SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I4P105

Volume 10 Issue 4, 46-56, April 2023
© 2023 Seventh Sense Research Group®

Original Article

Evolutionary Optimization Algorithm with Deep Echo State Network for Anomaly Detection on Secure Cloud Computing Environment

V. Sujatha Bai[1], M. Punithavalli[2]

[1,2]*Department of Computer Applications, Bharathiar University.*

[1]*Corresponding Author: sujathabaisaravanan@gmail.com*

Received: 23 February 2023 Revised: 21 March 2023 Accepted: 08 April 2023 Published: 27 April 2023

Abstract *- Cloud Computing (CC) is undoubtedly an indispensable technology across the world. It indicates a revolution in collaborative services and data storage. Yet, security problems have increased with the move to CC, which includes intrusion detection systems (IDS). Anomaly detection (AD) is a vital method that ensures the security of CC environments. It identifies unusual behaviour that specifies a security threat. AD is important in a secure CC environment for identifying breaches or attacks and monitoring system performance. The design of potential AD in a secure CC environment needs a combination of machine learning (ML) algorithms, continuous monitoring, and data analysis. Therefore, this article introduces a new Falcon Optimization Algorithm with Deep Echo State Network for Anomaly Detection (FOADESN-AD) technique for a secure CC environment. The presented FOADESN-AD technique exploits the DL model with a metaheuristic optimizer for anomaly or intrusion detection in the cloud platform. To accomplish this, the FOADESN-AD technique initially performs a Z-score normalization process. For anomaly detection, the FOADESN-AD technique uses the DESN classifier, which accurately detects the presence of anomalies in the cloud environment. Moreover, the FOA is utilized to finetune the hyperparameter values of the DESN model, achieving superior classification results. The performance analysis of the FOADESN-AD method is implemented on the CSE-CICIDS-2018 dataset. The experimental values stated the betterment of the FOADESN-AD method over other existing approaches.*

Keywords *- Cloud computing, Falcon optimization algorithm, Anomaly detection, Security, Deep learning.*

1. Introduction

In the field of Information Technology (IT), one of the latest service innovations is Cloud computing (CC). The benefit of CC is that it allows access without limitations of time and location [1]. CC allows control of storage capacity, provides minimal costs and supports mobile and collaborative services or applications. Further, cloud services are multisource, letting the users utilize service providers as per their necessities [2].

The CC usage even lessens maintenance requirements for on-site storage, capital expenditures, physical space and power usage. Many companies, governments and banks have implemented CC technology since its services became more common [3]. This evolution even exposed such systems to different types of cyber-threats by intruders and hackers, demanding robust security systems. Cloud service companies grant quite a few security services as applications [4]. The Amazon Web Service (AWS) stores render services with limited dates and validity dependent upon the service license period [6]. Owing to its distributed nature, the cloud platform is prone to intrusion attacks [7]. An intrusion detection system (IDS) is adopted to ensure the security of CC by checking the configurations, logs, and network flow [8]. Still, existing IDS is network-or-host-based and unrealistic for CC; meanwhile, IDSs could not identify hidden attack paths. Numerous ways are there to avoid hacker intrusion. Also, a firewall as the first and IDS as the second line of defence was utilized for monitoring network traffic for abnormal behaviour [10].

In advance, an ID gathers a huge volume of malicious attack datasets and compares their behaviour patterns and database attack features to effectively ascertain intrusion data to defend against ransomware. Deep learning (DL) encompasses neural networks, including a multi-level structure not similar to a machine-learning (ML) network since it may process and learn features by itself and make changes in feature values.

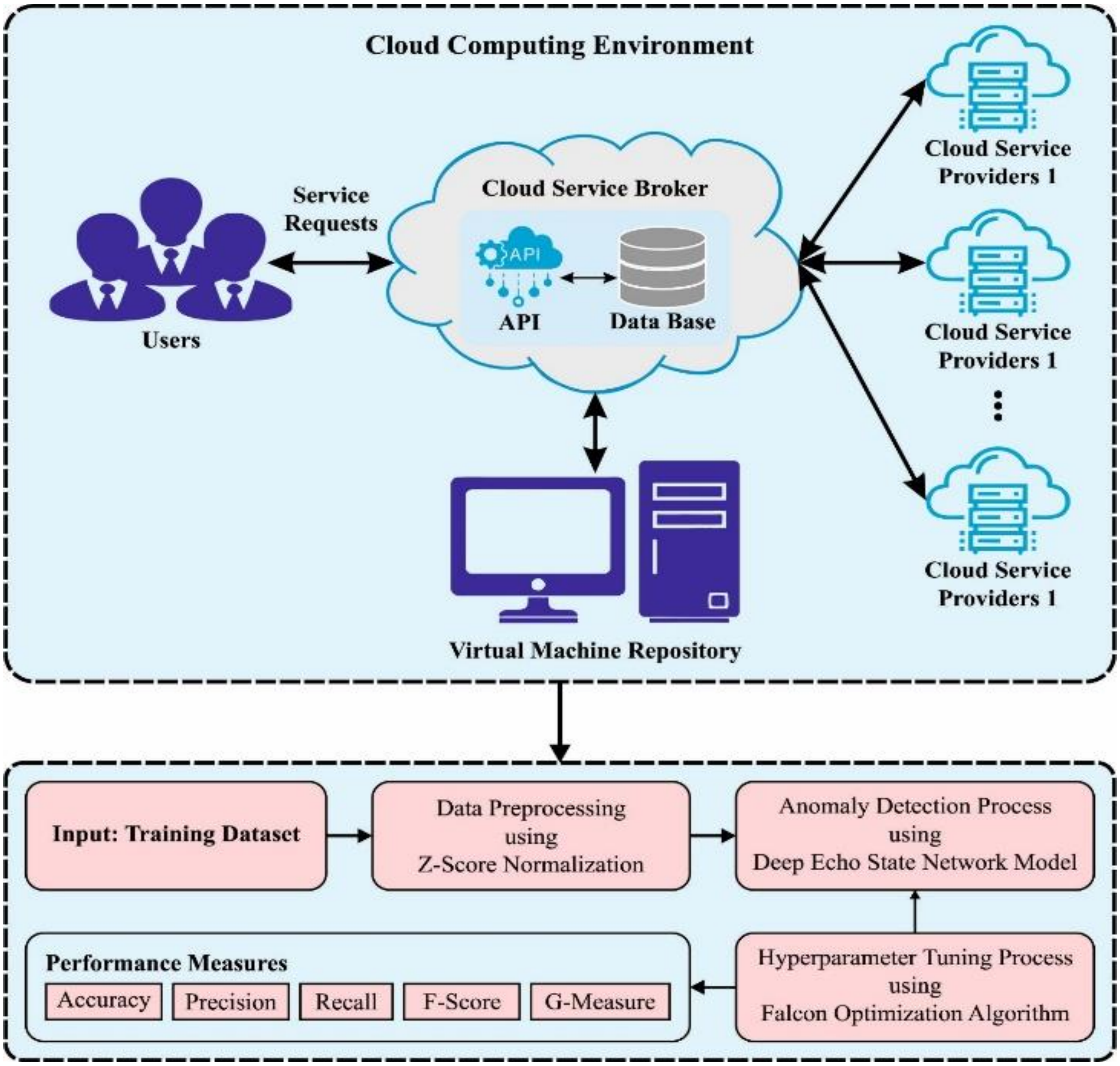

Cloud Computing Environment
Service Requests
Users
Cloud Service Broker
API
API
Data Base
Cloud Service Providers 1
Cloud Service Providers 1
Cloud Service Providers 1
Virtual Machine Repository
Input: Training Dataset
Data Preprocessing using Z-Score Normalization
Anomaly Detection Process using Deep Echo State Network Model
Performance Measures
Accuracy
Precision
Recall
F-Score
G-Measure
Hyperparameter Tuning Process using Falcon Optimization Algorithm

Fig. 1 Overall flow of FOADESN-AD approach

To deal with the fast expansion of big data [11], DL of automatic feature processing engineering is the production technique, and correct amalgamations of neurons and layers ought to be devised for extracting vital features and making decisions for large-scale data. The author has listed several articles on the DL application in network attack detection. Hence, it is appropriate to use DL to apply IDS [12].

This article introduces a new Falcon Optimization Algorithm with Deep Echo State Network for Anomaly Detection (FOADESN-AD) technique for a secure CC environment. The FOADESN-AD technique aims to perform accurate anomaly or intrusion detection in the cloud environment. Primarily, the FOADESN-AD technique carries out the Z-score normalization process. For anomaly detection, the FOADESN-AD technique uses the DESN classifier, which accurately detects the presence of anomalies in the cloud environment. Finally, the FOA is applied to finetune the parameter values of the DESN technique. The performance validation of the FOADESN-AD technique is carried out on the CICIDS-2017 dataset.

2. Related Works

Aldallal and Alisa [14] presented new ML-related hybrid IDS. It combines with Genetic Algorithm (GA) and Support Vector Machine (SVM) with a fitness function introduced to improve the accuracy level of the system. In this study, an SVM was employed degree, gamma utilizing various values of hyperparameters of the kernel function. Zhao et al. [15] present a technique based on ConvNeXt-Sf termed lightweight intrusion detection approach. Firstly, the 2D framework of ConvNeXt is minimalized to a 1D sequence. Next, to enhance ConvNeXt to make the latter more lightweight, the ShuffleNet are applied. Guezzaz et al. [16]

introduced an approach with the help of ML techniques named hybrid IDS for Edge-Based IIoT Security. Specifically, the KNN was merged to make effective decisions and enhance detection accuracy, and the PCA was exploited for enhanced feature training processes and engineering.

Hajimirzaei and Navimipour [18] suggest a new IDS depends on an integration of artificial bee colony (ABC), multilayer perceptron (MLP) network, and fuzzy clustering approaches. This MLP Abnormal and Normal network traffic packets were detected. In [19], the authors introduced a potential Hybrid clustering and classification method to implement a discrepancy-related IDS for malevolent attacks like normal (no intrusion), U2R, DoS, and R2L Probe, utilizing threshold-based functions, and with two threshold values, the results were tested.

Sakr et al. [20] devised an anomaly-based Network IDS (NIDS) where clouds analyse the network traffics flow. The network administrator must be notified regarding the nature of traffic to any intrusive network connections and block drops. As the classifier of network connection, SVM was adopted. The binary-based Particle Swarm Optimization (BPSO) is applied to choosing the relevant networking feature, while the standard-based PSO (SPSO) is implemented to tune the SVM control parameters. Issa and Albayrak [22] modelled a novel DL classifier approach by incorporating Short-Term Long Memory (LSTM) and Convolutional Neural Networks (CNNs).

3. The Proposed Model

In this article, a new FOADESN-AD technique was developed for accurate anomaly detection and classification in the secure CC platform using the DL algorithm with a hyperparameter tuning process. The presented FOADESN-AD technique encompasses three major processes: Z-score normalization, DESN-based detection, and FOA-based parameter tuning. The detailed working of these modules is explained below. Figure 1 represents the overall flow of the FOADESN-AD algorithm.

3.1. Z-Score Normalization

In the initial phase, the Z-score normalization is utilized to scale the input data into a uniform format. Here, the raw data gets modified using Eq. (1).

$$Z_{score} = \frac{X_i - X_{mean}}{S} \qquad (1)$$

Whereas X_i denotes the raw value that all the variables continue, X_{mean} implies the average of variable values, and S represents the standard deviation (SD). Therefore, raw ratios are normalizing, with its average as 0 and SD as a unit across samples.

3.2. Anomaly Detection using DESN Model

To detect the anomalies proficiently, the DESN model is exploited. The DESN is a deep network architecture where a single reservoir layer can be replaced with the ESN using a reservoir layer [23]. DESN is accomplished through the reservoir layer, input, and output layers. The input data series is fed into the overall network. The reserved layer computes and proceeds the dataset to attain the internal state. The output layer sets the internal state into the actual value to attain the model parameter. The DESN needs only matrix operation than the backpropagation (BP) model and does not need multiple iterations of the BP model. Moreover, it has a better prediction accuracy and feature extraction capability when compared to basic ESN.

Determine $u(n) = [u_1(n) \cdots u_k(n)]^T$ as an input sample at n time. The state of the reserved layer in the DESN is updated after every sample is inputted into the DESN:

$$x^{(l)}(n) = \left(1 - a^{(l)}\right)x^{(l)}(n-1) + a^{(l)} \tanh\left(W_{in}^{(l)} i^{(l)}(n) + \theta^{(l)} + \widehat{W}x^{(l)}(n-1)\right) \qquad (2)$$

$$i^{(l)}(n) = \begin{cases} u(n), l = 1 \\ x^{(l-1)}(n), l > 1 \end{cases} \qquad (3)$$

In Eq. (2) and (3), $i^{(l)}(n)$ denotes the input of the reserved layer once l is bigger than 1, then the input layer of the reserved layer is the state of the reserved layer, while l is 1, then the input of the reserved layer is the original sample. $a^{(l)}$ signifies the leaking rate of the l-th layer, $x^{(l)}(n)$ signifies the state attained by the l-th layer reserved layer once the initial sample is inputted. $\widehat{W}$ means the internal matrix of the reserved layer, tanh denotes the tangent function, and $W_{in}^{(l)}$ and $\widehat{W}$ shows the fixed parameter.

Lastly, the state of every reserved layer is collected to attain the state of overall networking.

$$x(n) = [x^{(1)}(n)x^{(2)}(n) \cdots x^{(l)}(n)]^T \qquad (4)$$

The state and the output weight collected by the network are multiplied by the matrix to obtain the output value of the last network.

$$y(n) = g\left(W_{out}x(n)\right) \qquad (5)$$

Where W_{oui} denotes the variable which needs to be only trained. The output weight W_{out} is the model parameter that should be evaluated. Arrange each $x(n)$ and $y(n)$ by row to attain X and Y, the final model parameter. W_{oui} is evaluated by the subsequent equation:

$$W_{out} = Y\gamma(XX^T + \alpha E)^{-1} \qquad (6)$$

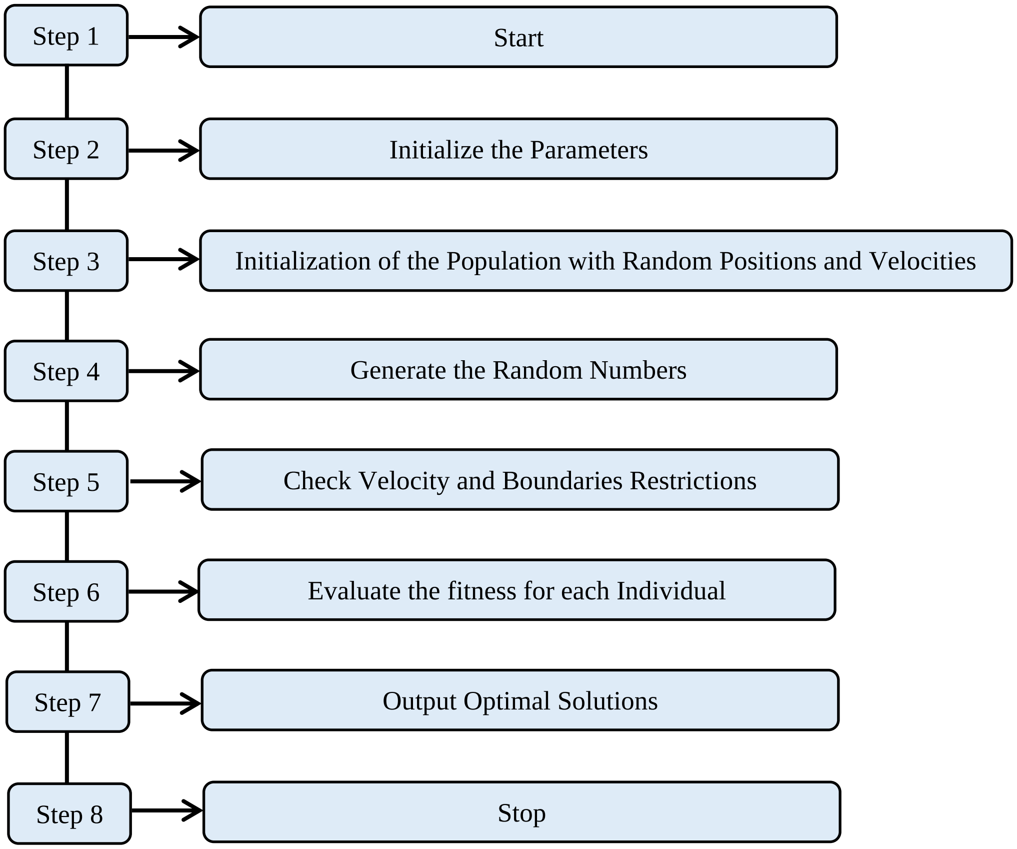

Step 1
Start
Step 2
Initialize the Parameters
Step 3
Initialization of the Population with Random Positions and Velocities
Step 4
Generate the Random Numbers
Step 5
Check Velocity and Boundaries Restrictions
Step 6
Evaluate the fitness for each Individual
Step 7
Output Optimal Solutions
Step 8
Stop

Fig. 2 Steps involved in FOA

In Eq. (6), the T superscript signifies the matrix transpose, α denotes the regularization coefficient, E represent the unit matrix, and -1 signifies the inverse of the smatrix. The derivation principle of Eq. (6) and is resultant of the loss function $\mathrm{L} = |\mathrm{W}_{\mathrm{out}}\mathrm{X} - \gamma| + \alpha\mathrm{W}_{\mathrm{out}}$, in which L represent the loss matrix.

3.3. Hyperparameter Tuning using FOA

The FOA is used for finetuning the hyperparameter value of the DESN. FOA is a more robust and reliable algorithm for the stochastic population-based problem that necessitates arrangement from the number of variables to its three-step action settlement [24]. The inspiration behind the FOA is the chasing strategy of falcons when looking for their prey during the fight. Falcon takes multiple paths to get closer towards the prey. All the routes have two major parts: the former is a logarithmic spiral where a falcon continuously keeps its head straight when looking toward the prey with maximum visible acuity. Thereby locomotion can be attained using the three stages: initially, falcons explore for prey; then enhance their dives through a logarithmic spiral; and later falcon dive itself (that might lead to success, viz., prey acquisition). Or else, a falcon rapidly reverses its action. Figure 2 depicts the steps involved in FOA.

The FOA implementation can be explained in the following five steps as follows.

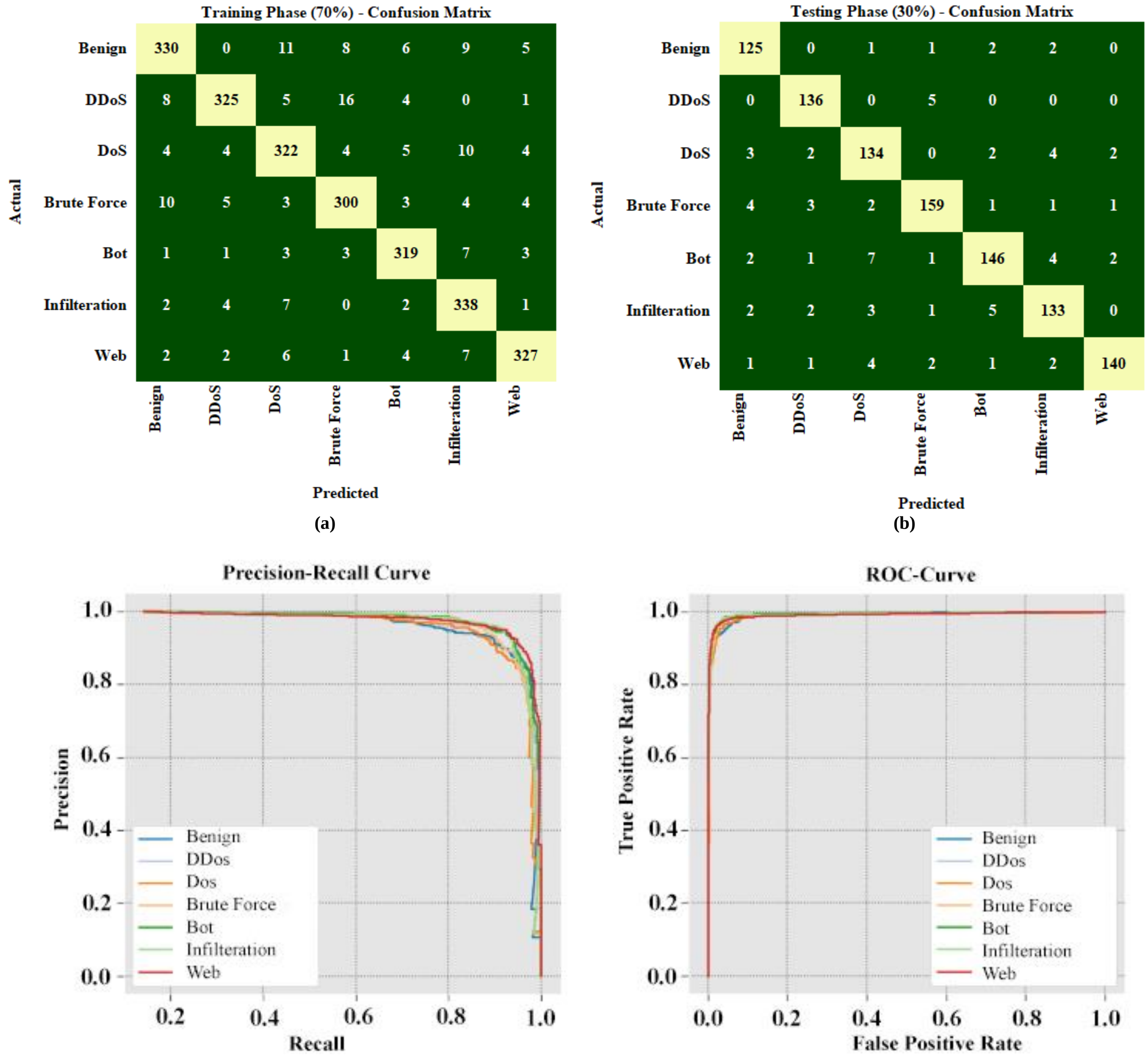
Step 1 **:** Begin the process by finetuning the parameter for the optimization problems involving the highest speed (Vmax), number of falcons (NP), social (sc) constant, cognitive rate (cc), dive probability (DP), following (fc) constant, and awareness probability (AP).

Step 2 **:** Randomly sets the velocity and location of the falcon in a D-dimensional space, where the location of all the falcons is determined by the amount of NP applicants within D dimension. The speed is produced arbitrarily amongst the Vmax, and Vmin limitations are correspondingly given as follows:

$$\mathrm{Vmax} = 0.1 \cdot \mathrm{ub} \quad (7)$$

$$\mathrm{Vmin} = -\mathrm{Vmax} \quad (8)$$

Where ub signifies the upper bound of every dimension. At first, the pairs of random integers $(\mathrm{pAP}, \mathrm{pDP})$ are generated for all the falcons for correspondence amongst the DP and AP.



(a)

(b)

(a)

(b)

Fig. 3 Classifier outcome of (a-b) Confusion matrices, (c) PR curve, and (d) ROC curve

Step 3 **:** Evaluate the fitness value and select the global (gbest) and best (xbest) sites. The position selected is for generating novel locations as pAP and pDP.

Step 4 **:** New location is generated, as well as updated the falcon position is. Next, pAP was compared to the AP; if AP is greater than pAP, then the falcon will move from looking for prey:

$$X_{iter+1} = X_{iter} + V_{iter} + cc(X_{best}, X_{iter} + sc(g_{best}, X_{iter}) \qquad (9)$$

In Eq. (9), V_{iter} denotes the present velocity, and If pAP is greater than AP, then compare a DP amongst pDP. If DP is lesser than pDP, then the target one is selected as prey by the falcon (X_{chosen}) and complete its essential step towards hunting:

$$x_{iter+1} = X_{iter} + |X_{chosen} - x_{iter}|.\, exp^{bt} cos(2\pi t) \qquad (10)$$

In Eq. (10), X_{iter} indicates the existing location of the falcon. b denotes the static number that defines the state of the spiral logarithm (equivalent to 1), and t shows the arbitrary integer within $(-1,1)$ that defines the next position with regard to its accurate destination. If AD is greater than pAP, a comparation of the score function of the falcon and the score function of desired prey.:

$$X_{iter+1} = X_{iter} + V_{iter+1} + fc \cdot \text{rand}(X_{chosen} - X_{iter}) \quad (11)$$

Then, the falcon continued to fly based on its better location:

$$X_{iter+1} = X_{iter} + V_{iter+1} + cc.\,\text{rand}(X_{best}, X_{iter}) \quad (12)$$

Later, the newest position is estimated concerns the location boundaries and velocities. Then, its newest score function is calculated, and the novel value of X_{est} and gbest are defined.

Step 5 : Last, succeeding evaluations of Step 4 are repeated until the maximum amount of iterations (itermax) is attained.

The FOA method also derives fitness functions for achieving superior accuracy of classification. It states a positive integer to indicate the enhanced accomplishment of the candidate solution. The dimension of classification error rate can be treated as the fitness function, as given as follows.

$$\text{fitness}(x_i) = \text{ClassifierErrorRate}(x_i)$$

$$= \frac{\text{number of misclassified samples}}{\text{Total number of samples}} * 100 \quad (13)$$

4. Results and Discussion

In this segment, the performance analysis of the FOADESN-AD technique is tested on the CSE-CICIDS 2018 Dataset [25], which holds 3500 samples with 7 class labels. Each class holds a set of 500 samples. It comprises the networking traffic and logs records of all the machines from the victim side, together with 80 networking traffic features extracted from captured traffic through CICFlowMeterV3.

Figure 3 illustrates the classifier outcome of the ABCFS-OHML method under the test dataset. Figures 3a-3b demonstrates the confusion matrix the ACC-CBOEFF method offers on 70:30 of TRP/TSP. The figure denoted that the ACC-CBOEFF method accurately recognized and classified all 7 class labels. Similarly, Figure 3c demonstrates the PR analysis of the ABCFS-OHML model. The figures reported that the ABCFS-OHML method had obtained maximum PR performance under all classes. Finally, Figure 3d illustrates the ROC investigation of the ABCFS-OHML model.

Table 1. Detection outcome of FOADESN-AD approach on 70% of TRP

Training Phase (70%)					
Class	**Accu$_y$**	**Prec$_n$**	**Reca$_l$**	**F$_{score}$**	**G$_{measure}$**
Benign	97.31	92.44	89.43	90.91	90.92
DDoS	97.96	95.31	90.53	92.86	92.89
DoS	97.31	90.20	91.22	90.70	90.71
Brute Force	97.51	90.36	91.19	90.77	90.77
Bot	98.29	93.00	94.66	93.82	93.83
Infilteration	97.84	90.13	95.48	92.73	92.77
Web	98.37	94.78	93.70	94.24	94.24
Average	97.80	92.32	92.31	92.29	92.30

Table 2. Detection outcome of FOADESN-AD method on 30% of TSP

Testing Phase (30%)					
Class	**Accu$_y$**	**Prec$_n$**	**Reca$_l$**	**F$_{score}$**	**G$_{measure}$**
Benign	98.29	91.24	95.42	93.28	93.31
DDoS	98.67	93.79	96.45	95.10	95.11
DoS	97.14	88.74	91.16	89.93	89.94
Brute Force	97.90	94.08	92.98	93.53	93.53
Bot	97.33	92.99	89.57	91.25	91.27
Infilteration	97.52	91.10	91.10	91.10	91.10
Web	98.48	96.55	92.72	94.59	94.61
Average	97.90	92.64	92.77	92.68	92.70

The figure depicted that the ABCFS-OHML method has resulted in proficient outcomes with the highest ROC values under dissimilar class labels.

In Table 1 and Figure 4, the detection outcomes of the FOADESN-AD method on 70% of TRP are given. The results represent the effectual detection ability of the FOADESN-AD technique on different classes. For instance, on benign class, the FOADESN-AD technique obtains $accu_y$ of 97.31%, $prec_n$ of 92.44%, $reca_l$ of 89.43%, F_{score} of 90.91%, and $G_{measure}$ of 90.92%. Concurrently, in the DoS class, the FOADESN-AD method obtains $accu_y$ of 97.31%, $prec_n$ of 90.20%, $reca_l$ of 91.22%, F_{score} of 90.70%, and $G_{measure}$ of 90.71%.

Simultaneously, in Bot class, the FOADESN-AD technique obtains $accu_y$ of 98.29%, $prec_n$ of 93%, $reca_l$ of 94.66%, F_{score} of 93.82%, and $G_{measure}$ of 93.83%. Finally, in web class, the FOADESN-AD method obtains $accu_y$ of 98.37%, $prec_n$ of 94.78%, $reca_l$ of 93.70%, F_{score} of 94.24%, and $G_{measure}$ of 94.24%.

In Table 2 and Figure 5, the detection outcomes of the FOADESN-AD approach on 30% of TSP are given. The results represent the effectual detection ability of the FOADESN-AD technique on different classes. For instance, on benign class, the FOADESN-AD method attains $accu_y$ of 98.29%, $prec_n$ of 91.24%, $reca_l$ of 95.42%, F_{score} of 93.28%, and $G_{measure}$ of 93.31%.

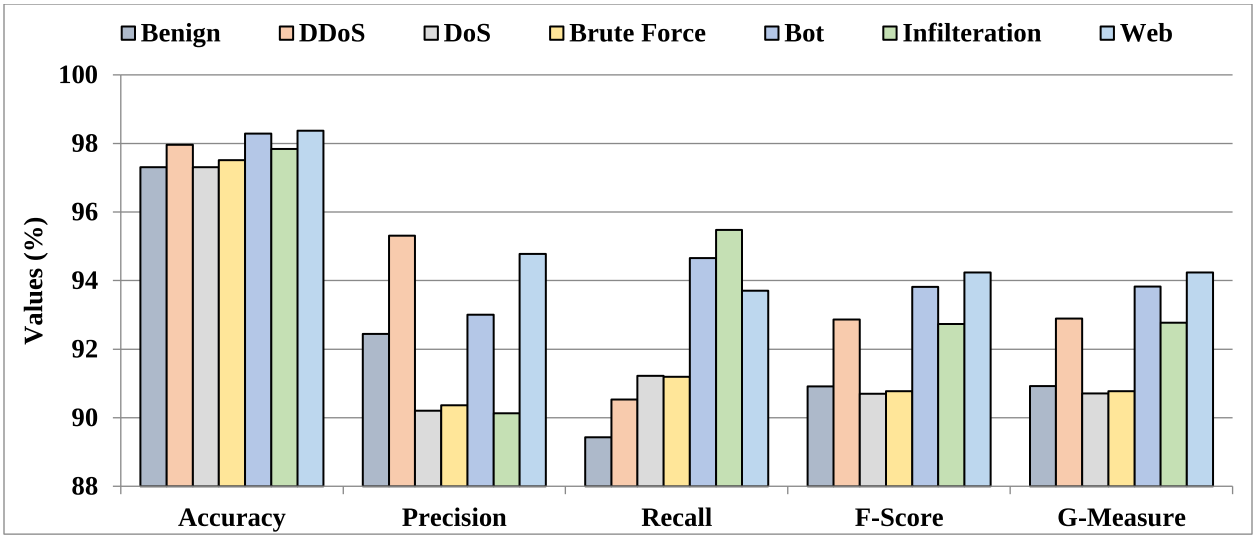


Fig. 4 Detection outcome of FOADESN-AD method on 70% of TRP

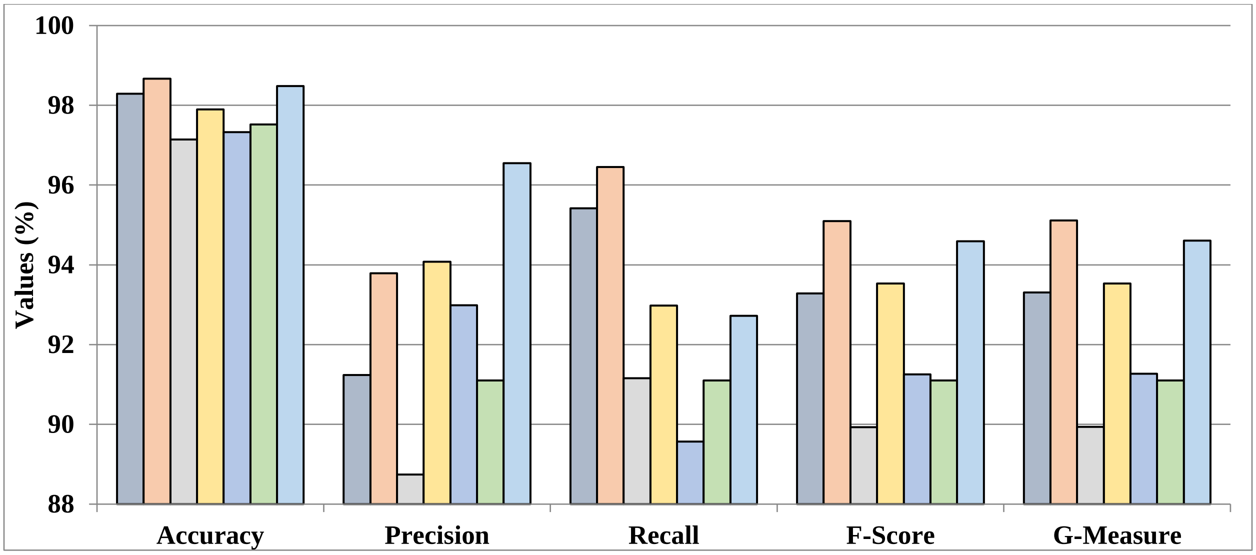


Fig. 5 Detection outcome of FOADESN-AD approach on 30% of TSP

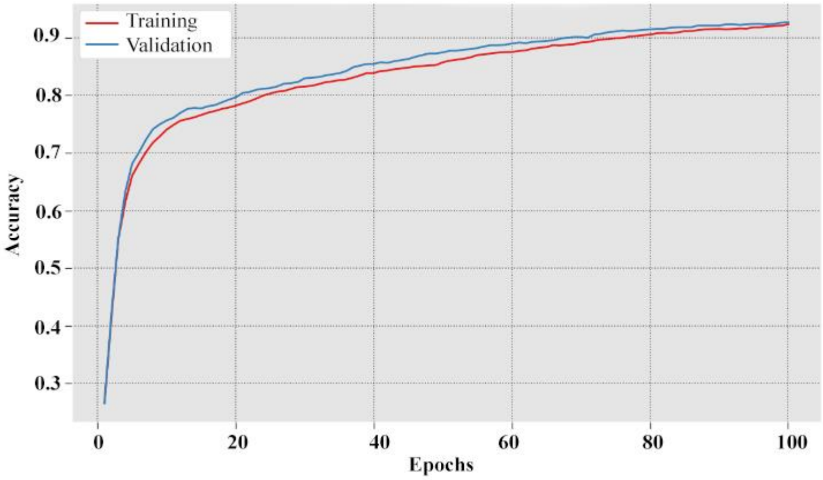


Fig. 6 Accuracy curve of the FOADESN-AD approach

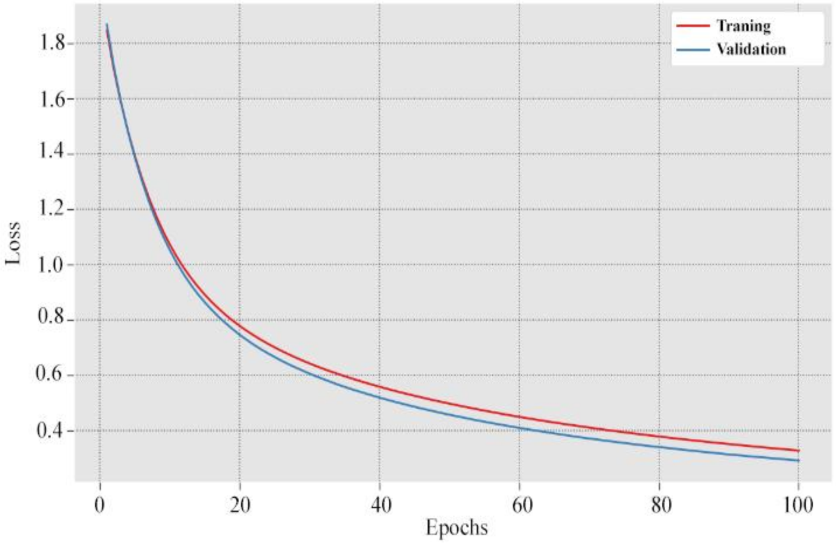


Fig. 7 Loss curve of the FOADESN-AD approach

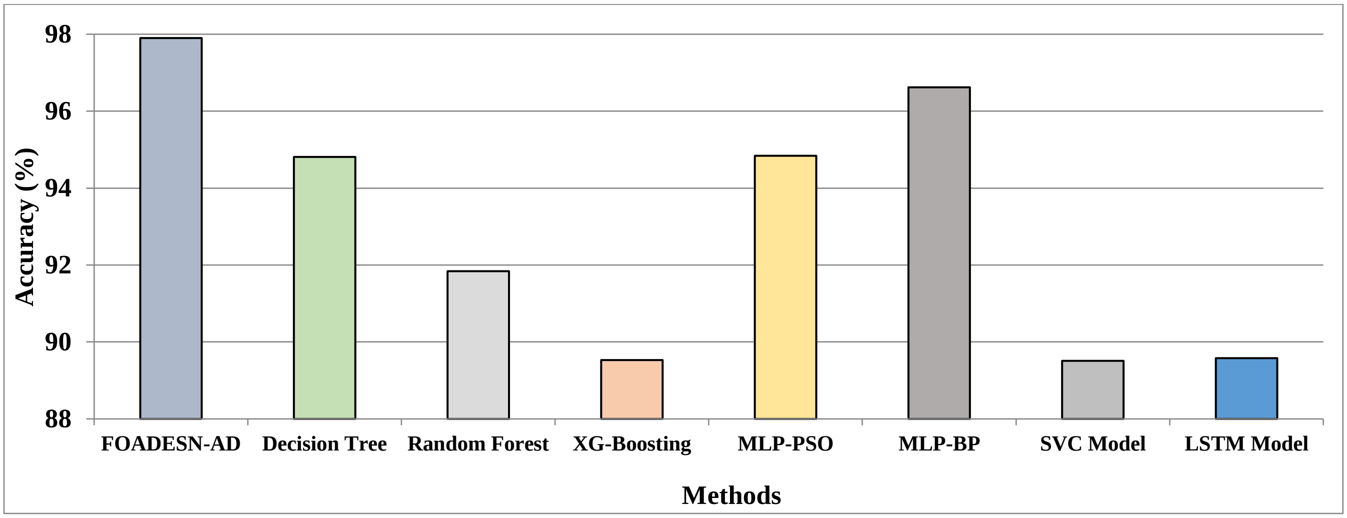


Fig. 8 Accu$_y$ the outcome of the FOADESN-AD approach with existing methods

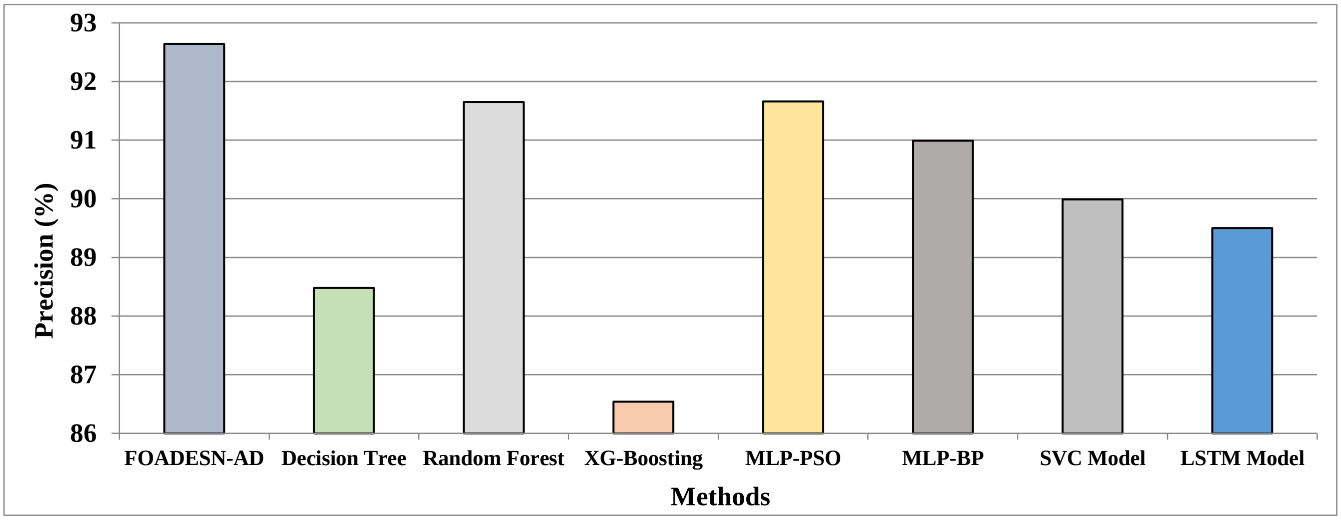


Fig. 9 Prec$_n$ the outcome of the FOADESN-AD approach with existing methods

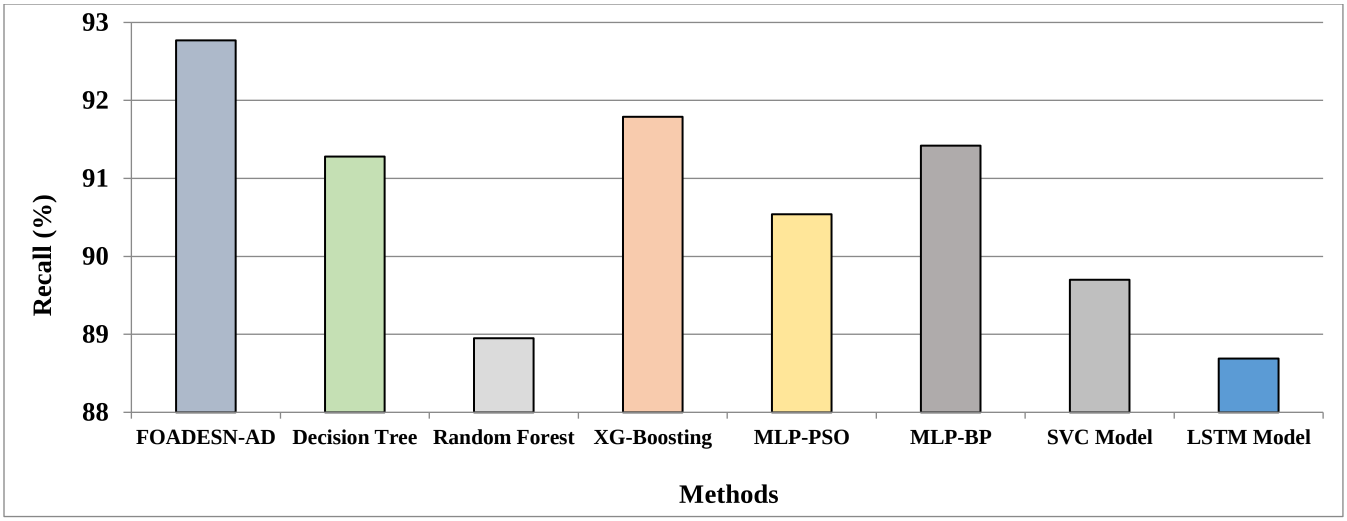


Fig. 10 Reca$_l$ the outcome of the FOADESN-AD approach with existing methods

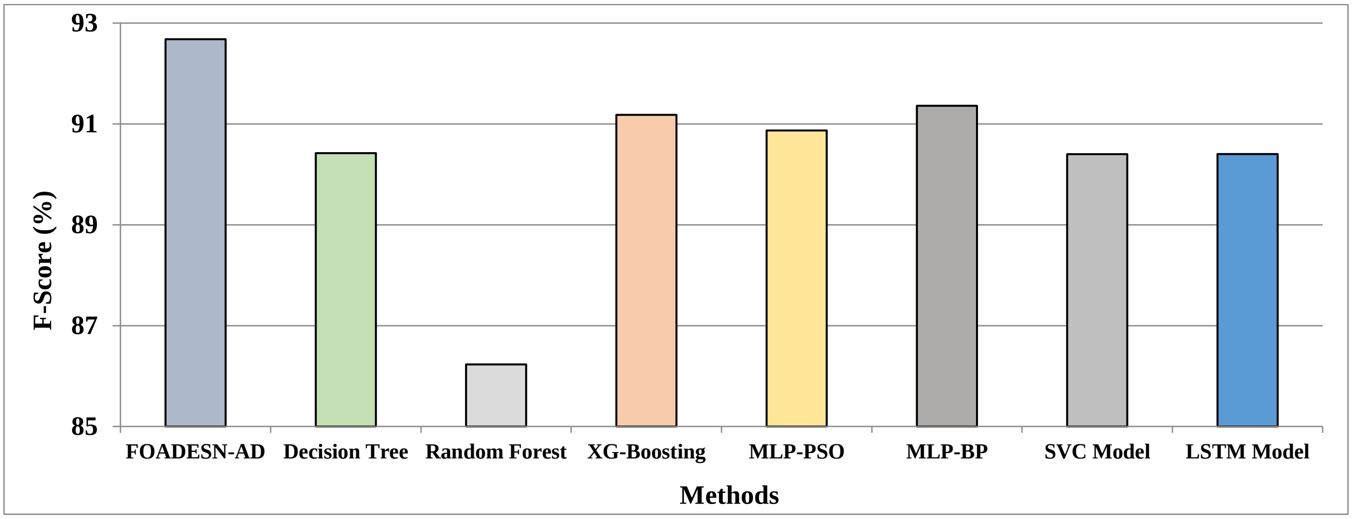


Fig. 11 F$_{score}$ the outcome of the FOADESN-AD approach with existing methods

Concurrently, in the DoS class, the FOADESN-AD method attains $\mathrm{accu_y}$ of 97.14%, $\mathrm{prec_n}$ of 88.74%, $\mathrm{reca_l}$ of 91.16%, $\mathrm{F_{score}}$ of 89.93%, and $\mathrm{G_{measure}}$ of 89.94%. Concurrently, in Bot class, the FOADESN-AD method attains $\mathrm{accu_y}$ of 97.33%, $\mathrm{prec_n}$ of 92.99%, $\mathrm{reca_l}$ of 89.57%, $\mathrm{F_{score}}$ of 91.25%, and $\mathrm{G_{measure}}$ of 91.27%. Finally, in web class, the FOADESN-AD technique obtains $\mathrm{accu_y}$ of 98.48%, $\mathrm{prec_n}$ of 96.55%, $\mathrm{reca_l}$ of 92.72%, $\mathrm{F_{score}}$ of 94.59%, and $\mathrm{G_{measure}}$ of 94.61%.

Figure 6 examines the accuracy of the FOADESN-AD technique during the training and validation process on the test dataset. The figure notifies that the FOADESN-AD method reaches increasing accuracy values over increasing epochs. In addition, the increasing validation accuracy over training accuracy exhibits that the FOADESN-AD technique learns efficiently on the test dataset.

The loss analysis of the FOADESN-AD method at the time of training and validation is demonstrated on the trial dataset in Figure 7. The outputs indicate that the FOADESN-AD method reaches closer values of training and validation loss. The FOADESN-AD technique learns efficiently on the test dataset.

The extensive results of the FOADESN-AD technique are compared with recent approaches in Table 3 [27-31]. Figure 8 provides a comparative $\mathrm{accu_y}$ inspection of the FOADESN-AD with existing methods. The experimental results indicate that the XG-Boosting, SVC, and LSTM models provide lower $\mathrm{accu_y}$ of 89.53%, 89.51%, and 89.58% respectively. Then, the DT, MLP-PSO, and RF models obtain closer $\mathrm{accu_y}$ of 94.81%, 94.84%, and 91.84%, respectively. Although the MLP-BP model attains reasonable $\mathrm{accu_y}$ of 96.62%, the FOADESN-AD technique gains higher $accu_y$ of 97.90%.

Table 3. Comparative outcome of FOADESN-AD approach with existing methods

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
FOADESN-AD	97.90	92.64	92.77	92.68
Decision Tree	94.81	88.48	91.28	90.42
Random Forest	91.84	91.65	88.95	86.23
XG-Boosting	89.53	86.54	91.79	91.18
MLP-PSO	94.84	91.66	90.54	90.87
MLP-BP	96.62	90.99	91.42	91.36
SVC Model	89.51	89.99	89.70	90.40
LSTM Model	89.58	89.50	88.69	90.40

Figure 9 provides a comparative $prec_n$ assessment of the FOADESN-AD with existing methods. The experimental results indicate that the XG-Boosting, DT, LSTM, and SVC methods provide lower $prec_n$ of 86.54%, 88.48%, 89.50%, and 89.99% correspondingly. Then, the RF and MLP-BP models obtain closer $prec_n$ of 91.65% and 90.99% correspondingly. Although the MLP-PSO model attains reasonable $prec_n$ of 91.66%, the FOADESN-AD technique gains higher $prec_n$ of 92.64%. Figure 10 provides a comparative $reca_l$ investigation of the FOADESN-AD with existing techniques. The experimental results indicate that the LSTM, RF, SVC, and MLP-PSO models provide lower $reca_l$ of 88.69%, 88.95%, 89.70%, and 90.54% correspondingly. Then, the DT and MLP-BP methods obtain closer $reca_l$ of 91.28% and 91.42%, correspondingly. Although the XG-Boosting method attains reasonable $reca_l$ of 91.79%, the FOADESN-AD technique gains higher $reca_l$ of 92.77%.

Figure 11 provides a comparative F_{score} inspection of the FOADESN-AD with existing methods. The experimental results indicate that the RF, SVC, LSTM, and DT methods provide lower F_{score} of 86.23%, 90.40%, 90.40%, and 90.42%, respectively. Then, the XG-Boosting and MLP-PSO method attains closer F_{score} of 91.18% and 90.87%, correspondingly. Although the MLP-BP method obtains reasonable F_{score} of 91.36%, the FOADESN-AD technique gains a higher F_{score} of 92.68%. These results confirmed the enhanced performance of the FOADESN-AD technique.

5. Conclusion

In this article, a new FOADESN-AD method was developed for accurate anomaly detection and classification in the secure CC platform using the DL model with a hyperparameter tuning process. The presented FOADESN-AD technique encompasses three major processes: Z-score normalization, DESN-based detection, and FOA-based parameter tuning. In the presented FOADESN-AD technique, the FOA is utilized to finetune the hyperparameter values of the DESN model, achieving improved classification results. The performance analysis of the FOADESN-AD method is implemented on the CICIDS-2017 dataset. The experimental analysis stated the betterment of the FOADESN-AD algorithm over other existing approaches. In the future, the dimensionality reduction technique can improve the FOADESN-AD method's efficiency.

References

[1] V. Kanimozhi, and T. Prem Jacob, “Artificial Intelligence Outflanks All Other Machine Learning Classifiers in Network Intrusion Detection System on the Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing,” *ICT Express*, vol. 7, no. 3, pp. 366-370, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Theyazn H. H. Aldhyani, and Hasan Alkahtani, “Artificial Intelligence Algorithm-Based Economic Denial of Sustainability Attack Detection Systems: Cloud Computing Environments,” *Sensors*, vol. 22, no. 13, p. 4685, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Haifeng Lin et al., “Internet of Things Intrusion Detection Model and Algorithm Based on Cloud Computing and Multi-Feature Extraction Extreme Learning Machine,” *Digital Communications and Networks*, vol. 9, no. 1, pp. 111-124, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] K.P. Sanal Kumar et al., “Security and Privacy-Aware Artificial Intrusion Detection System Using Federated Machine Learning,” *Computers & Electrical Engineering*, vol. 96, p. 107440, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dr. S.Veerapandi, Dr. R.Surendiran, and Dr. K.Alagarsamy, "Enhanced Fault Tolerant Cloud Architecture to Cloud-based Computing using Both Proactive and Reactive Mechanisms," *DS Journal of Digital Science and Technology*, vol. 1, no. 1, pp. 32-40, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Noah Oghenefego Ogwara, Krassie Petrova, and Mee Loong Yang, “Towards the Development Of A Cloud Computing Intrusion Detection Framework Using An Ensemble Hybrid Feature Selection Approach,” *Journal of Computer Networks and Communications*, *2022*, pp.1-16. [CrossRef] [Google Scholar] [Publisher Link]

[7] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco, “Practical Autoencoder-Based Anomaly Detection By Using Vector Reconstruction Error,” *Cybersecurity*, vol. 6, no. 1, p. 1, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Muhammad Asif et al., “MapReduce-based Intelligent Model for Intrusion Detection Using Machine Learning Technique,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9723-9731, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Dr. S.Veerapandi, Dr.R.Surendiran, and Dr. K.Alagarsamy, "Live Virtual Machine Pre-copy Migration Algorithm for Fault Isolation in Cloud Based Computing Systems," *DS Journal of Digital Science and Technology*, vol. 1, no. 1, pp. 23-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Maamar Ali Saud AL Tobi et al., "Machinery Faults Diagnosis using Support Vector Machine (SVM) and Naïve Bayes Classifiers," *International Journal of Engineering Trends and Technology*, vol. 70, no. 12, pp. 26-34, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Shashikala, and G. K. Ravikumar, "Deep Studying Signature for Obstruction Obscure in Copy Move Image Forgeries," *International Journal of Engineering Trends and Technology*, vol. 70, no. 10, pp. 262-270, 2022. [CrossRef] [Publisher Link]

[12] Stanislav Yamashkin et al., "Metageosystem Analysis Based on a System of Machine Learning and Simulation Algorithms," *International Journal of Engineering Trends and Technology*, vol. 70, no. 12, pp. 1-12, 2022. [CrossRef] [Publisher Link]

[13] Dr. R.Surendiran, and Prof. K. Raja, " A Fog Computing Approach for Securing IoT Devices Data using DNA-ECC Cryptography," *DS Journal of Digital Science and Technology*, vol. 1, no. 1, pp. 10-16, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ammar Aldallal, and Faisal Alisa, “Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning,” *Symmetry*, vol. 13, no. 12, p. 2306, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Guosheng Zhao, Yang Wang, and Jian Wang, “Lightweight Intrusion Detection Model of the Internet of Things with Hybrid Cloud-Fog Computing,” *Security and Communication Networks*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Azidine Guezzaz et al., “A Lightweight Hybrid Intrusion Detection Framework Using Machine Learning for Edge-Based Iiot Security,” *The International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 1-9, 2022. [Google Scholar] [Publisher Link]

[17] Ahmad Shokuh Saljoughi, Mehrdad Mehrvarz, and Hamid Mirvaziri, “Attacks and Intrusion Detection In Cloud Computing Using Neural Networks And Particle Swarm Optimization Algorithms,” *Emerging Science Journal*, vol. 1, no. 4, pp. 179-191, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Bahram Hajimirzaei, and Nima Jafari Navimipour, “Intrusion Detection for Cloud Computing Using Neural Networks and Artificial Bee Colony Optimization Algorithm,” *Ict Express*, vol. 5, no. 1, pp. 56-59, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] K. Samunnisa, G. Sunil Vijaya Kumar, and K. Madhavi, “Intrusion Detection System in Distributed Cloud Computing: Hybrid Clustering and Classification Methods,” *Measurement: Sensors*, vol. 25, p. 100612, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mahmoud M. Sakr, Medhat A. Tawfeeq, and Ashraf B. El-Sisi, “Network Intrusion Detection System Based PSO-SVM for Cloud Computing,” *International Journal of Computer Network and Information Security*, vol. 11, no. 3, p. 22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Pawan Jaybhaye, and Dr. Bandu B. Meshram, "Malware Detection and Prevention on Cloud," *International Journal of Computer and Organization Trends*, vol. 9, no. 4, pp. 5-10, 2019. [CrossRef] [Publisher Link]

[22] Ahmet Sardar Ahmed Issa, and Zafer Albayrak, “DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM,” *Acta Polytechnica Hungarica*, vol. 20, no. 2, 2023. [Google Scholar]

[23] Yu-Ting Bai et al., “Nonstationary Time Series Prediction Based on Deep Echo State Network Tuned by Bayesian Optimization,” *Mathematics*, vol. 11, no. 6, p. 1503, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Kareem, S.W. and Okur, M.C., 2021. Falcon optimization algorithm for bayesian network structure learning. *Computer Science*, *22*.

[25] [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/

[26] Phyu Thi Htun, Kyaw Thet Khaing "Anomaly Intrusion Detection System using Random Forests and k-Nearest Neighbor," *International Journal of P2P Network Trends and Technology*, vol. 3, no. 1, pp. 39-43, 2013. [Google Scholar] [Publisher Link]

[27] Hsiao-Chung Lin et al., “Ensemble Learning for Threat Classification in Network Intrusion Detection on a Security Monitoring System for Renewable Energy,” *Applied Science*, vol. 11, no. 23, p. 11283, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ogobuchi Daniel Okey et al., “BoostedEnML: Efficient Technique for Detecting Cyberattacks in IoT Systems Using Boosted Ensemble Machine Learning, *Sensors*, vol. 22, p. 7409, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Saud Alzughaib, and Salim El Khediri, “Cloud Intrusion Detection Systems Based on DNN Using Backpropagation and PSO on the CSE-CIC-IDS2018 Dataset,” *Applied Science*, vol. 13, p. 2276, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] Balajee R M, and Jayanthi Kannan M K, “Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm,” *Electronics*, vol. 12, p. 1423, 2023. [CrossRef] [Google Scholar] [Publisher Link]