A Framework for Providing Security for Cloud SaaS Model through an Enhanced Sea Lion Optimization Algorithm

Reddy Saisindhutheja*
School of Computing & Information Technology, REVA University, Bengaluru, Karnataka, India
Department of CSE, Sreenidhi Institute of Science & Technology, Hyderabad, Telangana, India. E-mail: thejasindhu@gmail.com
Gopal K Shyam
School of Computing & Information Technology, REVA University, Bengaluru, Karnataka, India.
E-mail: gopalkrishnashyam@reva.edu.in

ABSTRACT
The Cloud paradigm is increasing very rapidly due to its on-demand services. Software-as-a Service (SaaS) is one amongst the most outstanding and fastest-growing fields in the era of Cloud computing. Organizations are adopting SaaS solutions, which offer several advantages, mostly in minimizing cost and time. Over all the excitement around SaaS, security is one of the foremost critical issues for its growth in Cloud computing. Hence this paper introduces a novel framework for detecting the DoS attacks using an enhanced Sea Lion Optimization Algorithm (SLnO) known as Fitness updated Sea Lion Optimization Algorithm (FSLnO). The proposed work has two stages (i) feature selection using FSLnO and (ii) classification through Recurrent Neural Network (RNN). It ensures the separation of normal and compromised date. For evaluation KDD cup 99 dataset is used and evaluated in terms of Precision, Accuracy, False Positive and Negative rates. Results prove that the proposed work outperformed the other conventional models.

Keywords:
Software-as-a-Service, Optimization, Attack Detection, RNN
Article Received: 18 October 2020, Revised: 3 November 2020, Accepted: 24 December 2020

I. Introduction

The Cloud paradigm is increasing very rapidly due to its on-demand services [1]. Software-as-a Service (SaaS) is one amongst the fastest-growing fields in the era of Cloud computing. Organizations are adopting SaaS solutions, which offer several advantages mostly with respect to finance and time management. The key ingredient in the success of Software-as-a-Service (SaaS) is based upon client's satisfaction [2]. Security is found to be one of the foremost critical issue for SaaS in Cloud computing [3], [4], [5], [6]. SaaS provides software licenses, modest tools, centralized management of data, multi-tenant solutions, minimal maintenance, and scalability.

The existing works on security focuses on multi-factor authentication, verifying the access control, monitoring the data access, verifying the data deletion, controlling the consumer access devices, security check events, etc. [7], [8], [9]. These services are totally delayed due to a lot of attacks [10]. Worms, DoS attacks or botnets are the subsets of threats that frequently occur in the networks [11]. Further, there is a need to implement a holistic solution for detecting the attacks.

In [12], an Intrusion Detection system is proposed in Communication Networks using various classifiers, viz., Navies Bayes classifier, CNN classifier, SVM classifier, ANN classifier, and KNN classifier. Similarly, in [13], a Log-Based Intrusion Detection is introduced for Cloud Applications. [14], [15], [16], [17] proposed various intrusion detection systems.

Some of the optimization techniques for investigating the attacks are as follows. In [18], a novel intrusion detection technique, LE-MPSO-BP, is developed by merging Laplacian Eigenmaps (LE) and Modified Particle Swarm

Optimization-Back Propagation (MPSOBP). Initially, dimensionality is reduced using LE, for selecting the features MPSO-BP is used. The DDoS attack detection system is developed in [19] using C.4.5 algorithm, coupled with signature detection. Decision tree is used for attack detection. The existing algorithms lacks local optima and excessive time for computation. There is a need to implement solution for attack detection.

This paper introduces an attack detection system to provide security for SaaS. An enhanced machine learning algorithm called Fitness updated Sea Lion Optimization Algorithm is designed and developed for feature selection. Further for classification RNN is accomplished, which separates the normal and attack data and sends the normal data to Cloud Service Provider (CSP). The rest of the paper is organized as follows. The proposed work is explained in section 2. Experimental results are discussed in section 3. Finally, section 4 offers a conclusion.

II. Proposed Work

2.1 Architecture of the proposed work

This work is classified into three stages viz, (i) Data Pre-processing (ii) Selecting the features through FSLnO and (iii) Classification with RNN. As a benchmark, we employed the KDD cup 99 dataset [20]. It is an intrusion detector learning dataset consists of 41 features. Initially, the features are pre-processed. After pre-processing, the best features are selected using the FSLnO algorithm to increase the accuracy of the detection. Further, the behaviour of the data is predicted from the best attributes that use RNN. The entire process is divided into testing and training. For training, we have considered 80% of data and the remaining 20% for testing. Figure 1 depicts the architecture of the proposed work.

2.2 Feature selection using FSLnO: Training Phase

Selecting the most relevant features plays a vital role in the attack detection system. The complexity and evaluation of the model are reduced with feature selection. After the pre-processing, we use the proposed FSLnO for feature selection. To maximize the searching ability of SLnO, we update the fitness of using precision metric. Increased precision value decreases the false-positive rates and gives the best search capability of features in the dataset.

The standard SLnO algorithm was developed based on the hunting process of sea lions that come with groups and decides to hunt [21]. Their whiskers generally govern the process of hunting and finding of the prey (target) in sealions. The arithmetical model of FSLnO is as follows:

Stage 1-Modified version of Detection and chasing of the prey: Sea lions whiskers help to estimate the position of prey concerning its structure. If there is plenty of prey, then the sealions ask others to come and attack the prey. In this process of hunting, the sea lion that demands others is known as the leader. Eq. (1) and Eq. (2) show the prey update mechanism.

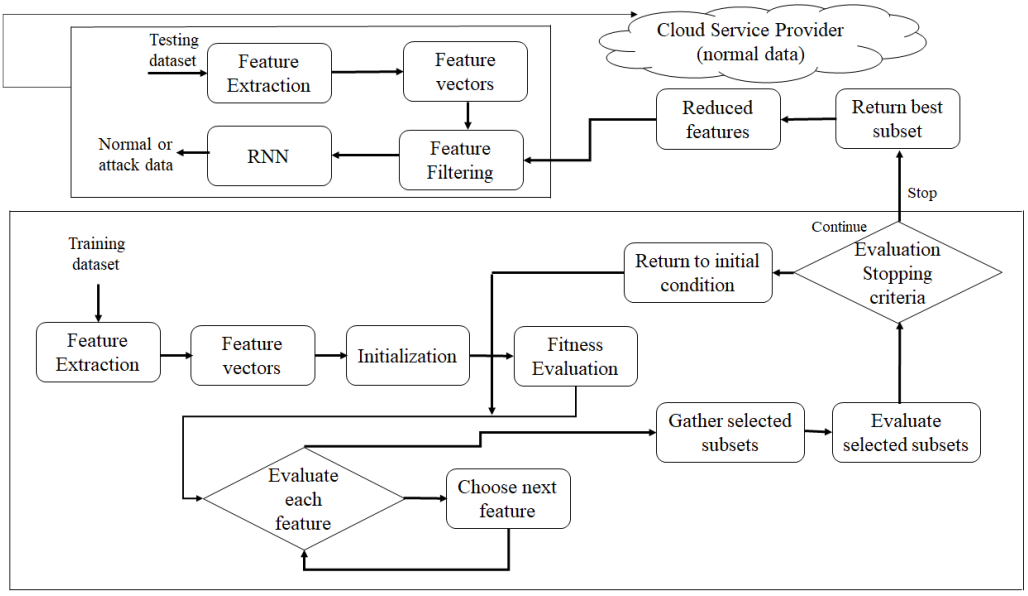


Figure 1: Architecture of the proposed work

The distance between target and sealion is assigned as $\overrightarrow{Distance}$. During the location update of sea lion, the outer product position $\overrightarrow{SL(t)}$ and target $\overrightarrow{Prey(t)}$ has given more prominence. Present iteration is denoted as *t*, and random vector $\vec{R}$ lies in between [0,1]. Sea lions change from present iteration to next *(t+1)*, when getting close to prey this is modeled in Eq. (2). After a course of iterations, there is a slight change of $\vec{Q}$ from 2 to 0.

$$\overrightarrow{Distance} = \left|2\vec{R}.\overrightarrow{Prey(t)} - \overrightarrow{SL(t)}\right| \quad (1)$$

$$\overrightarrow{SL(t+1)} = \overrightarrow{Prey(t)} - \overrightarrow{Distance}.\vec{Q} \quad (2)$$

Initially, the fitness *(Fit)* is found for the present search agent, and a median of total fitness Fit_{ev} is calculated. Further, *Fit >= median* (Fit_{ev}), a mid-value is given. If the solution obtained is greater than these mid-values, then it is evaluated using Eq. (3).

$$\overrightarrow{SL(t+1)} = mean\ (Prey(t): SL(t)) \quad (3)$$

Stage 2- Modified Vocalization: Sea lions communicate in both water and land. When it finds the prey, it calls others to join, and this is given in Eq. (4), Eq. (5) and Eq. (6), respectively.

The leader is represented as SL_{leader}, and noise in water, as well as, air is denoted as $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$.

$$\overrightarrow{SL_{leader}} = \left|(\overrightarrow{x_1}\,(1+\overrightarrow{x_2}))/\overrightarrow{x_2}\right| \quad (4)$$

$$\overrightarrow{x_1} = Sin\theta \quad (5)$$

$$\overrightarrow{x_2} = Sin\phi \quad (6)$$

The new solution is based on the new value. The median of total fitness Fit_{ov} is evaluated. If *(Fit* $\geqslant$ *median* (Fit_{ov})*)*, the solution that has maximum value when compared with mid values are updated using Eq. (7).

$$\overrightarrow{SL(t+1)} = mean\ (Prey(t): SL(t)) \quad (7)$$

Stage 3- Modified Attacking phase:

(a) Reduced encircling approach: This mainly depends on $\vec{Q}$ present in Eq. (2), and is decreasing from 2 to 0 in all rounds. This minimization leads the sea lions to go near the prey.

(b) Circle position update: During the process of hunting, sea lions chase a group of fishes and it is given in Eq. (8). The most acceptable optimal solution and the searching agents' distance is denoted as $\left|\overrightarrow{Prey(t)} - \overrightarrow{SL(t)}\right|$. The arbitrary number between [-1,1] is denoted as *rm*, and

cos(2πm) denote sea lions circular path surrounding the prey.

$$\overrightarrow{SL(t+1)} = \left| \overrightarrow{Prey(t)} - \overrightarrow{SL(t)} . \cos(2\pi l) \right| + \overrightarrow{Prey(t)} \quad (8)$$

At the time of updating regarding the position, the present fitness is compared over the median of total fitness Fit_{ov}. *If (Fit* $\geqslant$ *median* (Fit_{ov}), the result that has values greater than mid values are modified with Eq. (9).

$$\overrightarrow{SL(t+1)} = mean(P(t): S(t)) \quad (9)$$

The variation between SLnO and FSLnO is, in that the current fitness is analyzed with middle value of the complete fitness in each and every phase. If it is larger than mid value then they are discarded and the remaining are selected for further optimization. Additionally, the average is calculated for all the fitness values, and the best value is selected. Thus, this entire process maximizes the convergence speed of the algorithm and selects 11 best features out of 41.

2.3 Attack detection using RNN

Classification is performed through RNN. The weights of input to hidden layer are denoted as $(Wt_{11}, Wt_{12}, ..., Wt_{1n})$ *and* $(Wt_{21}, Wt_{22}, ..., Wt_{2n})$. The arbitrary weights of the recurrent layer and neuron of the output layer are generated with the interval $[Wt_{minimum}, Wt_{maximum}]$.

Step 1: Initially, the input layer is assigned with selected attributes and weights.

Step 2: RNN is explained with Eq. (10) and Eq. (11), where q_i and Wt_{ij} represent the activation state of neuron's *i* at time *t*, and weights are optimized. Activation function fun_i depends on inputs.

$$p_i(t) = \sum_j q_j(t) Wt_{ij}(t) \quad (10)$$

$$q_i(t) = p_i(x_i(t)) \quad (11)$$

Step 3: Sigmoid function determines the decision vector in Eq. (12), in which, i=1 to n and output is $q_{act} = Wt_{2i} fun_i$.

$$fun_i = \frac{1}{1+e^{-pi}} \quad (12)$$

Step 4: In back propagation regarding forward procedure, each neuron's output is computed using Eq. (13), Eq. (14), where *Hid, Cn, In, fun* denotes: hidden layer, neuron stored at previous network location, input neurons, activation function. Then p_j is the j_{th} input of the neuron and Td_{ij} is displacement in recurrent function.

$$q_i(t) = fun_i(pi(t), C_{ni}(t)) \quad (13)$$

$$p_i(t) = \sum_{j \in Hid} q_j(t) wt_{ij} + \sum_{j \in In} p_j(t) wt_{ij} + \sum_{j \in Cn} q_j(t - Td_{ij}) wt_{ij} \quad (14)$$

Step 5: By using Bayesian Regulation, the back propagation error can be minimized using Eq. (15), which is difference between predicted and actual value.

$$E_{rm} = P^{target} - P^{actual} \quad (15)$$

Step 6: Latest weights, as well as bias, are modified with Eq. (16).

$$E_d = \frac{1}{N} \sum_{i=1}^{N} ((Er_m)^2) \quad (16)$$

Step 7: Weights are updated using Eq. (17), where α and β are two arbitrary values, and Br is the new weight update.

$$B_r = \beta E_d + \alpha Er_m \quad (17)$$

Testing phase: The CSP examines the data whether it is attacked or normal. A score value is obtained in Eq. (18). Using this the decision is taken to determine it as normal or intruded.

$$output = \begin{cases} T_h \geq score, & \text{normal} \\ T_h < score, & \text{attack} \end{cases} \quad (18)$$

III. Experimental Results and Discussion

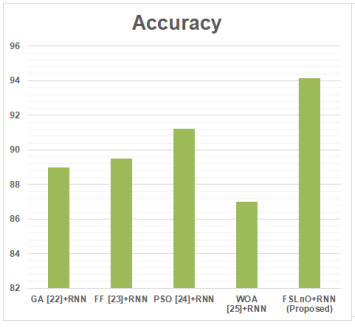
(i) Simulation Procedure: The proposed work is executed using a java platform and CloudSim.

Accuracy, precision, recall are positive measures that are to be at higher rates. Contrarily, FPR, FNR, FDR are negative measures that should be low to minimize errors. The evaluation is compared with GA [22], PSO [23], FF [24], WOA [25]. Further, NN [26], SVM [27], and CNN [28] are used for classifier evaluation. Table 1 shows the various metrics for evaluation.

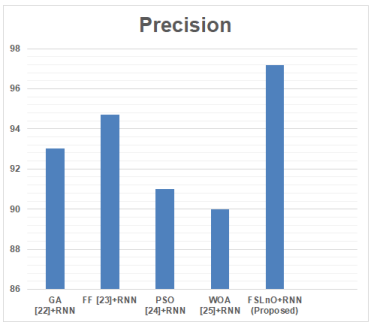
(ii) Evaluation of proposed and conventional algorithms: The existing algorithms like GA, PSO, FF, WOA are considered for comparison. Figure 2 - (a), (b), (c), (d), (e), (f) shows the performance of positive measures viz. accuracy, precision and recall; and the negative measures FPR, FNR and FDR over the other conventional algorithms. Results show that FSLnO outperformed when compared with other existing works. In GA, at each time, there may be a change in optimal value. WOA is not explored and cannot do efficient search space. PSO and FF has a high probability of local optima. In the proposed work, by taking mid and average values, the fitness is computed in FSLnO and performs a quick search resulting an optimal solution.

Metrics used	Definition	Formula
Accuracy	Ratio to the total of two correct predictions to the entire possible predictions	(TP + TN)/ (TP + TN + FP + FN)
Precision	Proportion of ordinary data identified to the entire normal and abnormal data	TP / (TP + FP)
Recall	Ratio of true positives and total elements that are in the class of true positive	TP / TP + FN
FPR (False Positive Rate)	Chance of falsely not accepting the null hypothesis.	FP / (FP + TN)
FNR (False Negative Rate)	Ratio of the positive values that gives the negative results while performing the test	FN / (FN + TP)
FDR (False Detection	It is an error of type I, which inappropriately reject null hypothesis	FP/ (TP + FP)

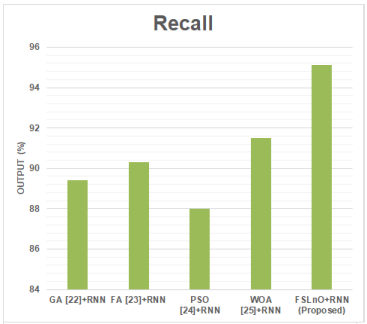
Table 1: Performance Metrics



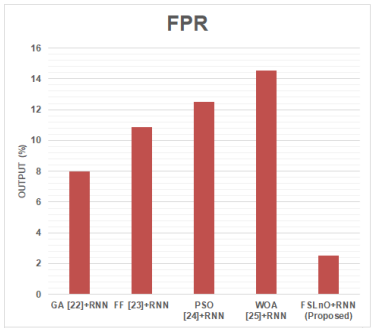
(a) Accuracy



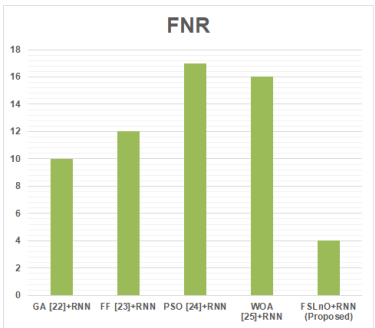
(b) Precision



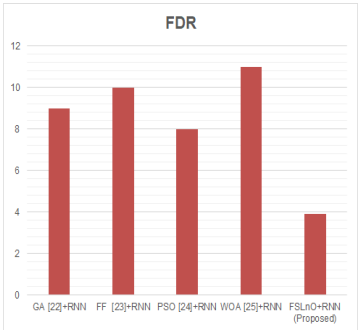
(c) Recall



(d) FPR



(e) FNR



(f) FDR

Figure 2: Evaluation of proposed and conventional algorithms

(iii) Evaluation of proposed and conventional classifiers: The proposed work is compared with different classifiers like NN, CNN, and SVM. The evaluation is performed with respect to positive and negative measures. Figure 3 Depicts the evaluation of proposed and the conventional classifiers. Results prove that the proposed work outperformed when compared with the other conventional classifiers with a precision of 97.3%, accuracy - 94.12%, recall - 94.01%, FPR- 2%, FNR - 4%, FDR - 1%.

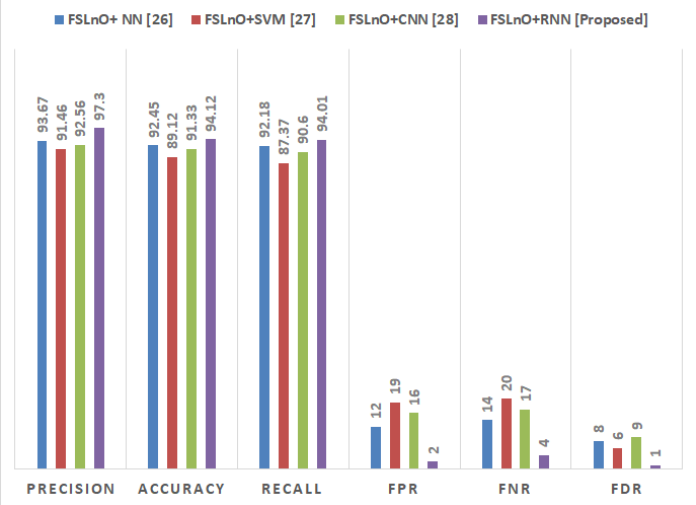


Figure 3: Evaluation of proposed and conventional classifiers

(iv) Convergence Proof: Feature selection plays a vital role in intrusion detection. Figure 4 depicts the convergence of FSLnO that selects 11 best features out of 41, whereas GA selects 13 features, FF retrieves 14 features, and WOA retrieves 16 features. One advantage of FSLnO is it has a high convergence speed as the mid-value is taken and the mean is calculated for the entire fitness. In this work we have calculated the fitness based on the precision metric, which minimizes the false-positive rates and results the best feature selection.

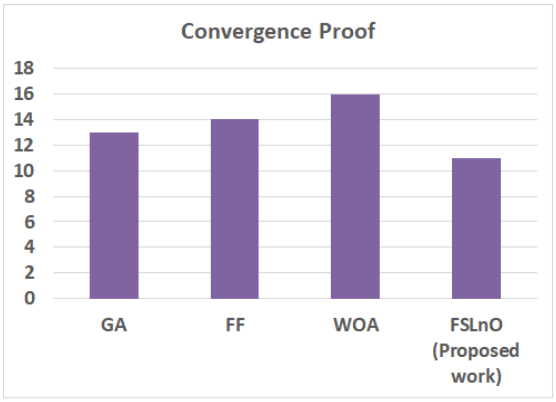


Figure 4: Convergence proof of FSLnO with respect to feature selection

IV.CONCLUSION

This paper has introduced a novel framework for DoS attack detection. Initially as a benchmark, KDD cup 99, dataset is considered and applied for pre-processing. After the pre-processing, the features are subjected to FSLnO algorithm to select the best features. For further classification, to separate the standard and attack data RNN is used. On the basis of score value, the usual and unusual data are separated, and usual data is sent to CSP. The proposed work was analyzed over the other conventional models by evaluating accuracy, precision, FPR and FNR. Results show that the proposed work outperformed the other existing works. In the future, we develop the attack mitigation system by considering the dynamic datasets.

REFERENCES

[1] Y. Zhang, H. Sheng, X. Wang, J. Hua, "User Security Authentication Scheme under SaaS Platform of Enterprises", International Conference on Virtual Reality & Intelligent Systems, pp.147-151, 2018.

[2] M. M. a. V. C. s. Mohamed Abdel-Basset, "NMCDA: A Framework for Evaluating Cloud Computing Services", Future Generation Computer Systems, vol. 86, pp. 12-29, 2018.

[3] P. S. a. K. Khajehmoogahi, "Towards continuous security certification of Software-as-a-Service applications using

web application testing techniques", Journal of Internet Services and Applications, vol. 4, no. 5, pp. 931-938, 2017.

[4] Khadija A, Hafiddi, Dahchour, "Policy-Driven Middleware for MultiTenant SaaS Services Configuration", International Journal of Cloud Applications and Computing, vol. 9, no. 4, 2019.

[5] Shweta K, Charu G,"Ensure Hierarchal Identity Based Data Security in Cloud Environment", International Journal of Cloud Applications and Computing, vol. 9, no. 4, pp. 21–36, 2019.

[6] Ansam Khraisat, Iqbal Gondal, Peter V & Joarder K, "Survey of intrusion detection systems: techniques, datasets and challenges", Cyber security, 2019, vol.2, no. 20, pp. 1-22.

[7] E. A. R.Barona, "A Survey on Data Breach Challenges in Cloudcomputing Security:Issues & Threats", International Conference on circuits Power & Computing Technologies, pp. 1-8, 2017.

[8] R. M. K. H. Eduardo B. Fernandez, "Building a security reference architecture for cloud systems", Journal of Requirements Engineering, vol. 21, no. 2, pp. 225-249, 2016.

[9] Amit Kr Mandal, Aniban S, "A Novel Meta-Information Management System for SaaS", International Journal of Cloud Applications and Computing (IJCAC), vol. 9, no.3, pp. 21, 2019.

[10] A. A, Bhupesh Kumar D, "Credential and Security Issues of Cloud Service Models", 2nd International Conference on Next Generation Computing Technologies, pp. 888-892, 2016.

[11] "Cyber security risks",http://www.adotas.com/2017/08/the-6-major-cyber-security-risks-to-cloud-computing/. [Accessed on 30 Dec 2021].

[12] M.N.G.S.Bhosale, "Intrusion Detection in Communication Networks Using Different Classifiers", 2nd International Conference on Advanced Technologies for Societal Applications, vol. 2, pp. 19-28, 2018.

[13] P. Jaron Fontaine, "Log-Based Intrusion Detection for Cloud Web Applications Using Machine Learning", International Conference on P2P, Parallel, Grid, Cloud & Internet Computing, vol. 96, pp. 197-210, 2019.

[14] AmarMeryem, Bouabid ELOuahidi, "Hybrid intrusion detection system using machine learning", In: Proceedings of the Network Security, Elsevier, vol. 2020, no. 5, pp. 8-19, 2020.

[15] Çavu¸soglu Ü, "A new hybrid approach for intrusion detection using machine learning methods", In: Proceedings of the Applied Intelligence, Springer, vol. 49, 2019.

[16] I. Aljamal, A. Tekeoglu, K Bekiroglu, S Sengupta, "Hybrid Intrusion Detection System Using Machine Learning Techniques in Cloud Computing Environments", 17th International Conference on Software Engineering Research, Management & Applications, pp. 84-89, 2019.

[17] Aharkhizan M, Azmoodeh A, HaddadPajouh H, Dehghantanha A, Parizi R.M, Srivastava G, "A Hybrid Deep Generative Local Metric Learning Method for Intrusion Detection", Handbook of Big Data Privacy, 2020.

[18] D. Q. H. L. Yiqun Liu, "The intrusion detection model utilizing LE modified PSO-BP", International Conference on Software Engineering & Service Science, pp. 318-321, 2017.

[19] S. E. K. N. A. Y. S. Marwane Zekri, "DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments", 3rd International Conference of Cloud Computing Technologies and Applications, pp. 1-7, 2017.

[20] "KDD cup 99," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, [Accessed on 4.nov.20].

[21] Raja Masadeh, Basel A. Mahafzah and Ahmad Sharieh,"Sea Lion Optimization Algorithm", International Journal of Advanced Computer Science and Applications, vol. 10, no.5, 2019.

[22] JohnMcCall, "Genetic algorithms for modelling and optimisation", Journal of Computational and Applied Mathematics, vol. 184, no. 1, pp. 205-222, 2005.

[23] M.R.Tanweer, S.Suresh, and N.Sundararajan, "Self-regulating particle swarm optimization algorithm", Information Sciences, vol. 294, pp. 182-202, 2015.

[24] IztokFister, IztokFisterJr, Xin-SheYang and JanezBrest, "A comprehensive review of firefly algorithms", Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.

[25] Seyedali Mirjalili,Andrew Lewisa,"The Whale Optimization Algorithm", Advances in Engineering Software, vol.95, pp.51-67, May 2016.

[26] Wei-Chi Ku, "Weaknesses & drawbacks of a password authentication scheme using neural networks for multi-server architecture", IEEE Transactions on Neural Networks, vol. 16, 2005.

[27] WMCampbell, DESturim, DA.Reynolds, "SVM using GMM supervectors for speaker verification", IEEE Signal Processing Letters, vol. 13, pp. 308-311, 2006.

[28] Y. Qin, J. Wei W. Yang, "Deep Learning Based Anomaly Detection Scheme in SDN", 20th Asia-Pacific Network Operations and Management Symposium, pp. 1-4, 2019.