Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.034193
Article

Tech Science Press

Coot Optimization with Deep Learning-Based False Data Injection Attack Recognition

T. Satyanarayana Murthy[1], P. Udayakumar[2], Fayadh Alenezi[3], E. Laxmi Lydia[4] and Mohamad Khairi Ishak[5,*]

[1]Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India
[2]Department of Computer Science and Engineering, Aditya Engineering College, Surampalem, Andhra Pradesh, India
[3]Department of Electrical Engineering, College of Engineering, Jouf University, Saudi Arabia
[4]Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
[5]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, 14300, Malaysia
*Corresponding Author: Mohamad Khairi Ishak. Email: khairiishak@usm.my
Received: 08 July 2022; Accepted: 11 October 2022

Abstract: The recent developments in smart cities pose major security issues for the Internet of Things (IoT) devices. These security issues directly result from inappropriate security management protocols and their implementation by IoT gadget developers. Cyber-attackers take advantage of such gadgets' vulnerabilities through various attacks such as injection and Distributed Denial of Service (DDoS) attacks. In this background, Intrusion Detection (ID) is the only way to identify the attacks and mitigate their damage. The recent advancements in Machine Learning (ML) and Deep Learning (DL) models are useful in effectively classifying cyber-attacks. The current research paper introduces a new Coot Optimization Algorithm with a Deep Learning-based False Data Injection Attack Recognition (COADL-FDIAR) model for the IoT environment. The presented COADL-FDIAR technique aims to identify false data injection attacks in the IoT environment. To accomplish this, the COADL-FDIAR model initially preprocesses the input data and selects the features with the help of the Chi-square test. To detect and classify false data injection attacks, the Stacked Long Short-Term Memory (SLSTM) model is exploited in this study. Finally, the COA algorithm effectively adjusts the SLTSM model's hyperparameters effectively and accomplishes a superior recognition efficiency. The proposed COADL-FDIAR model was experimentally validated using a standard dataset, and the outcomes were scrutinized under distinct aspects. The comparative analysis results assured the superior performance of the proposed COADL-FDIAR model over other recent approaches with a maximum accuracy of 98.84%.

Keywords: False data injection attack; security; internet of things; deep learning; coot optimization algorithm

1 Introduction

The Internet of Things (IoT) networking paradigm is rapidly developing and has turned out to be a fully-realized technology. The development of IoT systems denotes that billions of several smart devices can be connected to a network around the community and can collect, store and process the information collected through sensors [1]. The IoT network is considered a significant aspect of future advancements, and its applications are expanding across a wide range of businesses [2]. IoT plays an important role in enhancing performance outcomes in several fields: entertainment, manufacturing, healthcare, environmental protection, public security, agriculture, accommodation, industrial monitoring, traditional metering system and intelligent transportation [3]. However, little attention has been paid to adopting high-quality security measures in IoT gadgets. In such cases, insecure communication and the lack of authentication remain the foremost issues in several IoT gadgets [4]. The insiders bring several challenges due to the heavy growth of IoT gadgets.

Privacy and security issues have substantially increased in recent years, especially after the extensive application of technology in the execution of different activities in organizations [5]. Amongst the issues, insider attacks are considered the most hazardous and expensive. An insider attack can be defined as a type of malicious activity executed by users who have authorized access to the information mechanism of an organization [6]. Since authorization issues are prevalent in insider attacks, it is challenging to identify intruders. Organizations tend to lose their goodwill and business objectives if the issues are not overcome within the earliest possible time. A false Data Injection Attack (FDIA) can be defined as the insertion of commands or data at the source or the time of communication. Data injection manipulates the value produced by actuators, sensors and other gadgets.

On the other hand, the command 'injection' denotes different types of server-issued instructions [7]. FDIA is one of the common attacks launched at any type of critical infrastructure. It is executed by interrupting the interaction sessions among various gadgets.

FDIAs can damage external elements, incur a heavy economic loss, and produce serious scenarios [8]. Thus, it is important to detect and prevent FDIAs in critical infrastructures. But, the prevailing solutions are merely theoretical or adopted from the cyber ecosystem, such as Intrusion Prevention Systems (IPS) and Intrusion Detection Systems (IDS), which are generally utilized in the protection of conventional computer networks [9]. The existing methods lack specific security needs and critical infrastructures. At the same time, there is a need to have a robust method with a high rate of events, real-time communication and identification, proactive defence and complicated physical and cyber interfaces [10]. Thus, the current study attempts to overcome such constraints by implementing Machine Learning (ML) methods in detecting injection attacks.

The current study introduces a new Coot Optimization Algorithm with a Deep Learning-based False Data Injection Attack Recognition (COADL-FDIAR) model for the IoT environment. The presented COADL-FDIAR technique pre-processes the input data and selects the features with the help of the Chi-square test. To detect and classify false data injection attacks, the Stacked Long Short Term Memory (SLSTM) model is exploited in this study. Finally, the COA algorithm adjusts the hyperparameters of the SLTSM model effectually and accomplishes a superior recognition efficiency. The proposed COADL-FDIAR model was experimentally validated using a standard dataset, and the results were inspected under distinct aspects.

The rest of the paper is organized as follows. Section 2 provides the information on the literature review; Section 3 introduces the proposed model. Section 4 validates the results, and Section 5 concludes the study.

2 Literature Review

In literature [11], a simplified Neural Network (NN) was proposed to detect the False Data Injection (FDI) assaults that target the communication line overflows. When the proposed method was compared against the existing techniques, it was found that it not only concentrated on stealthy attacks bypassing the state estimation (SE) but also helped in the de-congestion of the communication lines in smart grids. Hu et al. [12] modelled a Cyber-Physical Moving Target Defense (CPMTD) approach focused on detection and attack prevention methods to mitigate static vulnerabilities. This method amalgamated the defence policies for power mechanisms. The proposed CPMTD method was designed to prevent attacks that mislead and disturb the attack preparation methods through the randomization of data acquisition processes. The proposed method controlled the change-over of multiple system dimensions based on the network programmability of Protocol Oblivious Forwarding (POF). A physical-MTD strategy was devised for attack detection. It enhanced the detection probability of the FDIA by occasionally varying the dimension matrices of the SE related to the gadgets' capabilities in terms of disturbing the communication line.

Dehghani et al. [13] devised a new cyber-attack detection method for the detection of FDIAs related to the Fast Fourier Transform (FFT) along with the Singular Value Decomposition (SVD) method. Since the concentration of the proposed method was to detect the FDIAs, it considered the effect of renewable energy resources. When malicious data was included in system-state vectors, the vectors' temporal data and the spatial data relationship drifted from the usual functioning circumstances. Srinivasan et al. [14] proposed a Deep Learning (DL)-related location detection method to identify the FDIA regions continuously. This study used False Data Detector (FDD) and a Convolutional Neural Network (CNN). The FDD can be installed to capture the fake data. As a multi-label classifier, the CNN method was used to evaluate co-occurrence dependency, irrespective of the power flow computations that occur due to possible attacks.

Kumar et al. [15] examined the efficiency of ML techniques in identifying the FDIC. Specifically, the authors focused on two commonly-employed critical infrastructures, the water treatment plants and the power systems. This research work emphasized dealing with two key technical problems, such as identifying optimal features set in a hybrid of methods and resolving class imbalance perplexity using the oversampling techniques. In the study conducted earlier [16], a new false data detection and prevention pattern was modelled for the dimension infrastructure in smart grids. In this study, two methods were projected for the management of cyber-attacks: the variable dummy value method and a fixed dummy value method. The limits of the fixed dummy value method are recognized and sorted out in the variable dummy value method.

3 Materials and Methods

This study developed a new COADL-FDIAR algorithm to recognise false data injection attacks in the IoT environment. To accomplish this, the proposed COADL-FDIAR model pre-processes the input data and selects the features with the help of the Chi-square test. At last, the COA is exploited with the SLSTM model in this study. Fig. 1 showcases the overall processes of the proposed COADL-FDIAR approach.

3.1 Data Pre-processing

The proposed COADL-FDIAR technique is applied at the initial level to pre-process the input IoT data. Initially, the missing values in the dataset are filled with null values. Then, a type casting set is executed in which the non-integer values are cast with numbers. The symbolic values (like the receiver, transmitter and destination) are mapped with the integers that are scaled to a value between 1 and max. Here, max denotes the quantity of the symbolic features. The hexadecimal values (i.e., values of the WEP initialization vector and integrity check) are mapped into the integers.

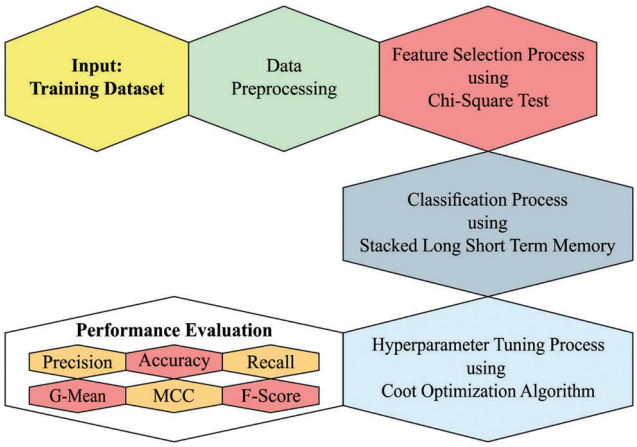

Figure 1: Overall process of the COADL-FDIAR approach

3.2 Chi-Square Feature Selection

To reduce the high dimensionality of the features, the Chi-square test is utilized in this study. Chi-square is a numerical test that measures the deviations in the predictable distribution, assuming that the feature event is independent of the class values [17]. The chi-square value is determined for the subsequent metrics such as the false positives (FP), false negatives (fn), true positives (tp), true negatives (TN), probability of the number of positive cases P_{pos} and the probability of the number of negative cases P_{neg}.

$$chi-square-metric = t\left(t_p,\ \left(t_p+f_p\right)P_{pos}\right)+t\left(f_n,\ \left(f_n+t_n\right)P_{pos}\right) + t\left(f_p,\ \left(t_p+f_p\right)P_{neg}\right)+t\left(t_n,\ \left(f_n+t_n\right)P_{neg}\right) \quad (1)$$

Here, t(count, expect) = (count−expect)/expect. Following is the list of stages followed in the chi-square technique; Identify the model, Study the sample data, devise an analysis plan and deduce the outcomes. Then, the model has stated, whereas the investigation plan identifies when the model employed is accepted or rejected. The plan must identify the following values; the researchers select an importance level in the 0.01, 0.05, or 0.10. The values lie somewhere in the range of 0 and 1. The chi-square test determines the independence levels to determine whether a considerable connection exists between the categorical elements. The sample data contains information that is examined to calculate the degrees of freedom, predictable frequency, test value, and P-value linked with the test data.

$$Degrees\ of\ freedom:\ DF=(r-1)*(c-1) \quad (2)$$

Here, r refers to the amount of one categorical variable level, and c signifies the count of levels to the rest of the categorical variables. The test statistic is determined as follows

$$\chi^2(f,\ c)=\left[\frac{N*(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}\right] \quad (3)$$

whereas A = Amount of times' features 't' and the co-occurrence of the class label 'c'.

B = number of times 't' acts without 'c'.

C = Count of times 'c' performs without 't'.

D = number of times both 'c' and 't' performs.

N = Entire amount of records.

3.3 SLTSM Based Classification

This study exploits the SLSTM model to recognize the false data injection attacks. Long Short-Term Memory (LSTM) is a modified Recurrent Neural Network (RNN) architecture that controls the memory of a time-sequence dataset with the addition of memory cells into the hidden state [18]. The data is passed between the cells in the hidden layer through a sequence of programmable gates such as the forget gate, input gate and output gate. The LSTM method maintains the cell state via gate mechanisms that can resolve short- and long-term memory dependence issues. This scenario helps in the prevention of explosions and the gradient vanishing problems. The input gate function monitors the current data in the memory cell; the output gate controls the distribution of the present data all over the network. The forget gate defines whether the data must be removed according to the status of the previous layer. Eqs. (4)–(11) explain the processes to update and implement the LSTM cell state and calculate the output of the LSTM.

$$F_t = \sigma\left(W_x f\, X_t + W_{hf} H_{t-1} + B_f\right) \tag{4}$$

$$I_t = \sigma(W_{xi} X_t + W_{hi} H_{t-1} + B_i) \tag{5}$$

$$\tilde{C}_t = \sigma(W_{xc} X_t + W_{hc} H_{t-1} + B_c) \tag{6}$$

$$C_t = F_t * C_{t-1} + I_t * \tilde{C}_t \tag{7}$$

$$O_t = \sigma(W_{xo} X_t + W_{ho} H_{t-1} + B_0) \tag{8}$$

$$H_t = o_t \tanh\,(C_t) \tag{9}$$

$$Y_t = \sigma\left(W_{hy} H_t + B_y\right) \tag{10}$$

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \tag{11}$$

In these expressions, Y_t = output vector; X_t = input vector; F_t = forget gate outcome; I_t = input gate outcome; C_t = finishing state in memory block; O_t = output gate outcome; σ = sigmoid function; W_{xf}, W_{xi}, W_{xc}, and W_{xo} refer to the input weight matrices; C_t = temporary, W_{hy} indicates the output weight matrix; W_{hf}, W_{hi}, W_hy and W_{ho} denote the recurrent weight matrices; and B_f, B_i, B_c, B_0, and B_y denoting the bias vectors.

The presented DL-hybrid SLSTM mechanism acts as a prototype for the sequence-to-sequence (seq2seq) paradigm [19]. Fig. 2 depicts the infrastructure of the SLSTM technique. Recently, the Seq2seq model has been widely applied in machine translation and is composed of encoder and decoder entities. When the information is received through the encoder, it compresses the data into a single vector, and the vector is known as the context vector. Then, the decoder makes use of it to generate the output sequence. Either LSTM or RNN is utilized as the encoder to transform the input data into a hidden layer.

Accurately, an encoder ϕ is generated through the hidden, and the input layers that compress the input dataset x from a high-dimension demonstration into a low-dimension demonstration, Z. Meanwhile, a decoder Ψ is generated through the hidden and output layers that reconstruct the input dataset χ' from a suitable code. The transition in the seq2seq learning is arithmetically calculated using a typical neural network system with a σ sigmoid function.

$$\phi: X \to Z \tag{12}$$

$$\chi \mapsto \phi(x) = \sigma(Wx + b) := z$$

$$\Psi: Z \to Z \tag{13}$$

$$z \mapsto \Psi(x) = \sigma(|\tilde{W}_{Z+\tilde{b}})_{:=\chi'}$$

W refers to the weight matrix, and b denotes the bias.

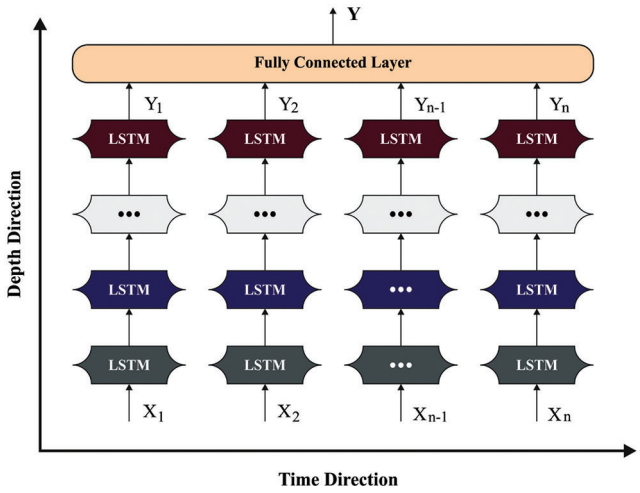

Figure 2: Architecture of the SLSTM approach

This study applies the decoder and encoder systems of the LSTM seq2seq paradigm are applied. The LSTM layer is stacked on the decoder and encoder systems. By stacking the LSTM, it is possible to enhance the prediction ability of the model to understand the complex representation of the time-sequence datasets in hidden layers by gathering data at different levels. Furthermore, x and 0 denote the input and output datasets, respectively, c = encoder context vector and ht and st = hidden state in the decoder and encoder are shown below.

$$h_t = LSTM_{enc}(x_t,\ h_{t-1}) \tag{14}$$

$$h_t = LSTM_{dec}(o_{t-1'}s_{t-1}) \tag{15}$$

Every encoder LSTM estimates a c context vector, and it is replicated and transferred to every decoder element.

3.4 Hyperparameter Tuning

In this study, the COA algorithm productively adjusts the SLTSM model's hyperparameters. The COA is a novel optimization technique developed based on coot bird behaviour [20]. The COA approach tries to mimic the collective behaviour of the coot birds. They are directed by some coots on the surface of the water. The four distinct behaviours observed amongst the coot birds include chain movement, random movement, adjusting the position based on the group leader and leading the group towards an optimum region. This behaviour must be mathematically modelled. Initially, a random population of the coot is produced. Consider a multi-dimension problem that should be resolved at the D dimension. The population of the N coots is produced by Eq. (16).

$$PosCoot(i) = random(1,\ D) \times (UB - LB) + LB,\ \ i = 1,\ 2,\ \ldots,\ N \tag{16}$$

Here, the location of the coots in a multi-dimension space is arbitrarily produced. The Upper Bound (UB) and Lower Bound (LB) layers are defined for every dimension. Therefore, the coots are prevented from going above or below the threshold. Also, an initial random population is estimated based on the Fitness Function (FF) given below.

$$F(i) = Fitness(PosCoot(i)), \quad i = 1,\ 2,\ \ \ldots,\ \ N \tag{17}$$

To model the random movement of the coots, an arbitrary location is generated at first based on the following equation.

$$R = rndom(1,\ \ D) \times (UB - LB) + LB \tag{18}$$

$$PosCoot(i) = PosCoot(i) \neq A \times RN2x(R - PosCoot(i)) \tag{19}$$

Here, RN2 represents a random value between 0 and 1. Here, A and B are defined as follows:

$$A = 1 - \left(T(i) \times \frac{1}{Iter\ \text{Max}}\right),\ B = 2 - \left(T(i) \times \frac{1}{Ite \cdot r\ \text{Max}}\right) i = 1,\ 2,\ \ \ldots IterMax \tag{20}$$

Here, $T(i)$ denotes the current iteration and IterMax refers to the maximal iteration count. To move a coot towards other costs for the implementation of the chain movement, the average location of two coots is applied as follows.

$$PosCoot(i) = 0.5 \times (PosCoot(i-1) + PosCoot(i)) \tag{21}$$

Also, a coot chooses a leader coot and follows it as given below.

$$L_{ind} = 1 + (iMODN_L) \tag{22}$$

Now, L_{ind} denotes the index of leaders and NL denotes the number of leaders defined as a variable. The p probability is determined. At last, the rules shown in Eq. (8) are applied to determine the position of the leader.

$$LeaderPos(i) = \left\{ \begin{array}{ll} B \times R3 \times cos(2R\pi) \times (gBest - LeaderPos(i)) + gBest & R4 < P \\ B \times R3 \times cos(2R\pi) \times (gBest - LeaderPos(i)) + gBest & R4 \geq P \end{array} \right\} \tag{23}$$

Now, R3 and R4 refers to the random numbers within [0, 1], $gBest$ denotes the present global best, and π is 3.14.

The COA approach derives the FF to achieve enhanced classification outcomes. It sets a positive value to denote a superior outcome of the candidate solution. Here, the classification error rate reduction is assumed as the FF as offered below in Eq. (24).

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \end{aligned} \tag{24}$$

4 Results and Discussion

The performance of the proposed COADL-FDIAR model was experimentally validated using a dataset with four classes. The dataset holds a total of 80,000 samples with 20,000 samples under each class as depicted in Table 1. The parameter settings are as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50 and activation: rectified linear unit (ReLU).

Table 1: Dataset details

Class	No. of instances
Normal (NOR)	20000
Injection (INJ)	20000
Impersonation (IMP)	20000
Flooding (FLO)	20000
Total number of instances	80000

Fig. 3 illustrates the confusion matrices generated by the proposed COADL-FDIAR model. The figure implies that the proposed COADL-FDIAR model proficiently recognized all the classes. For instance, on the entire dataset, the COADL-FDIAR model recognized 19,815, 19,460, 19,245 and 19,571 samples as NOR, INJ, IMP and FLO classes, respectively. Moreover, on 70% of the training (TR) dataset, the COADL-FDIAR technique categorized 13,927, 13,653, 13,313, and 13,755 samples under NOR, INJ, IMP and FLO classes correspondingly. At the same time, on 30% of the testing (TS) data, the proposed COADL-FDIAR model classified 5,888, 5,807, 5,932, and 5,816 samples under NOR, INJ, IMP, and FLO classes correspondingly.

The overall analytical results of the proposed COADL-FDIAR model under distinct aspects are shown in Table 2.

Fig. 4 shows the detailed classification results of the proposed COADL-FDIAR model on the entire dataset. The proposed COADL-FDIAR model identified the samples under normal (NOR) class with an $accu_y$ of 99.39%, $prec_n$ of 98.49%, $reca_l$ of 99.08%, F_{score} of 98.78%, Mathew Correlation Coefficient (MCC) of 98.37% and a G_{mean} of 99.28%. Similarly, the proposed COADL-FDIAR approach classified the samples under injection (INJ) class with an $accu_y$ of 98.74%, $prec_n$ of 97.64%, $reca_l$ of 97.30%, F_{score} of 97.47%, MCC of 96.63% and a G_{mean} of 98.25% correspondingly. Eventually, the proposed COADL-FDIAR algorithm categorized the samples under the impersonation (IMP) class with an $accu_y$ of 98.39%, $prec_n$ of 97.30%, $reca_l$ of 96.23%, F_{score} of 96.76%, MCC of 95.69% and a G_{mean} of 97.66%.

Fig. 5 depicts the detailed classification results of the proposed COADL-FDIAR model on 70% of the TR data. The COADL-FDIAR model identified the samples under NOR class with an $accu_y$ of 99.41%, $prec_n$ of 98.51%, $reca_l$ of 99.15%, F_{score} of 98.83%, MCC of 98.44% and a G_{mean} of 99.33%. Then, the proposed COADL-FDIAR method classified the samples under INJ class with an $accu_y$ of 98.69%, $prec_n$ of 97.62%, $reca_l$ of 97.14%, F_{score} of 97.38%, MCC of 96.50% and a G_{mean} of 98.17%. Moreover, the COADL-FDIAR approach identified the samples under IMP class with an $accu_y$ of 98.37%, $prec_n$ of 97.23%, $reca_l$ of 96.14%, F_{score} of 96.68%, MCC of 95.60% and a G_{mean} of 97.61%.

Fig. 6 demonstrates the detailed classification results of the proposed COADL-FDIAR model under 30% of the TS data. The proposed COADL-FDIAR model identified the samples under NOR class with an $accu_y$ of 99.33%, $prec_n$ of 98.43%, $reca_l$ of 98.89%, F_{score} of 98.66%, MCC of 98.22% and a G_{mean} of 99.18%. Then, the proposed COADL-FDIAR approach classified the samples under INJ class with an $accu_y$ of 98.85%, $prec_n$ of 97.68%, $reca_l$ of 97.68%, F_{score} of 97.68%, MCC of 96.91% and a G_{mean} of 98.45%. In the meantime, the proposed COADL-FDIAR technique categorized the samples under IMP class with an $accu_y$ of 98.43%, $prec_n$ of 97.44%, $reca_l$ of 96.42%, F_{score} of 96.93%, MCC of 95.88% and a G_{mean} of 97.77%.

Both Training Accuracy (TA) and Validation Accuracy (VA) values attained by the proposed COADL-FDIAR method on the test dataset are shown in Fig. 7. The experimental outcomes imply that the proposed COADL-FDIAR technique achieved the maximal TA and VA values, whereas the VA values were higher than the TA values.

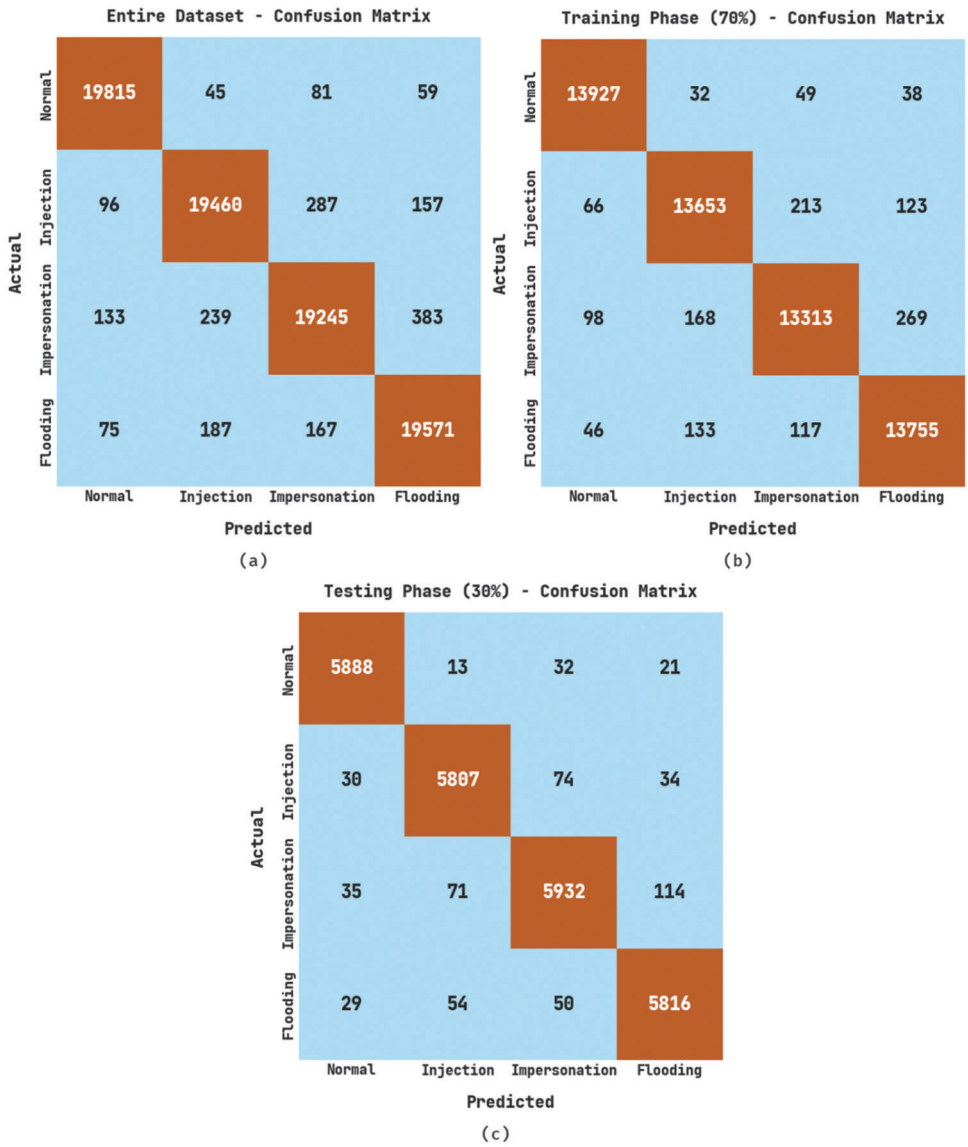

Figure 3: Confusion matrices of the COADL-FDIAR approach (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Table 2: Analytical results of the COADL-FDIAR approach under different measures and class labels

Class labels	Accuracy	Precision	Recall	F-Score	MCC	G-Mean
Entire dataset						
Normal	99.39	98.49	99.08	98.78	98.37	99.28
Injection	98.74	97.64	97.30	97.47	96.63	98.25
Impersonation	98.39	97.30	96.23	96.76	95.69	97.66
Flooding	98.71	97.03	97.86	97.44	96.58	98.43
Average	98.81	97.61	97.61	97.61	96.82	98.40

(Continued)

Table 2 (continued)

Class labels	Accuracy	Precision	Recall	F-Score	MCC	G-Mean
Training phase (70%)						
Normal	99.41	98.51	99.15	98.83	98.44	99.33
Injection	98.69	97.62	97.14	97.38	96.50	98.17
Impersonation	98.37	97.23	96.14	96.68	95.60	97.61
Flooding	98.70	96.97	97.89	97.43	96.56	98.43
Average	98.79	97.58	97.58	97.58	96.78	98.38
Testing phase (30%)						
Normal	99.33	98.43	98.89	98.66	98.22	99.18
Injection	98.85	97.68	97.68	97.68	96.91	98.45
Impersonation	98.43	97.44	96.42	96.93	95.88	97.77
Flooding	98.74	97.18	97.76	97.47	96.63	98.41
Average	98.84	97.68	97.69	97.68	96.91	98.45

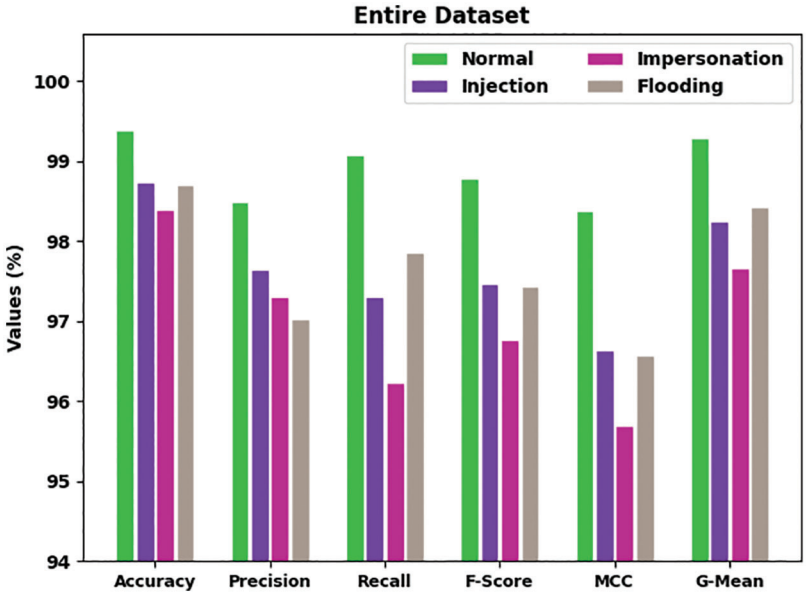

Figure 4: Analytical results of the COADL-FDIAR approach on entire dataset

Both Training Loss (TL) and Validation Loss (VL) values, acquired by the proposed COADL-FDIAR approach on test dataset, are displayed in Fig. 8. The experimental outcomes denote that the proposed COADL-FDIAR algorithm accomplished the least TL and VL values. In contrast, the VL values were lower than the TL values.

A clear precision-recall analysis was conducted on the COADL-FDIAR method using the test dataset. The results are exhibited in Fig. 9. The figure denotes that the proposed COADL-FDIAR method produced enhanced precision-recall values under all the classes.

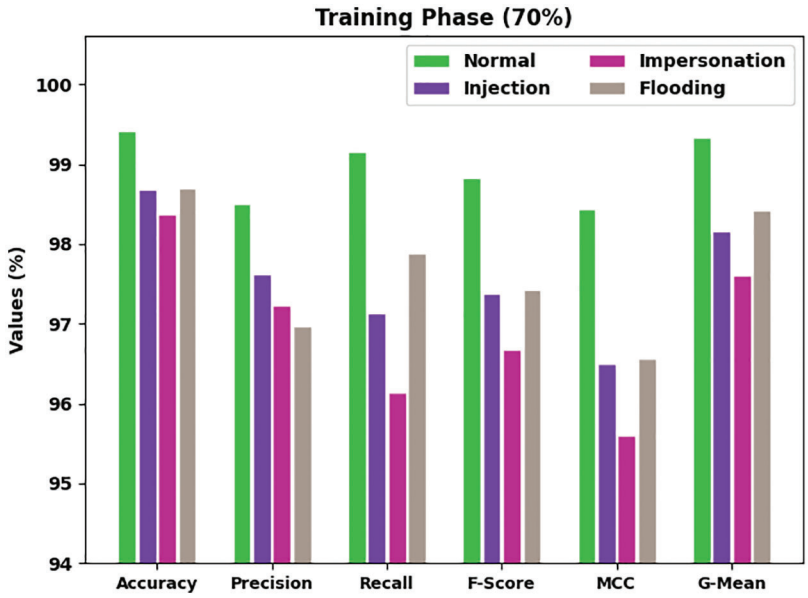

Figure 5: Analytical results of the COADL-FDIAR approach on 70% of the TR dataset

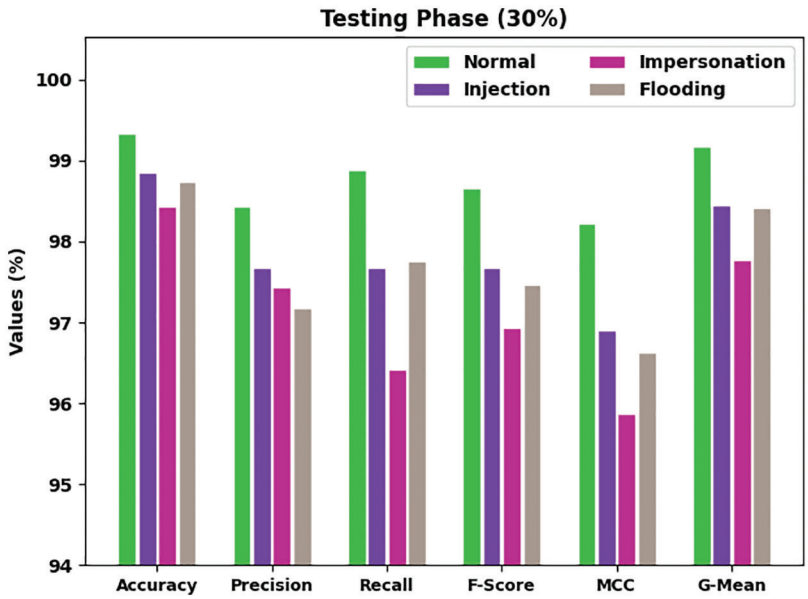

Figure 6: Analytical results of the COADL-FDIAR approach on 30% of the TS data

A brief Receiver Operating Characteristic (ROC) curve analysis was conducted on the proposed COADL-FDIAR technique using the test dataset. The results are shown in Fig. 10. The results denote that the proposed COADL-FDIAR approach established its ability to categorize the test dataset under distinct classes.

Table 3 displays the overall results of the proposed COADL-FDIAR model and other existing models under several measures [21].

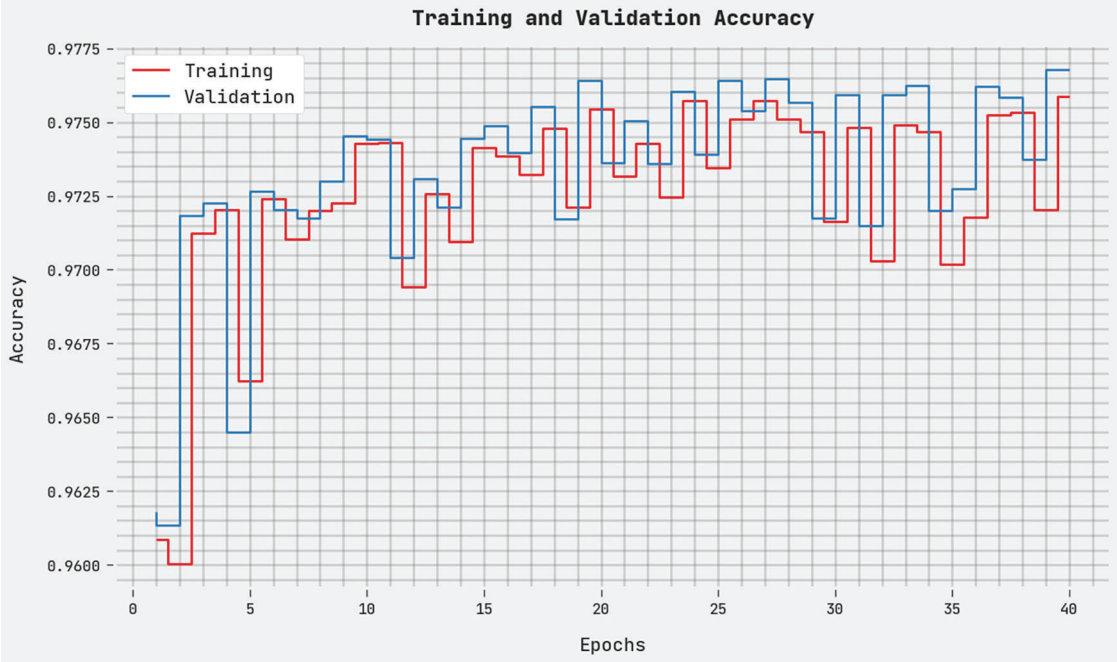

Figure 7: TA and VA analyses results of the COADL-FDIAR approach

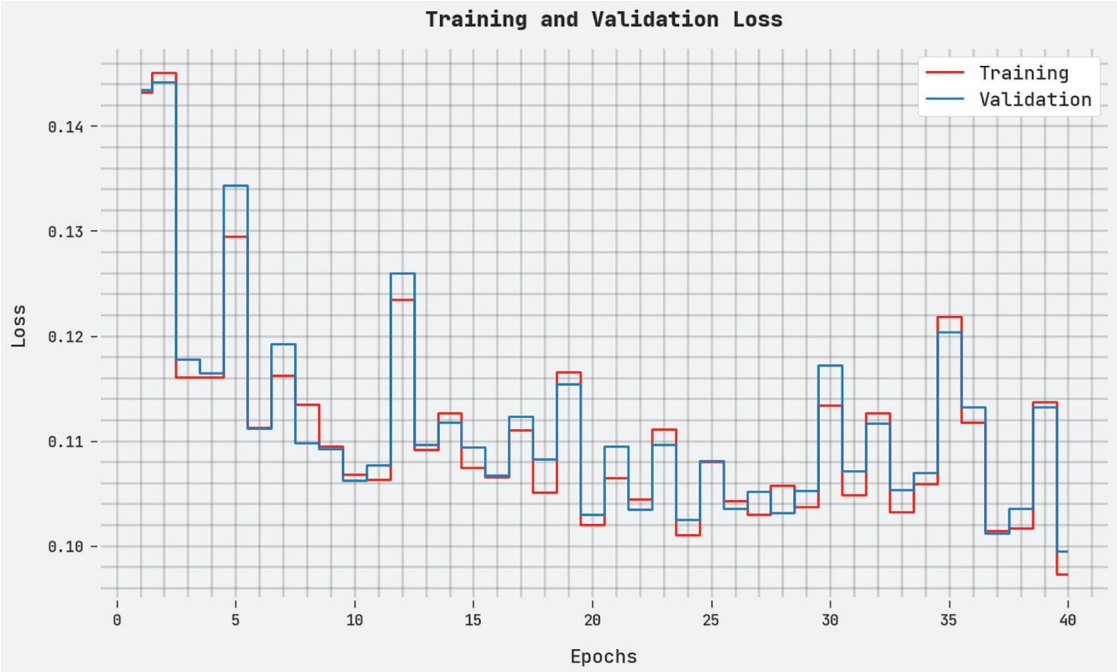

Figure 8: TL and VL analyses results of the COADL-FDIAR approach

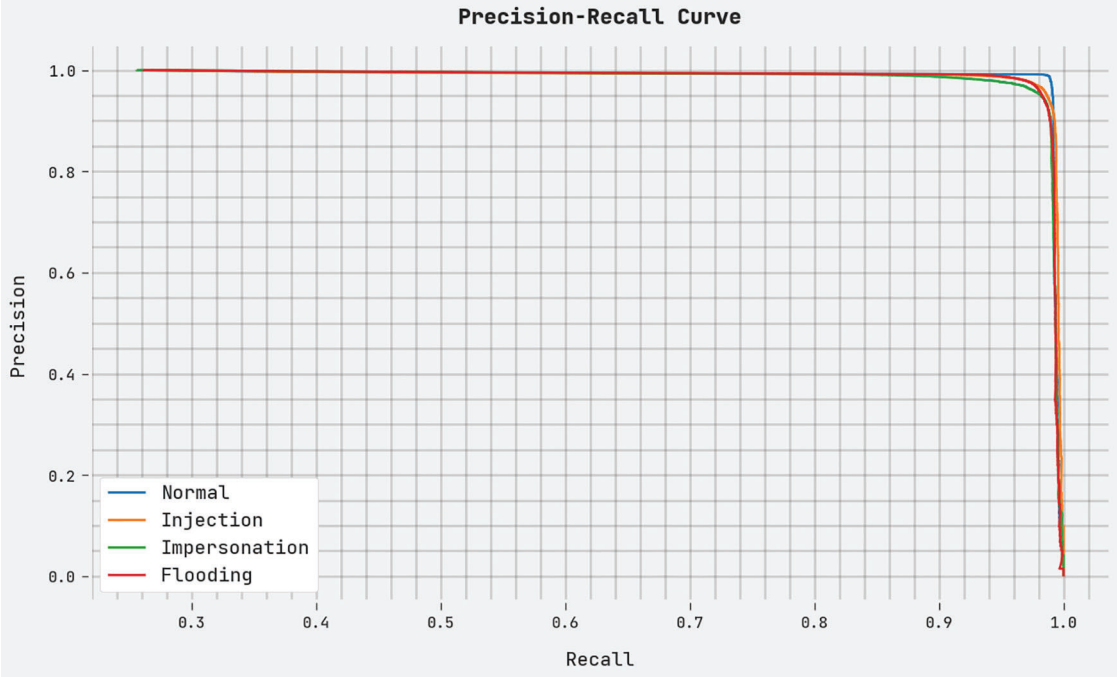

Figure 9: Precision-recall analysis results of the COADL-FDIAR approach

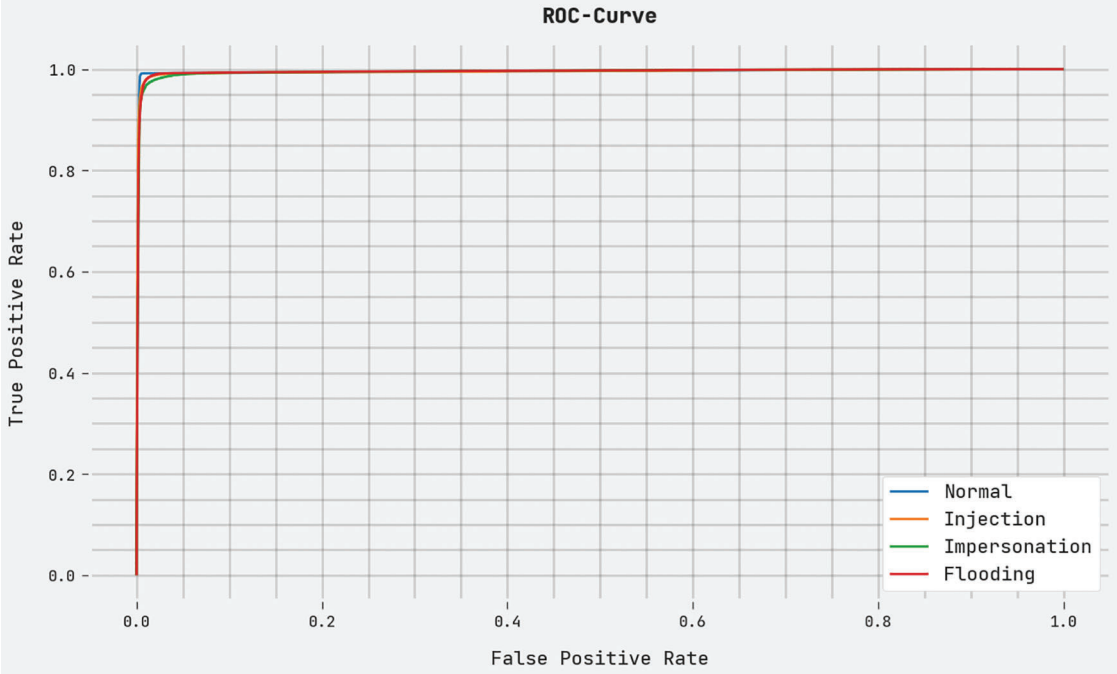

Figure 10: ROC analysis results of the COADL-FDIAR approach

Table 3: Comparative analysis results of the COADL-FDIAR approach and other existing algorithms

Methods	Accuracy	Precision	Recall	F-Score
COADL-FDIAR	98.84	97.68	97.69	97.68
Decision tree algorithm	96.30	97.36	94.31	95.37
SVM model	95.13	94.98	95.50	96.68
Random forest algorithm	96.79	94.59	95.73	95.96
Recurrent NN model	94.25	97.30	94.36	94.14
Deep NN model	95.96	97.05	96.90	97.11
LSTM model	96.78	95.02	95.90	96.98

Fig. 11 reports the brief $prec_n$ and $reca_l$ assessment results achieved by the proposed COADL-FDIAR and other existing models. The figure implies that the proposed COADL-FDIAR model produced increased $prec_n$ and $reca_l$ values. In terms of $prec_n$, the COADL-FDIAR technique achieved an enhanced $prec_n$ of 97.68%, whereas the Decision tree (DT), Support Vector Machine (SVM), Random Forest (RF), RNN, Deep Neural Network (DNN) and the LSTM approach produced the least $prec_n$ values such as 97.36%, 94.98%, 94.59%, 97.30%, 97.05% and 95.02% correspondingly. In terms of $reca_l$, the proposed COADL-FDIAR approach achieved an enhanced $reca_l$ of 97.69%, whereas the DT, SVM, RF, RNN, DNN and LSTM models accomplished low $reca_l$ values such as 94.31%, 95.50%, 95.73%, 94.36%, 96.90%, and 95.90% correspondingly.

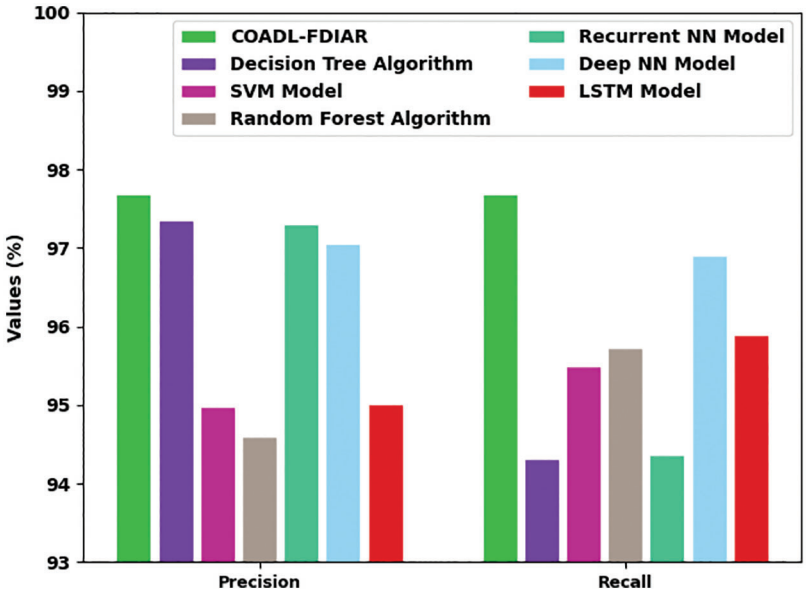

Figure 11: $Prec_n$ and $reca_l$ analyses results of the COADL-FDIAR approach and other existing algorithms

Fig. 12 demonstrates the detailed $accu_y$ and F_{score} analysis results of the proposed COADL-FDIAR and other existing approaches. The figure implies that the proposed COADL-FDIAR approach produced the increased $accu_y$ and F_{score} values. In terms of $accu_y$, the proposed COADL-FDIAR technique achieved an

enhanced $accu_y$ of 98.84%, whereas the DT, SVM, RF, RNN, DNN and the LSTM models achieved low $accu_y$ values such as 96.30%, 95.13%, 96.79%, 94.25%, 95.96% and 96.78% correspondingly.

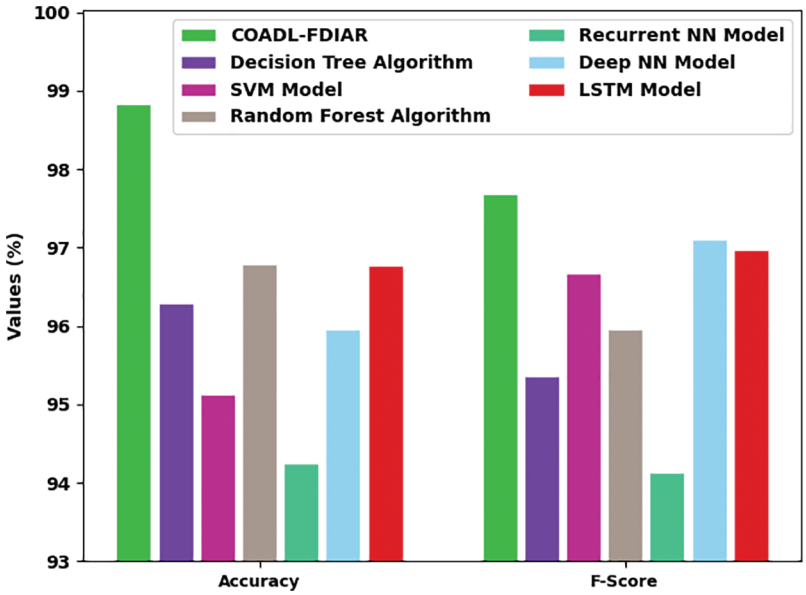

Figure 12: $Accu_y$ and F_{score} analyses results of the COADL-FDIAR approach and other existing algorithms

Moreover, in terms of F_{score}, the proposed COADL-FDIAR approach outperformed all other methods and achieved an enhanced F_{score} of 97.68%, whereas the DT, SVM, RF, RNN, DNN and LSTM methodologies achieved the least F_{score} values, such as 95.37%, 96.68%, 95.96%, 94.14%, 97.11% and 96.98% correspondingly. From these results, it can be inferred that the proposed COADL-FDIAR model achieved effectual outcomes over other models.

5 Conclusion

In this study, a new COADL-FDIAR algorithm has been developed to recognise false data injection attacks in the IoT environment. To accomplish this, the COADL-FDIAR model initially pre-processes the input data and selects the features using Chi-square test. For recognition and the classification of false data injection attacks, the SLSTM model is exploited in this study. Finally, the COA algorithm effectually adjusts the hyperparameters of the SLTSM model and accomplishes a superior recognition efficiency. The proposed COADL-FDIAR model was experimentally validated using a standard dataset and the outcomes were scrutinized under distinct aspects. The comparative analysis results confirmed the superior performance of the COADL-FDIAR model over other recent approaches with a maximum accuracy of 98.84%. In the future, the projected model can be tested on a large-scale real-time database.

Acknowledgement: The authors would like to thank Universiti Sains Malaysia (USM) and the Ministry of Higher Education Malaysia for providing the research grant, Fundamental Research GrantScheme (FRGS-Grant No: FRGS/1/2020/TK0/USM/02/1) that helped to carry out this research.

Funding Statement: This research was supported by the Universiti Sains Malaysia (USM) and the ministry of Higher Education Malaysia through Fundamental Research GrantScheme (FRGS-Grant No: FRGS/1/2020/TK0/USM/02/1).

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

References

[1] N. Kharlamova, S. Hashemi and C. Traholt, "The cyber security of battery energy storage systems and adoption of data-driven methods," in *2020 IEEE Third Int. Conf. on Artificial Intelligence and Knowledge Engineering (AIKE)*, Laguna Hills, CA, USA, pp. 188–192, 2020.

[2] B. Bostami, M. Ahmed and S. Choudhury, "False data injection attacks in internet of things," in *Performability in Internet of Things, EAI/Springer Innovations in Communication and Computing Book Series (EAISICC)*, Budapest, Hungary, pp. 47–58, 2018.

[3] H. T. Reda, A. Anwar, A. N. Mahmood and Z. Tari, "A taxonomy of cyber defence strategies against false data attacks in smart grid," arXiv preprint arXiv:2103.16085, 2021. https://doi.org/10.48550/arXiv.2103.16085.

[4] H. T. Reda, A. Anwar and A. Mahmood, "Comprehensive survey and taxonomies of false data injection attacks in smart grids: Attack models, targets, and impacts," *Renewable and Sustainable Energy Reviews*, vol. 163, pp. 112423, 2022.

[5] S. R. Mugunthan and T. Vijayakumar, "Review on IoT based smart grid architecture implementations," *Journal of Electrical Engineering and Automation*, vol. 1, no. 1, pp. 12–20, 2019.

[6] K. N. Qureshi, S. S. Rana, A. Ahmed and G. Jeon, "A novel and secure attacks detection framework for smart cities industrial internet of things," *Sustainable Cities and Society*, vol. 61, pp. 102343, 2020.

[7] Z. Qu, Y. Dong, N. Qu, H. Li, M. Cui *et al.,* "False data injection attack detection in power systems based on cyber-physical attack genes," *Frontiers in Energy Research*, vol. 9, pp. 644489, 2021.

[8] P. A. Giglou and S. N. Ravadanegh, "Defending against false data injection attack on demand response program: A bi-level strategy," *Sustainable Energy, Grids and Networks*, vol. 27, pp. 100506, 2021.

[9] C. Dunn, N. Moustafa and B. Turnbull, "Robustness evaluations of sustainable machine learning models against data poisoning attacks in the internet of things," *Sustainability*, vol. 12, no. 16, pp. 6434, 2020.

[10] M. Briland and F. Bouquet, "A language for modelling false data injection attacks in internet of things," in *2021 IEEE/ACM 3rd Int. Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT)*, Madrid, Spain, pp. 1–8, 2021.

[11] Z. He, J. Khazaei, F. Moazeni and J. D. Freihaut, "Detection of false data injection attacks leading to line congestions using neural networks," *Sustainable Cities and Society*, vol. 82, pp. 103861, 2022.

[12] Y. Hu, P. Zhu, P. Xun, B. Liu, W. Kang *et al.,* "CPMTD: Cyber-physical moving target defense for hardening the security of power system against false data injected attack," *Computers & Security*, vol. 111, pp. 102465, 2021.

[13] M. Dehghani, T. Niknam, M. Ghiasi, P. Siano, H. H. Alhelou *et al.,* "Fourier singular values-based false data injection attack detection in ac smart-grids," *Applied Sciences*, vol. 11, no. 12, pp. 5706, 2021.

[14] V. P. Srinivasan, K. Balasubadra, K. Saravanan, V. S. Arjun and S. Malarkodi, "Multi label deep learning classification approach for false data injection attacks in smart grid," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 6, pp. 2168–2187, 2021.

[15] A. Kumar, N. Saxena, S. Jung and B. J. Choi, "Improving detection of false data injection attacks using machine learning with feature selection and oversampling," *Energies*, vol. 15, no. 1, pp. 212, 2021.

[16] M. A. Shahid, F. Ahmad, F. R. Albogamy, G. Hafeez and Z. Ullah, "Detection and prevention of false data injection attacks in the measurement infrastructure of smart grids," *Sustainability*, vol. 14, no. 11, pp. 6407, 2022.

[17] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi-class SVM," *Journal of King Saud University—Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[18] A. Guo, A. Jiang, J. Lin and X. Li, "Data mining algorithms for bridge health monitoring: Kohonen clustering and LSTM prediction approaches," *The Journal of Supercomputing*, vol. 76, no. 2, pp. 932–947, 2020.

[19] F. Shahid, A. Zameer and M. J. Iqbal, "Intelligent forecast engine for short-term wind speed prediction based on stacked long short-term memory," *Neural Computing and Applications*, vol. 33, no. 20, pp. 13767–13783, 2021.

[20] A. S. Alqahtani, P. Saravanan, M. Maheswari and S. Alshmrany, "An automatic query expansion based on hybrid CMO-COOT algorithm for optimized information retrieval," *The Journal of Supercomputing*, vol. 78, no. 6, pp. 8625–8643, 2022.

[21] T. Gaber, A. E. Ghamry and A. Hassanien, "Injection attack detection using machine learning for smart IoT applications," *Physical Communication*, vol. 52, pp. 101685, 2022.