International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Automatic Intrusion Detection Using Optimal Features with Adaptive Bi-Directional Long Short Term Memory

G. Parimala[1], Kayalvizhi R.[2]

Submitted: 13/09/2023 **Revised**: 21/10/2023 **Accepted**: 08/11/2023

Abstract: Technology is advancing quickly, which not only makes life easier but also gives rise to various security problems. As the Internet has grown over time, so has the number of online attacks. To avoid the attacks, IDS (IDS) is developed. Nowadays, a lot of machine learning methods have been developed to detect attack. Although these jobs, decides the right feature and improving classification accuracy remains a challenging task. Therefore, in this paper, automatic attack detection is proposed. The presented approach is contains three stages namely, pre-processing, feature selection and classification. In pre-processing, the redundant and missing data are removed. After the pre-processing, important features are selected from each records using adaptive pelican optimization algorithm (APO). Then, the certain attributes are fed to the enhanced bi-directional long short term memory (EBi-LSTM). The efficiency of presented technique is discussed based on different metrics.

Keywords: *IDS, adaptive pelican optimization algorithm, enhanced bi-directional long short term memory, malicious attack and feature selection*

1. Introduction

The tremendous amount of data and information on the Internet today makes it challenging to maintain a secure and stable infrastructure. Although software upgrades and firewalls help provide security, dynamic systems are potentially vulnerable to misuse. IDSs are one, along with network analysers and other dynamic processes. The precise objective of intrusion detection is established [1]. In order to identify any deviations or other abnormalities—which are violations of computer security policies intrusion detection monitors and analyses the processes running in a computer system or network [2]. The misuse and anomaly techniques are the two types of intrusion detection. Misuse aims to identify attack signatures in the resource under observation. Understanding typical behaviour and any variation from it is necessary to identify anomalies [3].

Cloud computing intrusion detection is an NP-Hard issue. Consequently, based on evolutionary computing and meta-heuristic techniques, many algorithms can tackle this problem. Statistical approaches and rule-based expert systems are two common methods for investigating incursion at an early stage. The detection rate of some well-known intrusions by a rule-based expert IDS is high, but it struggles to identify novel intrusions and must manually and frequently update its signature database [4]. Different statistical techniques, including principal component analysis, cluster and multivariate analysis, Bayesian analysis, and frequency and simple important analysis, are used in statistical-based IDS. However, because complex network traffic necessitates the construction of a complex mathematical model, this kind of IDS is impracticable. To solve the problem, lot of machine learning and deep learning algorithms are developed [5,6].

Lot of researchers had developed intrusion detection on network using deep learning algorithm. Among them some of the works are analysed here; Olamatanmi et al. [7] had explained IDS based on machine leaning algorithm. This method was applied for underwater network. Here, to reduce the computation complexity, they applied the feature selection techniques. For classification, they applied C4.5 and Naïve Bayes (NB). Finally they conclude C4.5 algorithm got better results compared to NB classifier. Priya et al. [8] had developed deep reinforcement learning with binary Bat algorithm for IDS. Here, bat algorithm was used for feature selection and Q-learning algorithm was used for classification process. The root means square propagation (RMSProp) optimizer is utilized for the effectual hyper-parameter tuning procedure.

Mahzad et al. [9] had developed deep learning algorithm based IDS. They created a particular Long Short-Term Memory (LSTM) architecture, and this model was used to find intricate connections and long-term dependencies among incoming traffic packets. In doing so, they decreased the amount of false alarms and improved the developed IDS's accuracy. In addition, big data analytical approaches were used in this research to increase the pace of deep learning algorithms, which have a slow execution

[1,2]*Department of Networking and Communications, School of Computing, College of Engineering and Technology,*
SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, India
**Corresponding Author: Kayalvizhi . R, ,kayalvir@srmist.edu.in*
parimalg@srmist.edu.in

speed because of their enormous complexity. Moreover, Gavini et al. [10] had developed deep transfer learning based intrusion detection. To expand the general security of a cloud-based processing climate, this work fabricates a viable cloud IDS utilizing sandpiper-based include determination and broadened harmony profound exchange learning characterization. In view of the sandpiper improvement calculation (SOA), the quantity of elements from the furnished attack dataset is diminished with minimal measure of data lost. This model was then used to sort different assaults as per the best credits that were picked for them. Move learning utilizes a pre-prepared network called AlexNet for calibrating the qualities in convolution layers. To refresh the organization loads, the lengthy harmony enhancer was additionally utilized. Rahma et al. [11] had developed mobile Ad-hoc network for intrusion detection. The newly released Deep Learning-based IDS exploits the majority of potential assaults that affect routing services in Mobile Networks and concentrates on Denial of Services (DoS) attacks inside labelled datasets that are available for intrusion detection. Dilli et al. [12] had developed Quantum-Secure Ensemble Intrusion Detection Techniques for Cloud Based on Deep Learning. This system was created to accomplish two goals. It first examines the drawbacks of the current IDS before presenting a concept for improving IDS accuracy. Effective intrusion detection is achieved using the suggested Ensemble Intrusion Detection Model for Cloud Computing Using Deep Learning (EICDL)..

2. Proposed EBi-LSTM Based Intrusion Detection

The major purpose of the methodology is to detect the data packet as malicious or legitimate using deep learning model. The architecture of Proposed EBi-LSTM based intrusion detection is presented in figure 1. The presented approach consist of three stages namely, pre-processing, feature selection and classification. At first, the data are collected and pre-processed. After pre-processing, important attributes are selected using adaptive pelican optimization (APO) algorithm. Then, the selected attributes are fed to the enhanced bidirectional LSTM classifier to categorize a data as normal or malicious data.

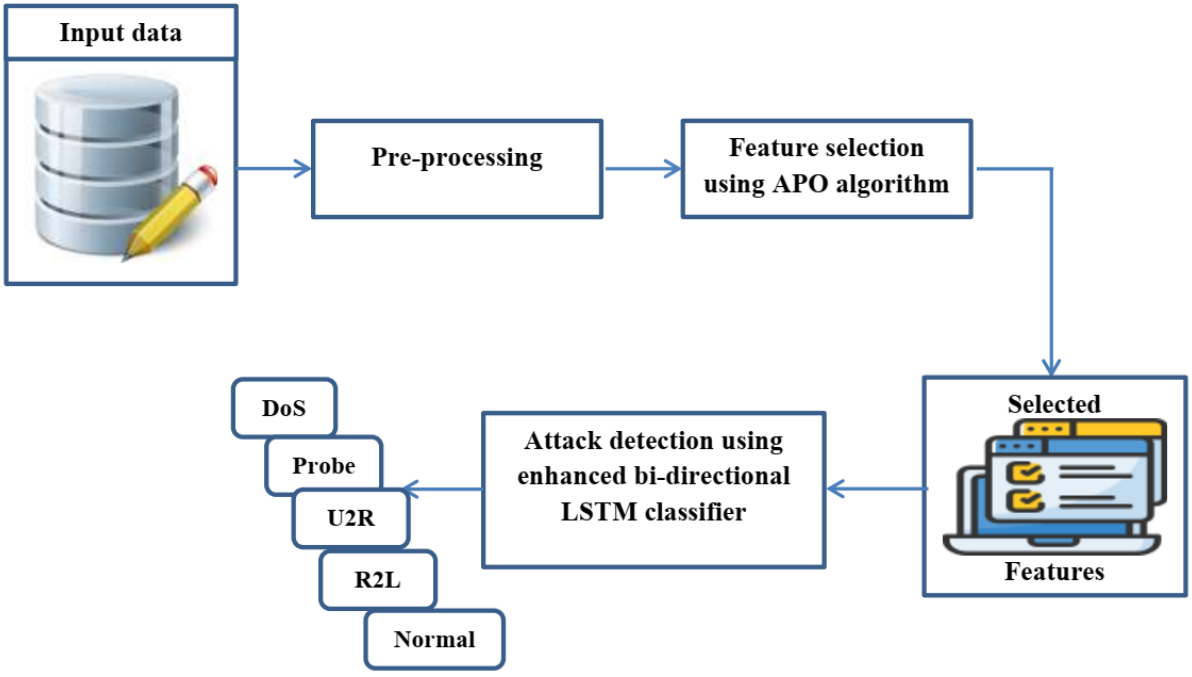

Fig 1: The architecture of Proposed EBi-LSTM based IDS model

2.1 Data pre-processing

Data pre-processing is an important process for data classification. First step of data-pre-processing is data filtering. Some of the data in the collection are numerical, while others are symbolic. To make the data processing procedure simpler, data was collected from several IDS sources and put into a numerical format. Numerical labels are created from symbolic attributes. Then, the data are normalized for further processing.

2.2 Feature selection using Adaptive pelican optimization algorithm

Feature selection is an important process for malicious data detection. The feature selection, select the important feature for detection process. It will decrease the computation complexity and execution time. For feature selection, in this paper, Adaptive pelican optimization algorithm is utilized (APO). APO is a hybridization of oppositional based learning (OBL) and APO. OBL strategy is mainly used for increases the searching ability

of agents. A brand-new stochastic optimisation technique that draws inspiration from nature is called the Pelican algorithm. The primary goal of the planned POA's design is to simulate pelicans' typical hunting behaviour [13]. In POA, pelicans serve as search agents by looking for food sources. In this algorithm, every population member is defined as a candidate solution. Every population member develops parameters for the optimization issue parameters related to their location in the search location. Below is an explanation of the feature selection process is clearly given.

Step 1: Solution initialization: For optimal feature selection process, at first, we define the initial solution randomly. In this paper, each feature is called as pelican and solution is represented as search agent. The initial solution is represented using equation (1).

$$S_{ij} = \{P_1, P_2, ..., P_n\} \quad (1)$$

$$P_1 = [F_1, F_2, ..., F_n] \quad (2)$$

Features / Records	F_1	F_2	F_3	F_4	….	F_{14}	F_{15}	….	F_{26}	F_{27}
P_1	S	N	S	N	….	S	S	….	N	S
P_2	N	S	N	S	….	N	N	….	S	N
P_3	S	N	S	S	….	S	N	….	N	N
….										
P_n	N	N	N	S	….	N	S	….	S	N

Table 1: Solution encoding Format

In table 1, F1, F_2,…,F_{27} represent the features, P_1,P_2,..,P_n represent the records, S represent the consequent characteristic is particular with N represent the consequent characteristic is not selected.

Step 2: Opposite solution generation: After solution initialization process, the explanation corresponding opposite solution is generated. The main purpose of opposite solution is to enlarge the penetrating facility. The opposite solution is generated based on the equation (3).

The opposite solution Y'_i can be defined as follows,

$$P'_i = x + y - P_i \quad (3)$$

Where, $P_i \in [x, y]$ is a real number.

Step 3: Fitness evaluation: once solutions are generated, strength of both the solution is designed. Strength is used to evaluate the solution quality. In this paper, maximum accuracy is measured as the fitness occupation. The fitness is calculated using equation (4).

$$Fitness = Max\left(\frac{TP + TN}{TP + TN + FP + FN}\right) \quad (4)$$

Step 4: Updation using APOA: To find out the best features, the solutions are updated using POA. The updation function is having two main functions such as exploration stage and exploitation stage king.

Travels to prey (exploration stage): In the initial stage, pelicans detect the prey location and then travel to this detected location. The pelican technique is modeled for scanning search space and power exploration in computing various locations of search space. The essential point in POA is prey location is created arbitrarily in the search location. This expands the exploration power of pelicans in the specific search for the issue-solving space. The travel to prey of the location is designed as follows;

$$X_{I,J}^{P1} = \{x_{I,J} + RAND.(P_J - i.x_{I,J}) \; f_p < f_I \; x_{I,J} + RAND.(x_{I,J} - P_J) \; Else \quad (5)$$

Here, i is defined as the random variable which equals 1 or 2, f_p is defined as the objective function parameter, P_J is defined as prey location and $X_{I,J}^{P1}$ is defined as the novel status of the pelican with its specific dimension. The parameter i is selected for every iteration and every

member. In the updating process, it is defined as effective updating, this technique is avoided from traveling to non-optimal locations. This procedure is designed by using the below equation,

$$x_I = \{x_I^{p1}, f_i^{p1} < f_i \; x_I \quad Else, \tag{6}$$

Here, f_i^{p1} is defined as the objective function parameter related to stage 1 and x_I^{p1} is defined as the novel status of the i[th] pelican.

Swimming on the water surface (exploitation stage): In the second stage, the pelican achieves the water surface and spread its wings on the water surface to move the fish upwards after that gathers the prey in its throat pouch. This technique achieves huge fish in the attacked location to be attacked by pelicans. Designing this characteristic of pelicans creates the POA to manage to best points in the hunting location. This procedure improves the exploitation ability and local search power of POA. With the consideration of the mathematical model, this algorithm should manage the points in the neighborhood of the pelican space to achieve the best solution. This characteristic of pelicans while hunting is designed using equation (7).

$$X_{I,J}^{P2} = x_{I,J} + r\left(1 - \frac{T}{t}\right).(2.RAND - 1).x_{I,J} \tag{7}$$

Here, $r\left(1 - \frac{T}{t}\right)$ is defined as coefficient parameters and radius of the neighborhood of the population members to search locally close to every member to achieve the best solution, t is defined as the maximum number of iterations, T is defined as the iteration counter, r is defined as the constant parameter that is equivalent to 0.2, $X_{I,J}^{P2}$ is defined as the novel status of the i[th] pelican.

These formulations are utilized to scan the location related to every member of the population with highly accurate steps and smaller ones. Hence, the POA can converge to solutions near the global optimal solutions related to the concept. In this stage, efficient updating is utilized to reject or accept the novel position which is designed as follows,

$$x_I = \{x_I^{p2}, f_i^{p2} < f_i \; x_I \quad Else, \tag{8}$$

Here, f_i^{p2} is defined as an objective function parameter, x_I^{p2} is defined as the novel status of the ith pelican.

Step 5: Termination criteria: The iteration is stop when the maximum iteration achieved. From the iterations, best solution is selected based on fitness value. The selected feature is used for classification process.

2.3 intrusion detection using EBi-LSTM classifier

After attribute selection, the selected attributes are fed to the EBi-LSTM classifier. One of the recurrent neural network architectures is the LSTM [14]. The output layer, hidden layer, and input layer are the three layers that make up an LSTM. In particular, the buried layer's memory cell has a connection loop with a constant weight value. There are state cells, input gates (IG), output gates (OG), and forget gates (FG) in this layer's block of memory cells. The data is held, discarded, or stored via these gates. The FG decides which data to keep and which to delete. The sigmoid function is used to process the data from the previous hidden state and the current input. The FG information is calculated using Equation (9).

$$FG_k = \sigma\left(We_f \; Y_k + Ve_f \; M_{k-1} + S_f\right) \tag{9}$$

Where;

FG_k □ FG value at time k

Y_k □ Memory cell of input vector at time k

We_f , Ve_f □ FG weight matrices

S_f □ FG bias vector

M_{k-1} □ Memory cell value at timestamp $k-1$

σ □ Sigmoid function

The input gate updates the state of the cell. The activation functions multiply the FG result by the point after receiving the result from the FG.

$$IG_k = \sigma\left(We_i \; Y_k + Ve_i \; M_{k-1}\right) + S_i \tag{10}$$

Where; IG_k □ IG value at timestamp k is defined as IG_k , Weight matrix is represented as We_i, Ve_i and bias vector of IG is represented as S_i . The new cell state is calculated using equation (11).

$$\tilde{c}_k = \tanh\left(We_c \; Y_k + Ve_c \; M_{k-1} + S_c\right) \tag{12}$$

$$c_k = FG_k \; c_{k-1} + i_k \; \tilde{c}_k \tag{13}$$

Where; c_k defined as the memory cell state, $\tilde{c}_k$ defined as candidate cell state, We_c, Ve_c defined as weight matrices, Z_c defined as bias value, FG_k defined as forget gate and $\tanh$ defined as activation function.

The output gate used to predict the output of the given input. The sigmoid function is used to process both ongoing information and previously buried information. The tanh function receives the updated cell position as input. To frame the information that will be sent by the

concealed state, the pointwise tanh and sigmoid results are multiplied together.

$$OG_k = \sigma(We_o\, Y_k + Ve_o M_{k-1} + S_o)$$
(14)

$$M_k = OG_k \tanh(c_k) \quad (15)$$

Where; OG_k is defined as the Value of OG, W_o, V_o defined as OG weight values, S_o defined as bias vector and M_k defined as memory cell. The final output of LSTM is calculated using equation (16).

$$O_k = \sigma(We_h M_k + S_h) \quad (16)$$

Where; O_k defined as the outcome of LSTM layer, W_h defined as weight matrix and S_h defined as the bias value. To improve the efficiency of LSTM classifier, we concatenate the both forward and backward propagation. The modified M_k value is calculated using equation (17).

$$M_k = M_k^F + M_k^B \quad (17)$$

Where; M_k defined as memory cell value, M_k^F defined as forwarding propagation and M_k^B defined as backward propagation.

The LSTM with attention mechanism is evaluated using equation (18).

$$M = \tanh(M_k) \quad (18)$$

We obtain the updated value of matrix M for both a forward and backward pass by substituting equation (17) into equation (18).

$$M' = \tanh\left(M_k^F + M_k^B\right)$$
(19)

Where; M' defined as modified value of matrix M. The M^* is calculated using equation (20).

$$M^* = \tanh\left(M\, \alpha^k\right) \quad (20)$$

$$\alpha = soft\max\left(w^k M\right) \quad (21)$$

Where; M^* is defined as the EBi-LSTM layer output, M represent the concatenation vector of backward and forwarding output of the hidden state, α^k defined as amount of attention. From equations (20) and (21), we calculate the M^*.

$$M'^* = \tanh\left(MH\, \alpha^k\right) \quad (22)$$

The final improved output is given by equation (23).

$$Ey_k = \sigma\left(W_H\, M'^* + S_h\right)$$
(23)

Where; Ey_k defined as Final enhanced classification output, We_H defined as weight matrix and S_h bias function.

3. Results and Discussion

In this division, the efficiency of the presented approach is analysed. The approach implemented using JAVA. The system setup used is a 64-bit version of Windows 2010 operating at 2 GHz on a dual-core computer with 4 GB of main memory. For experimental analysis, the KDD dataset and NSL-KDD dataset is used. For experimental analysis, we use different metrics namely, Accuracy, Precision, Recall, F-measure, False alarm rate and detection rate.

3.1 Experimental results

Experimental results obtained from the presented approach are explained in this section. Here, for feature selection APO algorithm is used and for malicious traffic detection EBi-LSTM classifier is used. We evaluate our suggested strategy against other methods to demonstrate its effectiveness.

Algorithms	Accuracy	Precision	Recall	F-measure	FAR	DR
GA+EBi-LSTM	90.32	88.5	91.2	90.1	13.32	90.23
PSO+ EBi-LSTM	91.42	90.62	92.5	91.6	11.52	91.34
GSA+ EBi-LSTM	94.3	93.40	94.6	94.6	8.52	94.23
APOA+ EBi-LSTM	98.35	97.52	98.3	96.7	2.32	98.1

Table 2: Experimental outcome based on different feature selection technique for KDD-Cup dataset

Algorithms	Accuracy	Precision	Recall	F-measure	FAR	DR
GA+EBi-LSTM	90.32	89.5	91.2	90.1	13.32	90.23
PSO+ EBi-LSTM	90.68	91.62	92.78	90.45	10.52	92.34
GSA+ EBi-LSTM	93.62	93.56	94.24	93.31	7.89	94.23
APOA +EBi-LSTM	98.2	97.25	98.1	97.23	2.53	98.8

Table 3: Experimental outcome based on different feature selection technique for NSL KDD-Cup dataset

Using the KDD-Cup dataset, we compare the efficiency of the presented feature selection technique with other techniques in Table 2. When analysing Table 2, our presented technique achieved a greater accuracy of 98.35%, which is 4.2% better than an IDS based on GSA+EBi-LSTM, 7.5% better than an IDS based on PSO+EBi-LSTM, and 8.8% better than an IDS based on GA+EBi-LSTM. Additionally, our suggested model had the highest levels of precision 97.52%, recall of 98.3%, F-measure of 96.7%, FAR of 2.32, and DR of 98.1%. According to Table 2, the suggested method outperforms individual feature selection techniques on the KDD-Cup dataset. This is due to the optimal feature selection and enhanced deep leaning model.Using the NSL KDD-Cup dataset, we evaluate the efficiency of our presented feature selection technique with various methods in Table 3. When evaluating Table 3, the presented approach achieved a maximum accuracy of 98.2%, precision of 97.25%, Recall of 98.1%, F-measure of 97.23%, FAR of 2.53 and DR of 98.8%. Table 3 shows that the presented method outperforms individual feature selection methods on the NSL KDD-Cup dataset. This is because of the optimal feature selection method.

Algorithms	Accuracy	Precision	Recall	F-measure	FAR	DR
APOA+ANN	90.45	91.56	91.32	91.97	13.26	91.67
APOA + RNN	92.34	94.67	93.45	92.32	10.3	93.12
APOA + LSTM	93.62	93.56	94.24	93.31	7.89	94.23
APOA+ EBi-LSTM	98.35	97.52	98.3	96.7	2.32	98.1

Table 4: Comparative analysis of hybrid features with various classifiers on the KDD-Cup dataset

Algorithms	Accuracy	Precision	Recall	F-measure	FAR	DR
APOA+ANN	87.38	92.4	92.67	92.54	41.31	92.67
APOA + RNN	92.26	95.5	95.28	95.39	23.61	95.28
APOA + LSTM	94.08	95.36	98.02	96.67	33.92	98.02
APOA+ EBi-LSTM	98.2	97.25	98.1	97.23	2.53	98.8

Table 5: Comparative analysis of hybrid features with various classifiers on the NSL KDD-Cup dataset

Table 4 shows the efficiency of presented approach by varying different classifier. When analysing table 4, presented approach attained the maximum accuracy of 98.35% which is 97.52%, recall of 98.3%, f-measure of 96.7%, FAR of 2.32% and DR of 98.1%. Compared to other methods, our method attained the better output. Moreover, in table 5, we analyse the efficiency based on NSL-KDD dataset. Here, also presented method attained the maximum output compared to other methods.

4. Conclusion

Automatic intrusion detection on network has been explained in this paper. For attack detection, EBi-LSTM classifier has been explained and for feature selection APOA has been introduced. The feature selection was used to reduce the computation complexity and increases the classification accuracy. For experimental analysis, two set of dataset has been used namely, KDD cup dataset and NSL KDD dataset. The efficiency of presented approach has been analysed based on different metrics. The effectiveness of presented approach compared with different method. Our presented approach attained the maximum accuracy of 98.35% for KDD cup dataset and 98.2% for NSL KDD-cup dataset. In future, we will apply hybrid deep learning algorithm and we will use intrusion prevention techniques.

References

[1] Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36, no. 1 (2013): 16-24.

[2] He, Ke, Dan Dongseong Kim, and Muhammad Rizwan Asghar. "Adversarial machine learning for network intrusion detection systems: a comprehensive survey." IEEE Communications Surveys & Tutorials (2023).

[3] Vinchurkar, Deepika P., and Alpa Reshamwala. "A review of intrusion detection system using neural network and machine learning." J. Eng. Sci. Innov. Technol 1 (2012): 54-63.

[4] Ugochukwu, Chibuzor John, E. O. Bennett, and P. Harcourt. An intrusion detection system using machine learning algorithm. LAP LAMBERT Academic Publishing, 2019.

[5] Almseidin, Mohammad, Maen Alzubi, Szilveszter Kovacs, and Mouhammd Alkasassbeh. "Evaluation of machine learning algorithms for intrusion detection system." In 2017 IEEE 15th international symposium on intelligent systems and informatics (SISY), pp. 000277-000282. IEEE, 2017.

[6] Kim, Jin, Nara Shin, Seung Yeon Jo, and Sang Hyun Kim. "Method of intrusion detection using deep neural network." In 2017 IEEE international conference on big data and smart computing (BigComp), pp. 313-316. IEEE, 2017.

[7] Mebawondu, O. J., Adetunmbi, A. O., Mebawondu, J. O., & Alowolodu, O. D. (2020, November). Feature Weighting and Classification Modeling for Network Intrusion Detection Using Machine Learning Algorithms. In International Conference on Information and Communication Technology and Applications (pp. 315-327). Cham: Springer International Publishing.

[8] Priya, S., & Kumar, K. (2023). Binary bat algorithm based feature selection with deep reinforcement learning technique for intrusion detection system. Soft Computing, 1-12.

[9] Mahdavisharif, M., Jamali, S., & Fotohi, R. (2021). Big data-aware intrusion detection system in communication networks: a deep learning approach. Journal of Grid Computing, 19, 1-28.

[10] Sreelatha, G., Babu, A. V., & Midhunchakkaravarthy, D. (2022). Improved security in cloud using sandpiper and extended equilibrium deep transfer learning based intrusion detection. Cluster computing, 25(5), 3129-3144.

[11] Meddeb, R., Jemili, F., Triki, B., & Korbaa, O. (2023). A deep learning-based intrusion detection approach for mobile Ad-hoc network. Soft Computing, 1-15.

[12] Salvakkam, D. B., Saravanan, V., Jain, P. K., & Pamula, R. (2023). Enhanced Quantum-Secure Ensemble Intrusion Detection Techniques for Cloud Based on Deep Learning. Cognitive Computation, 1-20.

[13] Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. Sensors, 22(3), 855.

[14] Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016, February). Long short term memory recurrent neural network classifier for intrusion detection. In 2016 international conference on platform technology and service (PlatCon) (pp. 1-5). IEEE.