Control and Cybernetics

vol. **47** (2018) No. 4

A technique for DoS attack detection in e-commerce transactions based on ECC and Optimized Support Vector Neural Network

by

Javed R. Shaikh[1] **and Georgi Iliev**[2]

[1]SKN Sinhgad Institute of Technology and Science, Lonavala, India
javedsheikh1987@gmail.com
[2]Technical University of Sofia, Bulgaria

Abstract: Cloud computing has become a significant area for business due to the high demand from people engaging in commerce and hence, protecting the entities from attacks, like Denial-of-Service (DoS) attacks is essential. Therefore, an effective DoS attack detection technique is required in the e-commerce transactions to provide security in this space. Accordingly, in this paper, a technique is developed for DoS attack detection for the e-commerce transactions by proposing Glowworm Swarm Optimization-based Support Vector Neural Network (GSO-SVNN) based authorization. The user and the server, who are registered for accessing the e-commerce web, are first registered and then, authenticated based on Elliptic Curve Cryptography (ECC) encryption with four verification levels follows. The proposed GSO-SVNN classifier, which is developed by incorporating the GSO algorithm in the training procedure of SVNN, determines the class of the user. The performance of the proposed technique is evaluated using four metrics, namely accuracy, precision, recall, and False Positive Rate (FPR), and the experimental results show that the maximum accuracy attained by the proposed DoS attack detection technique is 95.1%. This proves that the proposed technique is effective in detecting DoS attacks in e-commerce applications using the proposed GSO-SVNN based authorization.

Keywords: cloud computing, e-commerce transactions, DoS attack detection, SVNN, GSO

1. Introduction

The extensive sharing of resources in the cloud computing platforms has made it an attractive tool in recent years. The cost of accessing the cloud resources relies on the usage and the demand of the user. Cloud computing allows the users to utilize the resources in a considerable degree by statistical multiplexing, while dynamic scaling reduces the resource costs (Chen, Paxson and Katz, 2010). The Internet is becoming an unavoidable medium of business, with a large number of users being available in terms of electronic commerce (Gomez-Herrera, Martens and Turlea, 2014). Even though the usage of the cloud platforms is rising in the sense of business, the challenges it poses concerning security are equally increasing (Cao et al., 2015). Hence, a part of the users are reluctant to engage in the business via the Internet, as they are very concerned about the security and trustworthiness of respective entities (Gomez-Herrera, Martens and Turlea, 2014). A remedy, meant to prevent the entities from malicious attacks is to utilize adequate security and trust mechanisms. One of the approaches to protect the system with security service, like authentication, is known to be the hard security (Rasmusson and Jansson,1996). However, detecting entities that reveal ambiguous information after checking in legally in the e-commerce system is a challenging task. Hence, more advantageous and promising control mechanisms are required for the protection against various kinds of attacks. These mechanisms are categorized as soft security mechanisms (Rasmusson and Jansson, 1996), where trust and reputation are the two key factors of security (Zupancic and Trcek, 2017).

Over the past years, various trust models (Pinyol and Sabater-Mir, 2013; Josang, Ismail and Boyd, 2007) have been introduced by the researchers to design effective trust and reputation management schemes. These mechanisms, in particular, aid in developing a technique that measures the trust between different unknown entities in the cloud computing environment. The major reason for the development of such schemes is to improve the user interactions by avoiding wrong interactions and thereby alleviating the risks during the transactions. Moreover, the studies conducted have shown that the suppliers having high status on the market often provide their products at a higher cost (Resnick and Zeckhauser, 2002; Lucking-Reiley et al., 2007). Among a number of issues prevailing in the e-commerce transactions that are based on trust and reputation, excessive ratings are a common one (Hoffman, Zage and Nita-Rotaru, 2007; Yang et al., 2009). Unfair rating is a fundamental problem, as the trust in the e-commerce system is assumed by an entity depending on the ratings made by other entities. The trust, evaluated by an entity, can be misleading, when the ratings, given by the other entities, are incorrect. As false ratings may lead to wrong decisions with the related consequences, it is a necessity to resolve such problems in the e-commerce systems (Zupancic and Trcek, 2017).

Sharing and storing of the network resources in the e-commerce makes it easy for the potential aggressors in this environment to attack the confidentiality, availability, and authenticity of the cloud service. One of the main risks of security in the cloud platform is the Denial-of-Service (DoS) attack, which was ranked fifth in the year 2013 among various threats in the cloud by the cloud security alliance (TTW Group, 2013; Cao et al., 2015). The DoS attack allows the attackers to access the services from the server that are intended for the authenticated users, denying those services to the legitimate clients. It is the common attack, causing considerable losses in the cloud (Brustoloni, 2002). A kind of DoS attack is Distributed DoS (DDoS) attack (Udhayan and Anitha, 2009; Chun-Tao et al., 2012; Specht and Lee, 2004) that slows down the server in responding to the client request. Recently, the intensity of DDoS attacks in cloud security has been dangerously increasing. Usually, this kind of attack is initiated by the attacker from a group of systems, collectively called botnet, with the intention of exhausting the server resources, like Central Processing Unit (CPU), memory, etc. Therefore, the resources provided to normal users may be restricted or even sometimes unavailable. Although various techniques have been developed for the DoS attack mitigation, most of the existing techniques are not employed due to their poor performance (Prasad, Reddy and Rao, 2017). In the recent period, meta-heuristic algorithms (Karlekar and Gomathi, 2018; Ranjan and Prasad, 2018; Thomas and Rangachar, 2016; Menaga and Revathi, 2018) are commonly used for attack detection. Meta-heuristic algorithms can be utilized either to analyze attack database or to optimize and increase the accuracy of the classifiers.

This paper proposes a DoS attack detection technique in e-commerce transactions, using the ECC (Elliptic Curve Cryptography) based authentication with the proposed GSO-SVNN (Glowworm Swarm Optimization-Support Vector Neural Network) classifier. Here, the cloud server and the user are registered initially and then authenticated using various levels of verification. For the verification during authentication, several messages are transferred among the user, server, and the Authorization Center (AC), based on hashing and ECC encryption. Once they are authenticated, the features, such as duration, service, flag, verification status, num_access_files, wrong_fragment, src_bytes, are extracted based on the behaviour of the user from the web log file. With the features extracted as input, the proposed GSO-SVNN, which is developed by integrating the GSO algorithm in the SVNN classifier, authorizes the user. Thus, the proposed GSO-SVNN classifier performs DoS attack detection by generating two classes, namely authenticated user and attacker.

The major contributions of the paper are:

i) Developing a procedure for the registration and authentication, where four different levels of verification are designed using ECC and hashing function with the identity, password, private key and data profile of the three entities, namely the user, the server and the AC.

ii) Introducing a new classifier, named GSO-SVNN, by changing the training procedure of the SVNN using GSO algorithm so that the optimal weights are selected using the algorithm. The proposed GSO-SVNN identifies the attacker by appropriately classifying the user, based on user behaviour.

The structure of the paper is as follows: Section 1 presents the introduction to the cloud and e-commerce transactions and the necessity of attack detection techniques. In Section 2, the literature survey is presented with indication of the drawbacks of the existing techniques, followed by the system model in Section 3. Section 4 elaborates the proposed technique of DoS attack detection with the proposed authentication and authorization procedure. The results are discussed in Section 5, and the conclusion of the paper is given in Section 6.

2. Motivation

2.1. Literature review

In this section, published work, aimed at securing e-commerce transactions against various kinds of attacks, is presented.

Eva Zupancic and Denis Trcek (2017) developed a trust model, named Qualitative Assessment Dynamics Extended (QADE), by considering the various reported trust factors based on the trust's subjective nature. Here, the trust was computed by avoiding the trust values rated by the non-similar agents. The major limitation of this model is that it cannot deal with the time-related issues. An analytical model was developed by F. Al-Haidari, M. Sqalli and K. Salah (2015) to study the effect of Economic DoS (EDoS) attacks in cloud computing. Even though the simulation model was elaborated with consideration of various realistic parameters, it failed to include the pricing models, required for an effective analysis. Jiuxin Cao et al. (2015) presented an attack model involving initiating DoS attacks using malicious entities in the cloud. For the attack detection, a technique was developed based on the utilization of CPU and the network. However, the model developed was only meant for attack detection and still requires a mechanism for attack mitigation. Gaik-Yee Chan and associates (Chan, Lee and Heng, 2014) introduced a predictive model, Fuzzy Associative Rule Model (FARM), to resolve the issues regarding feature selection. FARM evaluated the anomalies based on the associative patterns, but is not suitable for larger data.

K. Munivara Prasad, A. R. Reddy and K. V. Rao (2017) developed a bio-inspired anomaly detection mechanism for real time application to detect DDoS Attack on Web. The mechanism developed utilized Cuckoo search and detected the intrusion behaviours by defining a feature metrics. Although the mechanism offered high detection accuracy, it could not detect other kinds of attacks, like Flash crowd. Guojun Wang et al. (2015) presented a distributed structured approach, called SybilTrust, for the detection of a Sybil attack. This approach utilized the neighbour trust similarity in a P2P e-commerce group to eliminate maliciousness among the peers. The advantage of the SybilTrust is that it provides for a better quality relationships in e-commerce transactions. In case there are more malicious peers, the honest peers are examined more times, and the chance arises that an honest peer is erroneously identified as a Sybil-malicious peer. A technique, named Cyber-Risk Assessment and Mitigation (CRAM) was designed by Mukhopadhyay et al. (2017) to establish the attack probability using Generalized Linear Models (GLMs), and to mitigate the attack probability by determining the required security technology, but the small sized dataset utilized for the experimentation made the training and the testing processes complex. Kamel Karoui (2016) developed a method for the risk assessment using likelihood and impact metrics that could resolve the weighting problem of the risk factors. The limitations of the method are that it cannot mitigate all kinds of attacks, and the time required for processing is long.

N. Hoque, H. Kashyap and D. K. Bhattacharyy (2017) developed a real-time DDoS detection approach, named NaHiD, by utilizing the correlation measure. This approach kept a normal traffic profile and calculated its correlation value with the incoming traffic sample. If the established threshold value is greater than the calculated correlation value then the attack alarm is generated. In the essence, this approach considered DDoS detection as a 2-class problem. Sahoo et al. (2018) introduced a Generalized Entropy (GE) based metric for detecting the low rate DDoS attacks. This metric utilized the flow based nature of the Software Defined Networks (SDN). The detection accuracy of this approach is high when compared to Shannon entropy. This approach, however, is not suitable for high rate DDoS attacks. Yin, Zhang and Yang (2018) introduced a framework for the software-defined Internet of Things (SD-IoT), based on the software-defined anything (SDx) paradigm. By using this framework, they introduced an algorithm for DDoS attacks detection and mitigation. This algorithm had good performance, and the SD-IoT framework strengthens the security of the IoT with various devices.

2.2. Challenges

The essential challenges, observed in the techniques developed in the literature, can be formulated are as follows:

- Many of the proposed approaches that focussed on preventing Sybil attacks in the e-commerce transactions in social and trusted networks did not perform, regarding the intended task, effectively (Wang et al., 2015).
- The CRAM framework, which was developed in Mukhopadhyay et al. (2017), has the following issues: as the Computer Security Institute–Federal Bureau of Investigation (CSI–FBI) dataset utilized in the work is small, it is difficult to perform training and testing in an optimal way. Moreover, the experiments with the framework were carried out with the assumption that all the cyber-attacks are independent, which is not true in the real cases.
- Employing security mechanisms in a network without studying their usefulness and impact on the ability of the network to mitigate attacks is inappropriate and may produce inadequate effects, causing even network performance degradation (Karoui, 2016).
- Most of the existing approaches are little efficient and are even not suitable for detecting DDoS attacks, since they require performing some basic computations between the user and the server, based on the user requests (Prasad, Reddy and Rao, 2017).
- Another challenge noticed appears in the QADE model of Zupancic and Trcek (2017), which cannot deal with real time issues, as the model fails to classify previous and current trust assessments.
- The FARM approach, developed by Chan, Lee and Heng (2014) is not appropriate for real-time e-commerce transactions in terms of differentiating between the normal user and the attacker, due to the tremendous increase in the size of the datasets.

The here proposed technique can overcome these challenges by differentiating the user from the attacker using the newly developed classifier and is applicable in real-time e-commerce applications.

3. System model

This section explains the system model for the DoS attack detection in the cloud during the e-commerce transactions. As depicted in Fig. 1, the e-commerce web server offers the cloud resources to the users, performing e-commerce transactions, for the user request. Let the users involved in the e-commerce transaction be represented as $U = \{U_1, U_2, \ldots, U_m\}$, where m is the total number of users, who are accessed by the n servers, given as $R = \{R_1, R_2, \ldots, R_n\}$. Before allowing the users to access the resources, the user and the server are registered and authenticated on the basis of various security parameters and verification levels, using the proposed authentication technique. This helps in e-commerce applications, where the user's information, like session password, number of cache access, etc., is stored in the authentication phase during the payment of the items purchased via credit card. The AC authorizes the user and offers the service required. Thus, an attacker, who tries to access the information, can be detected due to its unethical use of sensitive data from the cloud. Since DoS attack is a common risk-generating cyber attack in the e-commerce organizations, this work concentrates on detecting such attacks in the e-commerce transactions.

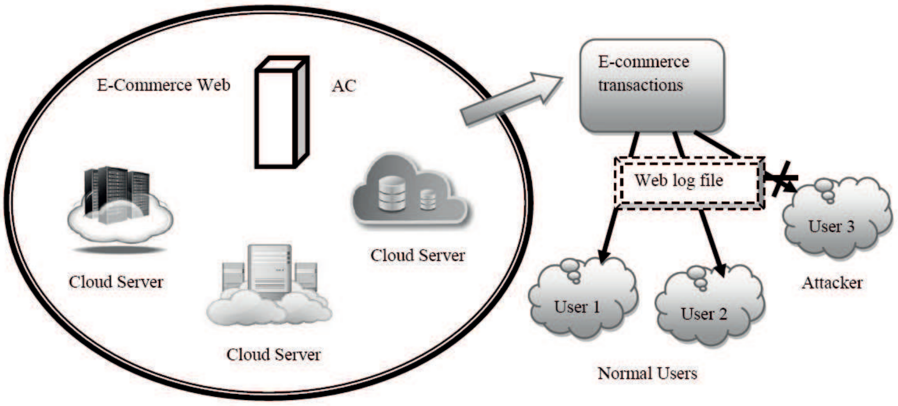


Figure 1. System model of IDS for e-commerce transactions in the cloud

4. The proposed DoS attack detection technique in the e-commerce transactions using ECC based authentication and Support Vector Neural Network

The proposed technique of DoS attack detection is based on the mutual authentication between the user and the server, using Elliptic Curve Cryptography (ECC) and the proposed GSO-SVNN classifier. The technique involves authentication and authorization mechanisms to detect the DoS during the e-commerce transactions. The authentication process includes various verification levels using ECC and hashing function, carried out after the registration of the user and the server. Then, the authorization is done by extracting the useful information, like duration, service, flag, verification status, num_access_files, wrong_fragment, src_bytes, from the web log file. Based on the features extracted, authorization is performed using the proposed GSO-SVNN, which is designed by modifying the training algorithm utilized in the SVNN. The here proposed GSO-SVNN classifier finds whether the user is an authenticated one or an attacker and thereby, detects DoS attacks in the e-commerce organizations. The block diagram of the proposed DoS attack detection technique is presented in Fig. 2.

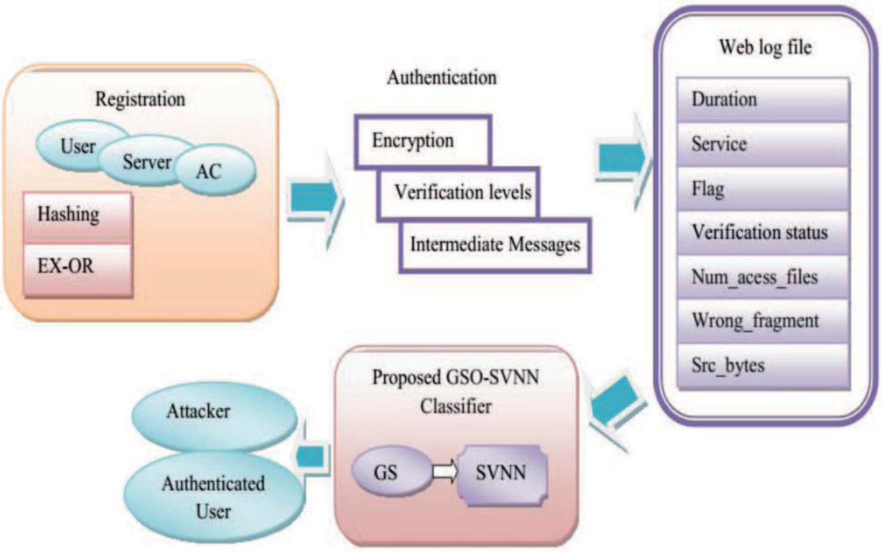


Figure 2. Block diagram of the DoS attack detection technique using the proposed GSO-SVNN classifier

The variables, which are utilized for the authentication are listed with their description in Table 1.

Table 1. Symbols and their descriptions

Symbols	**Description**
I^U	Identity of user
I^S	Identity of server
I^A	Identity of AC
K^U	The private key of the user
K^S	The private key of the server
K^A	The private key of the AC
K^P	Public key
P^U	Password of the user
P^S	Password of the server
P^A	Password of the AC
D^U	Data profile of the user
D^S	Data profile of the server
D^A	Data profile of the AC
I^U_+	Stored I^U
I^S_+	Stored I^S
I^A_+	Stored I^A
K^U_+	Stored K^U
K^S_+	Stored K^S
K^A_+	Stored K^A
P^U_+	Stored P^U
P^S_+	Stored P^S
P^A_+	Stored P^A
D^U_+	Stored D^U
D^S_+	Stored D^S
D^A_+	Stored D^A
V_j	Verification level
$E_{K^U_+}$	ECC based encryption using the key K^U_+
E_{K^U}	ECC based encryption using the key K^U
E_{K^S}	ECC based encryption using the key K^S

4.1. Registration of the user and the server for the e-commerce transactions

The registration procedure designed in the proposed technique follows the ECC approach. Before the authentication of the user, it is mandatory that the user and the server have to be registered. The proposed DoS attack detection in the e-commerce transactions includes three entities, namely the user, the server, and the AC. The registration between the user and the server is done based on the hashing operation and the EX-OR operation with the password, the private keys, and the identity of the three entities. The procedure involved in the registration is given as follows:

i) Initially, the server has its identity, I^S and password, P^S, which are stored in the AC as I^S_+ and P^S_+, to register in the cloud for the transaction.

ii) Then, the private key of the server is computed in the AC by taking the EX-OR of P^S_+ and the private key of the AC, and the resulting value is concatenated with I^S_+, as given below,

$$K^S = h\left(I^S_+ || \left(P^S_+ \oplus K^A\right)\right), \tag{1}$$

where P^S_+ is the stored password of the server, $\oplus$ is the EX-OR operation, and K^A is the private key of the AC.

iii) Using the computed K^S, K^A and the public key, the AC finds the data profile of the server as

$$D^S = \left(h\left(K^S \oplus K^A\right)\Theta K^P\right). \tag{2}$$

where K^P is the public key, $h(.)$ is the hashing function, and Θ is the element-wise multiplication.

iv) The private key and the data profile of the server, computed in the AC, are transferred to the server, where it stores K^S and D^S as K^S_+ and D^S_+. Also, the user identity, I^U, and the password, P^U, that the user holds, are stored in the server as I^U_+ and P^U_+.

v) Then, the server computes the user private key by hashing the I^U_+ that is concatenated with the EX-ORed P^U_+ and K^S:

$$K^U = h\left(I^U_+ || \left(P^U_+ \oplus K^S,\right)\right) \tag{3}$$

where I^U_+ and P^U_+ are the stored user identity and password, and K^S is the private key of the server.

vi) Moreover, the server computes the data profile of the user, similar to that of the server, as

$$D^U = \left(h\left(K^U \oplus K^S\right)\Theta K^P\right). \tag{4}$$

vii) Finally, K^U and D^U, calculated by the server, are received by the user and are stored as K^U_+ and D^U_+.

Thus, the user and the server in the cloud are registered and are subject to the authentication, for the purposes of DoS attack detection.

4.2. Authentication using ECC-based encryption

Following the registration of the user and the server, authentication is performed using ECC. ECC (see Seroussi, 1999; Hankerson, Menezes and Vanstone, 2006) is one of the methods based on public key encryption that encrypts the data with the keys using the concept of the elliptic curve. When compared to other encryption methods, the ECC has several advantages, such as very fast key generation, fast encryption, and decryption. Also, ECC employs a comparatively short encryption key, which requires less computing power than other first-generation encryption public key algorithms. The reduced size of the key and the high security offered make ECC a promising approach for the encryption. In ECC, an elliptic curve, often a plane curve, is chosen, and the key generation is done based on the points mapped on the curve. Then, the key is exchanged following a Diffie-Hellman key exchange procedure and finally, the encryption is performed. The proposed DoS attack detection technique performs several levels of verification by forwarding various intermediate messages, represented as S. Here, four messages and four verification levels are considered for the authentication. At first, the user concatenates I^U and P^U and encrypts the result using the stored private key of the user. The encrypted result is again concatenated with the user identity to create the message S_1:

$$S_1 = \left(E_{K_+^U}\left(I^U||P^U\right)||I^U\right), \tag{5}$$

where $E_{K_+^U}$ represents the encryption using ECC with K_+^U. The message generated by the user is verified at the server using the stored I^U and P^U, as

$$S_1^* = \left(E_{K^U}\left(I^U||P_+^U\right)||I_+^U\right), \tag{6}$$

where E_{K^U} is the encryption using ECC with K^U. If $S_1 = S_1^*$, it is considered to be the first level of verification, V_1. Then, the server constructs another intermediate message by encrypting the server identity and the password as follows,

$$S_2 = \left(E_{K_+^S}\left(I^S||P^S||I^S\right)\right), \tag{7}$$

where $E_{K_+^S}$ is the encryption using ECC with the stored private key of the server K_+^S.

The same message is verified at the AC using the stored key and identity of the server as

$$S_2^* = E_{K^S}\left(I^S||P_+^S\right)||P_+^S, \tag{8}$$

where E_{K^S} is the ECC encryption using K^S. If $S_2 = S_2^*$, it is the level 2 verification, V_2. For the third level of verification, a message is created in the AC by EX-ORing the hashed public key and the data profile of the server, as

$$S_3 = D^S \oplus h\left(K^P\right), \tag{9}$$

where D^S and $h\left(K^P\right)$ indicate the data profile of the server and the hashing function of the public key. Then, the message is verified at the server using the stored D^S as

$$S_3^* = D_+^S \oplus h\left(K^P\right), \tag{10}$$

where D_+^S is the stored data profile of the server. Thus, the level 3 verification, V_3, is complete if $S_3 = S_3^*$. The fourth message is generated at the server, similar to S_3, but with the data profile of the user, instead of D_+^S,

$$S_4 = D^U \oplus h\left(K^P\right), \tag{11}$$

where D^U is the data profile of the user. The user, with the message S_4, finds the data profile of the user as

$$D_*^U = h\left(K^P\right) \oplus S_4. \tag{12}$$

If the condition $D_*^U = D_+^U$ is satisfied, it is the level 4 verification, V_4. Thus, with the four levels of verification, the user can be authenticated when the verification condition is satisfied. If not, the user is meant to be an attacker.

4.3. Extracting user behavior

For the authorization, the user behavior characteristics, like duration, service, flag, verification status, num_access_files, wrong_fragment, and src_bytes, are recorded during the e-commerce transactions. These details, stored in the web log file, are considered as the features and are given as the input to the proposed classifier for the attack detection. Hence, the feature vector is represented as

$$u = \{u_1, u_2, \ldots, u_f, \ldots, u_7\}, f = 1, \ldots, 7, \tag{13}$$

where u_1 represents the duration of connection, u_2 is the service information, u_3 represents the flag, i.e. the connection status, u_4 indicates the verification status of the user, u_5 denotes the functions on access control files, u_6 is the number of fragments that are sent wrongly, and u_7 is the number of bytes in the packet transferred from the source to the destination.

4.4. Authorization using the proposed GSO-SVNN classifier

In this section, the proposed GSO-SVNN, developed for the authorization of the user and the server in the e-commerce web, is elaborated. SVNN (Ludwig, Nunes and Araujo, 2014) is composed of three layers, namely the input layer, the hidden layer, and the output layer. On the basis of the information obtained during the authentication phase, the SVNN classifies the user either as an intruder or an authenticated user. Hence, the input fed to the input layer of the SVNN is the user information, like session password and number of cache access. The proposed GSO-SVNN is developed by training the SVNN with the GSO (Kaipa and Ghose, 2017), in such a way that the performance of the classifier is improved due to the capacity of GSO of solving problems characterized by multimodal functions. The output of the proposed GSO-SVNN is given by the function

$$O = w_1 \cdot \log sig\left[\left(\sum_{f=1}^{7} u_f * w_2^l\right) + b_1\right] + b_2, \tag{14}$$

where u_f is the index variable of features, w_1 and w_2 indicate, respectively, the weights applied between the hidden and the output layers and between the input and the hidden layers, and b_1 and b_2 represent the biases at the hidden and the output layers, respectively. The architecture of the proposed GSO-SVNN is shown in Fig. 3.

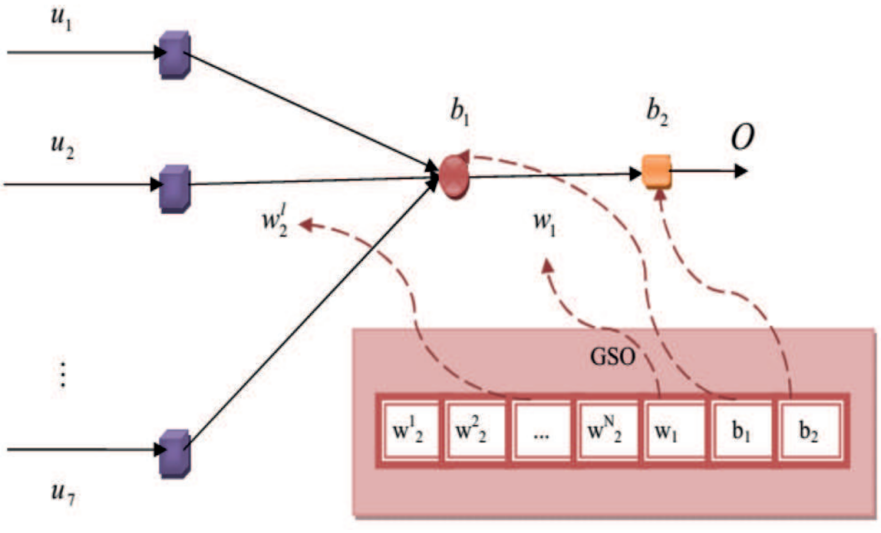


Figure 3. The architecture of the proposed GSO-SVNN

GSO-based training

GSO (Kaipa and Ghose, 2017) is an optimization scheme, which is designed on the basis of behaviour of glowworms following certain mechanisms, namely fitness evaluation, positive taxis, and neighbourhood update. The fitness of the actual glowworms or the agents is determined on the basis of a luminescent pigment, known as luciferin. However, in the proposed GSO-SVNN, the fitness is evaluated in terms of the error function. The positive taxis indicates the movement update, and in the neighbourhood update, each glowworm adjusts its neighbourhood for updating its position. The GSO algorithm has numerous advantages: It works with multi-modal and highly non-linear optimization problems. It does not utilize velocities, hence there are no issues associated with velocity. Also, the convergence speed of GSO is very high. Following are the algorithmic steps of the GSO algorithm:

i) Initialization: The primary step is the random initialization of the swarms in the search space, as given below:

$$G = \{G_1, G_2, \ldots, G_a, \ldots, G_M\} \tag{15}$$

where M is the number of solutions in the swarm.

ii) Finding the fitness of the solution: The fitness of each solution in the space is computed in the second step. As the fitness of the classifier is an error function, the fitness is considered as a function to be minimized, given as:

$$F = \xi_{\max} + \xi_{\min} + \frac{g}{2} \sum_{i=1}^{2} (O_i - O_i^*), \tag{16}$$

where $\xi = Eigen\left(w \times w^T\right)$

$$\xi_{\max} = Max\,(\xi) \tag{17}$$

$$\xi_{\min} = Min\,(\xi) \tag{18}$$

and where g is the regularization factor, O_i and O_i^* represent the output of the GSO-SVNN classifier and the predicted output, respectively.

iii) Update Process: The third step is the update phase, where the luciferin, the movement, and the neighbourhood range are updated.

Luciferin Update: For the luciferin update, the fitness of the solution at the iteration x+1 is utilized along with the luciferin estimated at the previous iteration, as

$$c_a\,(x+1) = (1-\rho)\,c_a\,(x) + \lambda F\,(G_a\,(x+1))\,, \tag{19}$$

where $c_a(x)$ is the luciferin level at iteration x, ρ is a constant with a value between 0 and 1, denoting the luciferin decay, λ is another constant for the luciferin enhancement and $F(G_a(x{+}1))$ is the fitness of the solution at x+1.

Movement Update: Then, the movement of the a^{th} glowworm is updated on the basis of the brightness of its neighbour, denoted b, according to a probability value, given as:

$$p_{ab}\,(x) = \frac{c_b\,(x) - c_a\,(x)}{\sum\limits_{d \in H_a(x)} c_d\,(x) - c_a\,(x)}, \tag{20}$$

where $H_a(x)$ is the neighbourhood of the a^{th} glowworm, such that $b \in H_a(x)$, and $c_a(x)$ and $c_b(x)$ are the luciferin levels of the a^{th} and b^{th} glowworms, respectively, at the current iteration. Thus, the movement update is given by:

$$G_a\,(x+1) = G_a\,(x) + v\left(\frac{G_b\,(x) - G_a\,(x)}{\|G_b\,(x) - G_a\,(x)\|}\right), \tag{21}$$

where $G_a(x)$ is the position of the a^{th} glowworm at iteration x, $G_b(x)$ is the position of the b^{th} glowworm at iteration x, v is the step size, and $\|.\|$ is the Euclidean distance between the glowworms.

Neighborhood Update: As each glowworm is correlated with a neighbourhood, it is necessary to update it adaptively

$$h_d^a(x+1) = \min\{h_e, \max\{0, h_d^a(x) + \delta(y_x - |H_a(x)|)\}\}, \tag{22}$$

where δ is a constant, y_x is the neighbour control parameter, and $0 < h_d^a \leq h_e$.

iv) Fitness Evaluation: The best solution is found on the basis of the fitness of the solution and the solution having the minimum error is chosen as the best solution.

v) Termination: The algorithm is terminated when the maximum iteration criterion is met. Let T be the maximum iteration number, and thus, before reaching the condition, for $x<T$, the solution with minimum error will be determined.

Thus, the GSO algorithm, used for training the SVNN, selects the weights and the biases optimally in such a way that the proposed GSO-SVNN finds to which class, i.e., attacker or normal user, the user belongs.

5. Results and discussion

This section deals with the results of the proposed method. The performance of the proposed method is evaluated with four setups in terms of four metrics, namely Accuracy, Precision, Recall, and FPR.

5.1. Experimental setup

The proposed method is implemented in MATLAB tool using the PC equipped with Windows 10 OS and Intel(R) i3 processor with 64-bit operating system and 4GB RAM. The experiment is done in four different simulation setups meant for collecting the data through simulation, given as follows:

Setup 1: The simulation setup with 50 users and 5 attackers.

Setup 2: The simulation setup with 75 users and 5 attackers.

Setup 3: The simulation setup with 100 users and 10 attackers.

Setup 4: The simulation setup with 150 users and 15 attackers.

5.1.1. Evaluation metrics

Accuracy: This notion refers to the degree, to which the result of a measurement, calculation, or specification conforms to the correct value or a standard:

$$ACC = \frac{T^P + T^N}{T^P + T^N + F^P + F^N}, \tag{23}$$

where T^P, T^N, F^P, and F^N represent the numbers of occurrences of the true positive, true negative, false positive, and false negative cases.

Precision: This notion refers to the fraction of relevant instances among all of the retrieved instances:

$$P = \frac{T^P}{T^P + F^P}. \tag{24}$$

Recall: The fraction of relevant instances that have been retrieved in the total number of relevant instances is the recall:

$$RE = \frac{T^P}{T^P + F^N}. \tag{25}$$

FPR: FPR is a measure that indicates that a given condition is considered to be present, when it actually is not:

$$FPR = \frac{F^P}{F^P + T^N}. \tag{26}$$

5.1.2. The techniques compared

The techniques, taken for the comparison of performance with the proposed technique are: Neighbor Similarity Trust (Wang et al., 2015), QADE (Zupancic and Trcek, 2017), ECC+SVM (with application of SVM instead of GSO-SVNN in the proposed technique), BARTD (Prasad, Reddy and Rao, 2017), ECC+SVNN (with application of SVNN instead of GSO-SVNN in the proposed technique).

5.2. Comparative analysis

Here, the comparative analysis of the proposed ECC+GSO-SVNN with the existing techniques, mentioned above: Neighbor Similarity Trust, QADE, ECC+ SVM, BARTD, ECC+SVNN, is reported, performed by varying the training percentages of data. The analysis was carried out using the four setups, also previously mentioned. The analysis of the models for each setup is described below:

5.2.1. Comparative analysis for Setup 1

In this section, the results, obtained for the simulation setup with 50 users and 5 attackers for the accuracy, precision, recall, and FPR of the attacker identification, are reported.

Figure 4a illustrates the accuracy metric values, obtained for the varying training datasets, denoted by k.

Thus, for the 80% training data, the corresponding accuracy values, obtained for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN, ECC +GSO-SVNN, are, respectively, 0.78, 0.816, 0.828, 0.832, 0.912 and 0.940, whereas for the 70% training data, the respective accuracy values are 0.742, 0.791, 0.8195, 0.831, 0.895 and 0.928. Hence, the proposed method predicts the result with the highest accuracy.

Figure 4b shows the precision values based on training data proportions varying from 0.3 to 0.8. For the data proportion of 80%, the precision measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN is, respectively, 0.899, 0.899, 0.911, 0.912, 0.969 and 0.978.

Next, Figure 4c shows the recall values calculated for Neighbor Similarity Trust, QADE, ECC+SVM, BAR TD, ECC+SVNN and the proposed ECC+ GSO-SVNN. For 80% data, the recall values measured are 0.837, 0.873, 0.906, 0.910, 0.93 and 0.952, respectively.

Finally, Fig.4d shows the FPR values, obtained for the Neighbor Similarity Trust, QADE, ECC+SVM, BAR TD, ECC+SVNN and the proposed ECC+ GSO-SVNN. The FPR values for Neighbor Similarity Trust, QADE, ECC+SVM, BAR TD, ECC+SVNN and the proposed ECC+ GSO-SVNN are 0.4, 0.4, 0.4, 0.315, 0.05 and 0.05, respectively, for the proportion of $k = 0.8$.

It can be concluded, therefore, that the proposed method shows generally better results than the existing methods used in the comparison.

5.2.2. Comparative analysis for Setup 2

In this section, the setup with 75 users and 5 attackers is taken into consideration by calculating the accuracy, precision, recall, and FPR of identification of the attack situations.

Thus, Fig. 5a illustrates the accuracy metric for the varying training dataset, denoted by k. For the 80% training data, the corresponding accuracy, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN, ECC +GSO-SVNN is equal, respectively, 0.6, 0.843, 0.866, 0.873, 0.929 and 0.951, whereas for the 70% training data, the accuracy measured for these methods is equal, respectively, 0.6, 0.842, 0.865, 0.873, 0.880 and 0.918. Hence, the proposed method predicts the proper result with the highest accuracy.

Figure 5b shows the precision values, obtained for the training data varying from 0.3 to 0.8. When the data is 80%, the precision values, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BAR TD, ECC+SVNN and the proposed ECC+ GSO-SVNN are 0.6, 0.928, 0.933, 0.934, 0.975 and 0.982.

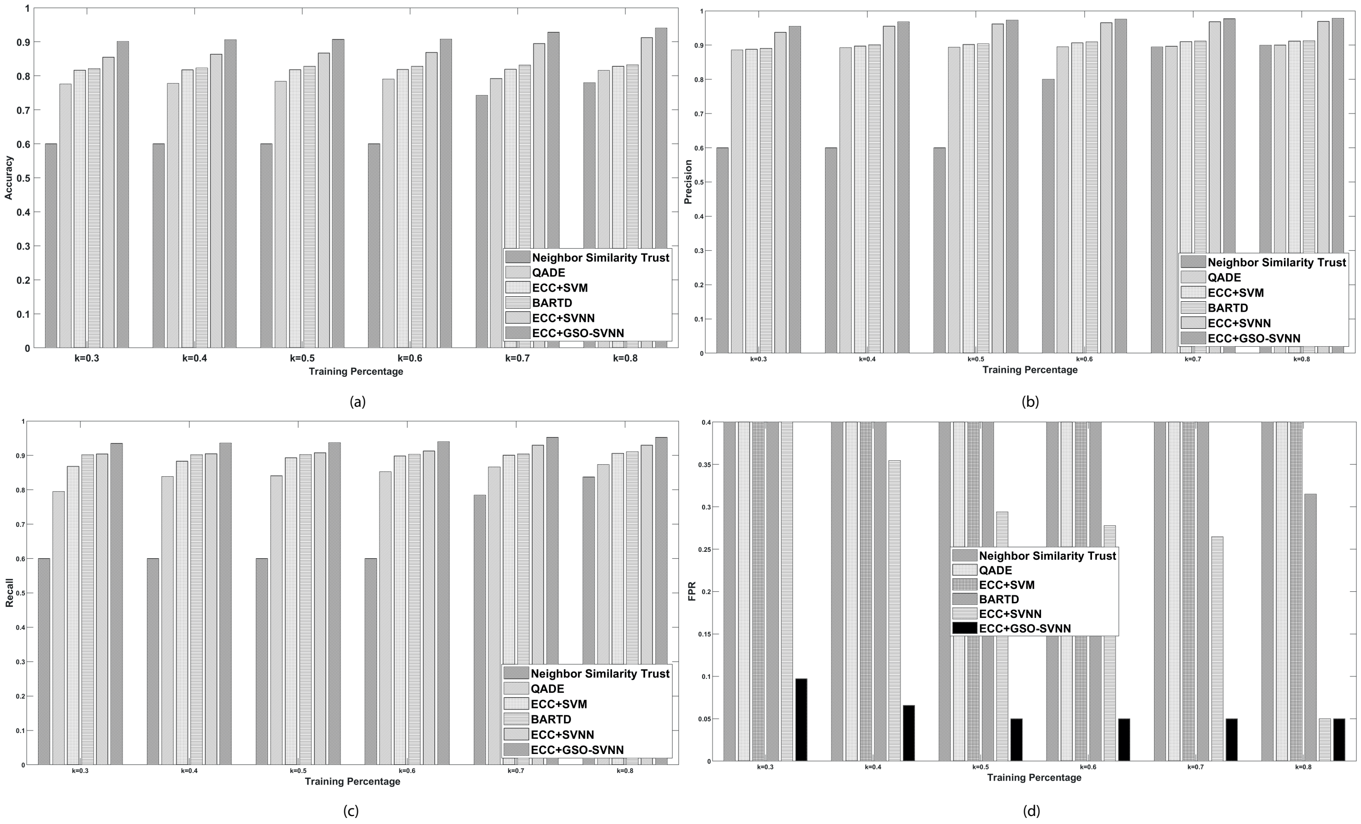
Figure 5c shows the recall values, calculated for the techniques considered. For the 80% data, the recall values, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN are 0.6, 0.910, 0.915, 0.940, 0.947 and 0.962, respectively.

Finally, Fig. 5d shows the FPR values, calculated for the results, obtained with Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN. The FPR values, calculated for these techniques, are 0.4, 0.4, 0.4, 0.4, 0.2875, and 0.05, respectively, for the proportion of $k = 0.8$.

5.2.3. Comparative analysis for Setup 3

This section shows the results from the comparative analysis for the setup with 100 users and 10 attackers, presenting the values of accuracy, precision, recall, and FPR, regarding the identification of the attack stuations.

Thus, Fig. 6a illustrates the values of the accuracy metric for the varying training dataset proportions, denoted by k. For the 80% training data,



(a) (b) (c) (d)

Figure 4. Analysis for Setup 1 based on a) accuracy b) precision c) recall, and d) FPR

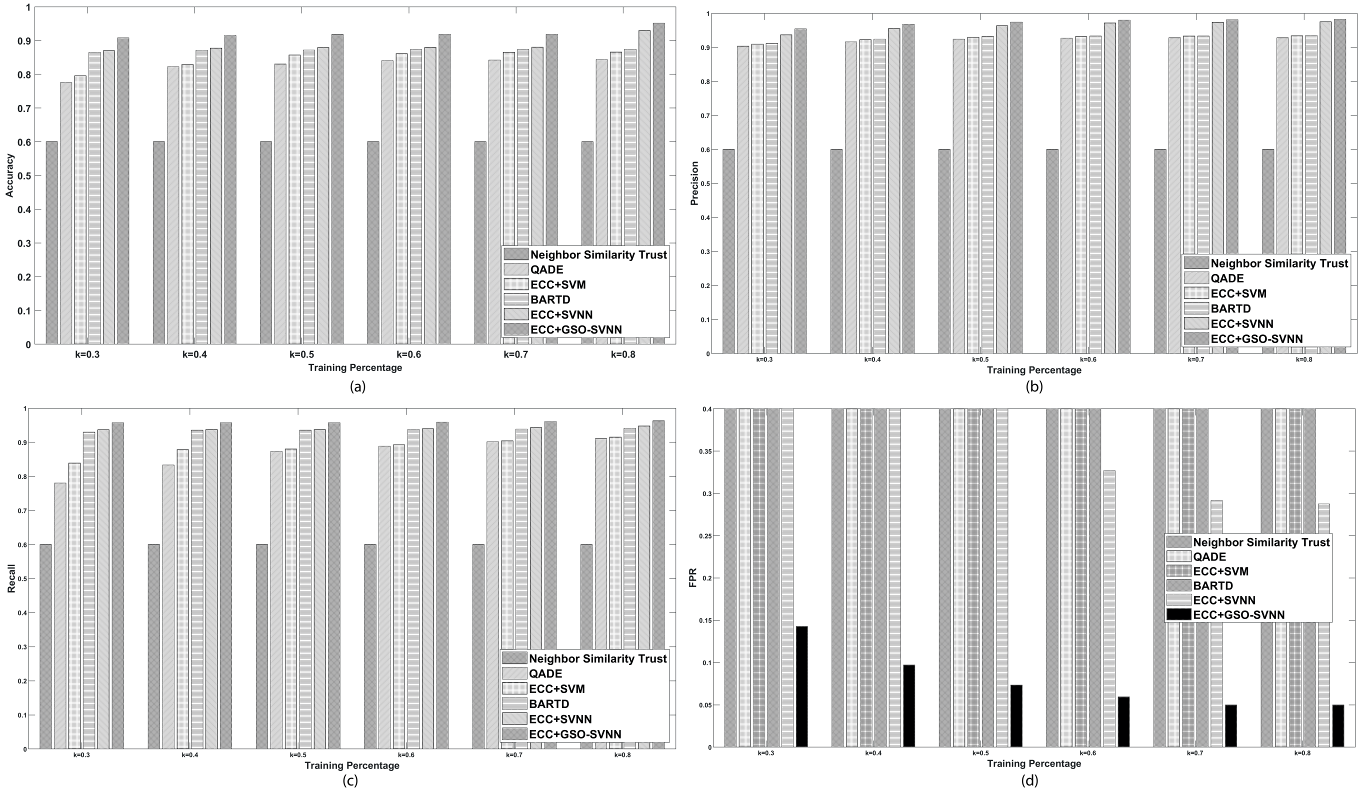


Figure 5. Analysis for Setup 2 based on a) accuracy b) precision c) recall, and d) FPR

the corresponding accuracy, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN is, respectively, 0.784, 0.818, 0.827, 0.871, 0.885 and 0.921, whereas for 60% training data, the accuracy, calculated for these techniques, is 0.769, 0.814, 0.824, 0.828, 0.882, 0.919, respectively. Hence, the proposed method predicts the actual result with the highest accuracy.

Figure 6b shows the precision values, calculated for the training data proportions varying from 0.3 to 0.8. When the data is 80%, the precision values, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN method are, respectively, 0.889, 0.893, 0.894, 0.894, 0.955, and 0.969.

Then, Fig. 6c shows the recall values, calculated for the compared techniques. For the 80% data, the recall values measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN are, respectively, 0.864, 0.896, 0.915, 0.924, 0.93 and 0.952.

Finally, Figure 6d shows the results in terms of the FPR indicator, where the minimum FPR, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN is 0.4, 0.4, 0.4, 0.315, 0.05 and 0.05, respectively.

It can, again, be concluded that the proposed method shows better results than the other existing methods, used in the comparison.

5.2.4. Comparative analysis for Setup 4

In this section, the results of the comparative analysis based on accuracy, precision, recall, and FPR for the setup with 150 users and 15 attackers, are shown.

And so, Fig. 7a illustrates the values of the accuracy metric, obtained with the varying training dataset proportions, denoted by k. For the 80% training data, the corresponding accuracy, measured for Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN is 0.764, 0.798, 0.8384, 0.857, 0.882 and 0.917, respectively, whereas for the 60% training data, these values are, respectively, 0.6, 0.777, 0.82, 0.838, 0.874 and 0.913. Hence, the here proposed method predicts the actual result with the highest accuracy.

Figure 7b shows the precision values, obtained, again for the training data proportions varying from 0.3 to 0.8. For the 80% data, the precision measured for the Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN is 0.846, 0.887, 0.888, 0.907, 0.965 and 0.976, respectively.

Figure 7c presents the recall values, calculated for the Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN. For the 80% data, the recall values, calculated for these techniques, are 0.877, 0.897, 0.920, 0.921, 0.93 and 0.952, respectively.

Finally, Fig 7d shows the comparison of the FPR values, obtained for the compared techniques, that is, Neighbor Similarity Trust, QADE, ECC+SVM,

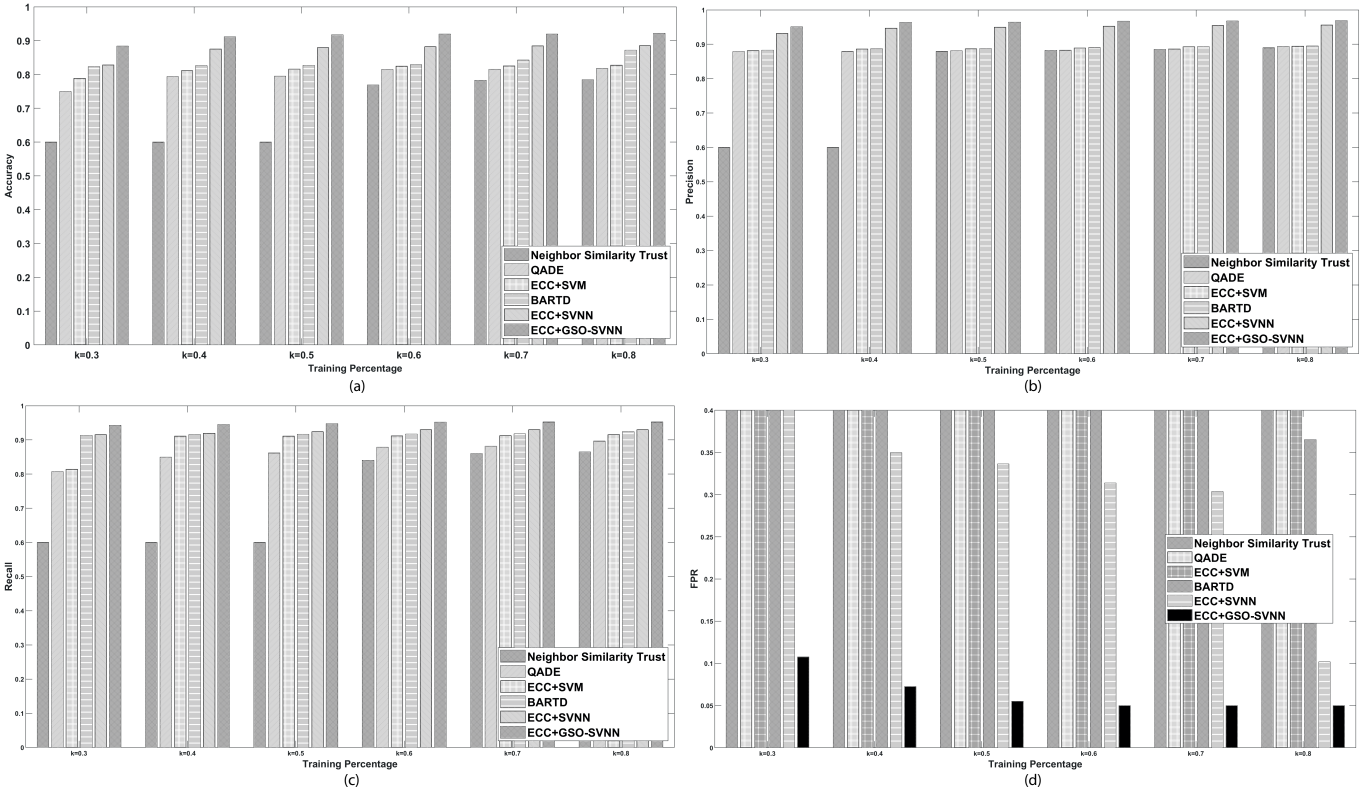


Figure 6. Analysis of Setup 3 based on a) accuracy b) precision c) recall, and d) FPR

BARTD, ECC+SVNN and the here proposed ECC+ GSO-SVNN. The FPR values, calculated for the Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+SVNN and the proposed ECC+ GSO-SVNN, are 0.4, 0.4, 0.4, 0.213, 0.05 and 0.05, respectively.

5.3. Discussion

This section sums up the results of comparative analysis of the considered techniques, i.e. Neighbor Similarity Trust, QADE, ECC+SVM, BARTD, ECC+ SVNN and the here proposed ECC+ GSO-SVN. The analysis refers to the overall performance of the methods in terms of accuracy, precision, recall and FPR. Table 2 provides the summary results of the comparative analysis of the existing and proposed methods. The results, obtained with the proposed ECC+GSO-SVNN in terms of accuracy, precision, recall and FPR are 0.951, 0.982, 0.962, and 0.05, and are in all these metrics better than those for any of the remaining techniques compared.

Table 2. Comparative results

Methods	***Accuracy***	***Precision***	***Recall***	***FPR***
Neighbor Similarity Trust	0.784	0.899	0.877	0.4
QADE	0.843	0.928	0.910	0.4
ECC+SVM	0.866	0.933	0.920	0.4
BARTD	0.873	0.934	0.940	0.4
ECC+SVNN	0.929	0.975	0.947	0.287
Proposed ECC+GSO-SVNN	**0.951**	**0.982**	**0.962**	**0.05**

6. Conclusion

In this paper, a technique is described, meant for DoS attack detection in e-commerce transactions, based on the proposed GSO-SVNN classifier. The proposed classifier is designed by modifying the SVNN using GSO algorithm in such a way that the respective weights are selected optimally. Initially, the user and the server are registered in the cloud for the authentication, which is designed using ECC encryption and hashing operations with four different verification levels and messages. Then, the user behaviour aspects, such as duration, service, flag, verification status, num_access_files, wrong_fragment, src_bytes, are extracted for the authorization. The proposed GSO-SVNN classifies the user as an authenticated user or an attacker based on the information extracted. The verifying experimentation is carried out using four metrics, namely accuracy, precision, recall, and FPR, in order to evaluate the performance of the proposed technique. The results suggest that the proposed technique of DoS

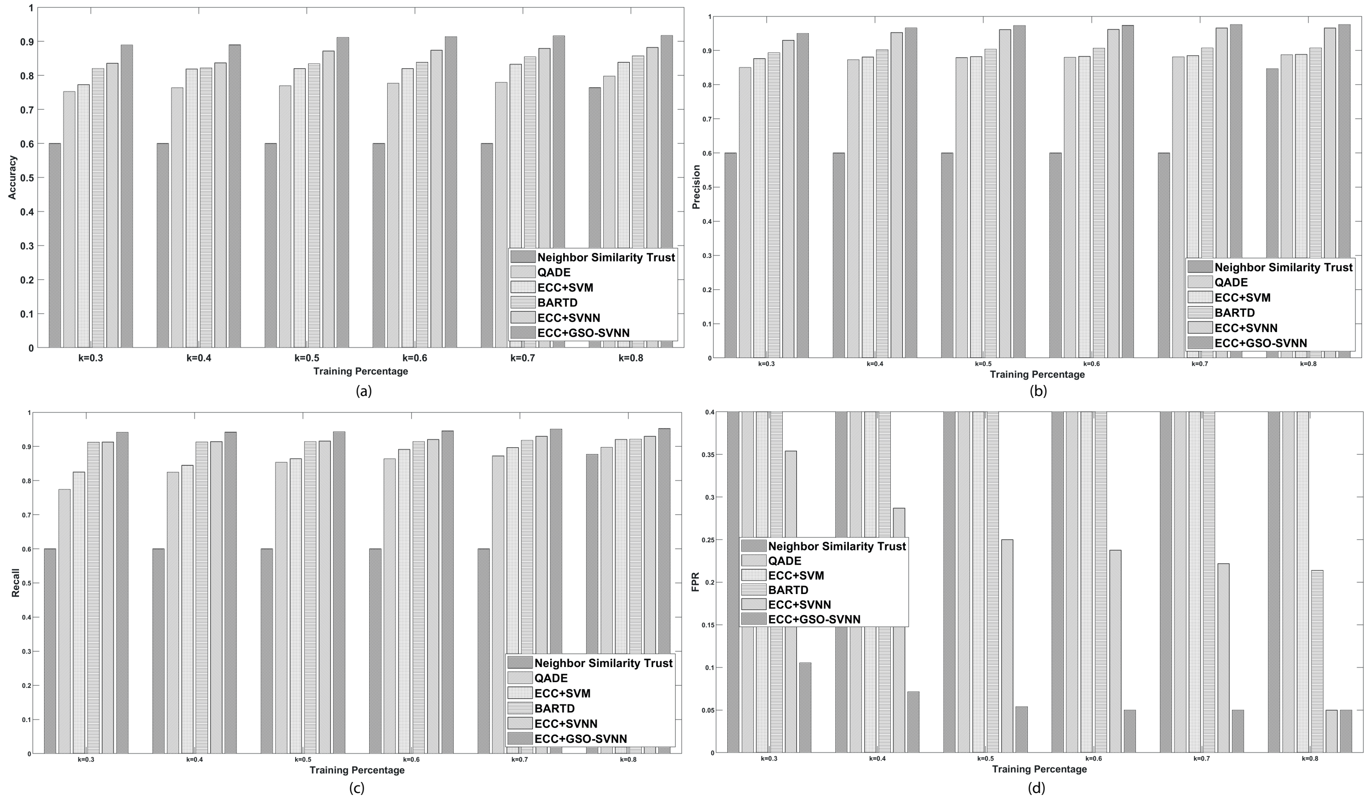


Figure 7. Analysis of Setup 4 based on a) accuracy b) precision c) recall, and d) FPR

attack detection is effective in attack detection in e-commerce transactions with 95.1% accuracy.

This work will be extended to perform attack mitigation by proposing an effective mechanism in the future.

References

Al-Haidari, F., Sqalli, M and Salah, K. (2015) Evaluation of the impact of EDoS attacks against cloud computing services. *Arabian Journal for Science and Engineering,* **40**(3) 773-785.

Brustoloni, J. (2002) Protecting electronic commerce from distributed denial-of-service attacks. In: *Proceedings of the 11th ACM international conference on World Wide Web,* ACM Digital Library, 553-561.

Cao, J., Yu, B., Dong, F., Zhu, X. and Xu, S. (2015) Entropy-based denial-of-service attack detection in cloud data center. *Concurrency and Computation: Practice and Experience,* **27**(18) 5623-5639.

Chan, G., Lee, C. and Heng, S. (2014) Defending against XML-related attacks in e-commerce applications with predictive fuzzy associative rules. *Applied Soft Computing,* 24, 142-157.

Chen, Y., Paxson, V. and Katz, R.H. (2010) What's new about cloud computing security. *University of California*, *Berkeley Report No.* UCB/ EECS-2010-5.

Chun-Tao, X., Xue-Hui, D., Li-Feng and Hua-Cheng, C. (2012) An algorithm of detecting and defending CC attack in real time. In: *Proceedings of 2012 IEEE International Conference on Industrial Control and Electronics Engineering (ICICEE),* IEEE, 1804-1806.

Gomez-Herrera, E., Martens, B and Turlea, G. (2014) The drivers and impediments for cross-border e-commerce in the EU. *Information Economics and Policy,* 28, 83-96.

Hankerson, D., Menezes, A.J. and Vanstone, S. (2006) *Guide to Elliptic Curve Cryptography.* Springer Science & Business Media.

Hoffman, K., Zage, D. and Nita-Rotaru, C. (2007) A Survey of attacks on Reputation Systems. *Computer Science Technical Reports.* Report No. 07-013, Purdue University, 1-17.

Hoque, N., Kashyap, H. and Bhattacharyy, D.K. (2017) Real-time DDoS attack detection using FPGA. *Computer Communications,* 110, 48-58.

Josang, A., Ismail, R. and Boyd, C. (2007) A survey of trust and reputation systems for online service provision. *Decision Support Systems,* **43**(2), 618-644.

Kaipa, K.N. and Ghose, D. (2017) *Glowworm Swarm Optimization: Theory, Algorithms and Applications. Studies in Computational Intelligence* **698**, Springer Verlag.

Karlekar N.P. and Gomathi, N. (2018) OW-SVM: Ontology and whale optimization-based support vector machine for privacy-preserved medical data classification in cloud. *International Journal for Communication Systems.* **31**(12), 1–18.

Karoui, K. (2016) Security novel risk assessment framework based on reversible metrics: a case study of DDoS attacks on an E-commerce web server. *International Journal of Network Management,* **26**(6), 553-578.

Lucking-Reiley, D., Bryan, D., Prasad, N. and Reeves, D. (2007) Pennies from eBay: The determinants of price in online auctions. *The Journal of Industrial Economics,* **55**(2), 223-233.

Ludwig, O., Nunes, U. and Araujo, R. (2014) Eigen value decay: A new method for neural network regularization. *Neurocomputing,* 124, 33–42.

Menaga, D. and Revathi, S. (2018) Least Lion Optimization algorithm (LLOA) Based Secret key Generation for Privacy Preserving Association Rule Hiding. *IET Information Security* **12**(4), 1-9.

Mukhopadhyay, A., Chatterjee, S., Bagchi, K.K., Kirs, P.J. and Shukla, G.K. (2017) Cyber Risk Assessment and Mitigation (CRAM) Framework Using Logit and Probit Models for Cyber Insurance. *Information Systems Frontiers,* 1-22.

Pinyol, I. and Sabater-Mir, J. (2013) Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review,* **40**(1), 1-25.

Prasad, K.M., Reddy, A.R.M. and Rao, K.V. (2017) BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web. *Journal of King Saud University-Computer and Information Sciences,* 1-15.

Ranjan, N.M. and Prasad, R.S. (2018) LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features. *Applied Soft Computing.* **71**, 994-1008.

Rasmusson, L. and Jansson, S. (1996) Simulated social control for secure Internet commerce. In: *Proceedings of the 1996 ACM workshop on new security paradigms,* ACM, 18-25.

Resnick, P. and Zeckhauser, R. (2002) Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In: *Proceedings of the Economics of the Internet and E-commerce*, Emerald Group Publishing Limited, 127-157.

Sahoo, K.S., Puthal, D., Tiwary, M., Rodrigues, J.J.P.C., Sahoo, B. and Dash, R. (2018) An Early Detection of Low Rate DDoS Attack to SDN Based Data Center Networks using Information Distance Metrics. *Future Generation Computer Systems,* 89, 685-697.

Seroussi, G. (1999) Elliptic curve cryptography. *Information Theory and Networking Workshop (Cat. No.99EX371), Metsovo,* 41, IEEE.

Specht, S.M. and Lee, R.B. (2004) Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. *Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems, September 15-17, 2004, San Francisco, CA,* 543-550.

Thomas, R. and Rangachar, M.J.S. (2016) Integrating GWTM and BAT algorithm for face recognition in low-resolution images. *The Imaging Science Journal,* **64**(8), 441-452.

TTW Group (2013) The notorious nine: Cloud computing top threats in 2013, Report. *Cloud Security Alliance.*

Udhayan, J. and Anitha, R. (2009) Demystifying and rate limiting ICMP hosted DoS/DDoS flooding attacks with attack productivity analysis. In: *Proceedings of IEEE International Advance computing conference (IACC 2009),* IEEE, 558-564.

Wang, G., Musau, F., Guo, S. and Abdullahi, M.B. (2015) Neighbor similarity trust against sybil attack in P2P e-commerce. *IEEE Transactions on Parallel and Distributed Systems,* **26**(3), 824-833.

Yang, Y., Feng, Q., Sun, Y.L. and Dai, Y. (2009) Dishonest behaviors in online rating systems: cyber competition, attack models, and attack generator. *Journal of Computer Science and Technology,* **24**(5), 855-867.

Yin, D., Zhang, L. and Yang, K. (2018) A DDoS Attack Detection and Mitigation with Software-Defined Internet of Things Framework. *IEEE Access,* 6, 24694 – 24705.

Zupancic, E. and Trcek, D. (2017) QADE: a novel trust and reputation model for handling false trust values in e–commerce environments with subjectivity consideration. *Technological and Economic Development of Economy,* **23**(1), 81-110.