2016). It efficiently examines the DDoS attacks by analysing the anomalous characteristics of the source IP address and destination IP address when they occur. However, the approach needs to be modified, and the threshold must be evaluated for better performance.

With the attention gained by the flash crowds, a model was suggested to detect DDoS attacks (Gera and Battula, 2018). This suggested model was based on examining an unusual traffic pattern. It was implemented to recognise DDoS attacks and extricate them from flash events. Although each traffic packet in this seems to be real, each traffic packet was inspected and monitored separately. A self-organising maps (SOM) technique was proposed to detect DDoS attacks by examining the flow statistics (Braga et al., 2010). It was reported that the method has a high detection rate and low time consumption, and that the extraction of time intervals is considered essential. However, the drawback of this strategy is that the attack action is not detected reliably and promptly due to some hysteresis in the detection.

The detection of DDoS attacks utilising TCP traffic was proposed as a real-time TCP-based DDoS detection solution by extracting useful aspects of TCP traffic. To identify DDoS attacks using the source IP address used by attackers, the model distinguishes between two attack mechanisms known as fixed source IP attacks (FSIA) and random source IP attacks (RSIA). The model applies decision tree classifiers to distinguish malicious traffic from benign traffic (Jiao et al., 2017).

By developing a DDoS detection system based on the C.4.5 algorithm, Zekri et al. (2017) suggested a methodology to mitigate the DDoS attack. In a DDoS attack, the attacker often employs zombies – innocent computers that have been compromised – to

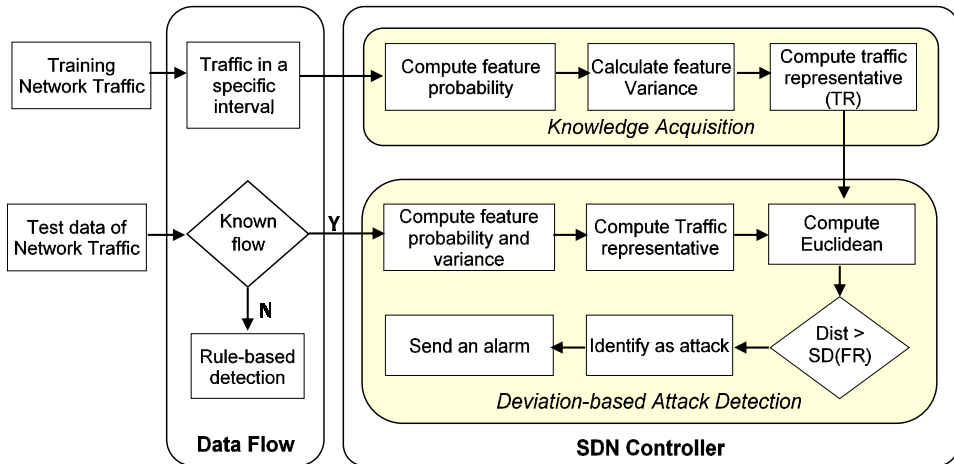
use known or unknown viruses and vulnerabilities to send a massive volume of packets from these already-captured zombies to a server. The suggested approach creates a decision tree that performs automatic, effective signature detection for DDoS flooding attacks when paired with signature detection techniques.

It was determined that information entropy (Liu et al., 2022) and the application of data mining, machine learning algorithms (Barbhuiya et al., 2021; Kachavimath and Narayan, 2021), and artificial intelligence (Fouladi et al., 2022) are widely used for DDoS attack detection in the cloud environment, with the SOM algorithm having the highest priority (Zhao et al., 2021). However, the SOM algorithms must determine the number of neurons in advance due to the high false-positive rate.

3 Proposed DDDM

The proposed DDDM offers a solution for identifying DDoS attacks in cloud computing. The idea of the work is inspired by the SBS-SDN-based solution suggested by Sindia and Dhas (2006). The proposed mechanism works in two stages, in which the training phase intends to acquire knowledge of the data traffic and the testing phase intends to differentiate the attack traffic from the normal traffic. The framework is implemented in software-defined networking, in which knowledge is acquired and traffic is analysed by the controller. Due to the varied capabilities and advantages in controlling and managing the networks, adjusting configuration settings, provisioning resources, and increasing network capacity without even using additional hardware, SDN is more widely used by researchers (Bawany and Shamsi, 2016).

Figure 1 Overall framework of the proposed deviation-based attack detection model (see online version for colours)



The first stage is knowledge acquisition, which helps to compute the representative network traffic by calculating the mean of the probability-based variance of features. This collected information is the knowledge base through which the model classifies the abnormal traffic patterns in real traffic. The traffic representation contains the probability-based variance of various features, including source and destination

addresses, as well as the packet rate for a particular period of time. The second stage is the deviation computation phase, in which the deviation between the traffic representation computed from current network traffic and that of the knowledge base is calculated. The idea behind using deviations is that they compute the packet rate variance for the source and destination addresses. However, this phase is carried out only when the flow is known, indicating that the source and destination addresses are available in the knowledge base. However, for the unknown flow, the proposed model applies rule-based attack detection. The overall framework of the proposed model is shown in Figure 1.

In some sort of flooding attack, the illegitimate bot system sends the request packet to one or more destinations (Saravanan et al., 2019; Somasundaram and Meenakshi, 2021). Similarly, in many cases, the single master bot system implants malware into the target system and makes it a zombie. These target systems will attack the victim when the particular trigger is sent by the master bot. Thus, the flooding attack on the victim will be made by sending request packets from different sources (Saravanan et al., 2016). Consecutively, the number of packets from the same source and the number of packets for the same destination address are evaluated for a time interval through deviation computation. The deviation will be high when the number of packets from or to the same host is very high, which indicates a DDoS attack. Moreover, the packet rate, which evaluates the total number of packets transmitted in a given period, is also used as an attribute for traffic representation. This deviation-based detection helps to determine whether the traffic is normal or abnormal in a very short span of time, which in turn detects the attack at an early stage.

The variance of a discrete probability distribution is used in computing the traffic representation for network features including the source and destination addresses (Cthaeh, 2020). Variance measures the variability or dispersion and calculates the expectation of the deviation of the intended variable from the average of its population. Along with the source and destination addresses, the packet arrival rate is also used. Here, when the number of packets from the same source to a different destination is high or the number of packets to a particular destination from various sources is high, it results in a higher variance. On the other hand, if the incoming packets from different sources or destinations are distributed evenly, then the variance will be low. The detailed procedure for the training and test phases is presented in the following sections.

3.1 Knowledge acquisition phase

In this training phase, the controller intends to create traffic representatives for normal traffic. It gathers the knowledge by applying variance with a discrete probability distribution to the source address, destination address, and packet arrival rate for a particular period of time. This knowledge base, created during the training phase, helps the controller distinguish between real and attack traffic during the detection phase.

With n being the total number of packets that arrived at a particular period of time and d being the distinct source addresses, the model computes the variance of a discrete probability distribution as given in equation (1).

$$Var(sa_i) = (m_n - c(sa_i))^2 \times P(sa_i) \quad (1)$$

$Var(sa_i)$ is the variance of the specific source address sa_i , $c(sa_i)$ is the count of packets with the specific source address sa_i , $P(sa_i)$ is the probability of packets with the source address sa_i and m_n is the mean of n packets arriving at a specific time interval.

The mean value m_n is computed by computing the average number of distinct packets that arrived in the specified period of time. As a result, the mean value is calculated by dividing the number of packets arrived by distinct source addresses as n / d .

The discrete probability distribution of packets arriving with the specific source address $P(sa_i)$ is calculated by dividing the count of packets $c(sa_i)$ with the source address sa_i by the total number of packets arrived n .

$$P(sa_i) = \frac{c(sa_i)}{n} \quad (2)$$

For illustration, consider the number of packets that arrived in a specific time interval n is 15 with $d = 4$ being the distinct sources addresses such as $sa_1 = 2$, $sa_2 = 4$, $sa_3 = 5$,

$sa_4 = 4$. Then $m_n = \frac{15}{4} = 3.75$. Thus, the probability of packets with distinct source

addresses are $P(sa_1) = \frac{2}{15} = 0.133$; $P(sa_2) = \frac{4}{15} = 0.267$; $P(sa_3) = \frac{5}{15} = 0.333$; $P(sa_4) =$

$\frac{4}{15} = 0.267$.

Finally, the variance of packets with source address sa_1 can be computed using the formula as $Var(sa_1) = (2.75 - 2)^2 \times 0.133 = 0.4083$. Similarly, the variance of other source addresses sa_2 , sa_3 and sa_4 will be 0.0167, 0.5208 and 0.0167. This implies that the number of packets with source addresses sa_1 and sa_3 deviates more than other source addresses. The same procedure will be carried out for the various destination addresses that arrived at a specific period and the packet rate for distinct destination addresses. This information is collected for different time periods t to compute traffic representations in the knowledge base. Finally, for each distinct source and destination address, the mean of the variance ($mVar$) and standard deviation (SD_Var) at various time intervals are assessed. Aside from the variance computations, the standard deviation is also calculated, which is the square of variance. This standard deviation is calculated to help determine the threshold for distinguishing between normal and abnormal traffic. The formula to compute the mean from the variance ($mVar$) and standard deviation (SD_Var) of distinct sources address is given in equation (3) and equation (4).

$$mVar(sa_i) = \frac{\sum_{t=1}^k Var(sa_i)_t}{k} \quad (3)$$

$$SD_Var(sa_i) = \sqrt{\frac{\sum_{t=1}^k (Var(sa_i)_t - mVar(sa_i))^2}{k}} \quad (4)$$

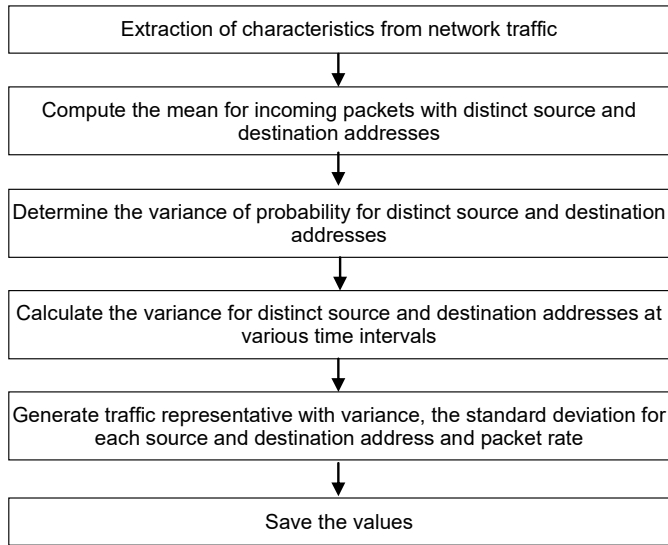
Here, t represents the time period that varies from 1 to k , where k indicates the number of the different time intervals in which the packets arrive with the source address sa_i . This is also computed for different destination addresses. Thus, the traffic representative for the source address and the destination address is stored in the knowledge base along with the average packet rate (Pkr). Equation (5) gives the format for the traffic representative for

the source address where $\bigcup_{t=1}^k Var(sa_i)$ represents the variance of sa_i at different periods t .

$$Traffic_{rep}(sa_i) = \left\langle \bigcup_{t=1}^k Var(sa_i), mVar(sa_i), SD_Var(sa_i), Pkr \right\rangle \quad (5)$$

The workflow of the knowledge acquisition phase carried out by the controller is shown in Figure 2.

Figure 2 Workflow structure of the knowledge acquisition phase



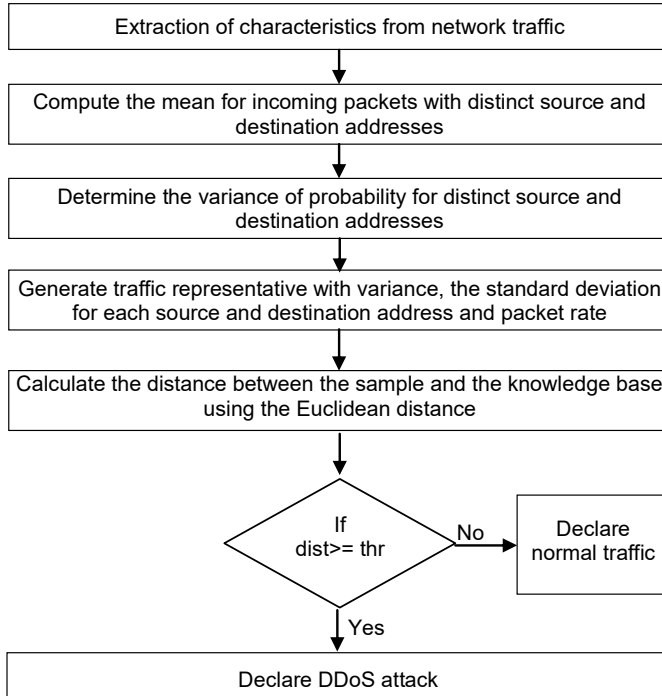
3.2 Deviation-based attack detection

In this testing phase, the incoming test samples are processed using the same procedure discussed in Section 3.1 for traffic representation. This process first checks to see if the flow is known by comparing the source and destination addresses of incoming packets to those in the knowledge base. The detection procedure continues only when the flow is known. The probability-based variance is computed for each distinct source and destination address in the input samples as given in equation (1). Then the mean of the computed variances is also evaluated for the incoming traffic as in equation (3). The double of the standard deviation of the traffic representation of the specific source address stored in the knowledge base, is chosen as the threshold for determining the real and attack traffic concerning the source and destination addresses.

However, the comparison between the mean-variance computed for the incoming samples from different sources and destinations and that of the knowledge base is analysed. The Euclidean distance is used to compare the mean variance of traffic representatives with the knowledge base. If the computed distance of the source or destination address is greater than that of the threshold value for the specific source and destination, then the network traffic with the given source or destination address is

considered abnormal. The flow diagram for the operational phase is shown in Figure 3. The pseudocode for deviation-based attack detection with source address is presented in Algorithm 1. However, the model can be extended to use various other features, such as source port, network port, packet size, etc.

Figure 3 Workflow structure of the operational phase



Algorithm 1 Known attack detection using deviation analysis

Input: Training samples of network traffic at various time intervals n_1, n_2, \dots, n_t , test sample n_{test}

Output: DDoS Attack Detection

// Knowledge Acquisition Phase

Procedure DeviationAnalysis()

Begin

Extract network traffic features from the training samples;

For j varies from n_1 to n_t **do**

 Compute the mean m_n for the number of packets based on the source address

For each unique source address sa_i **do**

 Compute discrete probability distribution $P(sa_i)$

 Compute variance $Var(sa_i)$

 Extract packet rate pkt_i ;

End For

End For

```

For  $j$  varies from  $n_1$  to  $n_t$  do
  For each  $sa_i$  in training samples do
    Compute the mean of the variance  $mVar$  and standard deviation  $SD\_Var(sa_i)$ 
    Create feature representation  $\left\langle \bigcup_{t=1}^k Var(sa_i), mVar(sa_i), SD\_Var(sa_i), Pkr \right\rangle$ 
  End For
End For
Extract network traffic features from the test sample  $n_{test}$ 
If the flow is known then
  Compute the mean for the number of packets based on the source address
  For each unique source address  $sa_i$  do
    Compute discrete probability distribution  $P(sa_i)$ 
    Compute variance  $Var(sa_i)$ 
    Extract packet rate  $pkt_i$ ;
  End For
End For
For each  $sa_i$  in test samples do
  Compute the Euclidean distance  $ED = Var(sa_i) - mVar(sa_i)$ 
  If  $(ED \geq 2 * SD\_Var(sa_i))$  then
    Declare an attack and send the alarm
  End If
End For
End If
End Procedure

```

3.3 Rule-based detection

The rule-based detection of DDoS attacks is carried out only when the flow is unknown. If the traffic samples at the specified interval are unknown, then the various simple rules are verified to identify the attacks. The modified version of the rules suggested by Somasundaram and Meenakshi (2021) is used in this work. The number of incoming packets, the synchronous flag, and the source IP address are verified in detecting the attacks. The number of incoming packets from the same sources is computed, and if it exceeds the given limit, then the scenario indicates attack traffic. So, the attack packets are discarded, and the IP address is blocked. Here, the limit for incoming requests per second is set at 25 (Somani et al., 2015). Similarly, the number of incoming packets from different sources is computed, and if it exceeds the given limit, the scenario indicates attack traffic, and the packets are discarded. Here, the limit for incoming packets from different source addresses per second is set at 40 (Somasundaram and Meenakshi, 2021). Consecutively, the packets are confirmed to have the synchronous flag and source IP address set for the incoming traffic. If either the synchronous flag or the source IP address is not set, then the packets are considered to be traffic with a spoofing attack and are rejected. The source code for the rule-based attack detection is given in Algorithm 2.

Algorithm 2 Rule-based DDoS attack detection for unknown flow

Input: Incoming requests
Output: Allow/disallow source packets
Procedure Rule_based_Detection()
Begin
Set limits for source_{same} and source_{diff}
Verify the incoming packet limit with the same and different sources
If (n(pkt_source_{same}) > source_{same}) **then**
 Consider as an attack packet and discard
Else if (n(pkt_source_{diff}) > source_{diff}) **then**
 Consider as an attack packet and discard
Else
 Check the packets for Synchronous Flag (SF) and source IP address
 If (SF ≠ 1) **then**
 Consider as an attack packet and discard
 Else if (SIP ≠ 1) **then**
 Consider as an attack packet and discard
 Else
 Allow the legitimate packets
 End If
End If
End Algorithm

4 Performance evaluation

The proposed model provides an SDN-based solution for detecting DDoS attacks in a cloud environment. This section aims at analysing the performance of the proposed model. The experiment is run on a system with an Intel Pentium 4 processor running at 2.40 GHz and 8 GB of RAM. For effective analysis, a private cloud network has been set up as a testbed using Oracle VM Virtual Box as a virtual environment. The proposed model is implemented in Python on a private cloud server.

This research aims to suggest an improved solution for detecting DDoS attacks in the cloud environment while providing good performance results. Standard performance metrics are used to assess the performance of the proposed approach from various perspectives. Thus, various metrics such as detection rate, false-positive rate, false-negative rate, and time consumption are used to analyse the results of the proposed model and compare them with the results of the existing models (Aborujilah and Musa, 2017).

4.1 Experimental setup

The proposed work is implemented in an environment containing group infrastructure architectures. Various hardware and software tools are utilised for performing the analysis and simulating the DDoS attacks. These include hardware such as Cisco 4000 ISR Series Routers and Cisco Nexus 5000 Series Switch for routing and switching, Big IP LTM-4200 for high-performance device traffic load control, Imperva Web Application Firewall Portal with Manager Console, Cisco Firepower FPR-2110, and HP DL-360. Various DDoS attack simulation software tools are also employed, such as Low Orbit Ion Canon (LOIC), High Orbit Ion Canon (HOIC), Packet Storm (HTTP Intolerable Load King), Are You Dead Yet (R.U.D.Y) for slow rate attacks, and TOR's Hammer for layer 7 DDoS attacks.

Group and alertness layer attacks are carried out on the networks using ICMP flooding and a thousand echo requests with buffer sizes ranging from 3,700 to 3,805 bytes. Legitimate users are denied access to the web software portal due to the use of various simulated DDoS attacks such as LOIC, R.U.D.Y, and slowloris. Real-world user monitoring records are used as standards while simulating DDoS attacks, and log parameters are collected to aid in the generation of DDoS attack graphs. These parameters are chosen as they specify the performance issues encountered by actual web users at any given time during an attack.

4.2 Result analysis

In general, for the detection model to be effective, the amount of detection time must be minimised. However, in many cases, though the models have a high detection rate, the time taken to perform the detection will be high. This is because the models have high computational complexity and require more processing time. Thus, reducing time consumption is a challenging task. One way to achieve reduced time is to use fewer yet significant features. Moreover, the attack detection system must also ensure a low FPR and false negative rate (FNR) (Osaniye, 2015; Cheng et al., 2018). Thus, with these metrics, the proposed model is analysed, and the results are compared with those of other existing models.

For performing the analysis, various features such as the network packet's destination IP, source IP, destination port, source port, and packet size fields are also considered. Table 1 specifies the sample attributes of the network traffic collected at various time intervals.

Table 1 Feature representative traffic data

<i>Application</i>	<i>SourceIP</i>	<i>Source port</i>	<i>Destination IP</i>	<i>Destination port</i>	<i>Action</i>	<i>Packet size</i>	<i>Time</i>
ftp	192.168.1.8	10	192.168.1.72	12	req	1,000	10 ms
ftp	192.168.1.8	10	192.168.1.72	12	rep	1,000	12 ms
http	192.168.1.12	22	192.168.1.83	35	req	2,376	22 ms
http	192.168.1.12	22	192.168.1.83	35	rep	2,376	19 ms
ftp	192.168.1.15	11	192.168.1.76	18	req	3,425	23 ms
ftp	192.168.1.15	11	192.168.1.76	18	rep	3,425	21 ms

Table 1 Feature representative traffic data (continued)

<i>Application</i>	<i>SourceIP</i>	<i>Source port</i>	<i>Destination IP</i>	<i>Destination port</i>	<i>Action</i>	<i>Packet size</i>	<i>Time</i>
smtp	192.168.1.11	13	192.168.1.74	17	req	1,294	34 ms
smtp	192.168.1.11	13	192.168.1.74	17	rep	1,294	38 ms
telnet	192.168.1.17	12	192.168.1.71	19	req	1,427	25 ms
telnet	192.168.1.17	12	192.168.1.71	19	rep	1,427	28 ms
http	192.168.1.56	14	192.168.1.65	15	req	1,526	32 ms
nntp	192.168.1.13	14	192.168.1.77	23	req	3,765	45 ms
nntp	192.168.1.13	14	192.168.1.77	23	rep	3,765	43 ms
ftp	192.168.1.10	16	192.168.1.75	20	req	2,715	16 ms
ftp	192.168.1.10	16	192.168.1.75	20	rep	2,715	20 ms

4.2.1 Detection rate

The performance of the proposed mechanism is evaluated and compared with various existing approaches. Some of the existing works used for the comparison are cluster analysis (Bhaya and Manaa, 2014), wavelet-based (Srihari and Anitha, 2014), entropy-based (Sindia and Dhas, 2017), and statistical analysis-based (Saravanan and Bama, 2020). The effective model will have the highest detection rate. An experimental analysis is made by varying the number of attack packets from 200 to 1,000, incremented by 200, with the real and attack requests in a ratio of 1:1. The results obtained for the above experiment are presented in Table 2.

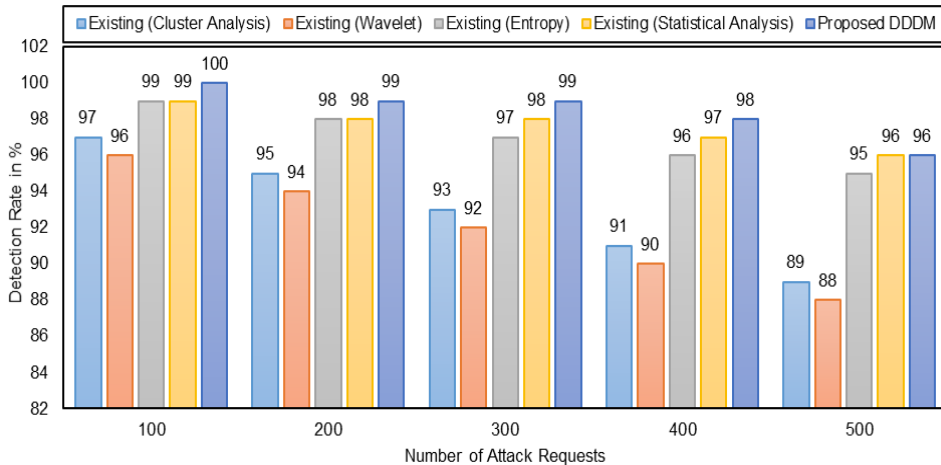
Table 2 Analysis of detection rate

<i>No. of requests</i>	<i>Detection rate in %</i>				
	<i>Existing mechanism</i>				<i>Proposed (DDDM) mechanism</i>
	<i>Cluster analysis</i>	<i>Wavelet</i>	<i>Entropy-based</i>	<i>Statistical analysis</i>	
200	97	96	99	99	100
400	95	94	98	98	99
600	93	92	97	98	99
800	91	90	96	97	98
1,000	89	88	95	96	96
Average	92	92	97	99	98

From the obtained results, it is clear that the average detection rate of the proposed model is above 98%, which is higher than the existing models under comparison. The cluster and wavelet models have an average detection rate of 92%, and the entropy-based and statistical analysis models have a detection percentage of around 97%. The wavelet-based detection model employs wavelet transformation to extract significant features and applies a semi-supervised learning technique for detecting DDoS attacks. On the other hand, the cluster-based detection analysis uses centroid rules for detecting DDoS attacks. However, both models have the lowest detection rate when compared with other entropy, count and deviation-based models. Moreover, the entropy-based detection mechanism

applies entropy variance for detecting DDoS attacks, and the statistical analysis-based model uses protocol entropy and count-based filtering for detecting attacks. Despite outperforming cluster and wavelet-based analysis, the detection rate of entropy and count-based models is lower than that of the proposed deviation-based analysis. The detection rates for various methods, including existing cluster analysis, wavelet mechanism, entropy-based, statistical analysis-based, and the proposed DDDM model, are compared and are depicted as a graph in Figure 4.

Figure 4 Comparison of detection rate with varied request (see online version for colours)



4.2.2 Time consumption

In general, the DDoS detection models use complex processing and procedures to improve the detection percentage, which in turn increases the execution time. Thus, time consumption is another significant metric to be analysed while evaluating the detection models. The attack detection procedure should be attentive to network traffic. The attack must be detected instantly to ensure the security of the underlying system by preserving its usual functionality. Thus, to analyse the execution time of the proposed model, an experiment is made by varying the number of attack packets from 100 to 500, incremented by 100, and the execution time is noted for the proposed and existing mechanisms such as cluster analysis (Bhaya and Manaa, 2014), wavelets (Srihari and Anitha, 2014), entropy-based (Sindia and Dhas, 2017), and statistical analysis based (Saravanan and Bama, 2020) detection mechanisms. The time taken to detect the attacks for cluster analysis, wavelet-based, entropy-based, statistical analysis models, and the proposed DDDM model is presented in Table 3.

The results obtained for the analysis of execution time for various proposed and existing models presented in Table 3 are depicted as a graph in Figure 5. From the analysis, it is clear that the execution time increases with the increase in the number of incoming requests.

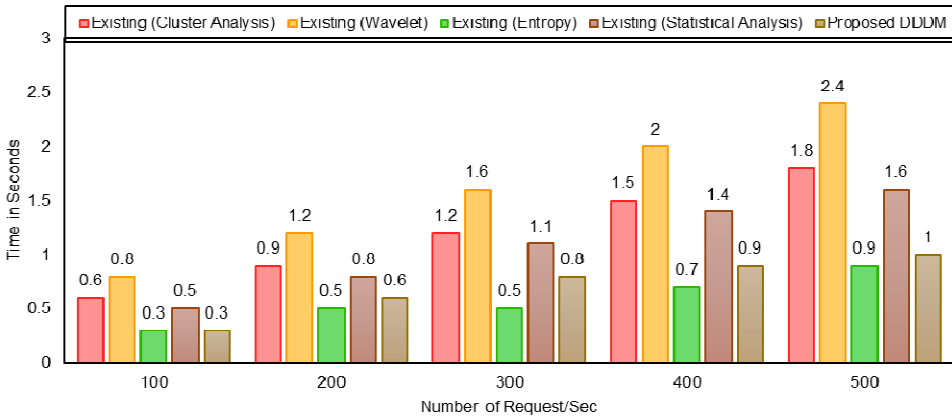
Moreover, the proposed DDDM model and the entropy-based models have minimum execution times when compared with that of the cluster, count and wavelet-based models. This is due to the fact that using less yet more significant features in performing the

traffic representative reduces the number of comparisons. The average execution time for the existing models such as cluster-based, wavelet-based, entropy-based and statistical analysis-based models are 1.2, 1.6, 0.58, and 1.08 seconds respectively. On the other hand, the execution time of the proposed model is 0.72 seconds. The entropy-based model has the shortest execution time since it takes only the destination address for performing detection analysis. However, the proposed deviation-based detection model makes use of source and destination addresses as well as the port address for calculating the traffic representatives. Thus, the execution time of the proposed model is higher than that of the entropy-based model. However, it is far less than the other cluster, statistical analysis and wavelet-based models.

Table 3 Analysis of time consumption

No. of requests	Execution time in seconds				
	Existing mechanism				Proposed (DDDM) mechanism
	Cluster analysis	Wavelet	Entropy-based	Statistical analysis	
100	0.6	0.8	0.3	0.5	0.3
200	0.9	1.2	0.5	0.8	0.6
300	1.2	1.6	0.5	1.1	0.8
400	1.5	2.0	0.7	1.4	0.9
500	1.8	2.4	0.8	1.6	1
Average	1.2	1.6	0.58	0.5	0.72

Figure 5 Comparison of attack detection time (see online version for colours)



4.2.3 FPR and FNR

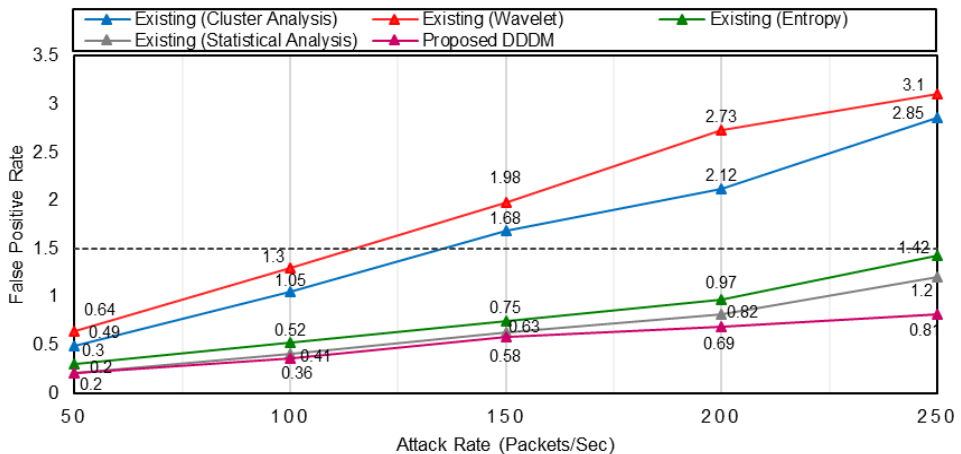
The difference between a false positive and a false negative is that with a false positive, the model identifies the regular traffic as an attack, and with a false negative, attack traffic is identified as regular. The FPR absorbs excessive computing resources and time, which disturbs the usual process of the system, and thus it intends to evaluate the efficiency of the detection model. The FNR compromises the trustworthiness of the

detection model by allowing the attack packets to be regular, and thus it intends to evaluate the reliability of the underlying detection model. For performing the analysis on FPR, an experiment is made by varying the number of attack packets from 50 to 250, incremented by 50. Table 4 displays the FPR assessed for the proposed and existing models such as cluster analysis (Bhaya and Manaa, 2014), wavelets (Srihari and Anitha, 2014), entropy-based (Sindia and Dhas, 2017), and statistical analysis-based (Saravanan and Bama, 2020) detection mechanisms. The obtained values are presented as a line graph in Figure 6. The results indicate that the proposed DDDM model, count-based statistics, and entropy-based detection model have fewer false positives than cluster analysis and wavelet-based mechanisms. Moreover, the average FPR for the entropy-based and statistical models is 0.8 and 0.65, respectively, whereas that of the proposed model is 0.5. This indicates that the proposed model has a more efficient performance than its competitors.

Table 4 Analysis of FPR

Attack rate (packets/sec.)	Existing mechanism				Proposed (DDDM) mechanism
	Cluster analysis	Wavelet	Entropy-based	Statistical analysis	
50	0.49	0.64	0.3	0.2	0.2
100	1.05	1.30	0.52	0.41	0.36
150	1.68	1.98	0.75	0.63	0.58
200	2.12	2.73	0.97	0.82	0.69
250	2.85	3.10	1.42	1.2	0.81

Figure 6 Comparison of FPR (see online version for colours)



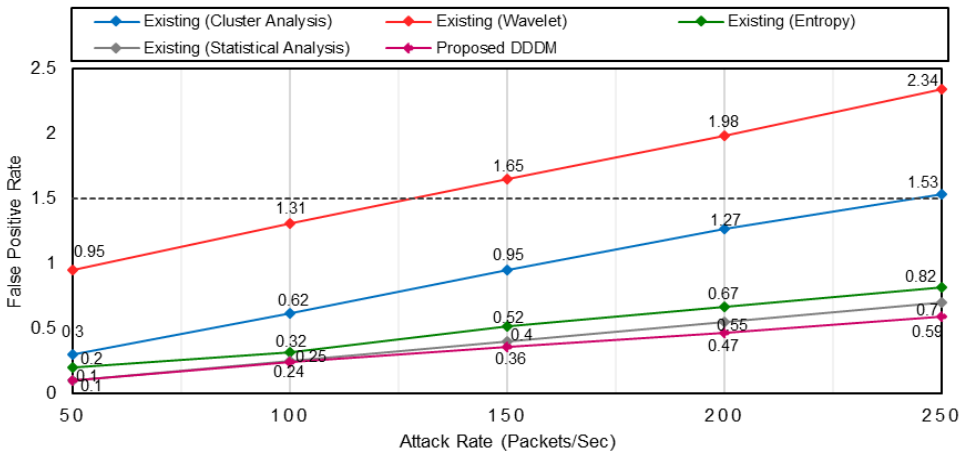
To analyse the reliability of the system, an experiment is made by varying the number of attack packets from 50 to 250, incremented by 50. The comparison of results is made for the proposed DDDM model with those of the existing detection models such as cluster analysis (Bhaya and Manaa, 2014), wavelets (Srihari and Anitha, 2014), entropy-based (Sindia and Dhas, 2017), and statistical analysis-based (Saravanan and Bama, 2020) detection mechanisms. The obtained values for the proposed and existing models are

presented in Table 5. The values presented in the table are also depicted as a graph in Figure 7. From the analysis, it is clear that the proposed model, an entropy-based model, and statistical analysis offer improved reliability with a lower FNR than the cluster analysis and wavelet-based existing models. Moreover, the average FNR for the proposed model is 0.35, whereas that of the entropy-based and statistical models are 0.5 and 0.4, respectively. This shows that the proposed model offers better results than other models under comparison.

Table 5 Analysis of FNR

Attack rate (packets/sec.)	Existing mechanism				Proposed (DDDM) mechanism
	Cluster analysis	Wavelet	Entropy-based	Statistical analysis	
50	0.30	0.95	0.2	0.1	0.1
100	0.62	1.31	0.32	0.25	0.24
150	0.95	1.65	0.52	0.4	0.36
200	1.27	1.98	0.67	0.55	0.47
250	1.53	2.34	0.82	0.7	0.59

Figure 7 Comparison of FNR (see online version for colours)



According to the extensive result analysis, the proposed deviation-based DDOS detection model outperforms competitors in terms of detection rate, a lower FPR, and a lower FNR. However, it is discovered that the proposed model takes slightly longer execution time than the entropy-based model. This is because the model utilises various features that arise from a single source address to increase the attack detection rate. Thus, the key advantages of the proposed work are reduced computational complexity and improved work efficiency.

5 Conclusions

As an emerging technology, several security risks exist in cloud computing that need to be resolved to ensure security and reliability. DDoS attacks are one type of security risk that aims to bring a server down by generating a large amount of traffic. This paper focuses on proposing a solution for detecting DDoS attacks from legitimate network traffic using the SDN framework. To detect attack traffic, a deviation-based DDOS detection model is proposed in which it initially collects the network traffic to form a knowledge base. The variance of the discrete probability distribution of the network features is computed and stored as the traffic representative. In the deviation-based detection phase, for known flows, the controller computes the variance, which is compared with the knowledge base by using Euclidean distance. If the difference is greater than the standard deviation that is set as a threshold, then the test sample of network traffic is considered an attack. Moreover, for the unknown flow, the model uses rule-based detection for identifying the attacks. With the experimental analysis, it was found that the proposed model offers an average detection rate of 98% with an average execution time of 0.72 seconds. The results of the FPR and FNR are also lower with high attack traffic than those of other competitors. Future work focuses on improving the attack detection rate. Further, other mathematical and statistical concepts can be used to achieve better results.

References

- Aborujilah, A. and Musa, S. (2017) 'Cloud-based DDoS HTTP attack detection using covariance matrix approach', *Journal of Computer Networks and Communications*, Vol. 2017, No. 1.
- Almseidin, M., Al-Sawwa, J. and Alkasassbeh, M. (2021) 'Anomaly-based intrusion detection system using fuzzy logic', in *2021 International Conference on Information Technology (ICIT)*, IEEE, July, pp.290–295.
- Alzahrani, S. and Hong, L. (2017) 'A survey of cloud computing detection techniques against DDoS attacks', *Journal of Information Security*, Vol. 9, No. 1, p.45.
- Barbhuiya, S., Kilpatrick, P. and S. Nikolopoulos, D. (2021) 'Linear regression based DDoS attack detection', in *2021 13th International Conference on Machine Learning and Computing*, February, pp.568–574.
- Bawany, N.Z. and Shamsi, J.A. (2016) 'Application layer DDoS attack defense framework for smart city using SDN', in *The Third International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM2016)*, May, p.1.
- Bhaya, W. and Manaa, M.E. (2014) 'A proactive DDoS attack detection approach using data mining cluster analysis', *Journal of Next Generation Information Technology*, Vol. 5, No. 4, pp.21–36.
- Braga, R., Mota, E. and Passito, A. (2010) 'Lightweight DDoS flooding attack detection using NOX/OpenFlow', in *IEEE Local Computer Network Conference*, IEEE, pp.408–415.
- Cheng, J., Zhang, C., Tang, X., Sheng, V.S., Dong, Z. and Li, J. (2018) 'Adaptive DDoS attack detection method based on multiple-kernel learning', *Security and Communication Networks*, Vol. 2018, No. 1, pp.1–19.
- Cthaeh (2020) 'Alternative variance formulas and their derivation', *Probabilistic World*, November [online] <https://www.probablisticworld.com/alternative-variance-formulas-derivation> (accessed January 2022).

- Deshmukh, R.V. and Devadkar, K.K. (2015) 'Understanding DDoS attack & its effect in cloud environment', *Procedia Computer Science*, Vol. 49, pp.202–210.
- Devaraju, S. and Ramakrishnan S. (2011) 'Performance analysis of intrusion detection system using various neural network classifiers', *IEEE International Conference on Recent Trends in Information Technology (ICRTIT 2011)*, pp.3–5.
- Devaraju, S. and Ramakrishnan, S. (2015) 'Detection of attacks for IDS using association rule mining algorithm', *IETE Journal of Research*, Vol. 61, No. 6, pp.624–633.
- Devaraju, S. and Ramakrishnan, S. (2019) 'Association Rule-mining-based intrusion detection system with entropy-based feature selection: intrusion detection system (Chapter 1)', *Handbook of Research on Intelligent Data Processing and Information Security Systems*, November, pp.1–24, p.24, IGI Global, ISBN: 9781799812906, DOI: 10.4018/978-1-7998-1290-6.
- Devaraju, S. and Ramakrishnan, S. (2020) 'Fuzzy rule-based layered classifier and entropy-based feature selection for intrusion detection system (Chapter 15)', *Handbook of Research on Cyber Crime and Information Privacy*, August, 2 Volumes, p.753, IGI Global, ISBN: 9781799857280, DOI: 10.4018/978-1-7998-5728-0.
- Dimolianis, M., Pavlidis, A. and Maglaris, V. (2021) 'Signature-based traffic classification and mitigation for DDoS attacks using programmable network data planes', *IEEE Access*, Vol. 9, No. 8, pp.113061–113076.
- Fouladi, R.F., Ermiş, O. and Anarim, E. (2022) 'A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN', *Computer Networks*, Vol. 214, No. 9, p.109140.
- Gera, J. and Battula, B.P. (2018) 'Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds', *EURASIP Journal on Information Security*, Vol. 2018, No. 1, pp.1–12.
- He, H., Hu, Y., Zheng, L. and Xue, Z. (2018) 'Efficient DDoS attack detection and prevention scheme based on SDN in cloud environment', *Journal on Communications*, Vol. 39, No. 4, p.139.
- Jiao, J., Ye, B., Zhao, Y., Stones, R.J., Wang, G., Liu, X. and Xie, G. (2017) 'Detecting TCP-based DDoS attacks in Baidu cloud computing data centers', in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, IEEE, pp.256–258.
- Kachavimath, A.V. and Narayan, D.G. (2021) 'A deep learning-based framework for distributed denial-of-service attacks detection in cloud environment', in *Advances in Computing and Network Communications*, pp.605–618, Springer, Singapore.
- Khorshed, M.T., Ali, A.S. and Wasimi, S.A. (2012) 'A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing', *Future Generation Computer Systems*, Vol. 28, No. 6, pp.833–851.
- Liu, Y., Zhi, T., Shen, M., Wang, L., Li, Y. and Wan, M. (2022) 'Software-defined DDoS detection with information entropy analysis and optimized deep learning', *Future Generation Computer Systems*, Vol. 129, pp.99–114.
- Luo, H., Lin, Y., Zhang, H. and Zukerman, M. (2013) 'Preventing DDoS attacks by identifier/locator separation', *IEEE Network*, Vol. 27, No. 6, pp.60–65.
- Mirkovic, J. and Reiher, P. (2004) 'A taxonomy of DDoS attack and DDoS defense mechanisms', *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2, pp.39–53.
- Mondal, H.S., Hasan, M.T., Hossain, M.B., Rahaman, M.E. and Hasan, R. (2017) 'Enhancing secure cloud computing environment by Detecting DDoS attack using fuzzy logic', in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, IEEE, pp.1–4.

- Mousavi, S.M. and St-Hilaire, M. (2015) 'Early detection of DDoS attacks against SDN controllers', in *2015 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, February, pp.77–81.
- Osaniye, O.A. (2015) 'Short paper: IP spoofing detection for preventing DDoS attack in cloud computing', in *2015 18th International Conference on Intelligence in Next Generation Networks*, IEEE, pp.139–141.
- Rahman, O., Quraishi, M.A.G. and Lung, C.H. (2019) 'DDoS attacks detection and mitigation in SDN using machine learning', in *2019 IEEE World Congress on Services (SERVICES)*, IEEE, July, Vol. 2642, pp.184–189.
- Ramakrishnan, S. and Devaraju, S. (2016) 'Attack's feature selection-based network intrusion detection system using fuzzy control language', *Springer-International Journal of Fuzzy Systems*, DOI: 10.1007/s40815-016-0160-6.
- Rawashdeh, A., Alkasassbeh, M. and Al-Hawawreh, M. (2018) 'An anomaly-based approach for DDoS attack detection in cloud environment', *International Journal of Computer Applications in Technology*, Vol. 57, No. 4, pp.312–324.
- Saied, A., Overill, R.E. and Radzik, T. (2016) 'Detection of known and unknown DDoS attacks using artificial neural networks', *Neurocomputing*, Vol. 172, No. 1, pp.385–393.
- Saravanan, A. and Bama, S.S. (2020) 'Multi-model anti-Ddos framework for detection and mitigation of high rate DDoS attacks in the cloud environment', *International Journal of Scientific & Technology Research*, Vol. 9, No. 3, pp.4503–4511.
- Saravanan, A., Irfan Ahmed, M.S. and Sathya Bama, S. (2016) 'Mitigation framework against DDoS attacks in cloud server', *ICT Innovations 2016 Web Proceedings*, pp.33–42.
- Saravanan, A., Sathya Bama, S., Kadry, S. and Ramasamy, L.K. (2019) 'A new framework to alleviate DDoS vulnerabilities in cloud computing', *International Journal of Electrical & Computer Engineering*, Vol. 9, No. 5, pp.2088–8708.
- Sindia, T.V. and Dhas, J.P.M. (2006) 'SBS-SDN based solution for preventing DDoS attack in cloud computing environment', *ARPN Journal of Engineering and Applied Sciences*, Vol. 12, No. 11, pp.3593–3599.
- Sindia, T.V. and Dhas, J.P.M. (2017) 'A bifold software defined networking based defence mechanism for DDOS attacks in the cloud environment', *International Journal of Applied Engineering Research*, Vol. 12, No. 20, pp.9467–9474.
- Somani, G., Gaur, M.S., Sanghi, D., Conti, M. and Buyya, R. (2017) 'DDoS attacks in cloud computing: issues, taxonomy, and future directions', *Computer Communications*, Vol. 107, No. 7, pp.30–48.
- Somani, G., Johri, A., Taneja, M., Pyne, U., Gaur, M.S. and Sanghi, D. (2015) 'DARAC: DDoS mitigation using DDoS aware resource allocation in the cloud', in *International Conference on Information Systems Security*, Springer, Cham, pp.263–282.
- Somasundaram, A. and Meenakshi, V.S. (2021) 'A novel three layer filtering (3L-F) framework for prevention of DDoS attack in cloud environment', *International Journal of Computer Networks and Applications*, Vol. 8, No. 4, pp.334–345.
- Srihari, V. and Anitha, R. (2014) 'DDoS detection system using wavelet features and semi-supervised learning', in *International Symposium on Security in Computing and Communication*, Springer, Berlin, Heidelberg, September, pp.291–303.
- Wang, B., Zheng, Y., Lou, W. and Hou, Y.T. (2015) 'DDoS attack protection in the era of cloud computing and software-defined networking', *Computer Networks*, Vol. 81, No. 4, pp.308–319.
- Wang, X., Chen, M., Xing, C. and Zhang, T. (2016) 'Defending DDoS attacks in software-defined networking based on legitimate source and destination IP address database', *IEICE Transactions on Information and Systems*, Vol. 99, No. 4, pp.850–859.
- Ye, J., Cheng, X., Zhu, J., Feng, L. and Song, L. (2018) 'A DDoS attack detection method based on SVM in software defined network', *Security and Communication Networks*, Vol. 2018, No. 1, pp.1–8.

- Zekri, M., El Kafhali, S., Aboutabit, N. and Saadi, Y. (2017) 'DDoS attack detection using machine learning techniques in cloud computing environments', in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, IEEE, pp.1–7.
- Zhao, K., Lu, B., Shi, H., Ren, G. and Zhang, Y. (2021) 'A DDoS attack detection and defense mechanism based on the self-organizing mapping in SDN', *Internet Technology Letters*, Vol. 7, No. 1, pp.1–4.
- Zhijun, W., Wenjing, L., Liang, L. and Meng, Y. (2020) 'Low-rate DoS attacks, detection, defense, and challenges: a survey', *IEEE Access*, Vol. 8, No. 2, pp.43920–43943.