Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033497
Article

Enhanced Coyote Optimization with Deep Learning Based Cloud-Intrusion Detection System

Abdullah M. Basahel[1], Mohammad Yamin[1], Sulafah M. Basahel[2] and E. Laxmi Lydia[3,*]

[1]Faculty of Economics and Administration, King Abdulaziz University, Jeddah, Saudi Arabia
[2]E-commerce Department, College of Administrative and Financial Sciences, Saudi Electronic University, Jeddah, Saudi Arabia
[3]Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
*Corresponding Author: E. Laxmi Lydia. Email: elaxmi2002@yahoo.com
Received: 18 June 2022; Accepted: 19 August 2022

Abstract: Cloud Computing (CC) is the preference of all information technology (IT) organizations as it offers pay-per-use based and flexible services to its users. But the privacy and security become the main hindrances in its achievement due to distributed and open architecture that is prone to intruders. Intrusion Detection System (IDS) refers to one of the commonly utilized system for detecting attacks on cloud. IDS proves to be an effective and promising technique, that identifies malicious activities and known threats by observing traffic data in computers, and warnings are given when such threats were identified. The current mainstream IDS are assisted with machine learning (ML) but have issues of low detection rates and demanded wide feature engineering. This article devises an Enhanced Coyote Optimization with Deep Learning based Intrusion Detection System for Cloud Security (ECODL-IDSCS) model. The ECODL-IDSCS model initially addresses the class imbalance data problem by the use of Adaptive Synthetic (ADASYN) technique. For detecting and classification of intrusions, long short term memory (LSTM) model is exploited. In addition, ECO algorithm is derived to optimally fine tune the hyperparameters related to the LSTM model to enhance its detection efficiency in the cloud environment. Once the presented ECODL-IDSCS model is tested on benchmark dataset, the experimental results show the promising performance of the ECODL-IDSCS model over the existing IDS models.

Keywords: Intrusion detection system; cloud security; coyote optimization algorithm; class imbalance data; deep learning

1 Introduction

Cloud Computing (CC) presents on-demand network accessibility to a group of configurable computing assets such as servers, networks, services, storage, and applications, that can be quickly

revealed with service provider interaction or lesser management attempts [1]. Whereas it renders several advantages, one such major difficulty for associations seeking to implement cloud-based solutions was safety. The reason behind this was due to the cloud infrastructure nature that is completely dispersed and open [2], therefore becoming more prone to attacks and threats. This atmosphere forms inducements for effective invaders for initiating assaults which target the gadgets with accessibility to data saved on a cloud [3]. The menaces because of assaults were to the availability, integrity, and confidentiality of sources and cloud services. For instance, a Distributed Denial of Service (DDoS) attack, was the one which focuses on preventing accessibility of data saved on a cloud, by choking bandwidth of the network via packet flooding [4]. Other effective assault kinds involve Port Scanning, Internet Protocol (IP) spoofing, Man in the Middle Attack, Domain Naming System (DNS) poisoning, and so on. Cloud security was an exciting active domain of research works and several heuristics emerged and were suggested. Fundamental security components like a firewall which gives protection to the internal network and adopts message encoding is used as primary lines of defence [5]. But a firewall cannot find an assault commenced by an insider.

Nowadays, the diverse intrusion detection system (IDS) methods and tools were accessible in the cloud platform for finding assaults on cloud infrastructure [6]. The conventional or prevailing IDS becoming unsuccessful and ineffective because of dynamic behavior and heavy traffic. IDS typically accompaniment a firewall for forming an effectual security solution. Such factors lead to a surge in developing a creative way of improving the IDS in CC [7]. That could observe network traffic and efficiently detects network intrusions. This study was aimed at network-related security assaults on the public cloud atmosphere and suggested solutions and proposals. Interpretation from the publication which does not cover the number of countless protections towards network assaults [8]. Irrespective of the substantial number of endeavours performed by security experts in time of recent 2 decades, the system security problem not searched. The reason for this was the speedy development in computation power and available sources to invaders that allows them in launching complicated assaults [9]. An IDS becomes the most commonly utilized method to detect assaults on cloud environments. The growth of effective IDS becomes unsuccessful for placement in cloud atmosphere because of its distributed and virtualized nature [10].

This article devises an Enhanced Coyote Optimization with Deep Learning based Intrusion Detection System for Cloud Security (ECODL-IDSCS) model. The ECODL-IDSCS model initially addresses the class imbalance data problem by the use of Adaptive Synthetic (ADASYN) technique. For detecting and classification of intrusions, long short term memory (LSTM) model is exploited. In addition, ECO algorithm is derived to optimally fine tune the hyperparameters related to the LSTM model to enhance its detection efficiency in the cloud environment. Once the presented ECODL-IDSCS model is tested on benchmark dataset, the experimental results show the promising performance of the ECODL-IDSCS model over the existing IDS models.

2 Related Works

In [11], cloud IDS at hypervisor layer can be advanced and evaluated for detecting the depraved activities in CC atmosphere. The cloud IDS employs a hybrid method that is a blend of improved fuzzy c-means (FCM) clustering method and Back propagation artificial Neural Network (ANN) for improving the accuracy of detection of cloud IDS. The suggested technique can be applied and compared with classic FCM and K-means. In [12], a host-oriented IDS (H-IDS) for protection of virtual machines in the cloud atmosphere was suggested. To end this, initially, significant features of every class were chosen by making use of logistic regression (LR) and such values were enhanced utilizing

the regularization method. Afterward, several attacks were categorized by leveraging a mixture of 3 distinct classifiers they are neural network (NN), decision tree (DT), and linear discriminate analysis (LDA) with bagging technique for every class.

Chiba et al. [13] suggest to optimize a very famous soft computing tool broadly utilized for IDS such as Back Propagation Neural Network (BPNN) employing an Improved Genetic Algorithm (IGA). Genetic algorithm (GA) can be enhanced via optimizing policies, like Fitness Value Hashing and Parallel Processing that minimize convergence period, performance period, and save processing power. As, learning rate and Momentum terminologies were amongst the most related parameters which influence the BPNN classifier performance, and used IGA for finding near optimal or optimal values of such 2 parameters that assure low false alarm rate, high detection rate, and high accuracy. Mayuranathan et al. [14] present an effectual feature subset selection on the basis of classifier method for identifying DDoS attacks. For detection of the DDoS attacks in IDS, optimal feature set can be chosen having maximal detection by using Random Harmony Search (RHS) optimization method. Once the features were chosen, a DL-related classification method employing Restricted Boltzmann Machines (RBM) can be implied for detecting the DDoS. To improve the detection rates of DDoS attacks, 7 extra layers set were involved amongst the visible and hidden layers of the RBM. Jaber et al. [15] suggest a new IDS which compiles an FCM method having support vector machine (SVM) for enhancing the accurateness of the detection system in CC atmosphere. The presented model can be applied and comparison made with prevailing systems.

3 The Proposed ECODL-IDSCS Model

In this study, a new ECODL-IDSCS model has been developed to accomplish security in the cloud environment. The presented ECODL-IDSCS model encompasses a series of processes such as data normalization and class imbalance data handling at the primary step. In addition, the LSTM model is utilized for the recognition and classification of intrusions. Finally, the ECO algorithm is derived to optimally fine tune the hyperparameters related to the LSTM model. Fig. 1 depicts the overall process of ECODL-IDSCS algorithms.

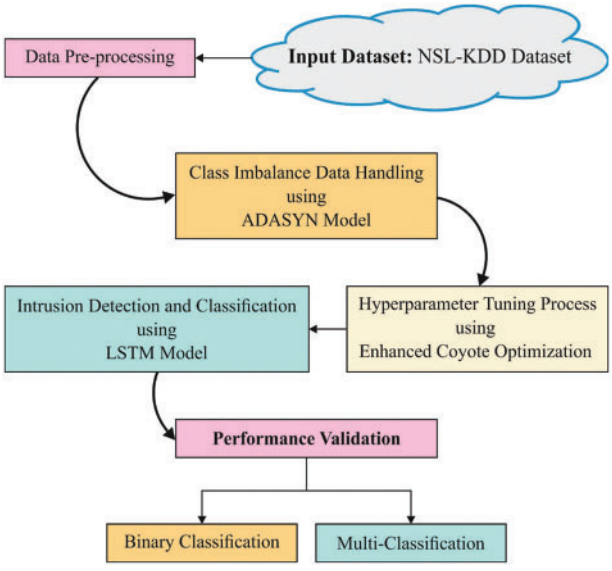


Figure 1: Overall process of ECODL-IDSCS approach

3.1 Data Pre-processing

The presented ECODL-IDSCS model performs data normalization at the primary step. A larger gap between different dimensional feature datasets within the data brings about problems like insignificant accuracy improvement and slow model training; thus, tackling these issues, the *MinMaxScaler* was adapted for mapping the dataset into the interval of (0, 1) in the following:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

In Eq. (1), x_{max} represents the maximal value, and $x_{\min}$ denotes the minimal value.

3.2 ADASYN Model

In this study, the class imbalance data problem can be addressed by the use of ADASYN technique. ADASYN is an adaptive oversampling technique based on the minority class sample [16]. In comparison to data expansion algorithm, it is represented as the fact that it produces lesser instances in feature space with high density and more instance in a special space with low density. This feature has the benefit of adaptively shifting decision boundary to difficult-to-learn instance, so the ADASYN is better suited when compared to augmentation techniques for handling network traffic with serious information imbalance. The process is implemented in the subsequent steps:

Step 1: Evaluate the number of instance to be synthesized as G that is formulated by

$$G = (n_b - n_s) \times \beta \tag{2}$$

In Eq. (2), n_b signifies the majority instance, n_s characterizes the minority instance, and $\beta \in (0, 1)$.

Step 2: For every minority instance, evaluate K neighbors through the Euclidean distance and represented as r_i the proportion of majority class instance comprised in the neighbor that is formulated by

$$r_i = k/K \tag{3}$$

In Eq. (3), K indicates the existing number of neighbors, as well as k denotes the majority class samples in the existing neighbors.

Step 3: Evaluate the sample count that should be combined for every minority instance based on G and combine the instances based on the following expression as follows

$$g = G \times r_i \tag{4}$$

$$Z_i = X_i + (X_{zi} - X_i) \times \lambda \tag{5}$$

In Eq. (5), g indicates the quantity to be synthesized, Z_i denotes the synthesized new instance, X_i denotes the existing minority instance, and X_{Zi} denotes an arbitrary minority sample amongst the k neighbors of X_i, $\lambda \in (0, 1)$.

3.3 LSTM Based Intrusion Detection and Classification

At this stage, the LSTM model is utilized for the recognition and classification of intrusions. The LSTM method was capable of storing the previous data so captured the prominent long-range dependency from the provided input [17]. The LSTM is executed significantly from the sequence

modelling task like text classifier, sentiment analysis (SA), time series forecast, and so on. There are 3 essential components from the LSTM technique containing input, forget, and output gates. The forget gate is resolved for forget or discard irrelevant data in the preceding cell state and novel input data. The sigmoid function was utilized from the trained for returning the values amongst zero and one. The value closer to 0 represents that the data is lesser significant for remembering. An input gate acts as filters for deciding that data was worth remember, so that upgrade as to next state. A value closer to 0 signifies that it can be lesser significant that upgraded. The resultant gate defines the data which must be the outcome from the next cell state.

The computations of single LSTM unit at a single time step t from the forget gate f_z, input gate i_z, output gate o_z, and cell state c_z are determined as:

$$f_z = \sigma\left(W_f X_z + U_f h_{z-1} + b_f\right) \tag{6}$$

$$i_z = \sigma\left(W_i X_z + U_i h_{z-1} + b_i\right) \tag{7}$$

$$o_z = \sigma\left(W_o X_z + U_o h_{z-1} + b_o\right) \tag{8}$$

$$\tilde{c}_z = \tanh\left(W_c X_z + U_c h_{z-1} + b_c\right) \tag{9}$$

$$c_z = f_z * c_{z-1} + i_z * \tilde{c}_z \tag{10}$$

$$h_z = o_z * \tanh\left(c_z\right) \tag{11}$$

whereas σ stands for the sigmoid function, X_z represents the input, $(W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c)$ and $(b_i, b_f, b_c b_o,)$ demonstrates the weighted matrix and bias from the input gate, forget gate, cell state, and output gate correspondingly. h_{z-1} and c_{z-1} were the resultant of LSTM at time step $z-1$. The function $*$ refers the element wise multiplication.

3.4 ECO Based Parameter Optimization

Lastly, the ECO algorithm is derived to optimally fine tune the hyperparameters [18–20] related to the LSTM model to enhance its detection efficiency in the cloud environment. The COA has population based metaheuristic to resolve global optimized problems [21]. This optimized technique was simulated by the social performance of Canis Latrans species which reside mostly in North America. The population of coyotes can be separated as G_p groups with C_c coyotes from all the groups. Thus, the population size is determined as the multiplication of $G_p * C_c$. All the coyotes are assumed that feasible (candidate) solution and their social conditions signified the objective function or decision variable. Accordingly, the social condition of c^{th} coyote from the p^{th} group at iteration t is demonstrated mathematically as follows:

$$SC_c^{p,t} = X = (x, x, \ldots, x_D) \tag{12}$$

whereas D implies the searching space dimensional.

This social behavior contains adapting coyotes to environment named $ft_c^{p,t} \in R$. The alteration of coyotes to their respective present social condition was validated utilizing the subsequent formula:

$$ft_c^{p,t} = f\left(SC_c^{p,t}\right) \tag{13}$$

Begin with procedure, the coyotes were arbitrarily allocated to groups. But, the coyotes are sometimes a tendency for abandon its groups for combining other groups or developed solitary. The probability P_a that a coyote leaves their group is provided below:

$$P_a = 0,005 * C_c^2 \tag{14}$$

While the value of P_a could not exceed unity, the amount of coyotes per group was restricted to 14. This process enhances the cultural interchange among every coyote from the global population. During all the groups, the leader of coyotes, also named as alpha coyote was the coyote which attains an optimum objective function. Assuming that minimized problems, the alpha coyote is defined mathematically by the subsequent formula:

$$alpha^{p,t} = \left\{SC_c^{p,t} | arg_{c=\{1,2,\ldots,Cc\}} minf\left(SC_c^{p,t}\right)\right\} \tag{15}$$

The cultural tendency of all the groups is demonstrated as:

$$cultr_j^{p,t} = \begin{cases} R^{p,t}_{\frac{Cc+1}{2},j}, & \text{Cc is odd} \\ \dfrac{R^{p,t}_{\frac{Cc}{2}j} + R^{p,t}_{(\frac{Cc}{2}+1)j}}{2}, & \text{Otherwise} \end{cases} \tag{16}$$

In which $R^{p,t}$ signifies the ranked social condition of coyote inside p^{th} group at iteration t to all the j from the searching space D. Afterward, determine the global optimum $alpha^{p,t}$ and the cultural tendency $cultr^{p,t}$ of all the groups, novel social conditions of coyotes were upgraded utilizing the subsequent formula:

$$new\_SC_c^{p,t} = \{SC_c^{p,t} + r1\left(alpha^{p,t} - SC_{cr_1}^{p,t}\right) + r2\left(cultr^{p,t} - SC_{cr_2}^{p,t}\right\} \tag{17}$$

whereas $SC_{cr_1}^{p,t}$ and $SC_{cr_2}^{p,t}$ were social conditions of random coyotes $cr1$ and $cr2$ correspondingly. $r1$ and $r2$ imply the arbitrary numbers from the range of zero and one. Fig. 2 illustrates the flowchart of COA.

The fitness value of novel social conditions is computed as:

$$ft_c^{p,t} = \begin{cases} newft_c^{p,t}, & if\ new\_ft_c^{p,t} < ft_c^{p,t} \\ ft_c^{p,t}, & \text{Otherwise} \end{cases} \tag{18}$$

The selective of final social conditions are dependent upon the computation of novel solutions. Everyone has related to the worse solution from all the groups and optimum one is then kept as can be demonstrated in the subsequent formula:

$$SC_c^{p,t} = \begin{cases} new, SC_c^{p,t} & if\ newft_c^{p,t} < ft_c^{p,t} \\ SC_c^{p,\mathrm{t}} & Otherwise \end{cases} \tag{19}$$

In COA, the 2 important biological events of life (that is, the death and birth) were modeled assuming that the age of all coyotes $age_c^{p,t} \in N$ from the group p. A novel coyote is born by relating 2 parents $SC_{r_1,j}^{p,t}$ and $SC_{r_2 j}^{p,t}$ selective arbitrarily. The procedure of birth is mathematically provided as:

$$pp_j^{p,t} = \begin{cases} SC_{r_1,j}^{p,t} & r_j < P_s\ or\ j = j1 \\ SC_{r_2,j}^{p,t}, r_j \geq P_s + P_z K_j & \text{Otherwise} \end{cases} \tag{20}$$

whereas r_1 and r_2 denotes the parents coyotes chosen arbitrarily in the groups p. j_1 and j_2 are 2 arbitrary dimensional problems. r_j and K_j implies the arbitrary numbers from the range of zero and one. P_s and P_z signifies the scatter and connection probability correspondingly. It can be computed as:

$$P_s = \frac{1}{D} \tag{21}$$

$$P_z = \frac{1 - P_s}{2} \tag{22}$$

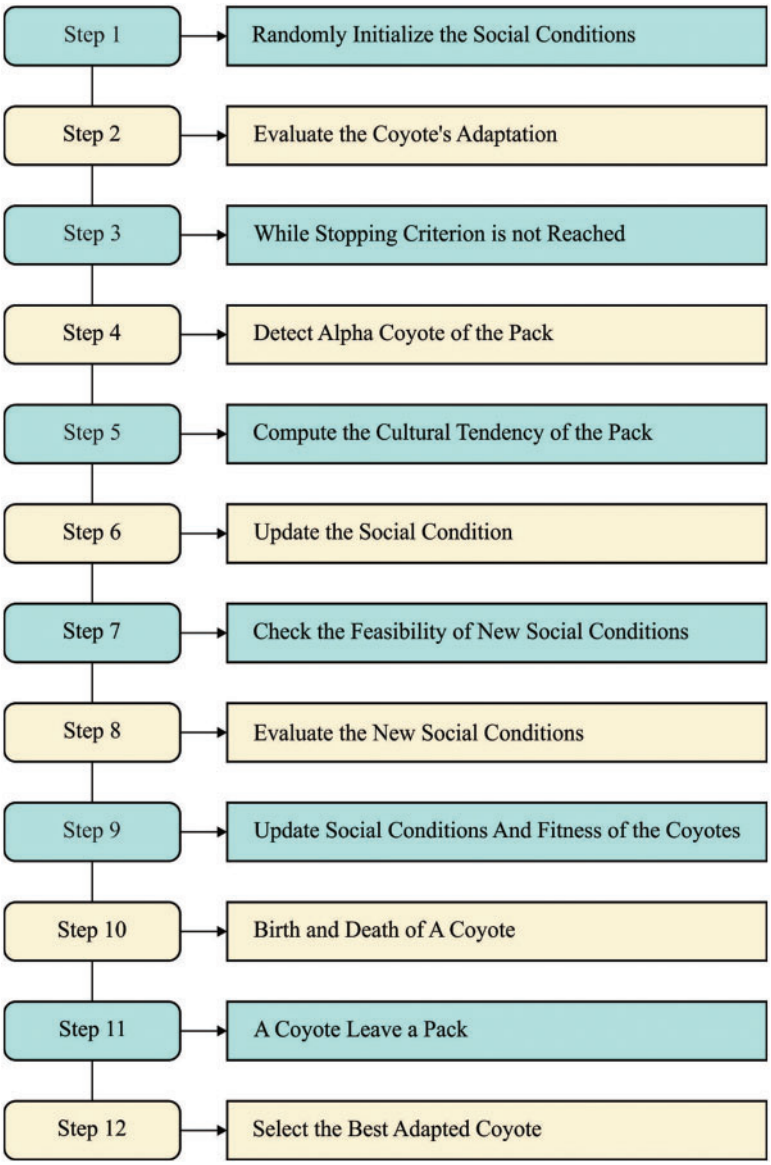


Figure 2: Flowchart of COA

Important steps of COA are provided under:

Step1: Create arbitrarily $G_p * C_c$ solutions and estimate the adaptive function.

Step2: Implement the subsequent functions inside all the groups.

- Identify the alphas of groups.
- Calculate the cultural tendency of groups.
- Upgrade the social condition of coyotes from the group.
- Estimate the novel social condition.
- Do the alteration.
- Progress the birth and death life cycle.

Step3: Change of coyotes amongst arbitrary groups.

Step4: Repeating step 2 still the end condition was attained.

Step5: Output the optimum social conditions of the modified coyote.

The ECO algorithm is developed by the use of late acceptance hill-climbing (HC) concept with the traditional COA. The HC signifies the most local search procedures [22]. Primarily, it starts with arbitrary solutions and iteratively transfers downhill in a present solution still the neighborhood solutions with optimum quality are reached. It can be utilized from several tries of optimized tasks. The HC begins with a randomized solution to candidate problems $H_i = \{H_i^1, H_i^2, \ldots, H_i^n\}$, afterward, it executes 2 operators, β operator, and Neighborhood navigation (N-operator), for iteratively taking novel solution $H_i' = \{H_i'^1, H_i'^2, \ldots, H_i'^n\}$. During all the iterations of HC, the N-operator stage was executed by the random walk approach for obtaining the neighboring solution (H_i') of solutions H_i as follows:

$$H_i'^k = H_i^k \pm U(0,1) \times BW \exists! k \in [1, n] \tag{23}$$

At this point, $BW = UB^k - LB^k$. Conversely, β operator was created for assigning a novel values to the novel solution H_i' utilizing original values of H_i or novel ones as follows:

$$H_i'^k = \begin{cases} H_i^r \ if \ rand \leq \beta \\ H_i^k \ otherwise \end{cases} \tag{24}$$

The β-operator is improve the exploration of HC by roaming from a searching zone to another one. In addition, exploitation was executed by N-operator phase which supports the neighborhood search for present solutions. By these operators, the HC is avoid trapping by the local minimum.

4 Results and Discussion

The experimental validation of the ECODL-IDSCS model is tested using the NSL-KDD dataset [23]. The ECODL-IDSCS model is examined under two aspects such as binary classification and multi-classification. Tabs. 1 and 2 depict the detailed description of two datasets.

Fig. 3 implies the confusion matrices formed by the ECODL-IDSCS model on binary classification dataset. On entire dataset, the ECODL-IDSCS model has identified 489 samples into normal and 1989 samples into abnormal. Also, on 70% of training (TR) data, the ECODL-IDSCS methodology

has identified 345 samples into normal and 1388 samples into abnormal. Besides, on 30% of testing (TS) data, the ECODL-IDSCS technique has identified 144 samples into normal and 601 samples into abnormal.

Table 1: Details on binary classification dataset

Class	No. of samples
Normal	500
Abnormal	2000
Total number of samples	2500

Table 2: Details on Multiclass classification dataset

Class	No. of Samples
Normal	500
Dos	500
Probe	500
Remote-to-local (R2L)	500
User to root (U2R)	500
Total number of samples	2500

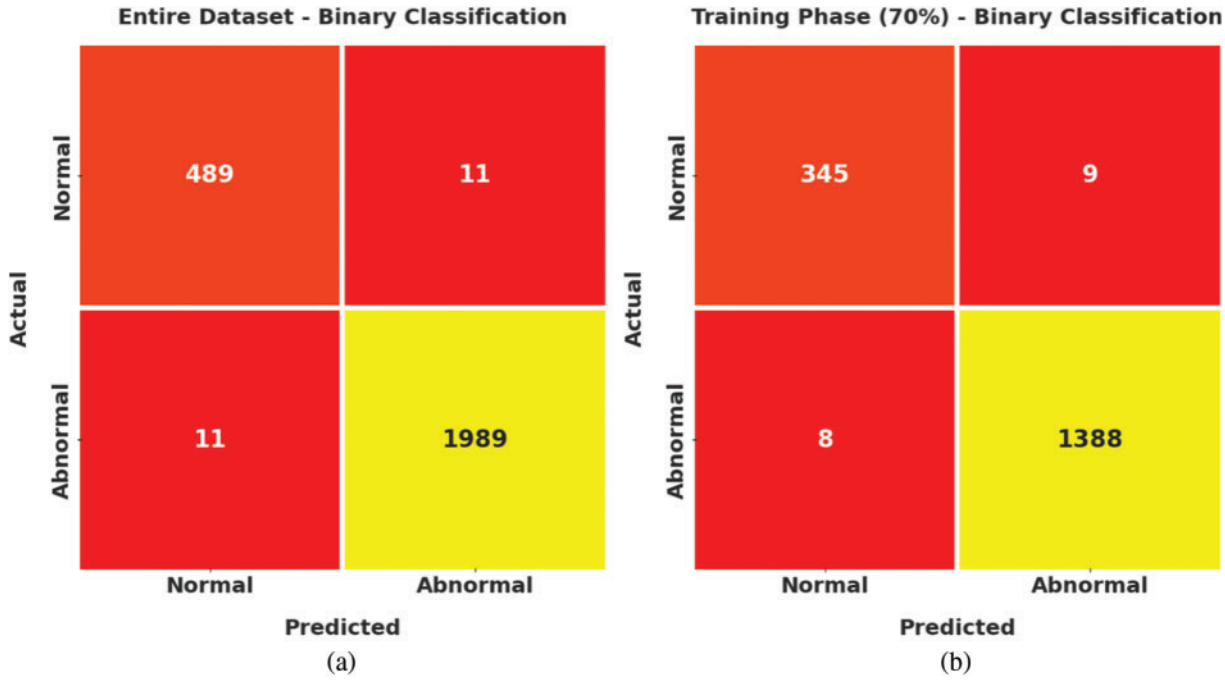
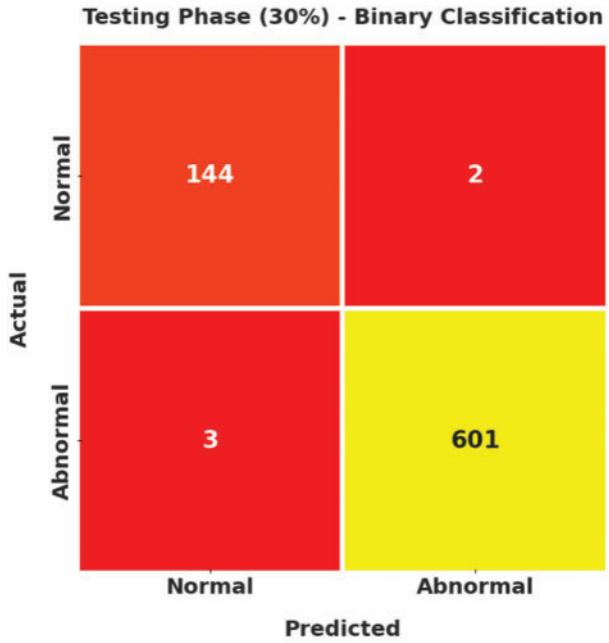


Figure 3: (Continued)



(c)

Figure 3: Confusion matrices of ECODL-IDSCS approach on binary classification (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Tab. 3 and Fig. 4 exhibits the classifier output of the ECODL-IDSCS model on the binary classification dataset. The obtained values represented that the ECODL-IDSCS model has accomplished effectual outcomes. For instance, on entire dataset, the ECODL-IDSCS model has resulted in average $accu_y$ of 99.12%, $prec_n$ of 98.63%, $reca_l$ of 98.63%, $F1_{score}$ of 98.63%, and AUC_{score} of 98.63%. Concurrently, on 70% of TR data, the ECODL-IDSCS approach has resulted in average $accu_y$ of 99.03%, $prec_n$ of 98.54%, $reca_l$ of 98.44%, $F1_{score}$ of 98.49%, and AUC_{score} of 98.44%. Eventually, on 30% of TS data, the ECODL-IDSCS method has resulted in average $accu_y$ of 99.33%, $prec_n$ of 98.81%, $reca_l$ of 99.07%, $F1_{score}$ of 98.94%, and AUC_{score} of 99.07%.

Table 3: Result analysis of ECODL-IDSCS approach on binary classification dataset

Labels	Accuracy	Precision	Recall	F1-Score	AUC score
Entire dataset					
Normal	99.12	97.80	97.80	97.80	98.63
Abnormal	99.12	99.45	99.45	99.45	98.63
Average	99.12	98.63	98.63	98.63	98.63
Training Phase (70%)					
Normal	99.03	97.73	97.46	97.60	98.44
Abnormal	99.03	99.36	99.43	99.39	98.44
Average	99.03	98.54	98.44	98.49	98.44

(Continued)

Table 3: Continued

Labels	Accuracy	Precision	Recall	F1-Score	AUC score
Testing phase (30%)					
Normal	99.33	97.96	98.63	98.29	99.07
Abnormal	99.33	99.67	99.50	99.59	99.07
Average	99.33	98.81	99.07	98.94	99.07

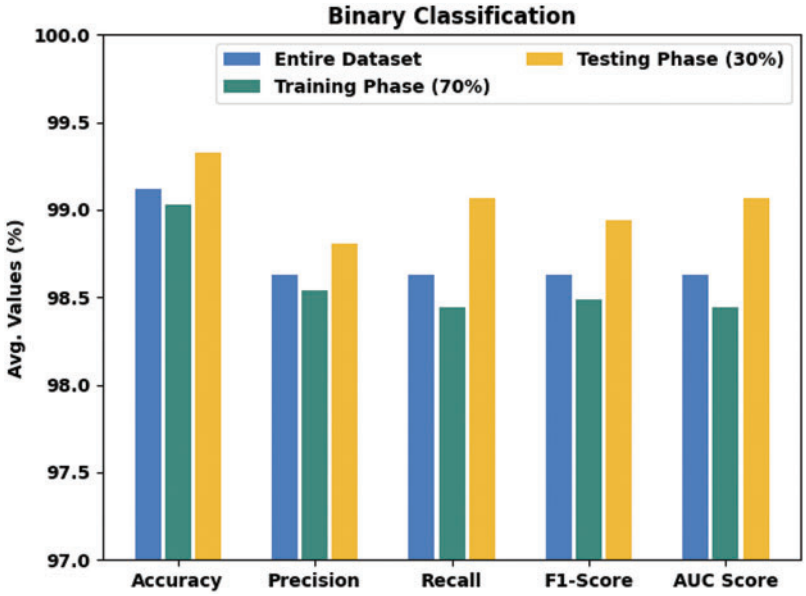


Figure 4: Result analysis of ECODL-IDSCS approach under binary classification dataset

A clear precision-recall analysis of the ECODL-IDSCS algorithm on binary classification dataset is represented in Fig. 5. The figure denoted the ECODL-IDSCS technique has resulted in enhanced values of precision-recall values under all classes.

A brief receiver operating characteristic (ROC) examination of the ECODL-IDSCS technique on binary classification dataset is shown in Fig. 6. The results represented the ECODL-IDSCS methodology has shown its ability in categorizing distinct classes on the binary classification dataset.

Fig. 7 signifies the confusion matrices formed by the ECODL-IDSCS method on Multiclass classification dataset. On entire dataset, the ECODL-IDSCS technique has identified 492 samples into normal, 494 samples into DoS, 474 samples into Probe, 484 samples into R2L, and 490 samples into U2R. Also, on 70% of TR data, the ECODL-IDSCS approach has identified 335 samples into normal, 337 samples into DoS, 334 samples into Probe, 358 samples into R2L, and 338 samples into U2R. Besides, on 30% of TS data, the ECODL-IDSCS methodology has identified 157 samples into normal, 157 samples into DoS, 140 samples into Probe, 126 samples into R2L, and 152 samples into U2R.

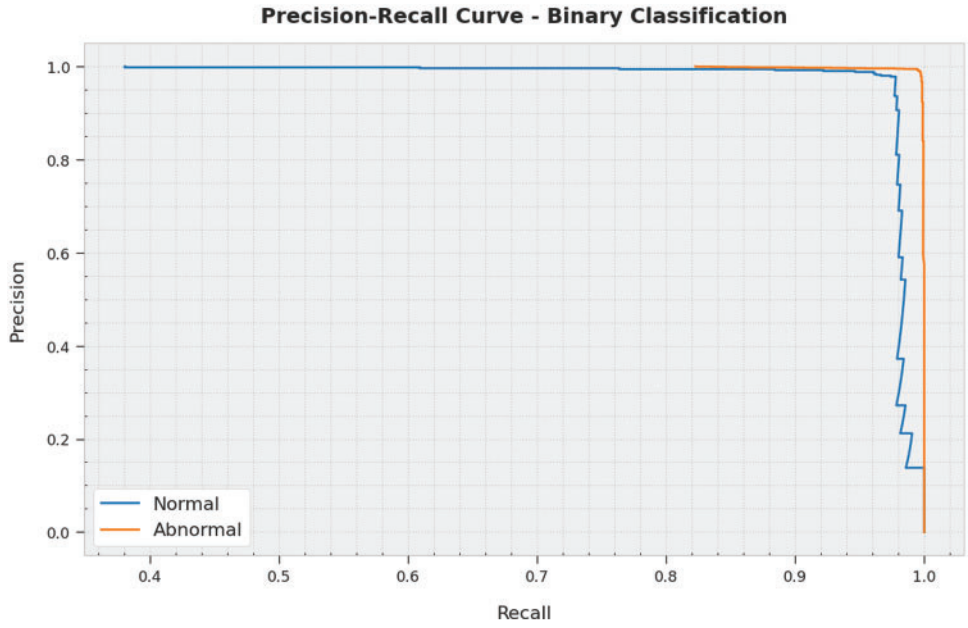


Figure 5: Precision-recall analysis of ECODL-IDSCS approach under binary classification dataset

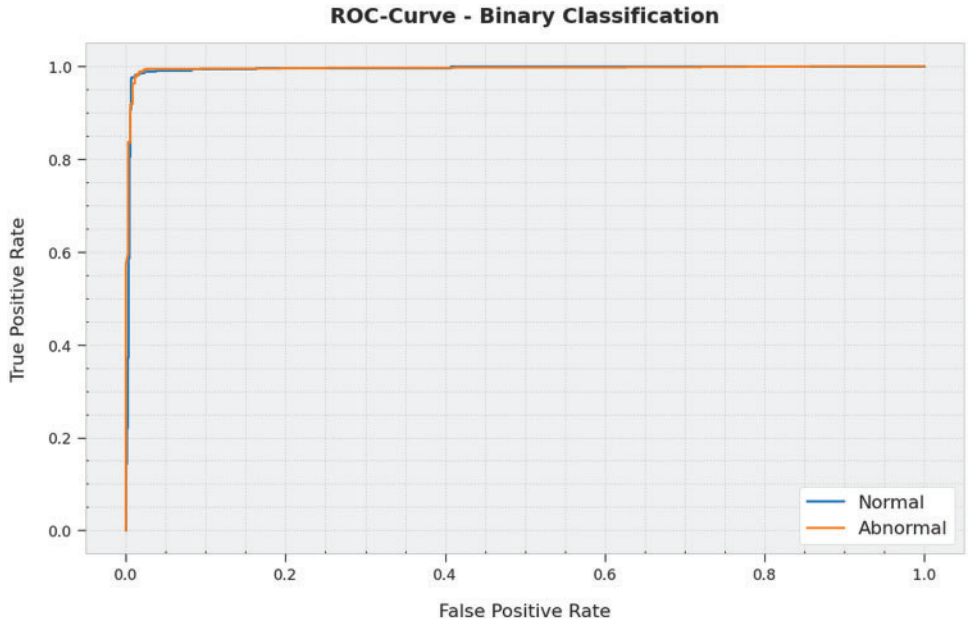


Figure 6: ROC analysis of ECODL-IDSCS approach under binary classification dataset

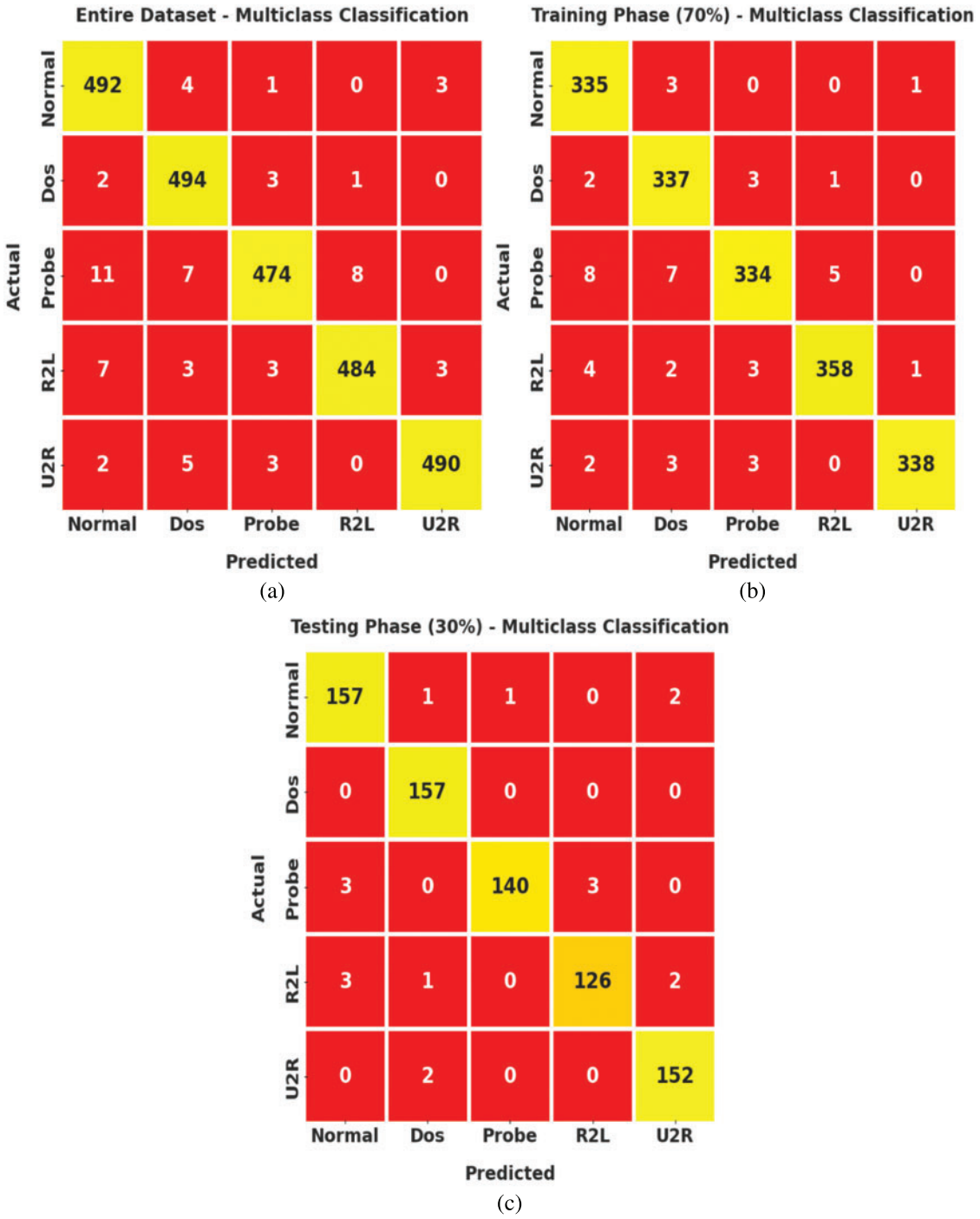


Figure 7: Confusion matrices of ECODL-IDSCS approach on Multiclass classification (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Tab. 4 and Fig. 8 show the classifier output of the ECODL-IDSCS method on the Multiclass classification dataset. The acquired values indicated that the ECODL-IDSCS approach has accomplished effectual outcomes. For example, on entire dataset, the ECODL-IDSCS method has resulted in average $accu_y$ of 98.94%, $prec_n$ of 97.38%, $reca_l$ of 97.36%, $F1_{score}$ of 97.36%, and AUC_{score} of 98.35%. Simultaneously, on 70% of TR data, the ECODL-IDSCS algorithm has resulted in average $accu_y$ of 98.90%, $prec_n$ of 97.26%, $reca_l$ of 97.28%, $F1_{score}$ of 97.26%, and AUC_{score} of 98.30%. Meanwhile, on 30% of TS data, the ECODL-IDSCS technique has resulted in average $accu_y$ of 99.04%, $prec_n$ of 97.65%, $reca_l$ of 97.51%, $F1_{score}$ of 97.57%, and AUC_{score} of 98.45%.

Table 4: Result analysis of ECODL-IDSCS approach on Multiclass classification dataset

Labels	Accuracy	Precision	Recall	F1-Score	AUC score
Entire dataset					
Normal	98.80	95.72	98.40	97.04	98.65
Dos	99.00	96.30	98.80	97.53	98.92
Probe	98.56	97.93	94.80	96.34	97.15
R2L	99.00	98.17	96.80	97.48	98.18
U2R	99.36	98.79	98.00	98.39	98.85
Average	98.94	97.38	97.36	97.36	98.35
Training phase (70%)					
Normal	98.86	95.44	98.82	97.10	98.84
Dos	98.80	95.74	98.25	96.98	98.59
Probe	98.34	97.38	94.35	95.84	96.85
R2L	99.09	98.35	97.28	97.81	98.42
U2R	99.43	99.41	97.69	98.54	98.77
Average	98.90	97.26	97.28	97.26	98.30
Testing phase (30%)					
Normal	98.67	96.32	97.52	96.91	98.25
Dos	99.47	97.52	100.00	98.74	99.66
Probe	99.07	99.29	95.89	97.56	97.86
R2L	98.80	97.67	95.45	96.55	97.48
U2R	99.20	97.44	98.70	98.06	99.02
Average	99.04	97.65	97.51	97.57	98.45

A clear precision-recall inspection of the ECODL-IDSCS methodology on Multiclass classification dataset is depicted in Fig. 9. The figure signifies that the ECODL-IDSCS technique has resulted in enhanced values of precision-recall values under all classes.

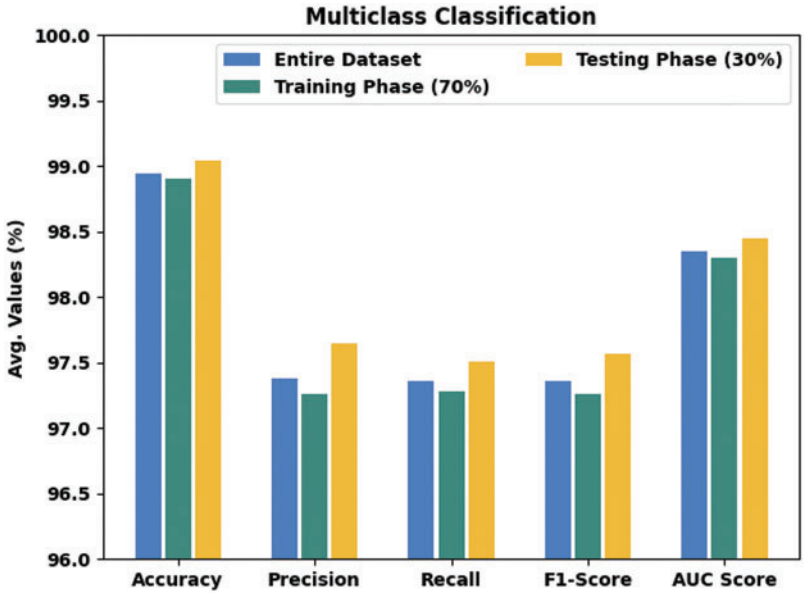


Figure 8: Result analysis of ECODL-IDSCS approach under Multiclass classification dataset

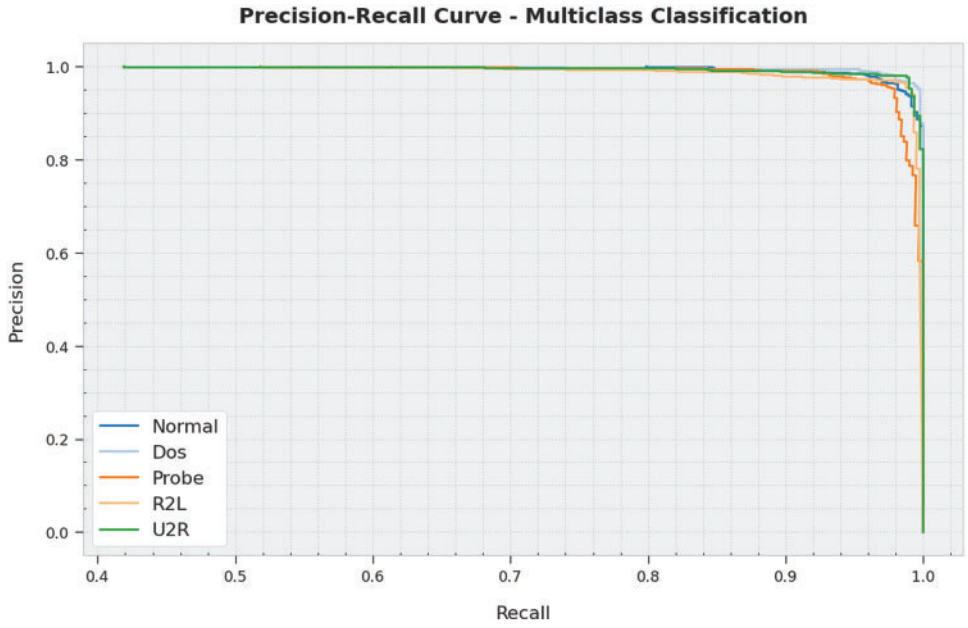


Figure 9: Precision-recall analysis of ECODL-IDSCS approach under Multiclass classification dataset

A brief ROC analysis of the ECODL-IDSCS technique on Multiclass classification dataset is portrayed in Fig. 10. The results show the ECODL-IDSCS algorithm has shown its ability in categorizing distinct classes on the Multiclass classification dataset.

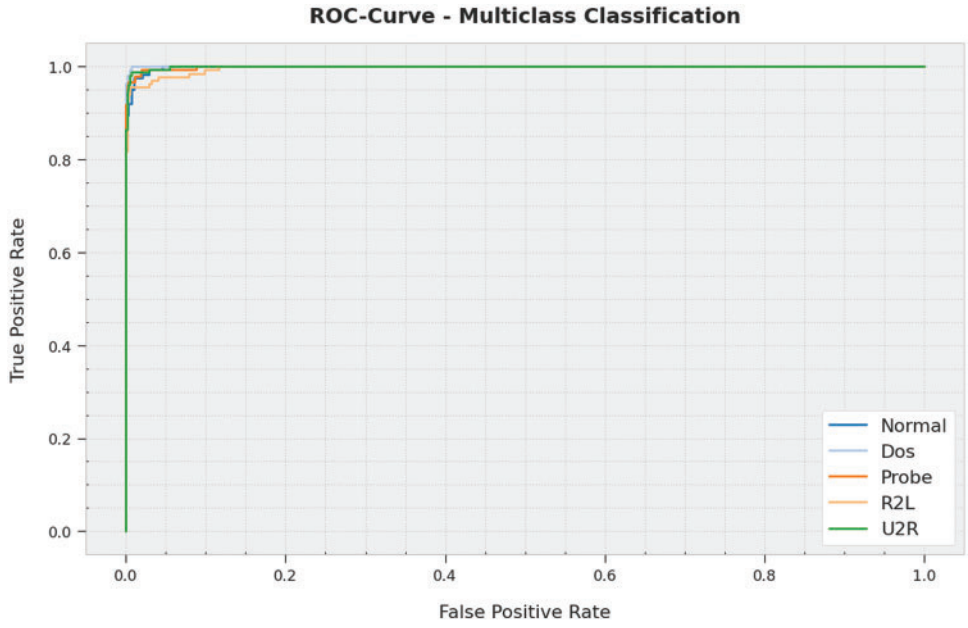


Figure 10: ROC analysis of ECODL-IDSCS approach under Multiclass classification dataset

To assure the enhanced performance of the ECODL-IDSCS model, a wide range of experiments were carried out in Tab. 5 [24]. The results implied that the adaptive ensemble, two-stage classifier ensemble based IDS (TES-IDS), GAR-Forest, and CNN-BiLSTM models have resulted to lower $accu_y$ values of 84.76%, 85.43%, 85.41%, and 83.89% respectively. Next to that, the Double-Layered Hybrid Approach (DLHA) technique has shown slightly improved $accu_y$ of 87.08%. Followed by, the deep learning model for network intrusion detection (DLNID) technique has accomplished reasonable $accu_y$ of 90.51%. However, the ECODL-IDSCS model has demonstrated higher $accu_y$ of 99.04%. Thus, the ECODL-IDSCS model can be applied to recognize intrusions in the CC environment.

Table 5: Comparative analysis of ECODL-IDSCS approach with existing methodologies

Methods	Accuracy	Precision	Recall	F1-Score
ECODL-IDSCS	99.04	97.65	97.51	97.57
DLNID	90.51	86.41	93.10	89.38
DLHA	87.08	88.17	90.44	89.30
Adaptive Ensemble	84.76	86.52	86.01	85.80
TES-IDS	85.43	88.59	86.53	87.54
GAR-Forest	85.41	88.10	84.77	84.54
CNN-BiLSTM	83.89	86.10	84.44	84.87

5 Conclusion

In this work, a novel ECODL-IDSCS model was projected to accomplish security in the cloud environment. The presented ECODL-IDSCS model encompasses a series of processes such as data

normalization and class imbalance data handling at the primary step. Additionally, the LSTM method was utilized for the recognition and classification of intrusions. Finally, the ECO algorithm is derived to optimally fine tune the hyperparameters related to the LSTM model to enhance its detection efficiency in the cloud environment. Once the presented ECODL-IDSCS model can be tested on benchmark dataset, the experimental results show the promising performance of the ECODL-IDSCS model over the existing IDS models. Thus, the ECODL-IDSCS model can be utilized as an effectual tool to achieve maximum security in the cloud platform. In future, the performance of the ECODL-IDSCS model can be improvised by the use of advanced DL classifiers.

Funding Statement: The Deanship of Scientific Research (DSR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia has funded this project, under grant no. KEP-1-120-42.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Deshpande, S. C. Sharma, S. K. Peddoju and S. Junaid, "HIDS: A host based intrusion detection system for cloud computing environment," *International Journal of Systems Assurance Engineering and Management*, vol. 9, no. 3, pp. 567–576, 2018.

[2] O. Achbarou, M. A. El Kiram, O. Bourkoukou and S. Elbouanani, "A new distributed intrusion detection system based on multi-agent system for cloud environment," *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, pp. 526, 2018.

[3] P. Wanda and H. J. Jie, "A survey of intrusion detection system," *International Journal of Informatics and Computation*, vol. 1, no. 1, pp. 1, 2020.

[4] S. Mehibs and S. Hashim, "Proposed network intrusion detection system in cloud environment based on back propagation neural network," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 26, no. 1, pp. 29–40, 2017.

[5] M. Hatef, V. Shaker, M. Jabbarpour, J. Jung and H. Zarrabi, "HIDCC: A hybrid intrusion detection approach in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 3, pp. e4171, 2017.

[6] D. G. Singh, R. Priyadharshini and E. Jebamalar Leavline, "Cuckoo optimisation based intrusion detection system for cloud computing," *International Journal of Computer Network and Information Security*, vol. 10, no. 11, pp. 42–49, 2018.

[7] S. Mehibs and S. Hashim, "Proposed network intrusion detection system based on fuzzy C mean algorithm in cloud computing environment," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 26, no. 2, pp. 27–35, 2017.

[8] M. Manickam and S. P. Rajagopalan, "A hybrid multi-layer intrusion detection system in cloud," *Cluster Computing*, vol. 22, no. S2, pp. 3961–3969, 2019.

[9] Z. Liu, B. Xu, B. Cheng, X. Hu and M. Darbandi, "Intrusion detection systems in the cloud computing: A comprehensive and deep literature review," *Concurrency and Computation*, vol. 34, no. 4, pp. e3835, 2022.

[10] P. Kanimozhi and T. Aruldoss Albert Victoire, "Oppositional tunicate fuzzy C-means algorithm and logistic regression for intrusion detection on cloud," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. e6624, 2022.

[11] P. Sharma, J. Sengupta and P. K. Suri, "WLI-FCM and artificial neural network based cloud intrusion detection system," *International Journal of Advanced Networking and Applications*, vol. 10, no. 1, pp. 3698–3703, 2018.

[12] E. Besharati, M. Naderan and E. Namjoo, "LR-HIDS: Logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, 2019.

[13] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri and M. Rida, "New anomaly network intrusion detection system in cloud environment based on optimized back propagation neural network using improved genetic algorithm," *International Journal of Communication Networks and Information Security*, vol. 11, no. 1, pp. 61–84, 2019.

[14] M. Mayuranathan, M. Murugan and V. Dhanakoti, "Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3609–3619, 2021.

[15] A. N. Jaber and S. U. Rehman, "FCM-SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.

[16] X. Gu, P. P. Angelov and E. A. Soares, "A self-adaptive synthetic over-sampling technique for imbalanced classification," *International Journal of Intelligent Systems*, vol. 35, no. 6, pp. 923–943, 2020.

[17] G. Jain, M. Sharma and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," *Annals of Mathematics and Artificial Intelligence*, vol. 85, no. 1, pp. 21–44, 2019.

[18] K. Shankar, S. Kumar, A. K. Dutta, A. Alkhayyat, A. J. M. Jawad *et al.,* "An automated hyperparameter tuning recurrent neural network model for fruit classification," *Mathematics*, vol. 10, no. 13, pp. 1–20, 2022.

[19] D. K. Jain, Y. Li, M. J. Er, Q. Xin, D. Gupta *et al.,* "Enabling unmanned aerial vehicle borne secure communication with classification framework for industry 5.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5477–5484, 2022.

[20] D. Venugopal, T. Jayasankar, M. Y. Sikkandar, M. I. Waly, I. V. Pustokhina *et al.,* "A novel deep neural network for intracranial haemorrhage detection and classification," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 2877–2893, 2021.

[21] S. Duman, H. T. Kahraman, U. Guvenc and S. Aras, "Development of a Lévy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems," *Soft Computing*, vol. 25, no. 8, pp. 6577–6617, 2021.

[22] N. Noman and H. Iba, "Inferring gene regulatory networks using differential evolution with local search heuristics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 634–647, 2007.

[23] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1–6, 2009.

[24] Y. Fu, Y. Du, Z. Cao, Q. Li and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, pp. 898, 2022.