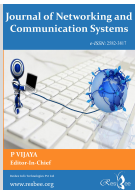
Vol.7 No.2 April 2024

Journal of Networking and Communication Systems

Received 22 February, Revised 25 March, Accepted 27 April



ECC based Authentication and Optimized Neural Network Assisted Access Control for Recognition of DoS Attacks in the Cloud

Jyothirmai Golla
University of Missouri Kansas City, United States
jyothirmai100@gmail.com

Abstract: Security is considered a major challenge for the applications implemented in cloud infrastructure. Cloud uses distributed technologies and is vulnerable to suspicious activities. Denial of Service (DoS) attack is a major issue for users, who use computers linked to the network. This paper presents a model for discovering DoS attacks in the cloud using authentication and access control. In authentication, mutual authentication between the cloud user and the server is executed with various security attributes. The security is provided using a hashing function and Elliptic Curve Cryptography (ECC) to resist various attacks. After authentication, the access control mechanism is performed to mitigate DoS attacks from the cloud. The user behaviors are recorded in the web log file, from which significant features are chosen. The selected features are subjected to the proposed Grasshopper Optimization Algorithm-based Neural Network (GOA-NN), which is designed by training the Neural network (NN) using GOA, to detect the DoS attack. The efficacy of GOA-NN is superior considering detection rate, accuracy, precision, and recall with values 0.857, 0.8, 0.866, and 0.875 respectively.

Keywords: Security, Dos Attack, Authentication, Access Control, Elliptic Curve Cryptography.

Nomenclature

Abbreviation	Expansion
PaaS	Platform as a Service
SaaS	Software as a Service
IaaS	Infrastructure as a Service
DoS	Denial of Service
DDOS	Distributed Denial of Service
SVM	Support Vector Machines
KNN	K-Nearest Neighbour
NN	Neural Network
GOA-NN	Grasshopper Optimization Algorithm-based Neural Network
RBFNN	Radial Basis Function Neural Network
FCC	Flow Correlation Coefficient
PFD	Protocol-Free Detection
RDoS	Reflection DoS
H-IDS	Hybrid Intrusion Detection System
ETT	Efficient Traceback Technique
DPPM	Dynamic Probability Packet Marking
ANN	Artificial Neural Network
MFE-ELM	Multi-Feature Extraction Extreme Learning Machine
BAT-RBF	Bat algorithm-Radial Basis Function
CSP	Cloud Service Provider

1. Introduction

Cloud is considered a gifted technology that offers resource abstraction. The cloud user does not need any external software or hardware to fulfill their desire to finish the task [3]. The cloud consists of a group that enables delivering of computing services and resource sharing in the network [14]. The cloud platform provides various infrastructures, like software, hardware, or other resource, which is inexpensive. In general, the cloud provides services to cloud users via third parties. The third parties are responsible for hiring the demanded service and the required resources using a pay-per-use basis [12]. Cloud computing is distinguished by different layers, which include PaaS, SaaS, and IaaS. The SaaS is the topmost layer, PaaS is the middle layer and IaaS is the bottommost layer [15] [16] and each layer is responsible for providing different services to the cloud users [13] [3]. Despite several advantages, cloud

https://doi.org/10.46253/jnacs.v7i2.a4

computing still poses several issues that not addressed can affect quick expansion. Cloud computing is an important research area for protecting the system from malicious users, as controlling data access has evolved as an issue for many years. Thus, several techniques [23] are introduced for implementing fine-grained access control that permits flexibility to specify differential access rights of each user [22].

Previous access control architectures consider the data owner and servers for storing the data in the same trusted domain, in which the servers are completely trusted as an omniscient reference and monitor [24], to describe and implement the access control policies. Moreover, this consideration is not needed in cloud computing as the cloud servers and data owners vary in different domains [22]. Several users adapt the cloud services over a network, which is not very trustworthy, and formulate the cloud to more attacks. The increasing applications of cloud results in more attacks and causes loss of useful data [1]. DoS attack is the most dangerous attack that produces severe hazards to the cloud environment, making services engaged to the targeted users by making a server from the computer network or building the network link congested by different packets [4]. Furthermore, DoS attacks degrade the available victim which can be a network, host, or router. They oblige rigorous computation tasks to the victim to exploit the system susceptibility or corrupt it with a large number of inadequate packets. The victim is forced to be out of service for a few minutes or even multiple days which causes severe damage to the services, which needs to be processed by the victim [9] [11]. Moreover, there exist several paralleled attacking flows, which are used for exhausting the resources in the targeted network. For initializing the attack, several computers are considered as an attack source [4].

The detection of DoS attacks needs to process a huge volume of traffic and requires detailed inspection. The DoS attack detection provides a bottleneck and interrupts other services if installed in traffic paths or if the power computation in detectors is inadequate [6]. The existing DoS or DDOS attacks [18] and their countermeasures are well-known attacks. There exist several sophisticated techniques [19] for addressing the most complicated DDOS attacks, in which the attacks are initiated by the behavior of humans. After detecting the DoS attack, trace back methods [20] are utilized for tracing and terminating the attack from the source [10]. Moreover, these techniques consider that the attacker is responsible for transmitting the packet directly to the application in such a way that the application can learn that the attack is ongoing [17]. The most frequently used classification techniques for detecting the intrusion include SVM, decision trees, associative classification, Bayesian classification, KNN classifiers, NN, and rule induction methods. Most of these methods are directly implemented as the standard method for the existing intrusion detection datasets. The standard classification algorithm does not perform well if the intrusions are lower than the usual behavior. In such situations, definite algorithms are developed and executed by the developers to address the intrusion detection issues. Moreover, the classification precision of existing techniques needs more improvement for detecting unknown attacks [21]. Thus, a scalable solution is required for detecting the attack and should be able to bedeployed in the most challenging infrastructures [6].

The aim is to design a technique to discover DoS attacks based on authentication and access control. Authentication is initiated by performing authorization between the cloud user and the cloud server with various security attributes. At first, the cloud user and server are registered in the Authorization Centre (AC) using five different messages, and different verification levels are done mutually. After authentication, the access control techniques are adapted to alleviate the DoS attack. The behavior of the user is recorded using certain features in the web log file. These features are subjected to the proposed GOA-NN for detecting the DoS attack by training the NN using GOA. Based on the trained weights, GOA-NN finds the presence of the DoS attacks.

The major contributions of this work towards the determination of DoS attacks are given below:

- Design a model for DoS attack discovery in the cloud. Here, the classifier, named GOA-NN, is developed for detecting the presence of DoS attacks by training the NN using GOA such that the weights are trained optimally.
- Design a model for DoS attack discovery in the cloud using two phases, namely the registration phase and authentication phase. Here, the security is provided using the hashing function and Elliptic Curve Cryptography (ECC) to resist various attacks.

The paper is structured as: Section 2 provides priorly devised models using DoS attack detection, Section 3 offers a model of the cloud, Section 4 elaborates on proposed GOA-NN for detecting DoS attack, Section 5 discusses simulation results, and finally, section 6 gives a conclusion using GOA-NN.

2. Literature Review

S. Velliangiri and J. Premalatha [1] designed a RBF-NN detector, for identifying DDoS attacks. The method provided improved accuracy, but the computational complexity became significantly high. Marwan Darwish*et al.* [2] developed a cloud-based authentication protocol for not only mitigating internal DoS attacks but also for the ability to defend against external DoS attackers. The method utilized a multilevel adaptive technique for dictating the efforts of protocol participants. The method failed to consider other possible DoS risks with other cloud layers. Michael R. Watson *et al.* [3] developed a method, named online cloud anomaly detection, which contains dedicated detection components of the cloud resilience architecture. Moreover, online cloud anomaly detection was applied SVM using features accumulated at the network. The method can detect suspicious strains with dedicated monitoring components per VM. The method requires less computational cost but was unable to determine network anomalies. Le Xiao *et al.* [4] developed an algorithm, named FCC-based PFD algorithm, for detecting the attacking flows effectively and was flexible. Thus, the algorithm can be used as a substitute for existing algorithms and determines the isolated attacking flows. However, the method failed to determine Real RDoS attacks in the computer network. Jorge Maestre Vidal *et al.* [5] designed artificial immune systems for mitigating DoS attacks. The method utilized constructing networks for distributed sensors for monitoring the environment. These components were used to identify the threats and react to the behavior of biological defense methods. It is proficient to follow different immune reactions and the construction of immune memory. The method was not applicable tomitigate the possible attacks if there are multiple routes in the network. Ognjen Joldzic *et al.* [6] designed a distributed, scalable solution for detecting lower-level DoS attacks, which are implemented by sending, devastating data for disturbing the network service. Scalability is attained by balancing the active traffic from various traffic processors and exploiting the flexibility that is takenin the Software Defined Networking paradigm. It was not feasible to develop a solution, which applies to all application-layer protocols. Ozge Cepheli *et al.* [7] designed a hybrid detection system, also known as H-IDS, for detecting DDoS attacks. The two datasets are taken into consideration for testing the detection performance and the method produced improved results. However, it needs more training data to detect the attack and thus, the performance is degraded. Rabia Latif *et al.* [8] developed an advanced method, named ETT, based on the DPPM method and replaced the MAC header in the IP header. At last, the path reconstruction technique was introduced for tracing an attacker. The total bytes allocated to the DPPM label were based on network topology for deployment.In 2023, Abdallah *et al.* [31] implemented ANN to detect DoS attacks in WLANs. Initially, the access point waits for the authentication request from the wireless station. The attackers changed the address of their MAC to match the victim's station. Then the management frames exchanged between the wireless devices were analyzed through ANN. Furthermore, they were trained to accurately detect the DOS and mitigate attacks. This method enhanced the security and reliability of WLANs. This method was not properly tested in real-world scenarios. In 2023, Morshedi et al. [32] executed Petri modeling to evaluate the security of the cloud. Initially, the server was analyzed and defined through official models to create an execution model. The security features were applied to evaluate the security of the server and other characteristics. Using the Petri modeling tool, the security was assessed. This method also used the OpenStack service system for modeling. Finally, OpenStack evaluated the behavior of its attack. This method was effective in assessing the security of the infrastructure systems. In 2023, Lin *et al.* [33] applied MFE-ELM for the identification of intrusion detection inIoT applications. Initially, the input data was preprocessed to remove unwanted noise and information. The features were then extracted from the image. MFE-ELM algorithm was used to detect and discover any intrusion to the cloud node. If it finds the intrusion then it automatically prohibits access to the cloud nodes. This method was effective in the detection of intrusion attacks. In 2023, Issa *et al.* [34] have ensembled CNN and LSTM for the detection of DDoS. Initially, NSL-KDD was preprocessed to remove unwanted data. Then the image was fed into the CNN layer to extract its features. The output from the CNN layer was sent to LSTM to capture temporal dependencies of the data and classification was done Finally, the data was trained and tested to evaluate its performance. This method obtained higher accuracy than other existing methods. In 2023, Al-Fayoumi *et al.* [35] have introduced the MQTT protocol for capturing LR-DDoS. Initially, data was preprocessed and split into training and testing data. In the training model, four ML models were used. Then the performance was evaluated from each model to evaluate its accuracy, precision, recall, and F1 score to get the evaluated output. This method provides effective detection of DoS attacks in real-time. In 2023, RM *et al.* [36] have rendered hybrid DL based on intrusion detection in the AWS cloud environment. Initially, the raw data were preprocessed and normalized. Then the feature was performed using PCA into attack and non-attack using SMO-FCM. The intrusion was identified through the Auto Encoder algorithm. Finally, the result was analyzed and accurate output was obtained. However, this method required computational resources due to the use of the DL algorithm. In 2023, Abu Bakar *et al.* [37] adopted Ml and sequential feature selection to get faster results in DDoS

attacks. Initially, this method used CICDDoS2019 as a custom-generated dataset. The system performed automatic feature extraction. In the learning phase, the system will automatically detect the DDoS detect. In addition to this ML continuously monitors and generates an alert to all nodes. Finally, the static agent controls the end host and other agents transmit the warning to intermediate nodes. This helps the process to detect the DDoS quickly and make a joint decision with the adjutant routers. In 2022, Alduailij *et al.* [38] have introduced ML for the detection of DDoS. Initially, the dataset was selected and preprocessed to remove inconsistencies and irrelevant data. Then the feature was extracted through MI and RFFI. Using five classification techniques the DDoS attack was classified. Finally, the datasets were evaluated and an accurate output was obtained. This method provides high accuracy in the detection of DDoS. However, this method did not apply to unknown DDoS attacks.

2.1 Challenges

The issues confronted by prior models are listed,

- Security is the major problem in the cloud platform and is amain problem in cloud execution. The network attacker adapts network vulnerability and its protocols to damage data and applications [1].
- In [1], BAT-RBF is developed for intrusion detection in the cloud environment. This method can determine the appropriate network structure and network parameters for obtaining the best solution. However, this method has high computational complexity and was unable to examine the other attacks.
- The confidential information uploaded to the cloud is susceptible to several security issues, which include confidentiality, integrity, and availability. Moreover, the uninterrupted cloud service gains the attention of intruders and misuse those resources and services offered by the CSP [21].
- The major drawbacks of authentication protocols for detecting DoS attacks are credentials, which are exposed to third parties for accessing restricted resources. Here, the first drawback includes access data, in which a password is saved by third parties for future access. The second drawback is that the resource owner cannot limit third-party access and cannot control the access duration [2].

3. System Model

The model of cloud used in DoS attack detection is represented in Figure 1. The model contains three important aspects, which are the Authentication Center (AC), data owner, cloud server, and cloud user. The users, who are responsible for obtaining information saved in the cloud server, are termed as cloud users. The cloud user is responsible for requesting a service, which is matched with a user query in the database using a querymapper and request handler for accessing the needed information. The cloud server consists of management servers, which contain physical servers and virtual servers. Thus, an effective technique is required for the owners to preserve the data and to send cloud users or end users by preventing malicious attacks. The end users use the information for developing the business, but the sensitive data is hidden from the individual. Thus, it is a major challenge to convert the original data to sanitized data by hiding the sensitive information. In addition, protecting sensitive information requires data maintenance so that the data given to the cloud user is valuable.

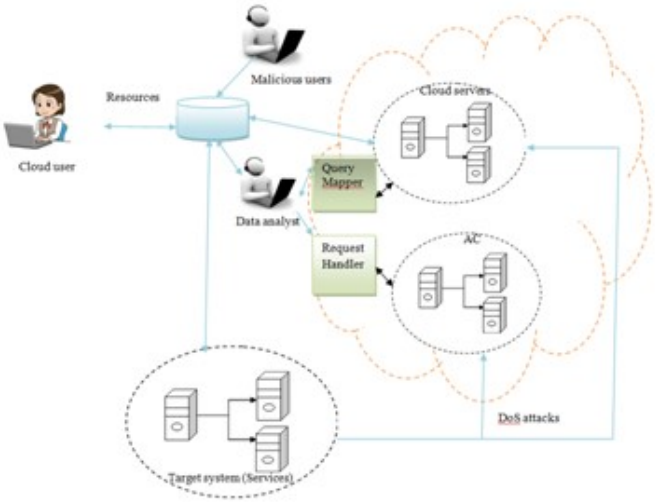


Fig. 1. *Cloud model for detecting DoS attack*

4. Proposed GOA-NN for Dos Attack Discovery

The technique of DoS attack detection by adapting the proposed GOA-NN is elaborated. The proposed technique contains two phases, which include the registration and authentication phases. Here, mutual authentication and several verification levels are employed to protect the cloud resources from unauthorized users when the cloud users communicate with the cloud server and AC. In registration, the cloud users are registered under the cloud server and AC. In the authentication phase, a mutual authentication is carried out for authenticating the cloud users.

4.1 Registration Phase

In the registration phase, the user and server share the distinctive identity and credential particulars to register on the server. It is started after the cloud user gives all the information to the cloud server. The first step is to register cloud users under cloud servers and an authentication center.

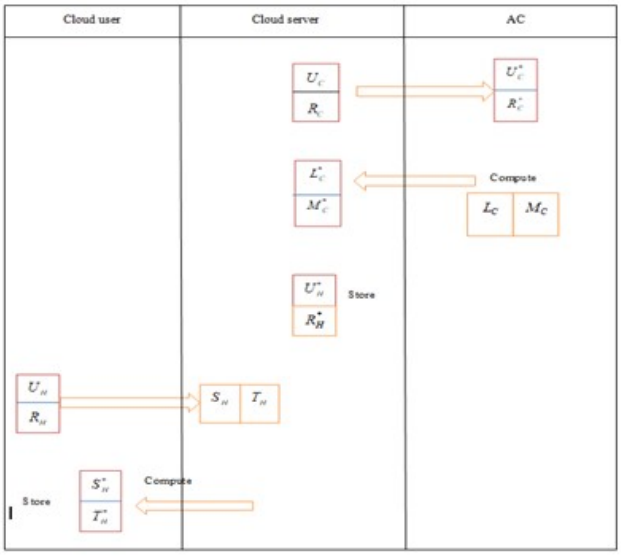


Fig. 2.*Registration phase*

The server is registered under AC using the username and password of the server. At first, the username U_C and password R_C of the server are sent to the AC. Here, the AC stores U_C and R_C in the database. Then, the AC computes encoded messages L_C and encrypted messages M_C using ECC and hashing function. L_C is computed by the values obtained by XORing Q_g and R_C^*.Then, the convolution operation is performed between the obtained value and the ECCECC-applied stored username U_C^*. The message computed at the AC is represented as,

$$L_C = h(Q_g \oplus R_C^*)\Theta ECC(U_C^*) \tag{1}$$

where h represents a hashing function, Q_g indicates AC key, R_C^* denotes stored server passwords, U_C^* indicates stored server usernames.

The encrypted message M_C is computed by XO Ring the message L_C and the public key Q_R and the result is encrypted using ECC [27]. The encrypted message computation is given as,

$$M_C = ECC(L_C \oplus Q_R) \tag{2}$$

where, L_C represents an encrypted message, and Q_R represents a public key.

The computed messages are sent to the server using a secure communication channel. Then, the username U_H and password R_H of the user is registered.

The server computes coded messages S_H by performing XOR operation between Q_C and R_H^* and hashing is adapted on the obtained result. The hashed result and the ECC of U_H^* adapts convolution operation for computing S_H are represented as follows,

$$S_H = h(Q_C \oplus R_H^*)\Theta ECC(U_H^*) \tag{3}$$

Where Q_C represents the server key, R_H^* indicates the stored password of users, and U_H^* indicates the stored username.

The enciphered message T_H is computed by XORing the coded message S_H and public key Q_R and further, the result is applied with the hashing function.

$$T_H = ECC(S_H \oplus Q_R) \tag{4}$$

where, S_H represents the coded message, and Q_R denotes the public key.

4.2. Authentication Phase

The authentication is executed after finishing the registration phase. To perform authentication, users, servers, and AC are validated using various verification levels, as elaborated in Figure 3.

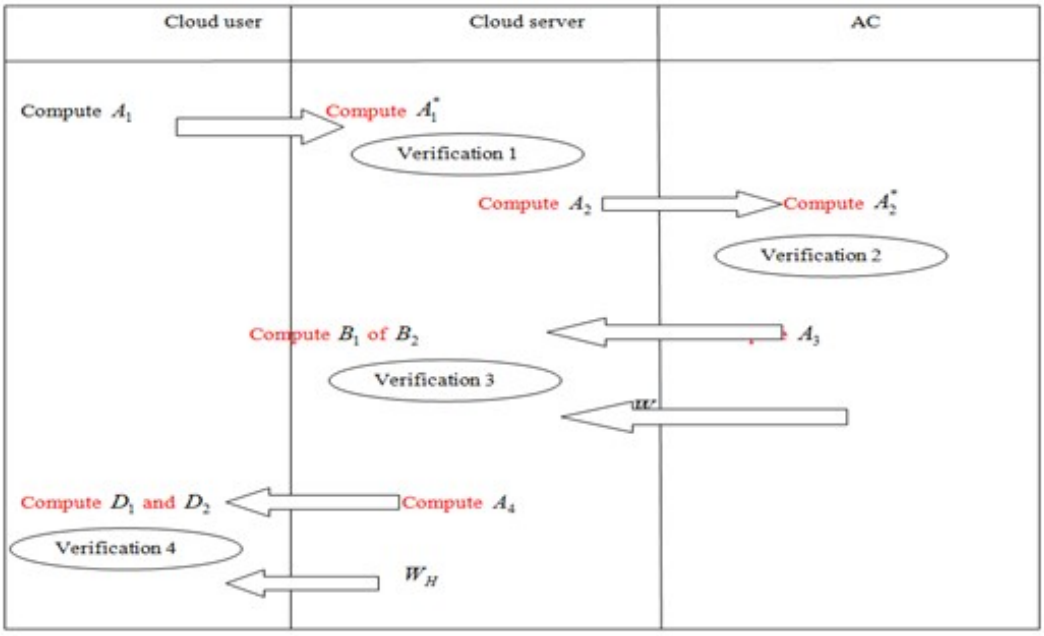


Fig. 3. *Authentication phase*

The steps involved in the authentication phase are described as follows,

Step1: First level verification using authorized messages A_1 and A_1^*: If a cloud user wants authentication from the server side, then the cloud user first calculates A_1 and forwards it to the server. Thereafter, the server evaluates A_1^* at the server side and both computed messages are verified for a match. Here, the A_1 is evaluated as follows, Initially, the convolution operation is adapted between U_H and R_H . Then the hashing is performed S_H^* . Then results are Xored and then the ECC is applied for computing on A_1. Similarly, A_1^* is computed as follows, Initially, the convolution operation is applied between the U_H^* and R_H^* . Then the hashing is performed on S_H . Then results are Xored and then the ECC is applied for computing A_1^*.

$$A_1 = ECC([U_H \Theta R_H] \oplus h(S_H^*)) \tag{5}$$

$$A_1^* = ECC([U_H^* \Theta R_H^*] \oplus h(S_H)) \tag{6}$$

If the authorized messages computed by the cloud user and cloud server are the same, i.e. $A_1 \equiv A_1^*$, the first level verification is done.

Step2: Second level verification using A_2 and A_2^*: For the authentication of the cloud server and the AC, the server calculates A_2 and forwards it to the AC, which computes A_2^*.Here, the A_2 is evaluated as follows, Initially, the convolution operation is adapted between the U_C and R_C . Then the hashing is

performed on L_C^*. Then results are Xored and then the ECC is applied for computing A_2. Similarly, A_2^* is enumerated as follows, Initially, the convolution operation is adapted between U_C^* and R_C^*. Then the hashing is performed on L_C. Then results are Xored and then the ECC is applied for computing A_2^*.

$$A_2 = ECC([U_C \Theta R_C] \oplus h(L_C^*)) \tag{7}$$

$$A_2^* = ECC([U_C^* \Theta R_C^*] \oplus h(L_C)) \tag{8}$$

If $A_2 \equiv A_2^*$, then second-level verification is done.

Step3: Third level verification: The AC provides authentication to the server by computing A_3 at AC and then, sending it to the server. Here, A_3 is computed as follows: Initially, L_C and M_C are XOR-ed and the result is further XOR-ed with the hashing of the concatenated value of U_C^*, R_C^* and W_C.

$$A_3 = (L_C \oplus M_C) \oplus h(U_C^* \| R_C^* \| W_C) \tag{9}$$

Then, the server computes certified messages B_1 and B_2, then both the computed messages are matched. Here, B_1 is computed as follows: Initially, the hashing function is applied on U_C, R_C and W_C^*. Then, the result obtained after the hashing is XOR-ed with A_3^*.

$$B_1 = A_3^* \oplus h | (U_C \| R_C \| W_C^*) \tag{10}$$

The certified message B_1^* is computed by performing the XOR operation of L_C^* and M_C^*, then it is formulated as follows,

$$B_1^* = L_C^* \oplus M_C^* \tag{11}$$

$$B_1 \equiv B_1^* \text{ Verified} \tag{12}$$

Now B_1 is verified with B_1^* for the successful verification. Then B_2^* is computed by XOR-ing B_1^* and L_C^*.

$$B_2^* = B_1^* \oplus L_C^* \tag{13}$$

$$B_2 \equiv L_C^* \text{ Verified} \tag{14}$$

Here, B_2 is verified with L_C^* and if the verification is successful, A_4 is computed, similar to A_3, but with coded message S_H and enciphered message T_H and the hashing of concatenated values at the user.

$$A_4 = (S_H \oplus T_H) \oplus h(U_H^* \| R_H^* \| W_H) \tag{15}$$

Step 4: Fourth level verification: Then, the endorsed message 1 D_1 is computed as follows, At first, the hashing of $U_H^*, R_H^*, and\ W_H^*$ is done. Then, the produced outcome is XORed with A_4. The mathematical representation of D_1 is formulated as follows,

$$D_1 = A_4^* \oplus h | (U_H^* \| R_H^* \| W_H^*) \tag{16}$$

Here, the D_1^* is computed by applying the Xor operation between S_H^* and T_H^* and is represented as follows,

$$D_1^* = S_H^* \oplus T_H^* \tag{17}$$

$$D_1 \equiv D_1^* \text{ Verified} \tag{18}$$

Then, the endorsed message 2 D_2 is computed by taking the XOR operation between D_1^* and T_H^* and is formulated as follows,

$$D_2^* = D_1^* \oplus T_H^* \tag{19}$$

$$D_2 \equiv T_H^* \tag{20}$$

After completing fourth-level authentication, the cloud user is authenticated. After authenticating the cloud user and the server, the access control mechanism using the proposed GOA-NN is carried out to alleviate DoS attacks in the cloud.

4.3Proposed GOA-NN For Dos Attack Detection

Once the user and the server are authenticated, the access control mechanisms are carried out to mitigate DoS attacks in the cloud. Here, GOA-NN is adapted to discover the existence of DoS attacks. Thus, the behavior of the user is recorded using various parameters on the web log file. The features mined are subjected to NN that employs GOA for the training to detect the DoS attack. Figure 4depicts the block diagram of the proposed GOA-NN-based DoS attack detection technique.

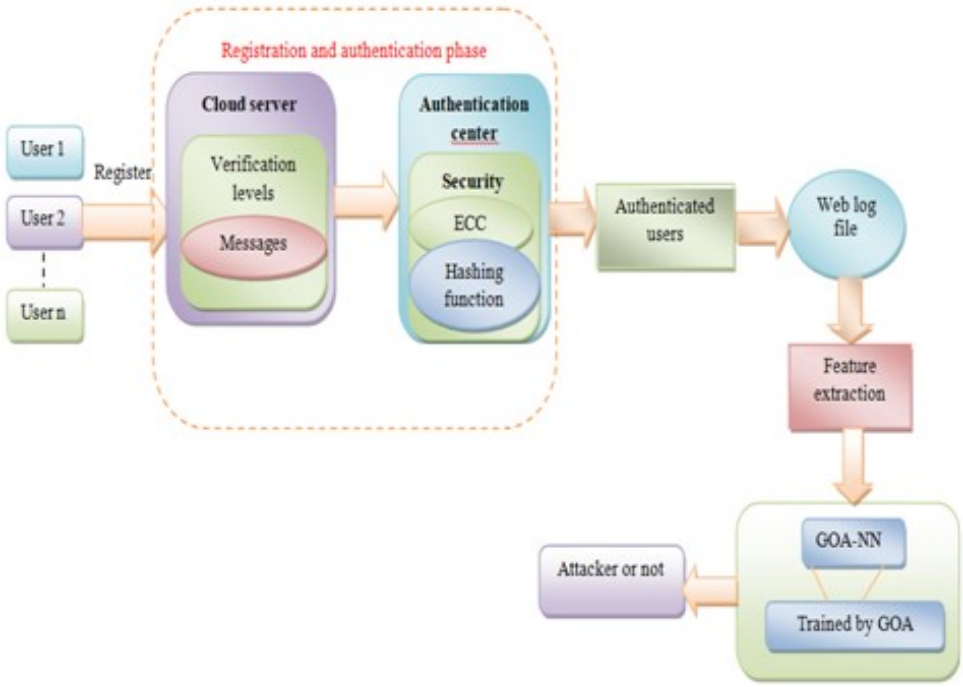


Fig. 4. *Block diagram of DoS attack detection using proposed GOA-NN*

a) Architecture of NN

The NN [26] contains three layers named input layer, the hidden layer, and the output layer. The input layer represents input features produced by examining the behavior of users. Thereafter, normalization of produced features is considered to improve network precision. The hidden layers comprise neurons to mine features considering the internal operation of the network. Finally, the output layer produces output and each layer is linked to each other to establish a network. The architecture of the neural network is given in Figure 5.

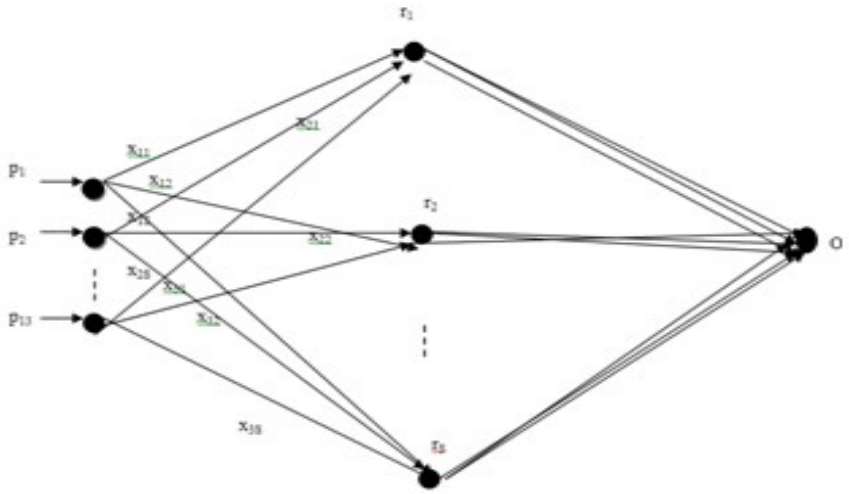


Fig. 5. *Architecture of NN*

Let us consider the p_j number of input features where $1 \le j \le 13$.

$$p = \{p_1, p_2, \ldots, p_j, \ldots, p_{13}\} \tag{21}$$

The hidden layers in the NN are represented as,

$$R = \{f_1, f_2, \ldots, f_j, \ldots, f_Z\} \tag{22}$$

where, Z number of hidden neurons present in the NN. The output at the hidden layer is represented by,

$$f_j = \frac{1}{c}\sum_{i=1}^{c} r_i \; p_i \tag{23}$$

where, r_i is the weight corresponding to the input feature p_i. The weights of the NN are represented as follows,

$$Y = \{r_1, r_2, \ldots, r_S\} \tag{24}$$

where, S refers to total weights in NN. Now, the output of the neural network is,

$$O_j = \sum_{j=1}^{Z} f_j \times r_j \tag{25}$$

where, O_j signifies output neuron which whether the presence of attack, f_j represents j^{th} hidden layer. The output layer is also represented as $O_j = F(r_z, p)$.where r_z denotes the current iteration. After detecting the current NN output $O_j = F(r_z, p)$**,** the error of the output gets computed. Here, the error is modeled as,

$$E_t = \sum_{j=1}^{i}\left(O_j^t - G_j\right) \tag{26}$$

where, G_j refers targeted output of the neurons in NN.

b) Training by GOA

Once the NN is initialized, then the weight values are initialized with random values. At first, the weights are initialized arbitrarily and then the weights are updated dynamically for each iteration using GOA.

The GOA [25] is inspired by the swarming behavior of grasshoppers to solve optimization issues. The weights are updated using GOA and are represented as follows,

$$T_m^l = n\left(\sum_{\substack{t=1 \\ t\neq m}}^{J} n\frac{N_l - F_l}{2} d(|\, y_t^l - y_m^l\,|)\frac{y_t^l - y_m^l}{l_{m,t}}\right) + \hat{P}_l \tag{27}$$

where, J denote number of grasshoppers, N_l represents upper bound, F_l indicates lower bound, y_t denote t^{th} grasshopper, y_m indicates m^{th} grasshopper, $l_{m,t}$ represents distance between m^{th} grasshopper and t^{th} grasshopper, n represents a decreasing coefficient and is given as,

$$n = n_{\max} - z\frac{n_{\max} - n_{\min}}{Z} \tag{28}$$

Where, $n_{\max}$ and $n_{\min}$ represents maximum and minimum value, z denote current iteration, Z indicates maximum number of iterations.

The pseudo-code of the proposed GOA-NN is illustrated in Table 1.

Table 1 Pseudo code of the proposed GOA-NN

Input: Population Y
Parameters: $n_{\max}$, $n_{\min}$, z, Z
Output : Y_{z+1}
while($z < Z$)
Update n using equation (28)
for each search agent,
Update the position of the present search agent with expression (27)
end for
Update Y if there exist best solution
$z = z + 1$
end while
Return Y

5. Results and Discussion

In this section, the results of the proposed GOA-NN-based attack detection technique to the existing methods Ning, H *et al.* [28], Hu, T *et al.*[29], and V. Jog [30] is described.

5.1 Experimental Setup

It is implemented in JAVA with the cloudsim tool.

5.2 Performance Metrics

Each of the metrics used is defined below.

Accuracy: It refers to the degree to which the results of a measurement, calculation, or specification conform to the correct value or standard.
Precision: It refers to the closeness of two or more measurements to each other.
Recall: The ratio of pertinent samples that are retrieved over total instances is known as recall.
Detection rate: It describes the ratio of the malicious attack count that is detected correctly to the total attack count.

5.3 Comparative Methods

The efficacy of GOA-NN is inspected by comparing GOA-NN with prior methods. Here, three existing methodologies, such as [28] [29],and [30] are taken for comparison.

5.4.1 Comparative Analysis

The comparative assessment of GOA-NN is carried out with prior methods using metrics, like accuracy, precision, recall, and detection rate, considering the DoS attack, and all attack scenarios.

a) Analysis Considering the DoS Attack

The comparative analysis of the proposed GOA-NN with existing, Ning, H *et al.* [28], Hu, T *et al.*[29], and V. Jog [30] in terms of detection rate, accuracy, precision, and recall considering the DoS attack scenario is illustrated in Figure 6. Figure 6a illustrates detection rate analysis with varying numbers of users. When the number of users is 140, the corresponding values of the detection rate are measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.5, 0.571, 0.714, and 0.857, respectively, whereas when the number of users is130, the values of the detection rate measured by Ning, H *et al.*[28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are0.357, 0.571, 0.714, and 0.714 respectively. Figure 6b illustrates the analysis based on accuracy metric with the varying number of users. When the number of users is 100, the corresponding accuracy is measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN is 0.36, 0.766, 0.783, and 0.833, respectively, whereas when the number of users is 110, the accuracy values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.36, 0.716, 0.733, and 0.833, respectively. Figure 6c reveals precision values varying from 100 to 140 users. When the number of users is 120, the precision values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.088, 0.315, 0.6125, and 0.65, respectively. Similarly, when the number of users is 130, the corresponding precision values are measured by existing techniques, Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.108, 0.263, 0.633, and 0.712, respectively. Figure 6d shows the recall measure calculated by the comparative techniques based on varying numbers of users. With 140 users, the recall values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.714, 0.847, 0.867, and 0.875, respectively. Similarly, when the number of users is 110, the corresponding recall values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.5, 0.714, 0.857, and 0.869, respectively.

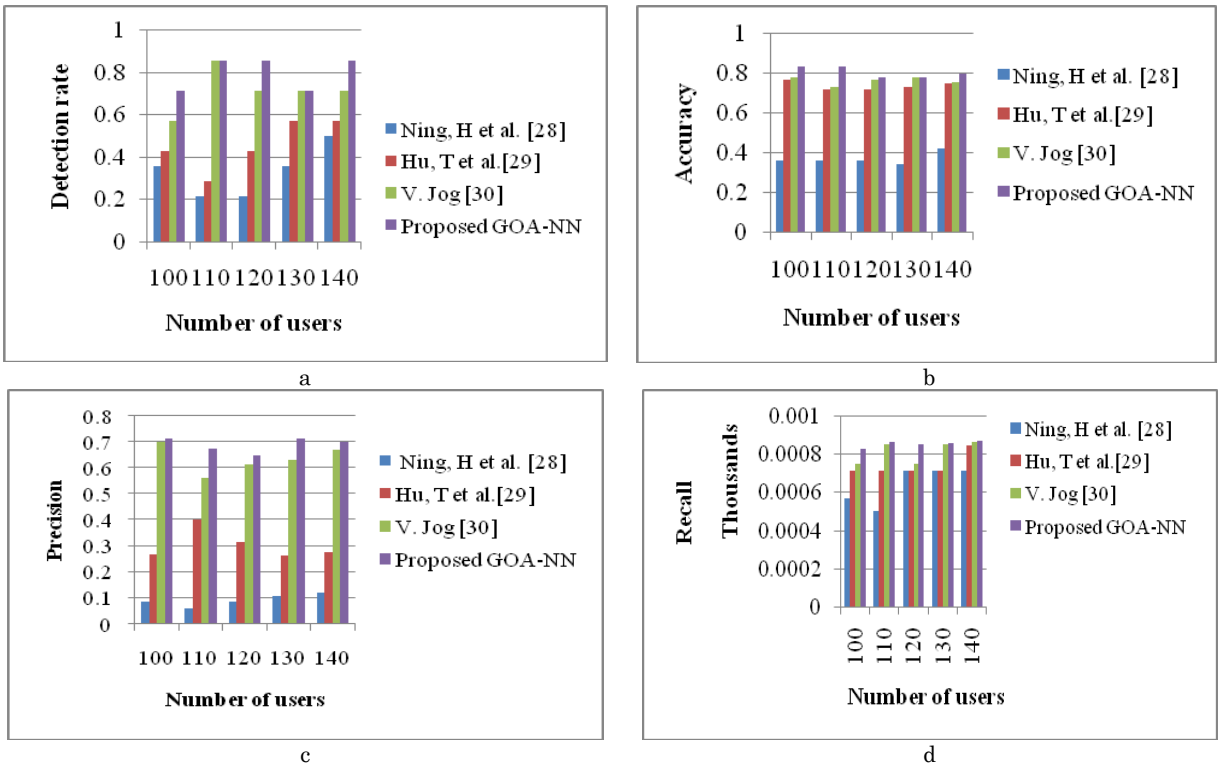


Fig. 6. *Analysis based on DoS attack based on a) Detection rate b) Accuracy c) Precision d) Recall*

b) Analysis considering all attacks

Figure 7 illustrates the analysis of proposed GOA-NN with existing methods, like Ning, H *et al.* [28], Hu, T *et al.*[29], and V. Jog [30] based on detection rate, accuracy, precision, and recall. The detection rate analysis with varying numbers of users is exposed in Figure 7a. When the number of users is 100, the corresponding values of the detection rate are computed by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.456, 0.695, 0.813, and 0.845, respectively, whereas, when the number of users is 120, the values of detection rate calculated by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN, are 0.357, 0.571, 0.571, and 0.845, respectively. The detection rate attained by the proposed GOA-NN method proves that it has the maximum detection rate. The analysis based on the accuracy metric with the varying number of users is depicted in Figure 7b. When the number of users is 140, the corresponding accuracy measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are0.6, 0.683, 0.733, and 0.8, respectively, whereas when the number of users is 130, the accuracy values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.36, 0.75, 0.78, and 0.8, respectively. The precision analysis is explicated in Figure 7c. When the number of users is 140, the precision values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.538, 0.6, 0.85, and 0.866, respectively. Similarly, when the number of users is 100, the corresponding precision values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.410, 0.560, 0.766, and 0.824, respectively. The analysis is based on recall value calculated by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN with varying numbers of users is depicted in Figure 7d. For 130 users, the recall values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.5, 0.571, 0.714, and 0.857, respectively. Similarly, when the number of users is 140, the corresponding recall values measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.613, 0.713, 0.856, and 0.869, respectively.

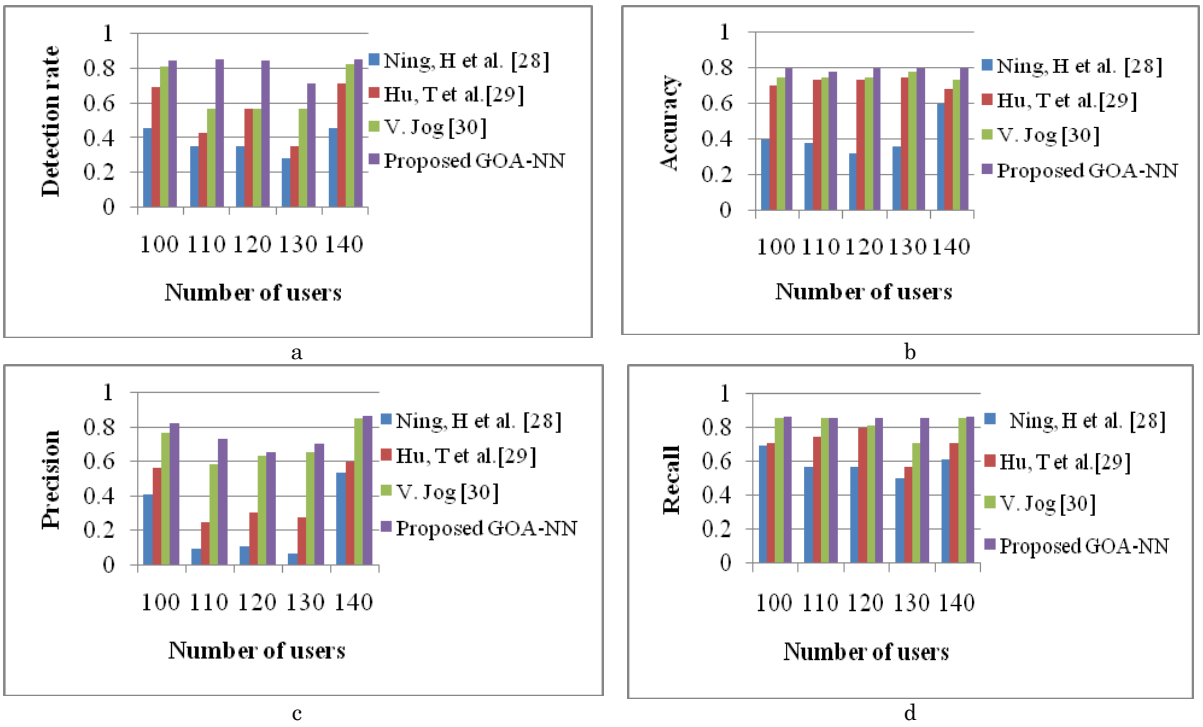


Fig. 7. *Analysis based on all attacks based on a) Detection rate b) Accuracy c) Precision d) Recall*

5.3. Discussion

Table 2. *Comparative Performance Discussion Table*

Methods	*Detection Rate*	*Accuracy*	*Precision*	*Recall*
Ning, H *et al.* [28]	0.5	0.6	0.538	0.714
Hu, T *et al.*[29]	0.713	0.75	0.6	0.847
V. Jog [30]	0.825	0.756	0.85	0.867
Proposed GOA-NN	**0.857**	**0.8**	**0.866**	**0.875**

Table 2 describes the overall performance of the proposed method in terms of accuracy, precision, detection rate, and recall values with that of the existing techniques, such as Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30], used for detecting the DoS attack. The values in terms of detection rate measured by existing methods, Ning, H *et al.* [28], Hu, T *et al.*[29], and V. Jog [30] are 0.5, 0.713, 0.825, and the proposed GOA-NN is 0.857. The values in terms of accuracy metric measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.6, 0.75, 0.756, and 0.8. The values attained are based on the precision metric measured by Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.538, 0.6, 0.85, and 0.866. The values in terms of recall metric computed by the methods, Ning, H *et al.* [28], Hu, T *et al.*[29], V. Jog [30] and proposed GOA-NN are 0.714, 0.847, 0.867, and 0.875, respectively.

6. Conclusion

This paper illustrates the access control mechanism to prevent DoS attacks using two methods, namely authentication and access control. The authentication is initiated with the mutual authentication amidst cloud user and cloud server based on four different messages. This work presents the need for effective authentication to ensure security in the cloud environment such that communication among users in the cloud is enhanced. Authentication is carried out by verifying the messages for authenticating the users. The hashing function and ECC are adapted in the proposed technique for providing security against different kinds of attacks. After the authentication of users, the behavior of authenticated users is recorded using certain parameters. Based on the above features, the proposed GOA-NN, which is designed by performing GOA-based training in NN, identifies the DoS attackers. Here, the efficacy of the developed model shows superior performance in terms of detection rate, accuracy, precision, and recall with values of 0.857, 0.8, 0.866, and 0.875, respectively.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] S.Velliangiri, and J. Premalatha, "Intrusion detection of distributed denial of service attack in the cloud", Cluster Computing, pp. 1-9, 2017.

[2] Marwan Darwish, AbdelkaderOuda, and Luiz Fernando Capretz, "A cloud-based secure authentication (CSA) protocol suite for defense against Denial of Service (DoS) attacks," Journal of Information Security and Applications, vol. 20, pp. 90-98, 2015.

[3] Michael R. Watson, Angelos K. Marnerides, Andreas Mauthe, and David Hutchison, "Malware detection in cloud computing infrastructures," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 192-205, 2016.

[4] Le Xiao, Wei Wei, Weidong Yang, YulongShen, and Xianglin Wu, "A protocol-free detection against cloud-oriented reflection DoS attacks," Soft Computing, vol. 21, no. 13, pp. 3713-3721, 2017.

[5] Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, and Luis Javier Garcia Villalba, "Adaptive artificial immune networks for mitigating DoS flooding attacks," Swarm and Evolutionary Computation, 2017.

[6] OgnjenJoldzic, ZoranDjuric, and PavleVuletic, "A transparent and scalable anomaly-based DoS detection method," Computer Networks, vol. 104, pp. 27-42, 2016.

[7] OzgeCepheli, SalihaBuyukcorak, and GunesKarabulut Kurt, "Hybrid intrusion detection system for DDoS attacks," Journal of Electrical and Computer Engineering, vol. 2016, pp. 1-8, 2016.

[8] RabiaLatif, Haider Abbas, SeemabLatif, and Ashraf Masood, "Distributed denial of service attack source detection using efficient traceback technique (ETT) in a cloud-assisted healthcare environment," Journal of medical systems, vol. 40, no. 7, pp. 161, 2016.

[9] HasinAfzal Ahmed, PriyakshiMahanta, Dhruba Kumar Bhattacharyya, and Jugal Kumar Kalita, "Shifting-and-scaling correlation based biclustering algorithm," IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 11, no. 6, pp. 1239-1252, 2014.

[10] Kalita, "Network attacks: Taxonomy, tools and systems," Journal of Network and Computer Applications, vol. 40, pp. 307-324, 2014.

[11] Zhiyuan Tan, ArunaJamdagni, Xiangjian He, Priyadarsi Nanda, and Ren Ping Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," IEEE Transactions on parallel and distributed systems, vol. 25, no. 2, pp. 447-456, 2014.

[12] Hassen Mohammed Alsafi, Wafaa Mustafa Abduallah, and Al-Sakib Khan Pathan, "IDPS: an integrated intrusion handling model for the cloud computing environment." International Journal of Computing & Information Technology, vol. 4, no. 1, pp. 1-16, 2012.

[13] OpeyemiOsanaiye, Kim-Kwang Raymond Choo, and MqheleDlodlo, "Distributed denial of service (DDoS) resilience in the cloud: review and conceptual cloud DDoS mitigation framework," Journal of Network and Computer Applications, vol. 67, pp. 147-165, 2016.

[14] Yang Liu, and Wei Wei, "A replication-based mechanism for fault tolerance in MapReduce framework," Mathematical Problems in Engineering, vol.20, pp. 1-7, 2015.

[15] Opeyemi A. Osanaiye, and MqheleDlodlo, "TCP/IP header classification for detecting spoofed DDoS attack in Cloud environment," in Proceedings of EUROCON 2015- IEEE International Conference on Computer as a Tool (EUROCON), pp. 1-6, 2015.

[16] Opeyemi A Osanaiye, "Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing," in Proceedings of 2015 IEEE 18th International Conference onIntelligence in Next Generation Networks (ICIN), pp. 139-141, 2015.

[17] Huan Liu, "A New Form of DOS Attack in a Cloud and Its Avoidance Mechanism," in Proceedings of the 2010 ACM workshop on Cloud computing security workshop, pp. 65-76, 2010.

[18] Stuart Staniford, Vern Paxson, and Nicholas Weaver, "How to 0wn the Internet in Your Spare Time," in Proceedings of the 11th USENIX Security Symposium, pp. 1-20, August 5-9, 2002.

[19] SrikanthKandula, Dina Katabi, Matthias Jacob, and Arthur Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds," in Proceedings of 2nd Symposium on Networked Systems Design & Implementation (NSDI '05), pp. 1-14, 2005.

[20] Abraham Yaar, Adrian Perrig, and Dawn Song, "FIT: Fast Internet Traceback," in Proceedings of IEEE Infocom, pp. 1-12, March 2005.

[21] N.Pandeeswari, and Ganesh Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," Mobile Networks and Applications, vol. 21, no. 3, pp. 494-505, 2016.

[22] Shucheng Yu, Cong Wang, KuiRen, and Wenjing Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proceedings of IEEE Infocom, pp. 1-9, 2010.

[23] Ting Yu, and Marianne Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proceedings of Security and Privacy 2003 IEEE Symposium on, pp. 110-122, 2003.

[24] James P Anderson, “Computer Security Technology Planning Study,” Air Force Electronic Systems Division, Report ESD-TR-73-51, 1972.
[25] ShahrzadSaremi, SeyedaliMirjalili, Andrew Lewis, "Grasshopper Optimisation Algorithm: Theory and application", Advances in Engineering Software, Vol. 105, pp. 30-47, March 2017.
[26] Kumar, M. and Bhatnagar, C., "Crowd Behavior Recognition Using Hybrid Tracking Model and Genetic Algorithm Enabled Neural Network," International Journal of Computational Intelligence Systems, vol.10, no.1, pp.234-246, 2017.
[27] Alowolodu, O.D., Alese, B.K., Adetunmbi, A.O., Adewale, O.S. and Ogundele, O.S., "Elliptic curve cryptography for securing cloud computing applications," International Journal of Computer Applications, vol.66, no.23, 2013.
[28] Ning, H., Liu, H. and Yang, L., "Aggregated-proof based hierarchical authentication scheme for the Internet of things," IEEE Transactions on Parallel & Distributed Systems, vol.1, pp.1-1, 2015.
[29] Hu, T., Wang, J., Zhao, G. and Long, X., "An improved mutual authentication and key update scheme for multi-hop relay in Internet of Things," In proceedings of 7th IEEE Conference on Industrial Electronics and Applications, pp.1024-1029, 2012.
[30] V. Jog, "Memory and machine attributes-based profiling and elliptic curve cryptography-based multi-level authentication for the security of Internet of Things," International Journal of Intelligent Computing and Cybernetics, vol.10, no.2, pp.241-256, 2017.
[31] Abdallah, A.E., Hamdan, M., Gismalla, M.S., Ibrahim, A.O., Aljurayban, N.S., Nagmeldin, W. and Khairi, M.H., “Detection of Management-Frames-Based Denial-of-Service Attack in Wireless LAN Network Using Artificial Neural Network”, Sensors, Vol. 23(5), pp.2663, 2023.
[32] Morshedi, R. and Matinkhah, S.M., “Providing a method for security analysis and evaluation of infrastructure services in cloud computing”, 2023.
[33] Lin, H., Xue, Q., Feng, J. and Bai, D., “Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine”, Digital Communications and Networks, Vol. 9(1), pp.111-124, 2023.
[34] Issa, A.S.A. and Albayrak, Z., “DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM”, Acta PolytechnicaHungarica, Vol. 20(2), 2023.
[35] Al-Fayoumi, D. and Abu Al Haija, Q., “Capturing Low-Rate DDoS Attack based on MQTT Protocol in Software Defined-IoT Environment”, Qasem S., Capturing Low-Rate Ddos Attack Based on Mqtt Protocol in Software Defined-Iot Environment.
[36] RM, B. and MK, J.K., “Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm”, Electronics, Vol. 12(6), pp.1423. 2023.
[37] Abu Bakar, R., Huang, X., Javed, M.S., Hussain, S. and Majeed, M.F., “An Intelligent Agent-Based Detection System for DDoS Attacks Using Automatic Feature Extraction and Selection”, Sensors, Vol. 23(6), pp.3333, 2023.
[38] Alduailij, M., Khan, Q.W., Tahir, M., Sardaraz, M., Alduailij, M. and Malik, F., “Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method”, Symmetry, Vol. 14(6), pp.1095, 2022.