Feature Subset Search for Cybersecurity in Industrial Internet of Things Environment Using Coot Optimization Algorithm

Adil. O. Y. Mohamed[1,*], Yousef Asiri[2], Manahill I. A. Anja[3], Bandar M. Alghamdi[4], Abdelgalal O. I. Abaker[5], Mnahil M. Bashier[6]

[1]Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

[2]Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran, 61441, Saudi Arabia

[3]Computer Sciences Program, Department of Mathematics, Turabah University College, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

[4]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

[5]Department of Administrative Sciences, Applied College in Khamis Mushait, King Khalid University, Abha, Saudi Arabia

[6]Department of Mathematics, Faculty of Science, Northern Border University, Arar, Saudi Arabia

Emails: adi.mohamed@qu.edu.sa; yasiri@nu.edu.sa; meanja@tu.edu.sa; bmmalghamdi@kau.edu.sa; aoadrees@kku.edu.sa; mnahil.elradi@nbu.edu.sa

Abstract

The Industrial Internet of Things (IIoT) is the incorporation of industrial processes with smart technology and interconnected devices to improve productivity and efficiency. The need for robust cybersecurity measures is crucial as the IIoT environment becomes vital to critical infrastructure in industries. Cybersecurity in IIoT is paramount to secure against possible threats, which ensures the integrity and resilience of industrial operations. Intrusion detection systems (IDSs) are instrumental in detecting anomalies, unauthorized access, or malicious activities. The incorporation of deep learning (DL) further reinforces the cybersecurity posture of the IIoT network. DL approach excels in analyzing complex and large datasets, which enables the detection of complex cyber threats by learning anomalies and patterns. Industrial processes can operate with heightened security, securing sensitive information, and critical infrastructure, and maintaining the reliability of a connected system in the industrial landscape by combining IIoT cybersecurity with innovative intrusion detection and DL technologies. Therefore, this article proposes an Integration of Coot Optimization Algorithm-based Feature Subset Search with Deep Learning for Cybersecurity (COAFSS-DLCS) technique on IIoT network. The objective is in the effectual recognition and classification of cyberattacks in the IIoT environment. Initially, the COAFSS-DLCS method uses min-max scalar to transform the input dataset into a suitable format. Furthermore, the COAFSS-DLCS employs the COAFSS approach for choosing an optimal feature subset. Additionally, the stacked long short-term memory autoencoder (SLSTM-AE) model is employed for classification. Moreover, the parameters of the SLSTM-AE classifier are fine-tuned using the Arithmetic Optimization Algorithm (AOA) for improved performance. A comprehensive empirical validation of the COAFSS-DLCS approach is performed under the UNSW_NB15 and UCI_SECOM datasets. The simulation outputs inferred the power of the COAFSS-DLCS over other methods.

Keywords: Industrial Internet of Things; Cybersecurity; Coot Optimization Algorithm; Hyperparameter Selection; Deep Learning

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

1. Introduction

The IIoT is considered a world where vital surroundings like water systems, power plants, and industries are controlled and monitored across the network [1]. The implementation of IIoT can enable the optimization of several industrial sectors, comprising the more intricate industrial surroundings. However, all of this interconnection should expand the cyberattack surface to different levels [2]. The risks are increased by vulnerabilities in the devices and their situation to attacks like unprotected hardware and firmware, insufficiently designed software, and unsafe physical access that may cause loss to equipment, industrial surveillance, and interruption of process controls, reduction of users and corporate information, and stealing of intellectual property [3]. The collection of cybersecurity attacks is developed in the digital era that will be mostly targeted to industries and organizations at huge [4]. Numerous cybersecurity issues and solutions are handled in these surroundings to mitigate security attacks and vulnerabilities. There are well-developed security substructures for all IIoT implementations; an organization must satisfy a significant number of losses to its resources and reputation over time [5]. Meanwhile, confidential and sensitive data are continuously transferred through the networks; the most dominant issue in IIoT networks is the data security problems [6].

Intruders will initiate potential attacks in the nodes interconnected with the wireless network. Moreover, a cloud platform must be employed for controlling the IoT device linked via a gateway [7]. Accordingly, once the gateway node is cooperated by an attacker, the entire IIoT model is miserable and developed as significant one. Every end-to-end section will be encrypted in the end-points to prevent security attacks across the networks [8]. The shortage of developed industrial processes, advanced technologies, and expansion of various device criteria make difficulties in the IIoT field and complex to handle all types of security attacks. As a result, a well-designed cybersecurity solution becomes crucial for protecting the IIoT environment. In these sceneries, among the above-mentioned machine learning (ML), DL techniques, and their applications, anomaly detection is a major interest [9]. During the method of detection of anomalies in IoT-generated data, a few authors considered identifying errors in device functions or communicational errors in an intricate IoT platform for example, smart city, whereas alternative emphasis on recognizing various categories of security attacks, like distributed denial-of-service (DDoS) attacks, IoT pivot, device tampering, malware analysis and botnet [10]. A few of the DL-based methods employed for anomaly detection comprise recurrent neural networks (RNNs), and autoencoders (AE).

This article proposes an Integration of Coot Optimization Algorithm-based Feature Subset Search with Deep Learning for Cybersecurity (COAFSS-DLCS) technique on IIoT network. The objective is in the effectual recognition and classification of cyberattacks in the IIoT environment. Initially, the COAFSS-DLCS method uses min-max scalar to transform the input dataset into a suitable format. Furthermore, the COAFSS-DLCS employs the COAFSS approach for choosing an optimal feature subset. Additionally, the stacked long short-term memory autoencoder (SLSTM-AE) model is employed for classification. Moreover, the parameters of the SLSTM-AE classifier are fine-tuned using the Arithmetic Optimization Algorithm (AOA) for improved performance. A comprehensive empirical validation of the COAFSS-DLCS approach is performed under the UNSW_NB15 and UCI_SECOM datasets.

2. Related Works

Maghrabi et al. [11] proposed a Bald Eagle Search Optimizer alongside the Hybrid DL-driven botnet detection (BESO-HDLBD) model. The main goal is to classify the botnets in an IoT infrastructure. The approach deploys the BESO for feature selection (FS) procedure to decrease the higher dimensional issue. The technique also utilizes HDL, an integration of CNN, Bi-LSTM, and attention idea. In [12], a DL method called DeBot is developed. DeBot utilizes a novel Cascade Forward Backpropagation Neural Networks (CFBPNNs) technique with a feature subset utilizing the Correlation-based FS (CFS) method. The method also presented the usage of optimum FS and combination with the cascading technique of DL. Soliman et al. [13] developed an intelligent recognition method. This approach employs the SVD model to increase recognition outcomes and decrease data features. In addition, the technique leveraged the SMOTE approach to evade under-fitting and overfitting concerns that outcome in the identification of bias. Many ML as well as DL approaches are employed for categorizing data for dual as well as multi-class classification. Latif et al. [14] presented an enhanced IDS based on the DTL technique. This structure uses a tri-layer architecture model that synergistically merges CNN, Genetic Algorithms (GA), and bootstrap aggregation ensemble models. This model was implemented in 3 vital phases: At initial, the cybersecurity dataset is transformed. Then, GA is employed to modify the hyperparameters. At last, the output of the best-performing techniques is combined utilizing ensemble models.

In [15], a novel Privacy-Preserving BC with DL method for Industrial IoT (PPBDL-IIoT) approach is introduced. This approach contains the project of Chaos Game Optimizer (CGO) alongside a Bi-GRNN model. In addition, the CGO model is used to alter the hyperparameters. CGO model is also then used. Furthermore, the Blockchain-enabled Integrity Check (BEIC) structure is presented. The authors [16] developed a novel cloud-empowered cyber-attack recognition framework dependent upon the Ensemble Bagged Trees Detection (EBTD) process. The

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

presented architecture executes the ANOVA technique and the priority-based FS and extractor technique to discover the optimum features with extremely reliant on the computational time, malicious behaviours, network traffic, and other kinds of attack. In [17], a novel IoT model is presented by leveraging a deep CNN. The developed methodology manages the signal process by adapting the RSSI signal into imagery. The 1D RSSI signal is changed into 2D imagery data for generating the novel feature dependent upon continuous wavelet transform (CWT), and the suggested DCNN is employed. Jalil Piran et al. [18] presented the hyperdimensional computing (HDC) technique. Integrating a brain cell regeneration model further enhances learning performance and reduces memory usage.

3. Proposed Methodology

In this article, the COAFSS-DLCS technique is proposed. It contains four major sub-processes namely data pre-processing, COA-based feature reduction, SLSTM-AE-based attack classification, AOA based tuning. Fig. 1 represents the flow of the COAFSS-DLCS model.

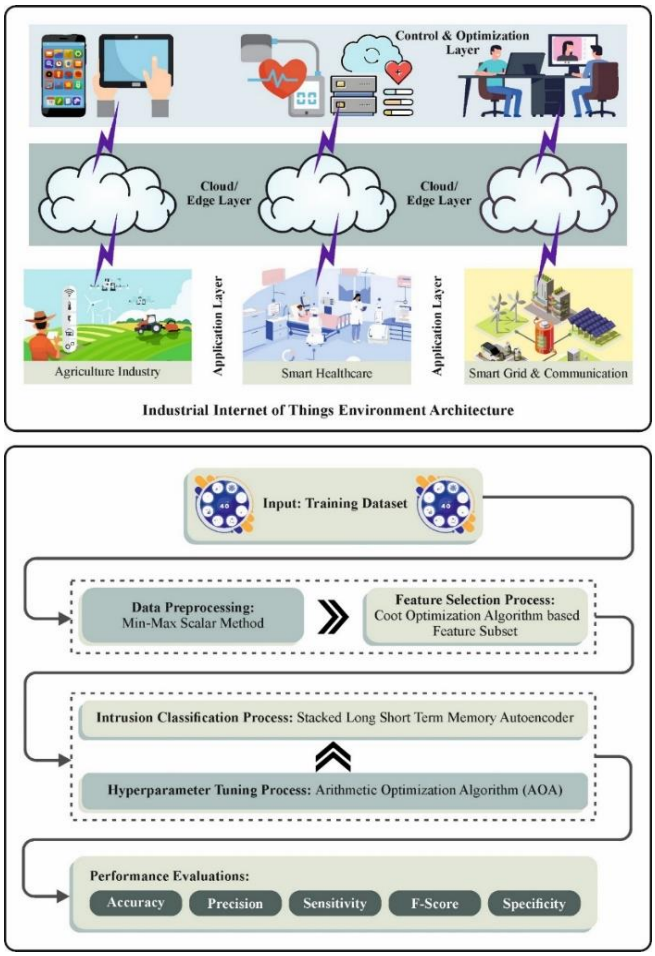

Figure 1. Overall flow of COAFSS-DLCS model

A. Data Pre-processing

At first, the COAFSS-DLCS method uses min-max scalar to transform the input dataset into a suitable format. This is a classical data pre-processing method used to normalize the range of numerical data features [19].

The process includes transforming the data point by dividing the result (the dissimilarity between minimal and maximal values) and subtracting the minimal value of the feature. This ensures that the value in the feature lies in [0,1]. Min-Max scaling is especially effective while working with ML techniques to the scale of input features, as it prevents features with large scales from dominating the model. Min-Max scaling contributes to improved model performance and convergence, promoting fair comparisons between variables in dissimilar datasets by bringing all features to a standardized range. This normalization method is extensively used in fields including clustering, image processing, and neural networks, where maintaining consistent feature scales is crucial for effective modelling and analysis.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

B. Design of COAFSS Technique

The COAFSS approach is used to elect an optimal feature subset. COA is inspired by the group behaviours of coots [20]. These movements include both irregular and regular patterns on the water surface. The group member tries to move towards the prey. Hence, the existing position is updated by considering the leader's position in the group. COA operates through four key mechanisms: random walking, chain movement, location updates based on group leaders, and leader-guided movement toward the optimal region. The population initialization ($\vec{x} = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3}, \dots, \overrightarrow{x_n}\}$) is generated at random. The initial location of the individual is randomly formed using the following equation:

$$CootPos(i) = rand(1, d) * (ub - lb) + lb \quad (1)$$

In Eq. (1), $CootPos(i)$ implies the the i^{th} coot location, i^{th} indicates the index number of the present individual, d shows the amount of decision variables, the upper and lower limitations within the problem space are represented as ub and lb, correspondingly.

Random walking

Using Eq. (2), a random location ($RandPos$) is produced in the ub and lb for changing the location of the existing individuals.

$$RandPos = rand(1, d) * (ub - lb) + lb \quad (2)$$

The updating process of the existing location with random walking enables it to depart from the local optimal location. The location update rule is evaluated by Eq. (3) to attain the new location of the existing individual:

$$CootPos(i) = CootPos(i) + A \times R2 \times \big(RandPos - CootPos(i)\big) \quad (3)$$

In Eq. (3), $R2$ is a random integer [0,1]; A value is evaluated in Eq. (4).

$$A = 1 - L \times \left(\frac{1}{Iter}\right) \quad (4)$$

Where L and $Iter$ are the existing and maximum amount of iteration.

Chain movement

The distance vector between two individual coots is evaluated once the chain movement is used, then; the first individual goes towards the others by about half of the distance vectors. The new location of the present individual coot is produced using chain movement as follows.

$$CootPos(i) = 0.5 \times \big(CootPos(i-1) + CootPos(i)\big) \quad (5)$$

Updating the position based on the group leaders

Some individuals in a group adjust the existing position towards the quality food source, and others change the existing location based on the group leader's position. This is mathematically given in the following:

$$K = 1 + (iMODNL) \quad (6)$$

Where the group leader index number is expressed by K, and NL indicates the leader count in the population. $coot(i)$ has updated the existing location with the information of k^{th} leaders. The chosen leader evaluates the next location of the individuals:

$$CootPos(i) = LeaderPos(k) + 2 \times R1 \times Cos(2R\pi) \times \big(LeaderPos(k) - CootPos(i)\big) \quad (7)$$

In Eq. (7), the R-value is an arbitrary integer range $[-1, +1]$, and the constant value π is set as 3.14. The index number of the existing leader is represented as k, $LeaderPos(k)$ denotes the designated leader location, $R1$ value shows a random value within [0,1],

The leader guides the group toward the optimal region (leader movement)

In COA, the coot individuals should be within the search range. If any dimension of a coot individual goes beyond the search range, it is adjusted back within the range. In the optimization problem, the individuals keep their location using the group leader's location within the search range.

$$LeaderPos(i)$$

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

$$= \begin{Bmatrix} B \times R3 \times & cos(2R\pi) \times (gBest - & LeaderPos(i)) + & gBest & R4 < 0.5 \\ B \times R3 \times & cos(2R\pi) \times (gBest - & LeaderPos(i)) - & gBest & R4 >= 0.5 \end{Bmatrix} \quad (8)$$

The FF applied in the COAFSS technique is intended to strike a balance between the attributes chosen in the classification accuracy (maximum) and each solution (minimum) attained by the selected attributes.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (9)$$

Where $\gamma_R(D)$ depicts the classifier error rate. α and β represent the significance of classifier quality and subset length. $\in$ [1,0] and $\beta = 1 - \alpha$. $|R|$ denote the cardinality of the chosen subset and $|C|$ refers the overall feature quantity.

C. Cyberattack Detection using SLSTM-AE

In this work, the SLSTM-AE technique is used for cyberattack detection. As an unsupervised learning algorithm, the LSTM-based AE is that exclusively exploits the LSTM model for the encoder and decoder [21]. AE is used to learn encoder-decoder from high-dimension input through the FFNN model where the high-dimensional data is inputted to the hidden state, and then encoded to generate lower-dimensional data. The encoder accepts an input series of x_i and encodes them into a latent depiction of h_i as follows:

$$Encoding: h_i(x) = E(w_e x + b_e) \quad (10)$$

The h_i hidden data is inputted to the decoder that reconstructs the actual data from it based on the following equation

$$Decoding: \hat{x} = D(W_{cl} h_i(X) + b_d) \quad (11)$$

In Eq. (11), $h_i(x)$ portrays the i^{th} hidden output vector from the x input dataset, and $\hat{x}$ shows the last output represented by the letters E and D. The weight matrix of the encoder and decoder is w_e and w_d, whereas b_e and b_d are the bias values for the encoder and decoder, correspondingly. Fig. 2 shows the SLSTM architecture.

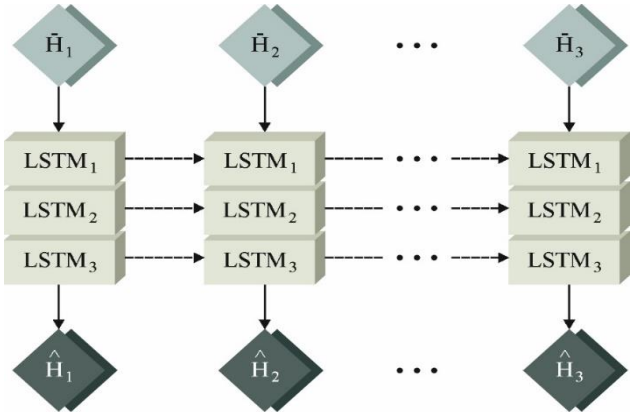

Figure 2. Structure of SLSTM

The SLSTM-AE model learns complex dynamics with the temporal sequence of input. Therefore, LSTM exploits memory to retain the information over a long input sequence. It effectively captures the temporal connection in multidimensional data. All the layers are separately trained, and the BP model to enhance the performance executes finetuning. A supervised learning method is used to finetune for minimise the prediction errors. The model focuses on learning hidden features and decreasing the reconstructed errors. Once the reconstructed error evaluated by Eq. (6) is smaller. Now, the decoder is eliminated, which leaves the encoder for extracting the hidden features of the input dataset. In the proposed work, the weather factor is encoded to deliver the optimum outcome.

$$Reconstruction\ Error = ||x - \hat{x}||^2 \quad (12)$$

D. Hyperparameter Selection using AOA

Lastly, the AOA is exploited to select the hyperparameter value of the SLSTM-AE model. In general, mathematical operators such as multiplication, addition, subtraction, and division are employed in a metaheuristic technique recognized as AOA [22]. To perform the optimizer over many search fields, they are employed and demonstrated. The population-based algorithms (PBA) normally begin the procedure of enhancing their systems by arbitrarily

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

choosing some candidate models. An exact objective function gradually estimates this specified response while using a set of optimizer values to slowly increase it. The best common solution to the issue is elevated by the accessibility of substitute solutions. The optimizer procedure is separated into dual cycles namely exploitation and exploration, by taking differences among metaheuristic techniques in PBA. Algebra and geometry mathematics are the most significant modules. Usually, Arithmetic operators (AO) are employed in the research of numbers. A few simple mathematical processes are applied while using optimization to discover ideal components, particularly with selected solutions. The foremost driving force is the usage of AO to find out the issues. The method of optimization begins with some appropriate sets that are signified by B in Eq. (28). In a perfect set, every iteration is arbitrarily produced.

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & \cdots & b_{1,i} & b_{1,1} & b_{1,m} \\ b_{2,1} & b_{2,2} & \cdots & \cdots & a_{2,i} & \cdots & b_{2,m} \\ b_{2,1} & b_{3,2} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_{M-1,1} & \cdots & \cdots & \cdots & b_{M-1,i} & \cdots & b_{M-1,m} \\ b_{M,1} & \cdots & \cdots & \cdots & a_{M,i} & b_{M,m-1} & b_{M,m} \end{bmatrix} \tag{13}$$

Exploitation and Exploration should be cautiously considered at the beginning of AOA. The mathematics optimizer's enhanced co-efficient is definite by the formula as follows.

$$MOA(D_{iter}) = \text{Min} + D_{iter}y\left(\frac{\text{Max} - \text{Min}}{E_{iter}}\right) \tag{14}$$

Here, D_{iter} is signified as the present iteration. Max and Min specify the accelerated function values. $MOA(D_{iter})$ represents the function value of k^{th} iteration, $and\ E_{iter}$ is denoted as the highest iteration count.

The AOA experimental module is tested and observed that, as per AO, calculations employing the multiplication or division operations have produced great distribution value that supports the exploration model. The operators of multiplication and division never effortlessly achieve the goal when compared to other operators due to the higher distribution of addition and subtraction operations. Using the search area generally during numerous fields, AOA exploration operators are a superior choice for employing the dual major search tactics such as multiplication and division as presented in the formulation below.

$$b_{k,i}(D_{iter}+1) = \begin{cases} bestb_i \div (MOP \div \omega) \times ((TV_i - UV_i) \times \lambda + TV_i), p_2 < 0.5 \\ bestb_i \times MOP \times \left((TV_i - UV_i) \times \lambda + UV_i\right), otherwise \end{cases} \tag{15}$$

Here, λ signifies the control parameter ≤ 0.5, $b_k(D_{iter}+1)$ specifies k^{th} solution of subsequent iteration ω directs the minimum integer count, $b_{k,i}(D_{iter}+1)$ represents the i^{th} location in the existing iteration, $bestb_i$ denotes the k^{th} location of optimal solution achieved up till now, TV_j and UV_i is the upper and lower bound.

$$MOP(D_{iter}) = 1 - \frac{D_{iter}^{\frac{1}{\beta}}}{S_{iter}^{\frac{1}{\beta}}} \tag{16}$$

In Eq. (16), M_{iter} designates the Max iterations ≤ 5, $MOP(D_{iter})$ signified as k^{th} an iteration function value, math optimizer probability (MOP) specifies co-efficient. D_{iter} refers to the existing iteration. As per AO mathematical formulations, which formed higher-density outcomes whether using subtraction or addition, the nature of AOA is surveyed. AOA exploitation operations utilize dual key search methods to carefully examine the search area over many locations of a superior alternative. Addition and subtraction search models are mentioned in the Eq. (17),

$$b_{k,i}(D_{iter}+1) = \begin{cases} bestb_i - MOP \times \left((TV_i - UV_i) \times \lambda + UV_i\right), p_3 < 0.5 \\ bestb_i + MOP \times \left((TV_i - UV_i) \times \lambda + UV_i\right), otherwise \end{cases} \tag{17}$$

The AOA model uses a fitness function (FF) to improve classifier efficiency, where a positive integer indicates better candidate performance. Minimizing classification error acts as the basis for computing the FF.

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{No.of\ misclassified\ instances}{Overall\ instances} \times 100 \end{aligned} \tag{18}$$

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

4. Experimental Validation

The performance study of the COAFSS-DLCS method takes place on UNSW_NB15 [23] and UCI_SECOM [24] datasets. Tables 1 and 2 describes the dataset.

Table 1: Dataset description

UNSW_NB15 Dataset	
Class	Instance Numbers
"Normal"	"1000"
"Generic"	"1000"
"Exploits"	"1000"
"Fuzzers"	"1000"
"DoS"	"1000"
"Reconnaissance"	"1000"
"Analysis"	"1000"
"Backdoor"	"1000"
"Shellcode"	"1000"
"Worms"	"1000"
Overall Instances	10000

Table 2: Dataset description

UCI_SECOM Dataset	
Class	Sample Numbers
Class1	2500
Class2	2500
Overall Instances	5000

The classifier results of the COAFSS-DLCS method under the UNSW_NB15 dataset are demonstrated in Fig. 3. The confusion matrices of the COAFSS-DLCS on 70%:30% of TRAST/TESST are portrayed in Figs. 3a-3b. The outcome denoted that the COAFSS-DLCS approach precisely detects and classifies all 10 classes. The classification detection outcome of the COAFSS-DLCS method is shown in Figs. 3c-3d. These results point out that the COAFSS-DLCS approach has reached effectual recognition rates.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

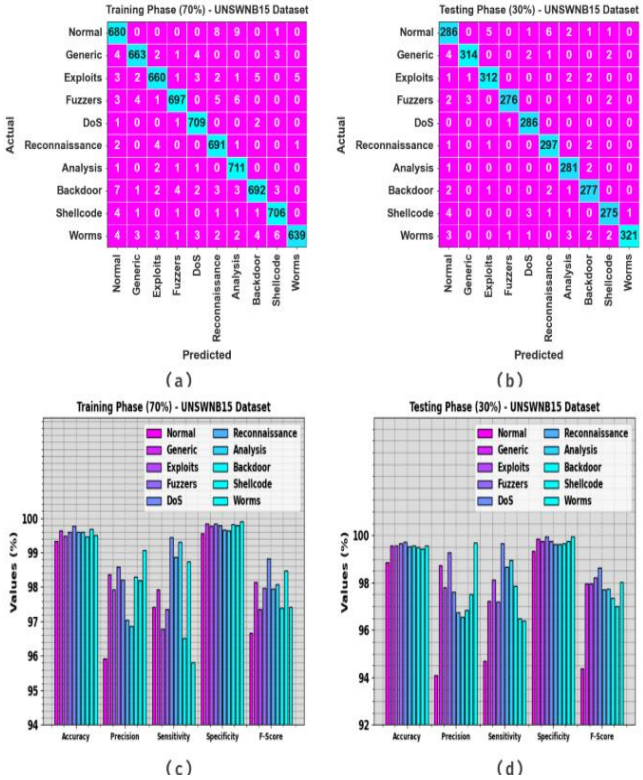

Figure 3. UNSW_NB15 dataset (a-b) confusion matrices and (c-d) classifier results under 70%TRAST and 30%TESST

The detection performance of the COAFSS-DLCS method under the UNSW_NB15 dataset is specified in Table 3 and Fig. 4. The outputs demonstrate that the COAFSS-DLCS approach provides high recognition performance on both TRAST and TESST datasets, emphasizing robust values across evaluations.

Table 3: Detection output of COAFSS-DLCS method at UNSW_NB15 dataset

UNSW_NB15 Dataset					
Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
TRAST (70%)					
Normal	99.33	95.91	97.42	99.54	96.66
Generic	99.64	98.37	97.93	99.83	98.15
Exploits	99.49	97.92	96.77	99.78	97.35
Fuzzers	99.59	98.59	97.35	99.84	97.96
DoS	99.76	98.20	99.44	99.79	98.82
Reconnaissance	99.59	97.05	98.86	99.67	97.94
Analysis	99.60	96.87	99.30	99.63	98.07
Backdoor	99.47	98.30	96.51	99.81	97.40
Shellcode	99.69	98.19	98.74	99.79	98.47

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

Worms	99.51	99.07	95.80	99.91	97.41
Average	99.57	97.85	97.81	99.76	97.82
TESST (30%)					
Normal	98.87	94.08	94.70	99.33	94.39
Generic	99.57	98.74	97.21	99.85	97.97
Exploits	99.57	97.81	98.11	99.74	97.96
Fuzzers	99.67	99.28	97.18	99.93	98.22
DoS	99.73	97.61	99.65	99.74	98.62
Reconnaissance	99.53	96.74	98.67	99.63	97.70
Analysis	99.57	96.56	98.94	99.63	97.74
Backdoor	99.50	96.85	97.88	99.67	97.36
Shellcode	99.43	97.52	96.49	99.74	97.00
Worms	99.57	99.69	96.40	99.96	98.02
Average	99.50	97.49	97.52	99.72	97.50

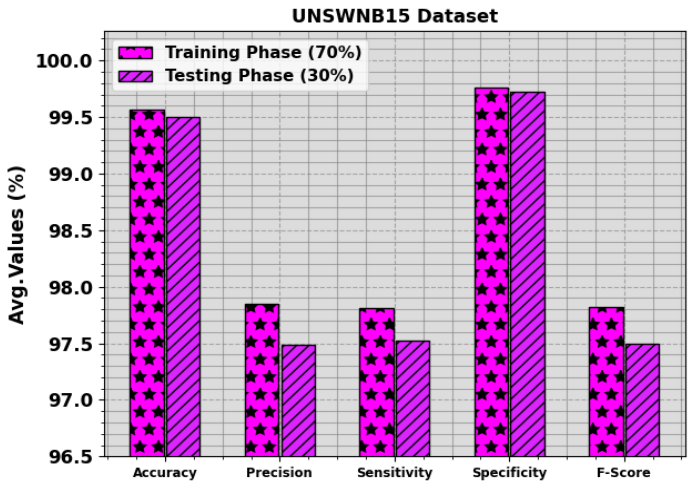

Figure 4. Average of the COAFSS-DLCS method on UNSW_NB15 dataset

Fig. 5 shows the efficiency of the COAFSS-DLCS approach on the UNSW_NB15 dataset through training accuracy (TRAA) and validation accuracy (VALA) accuracy. The results highlight its effective learning and robust ability to generalize to unseen data.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

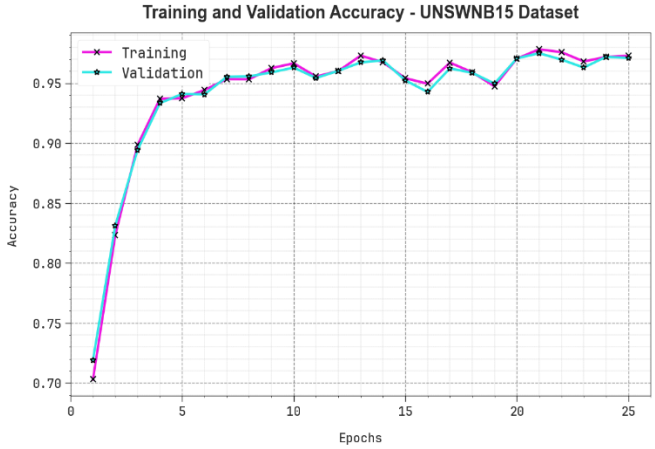

Figure 5. $Accu_y$ Curve of the COAFSS-DLCS model under UNSW_NB15 dataset

Fig. 6 shows a demonstration of the training loss (TRLA) and validation loss (VALL) outputs of the COAFSS-DLCS method under the UNSW_NB15 dataset in distinct epochs. The figure states a well-defined interpretation of the COAFSS-DLCS technique connected to the TRA data, underlining its effectiveness.

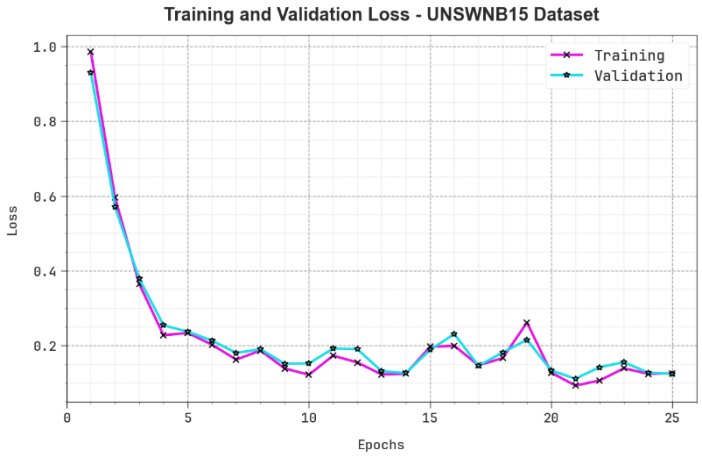

Figure 6. Loss curve of the COAFSS-DLCS model under UNSW_NB15 dataset

Fig. 7 depicts the PR curve of the COAFSS-DLCS methodology under the UNSW_NB15 dataset, which increasingly achieves better value of PR in every class. It confirms the upgraded capacities of the COAFSS-DLCS in 10 class's detection.

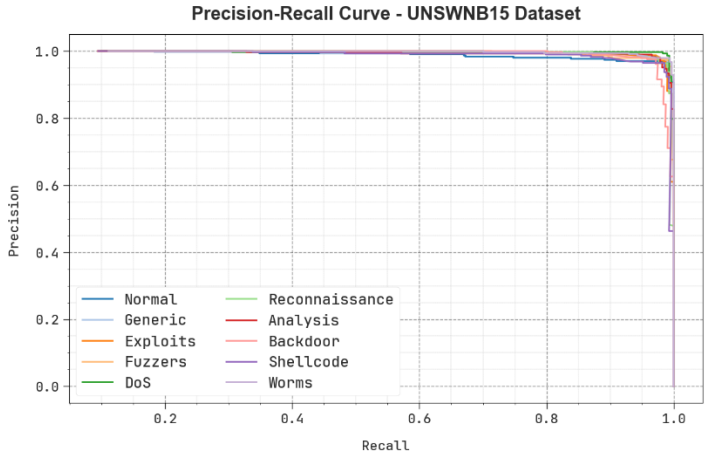

Figure 7. PR curve of the COAFSS-DLCS model under UNSW_NB15 dataset

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

Moreover, in Fig. 8, ROC curves of the COAFSS-DLCS model under the UNSW_NB15 dataset outperformed the distinct class classification. The figure gives emphasis to the superior classifier outcomes of the COAFSS-DLCS methodology on each class, indicating the proficiency in overcoming several classification complexities.

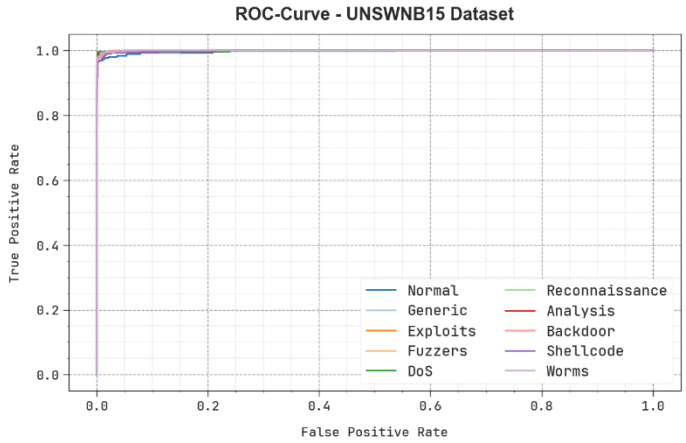

Figure 8. ROC curve of the COAFSS-DLCS model on UNSW_NB15 dataset

The comparative outcomes of the COAFSS-DLCS approach on the UNSW_NB15 dataset are portrayed in Fig. 9 [25]. These experimentation outcome values show that the ANN and SVM techniques have reported worse performance. Simultaneously, the KNN, DT, SSA-CRNN, and MFSDL-ADIIoT methods reached somewhat boosted results. Meanwhile, the GJODL-CADC model has accomplished reasonable performance. Nevertheless, the COAFSS-DLCS technique shows maximum performance with $prec_n$ of 97.85%, $reca_l$ of 97.81%, $accu_y$ of 99.57%, and F_{score} of 97.82%.

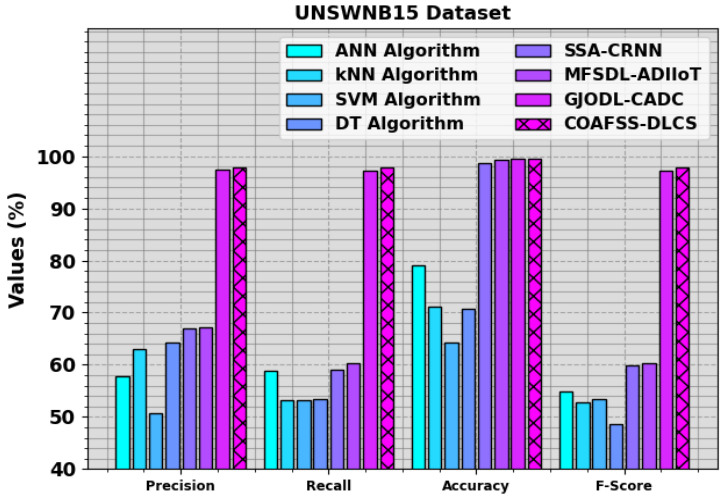

Figure 9. Comparison assessment of the COAFSS-DLCS approach under the UNSW_NB15 dataset

The classifier outcomes of the COAFSS-DLCS method UCI_SECOM dataset. Figs. 10a-10b demonstrates the confusion matrices of the COAFSS-DLCS approach on 70%:30% of TRAST/TESST. The classification detection outcome of the COAFSS-DLCS approach is portrayed in Figs. 10c-10d. These outcomes point out that the COAFSS-DLCS approach has reached effectual recognition rates.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

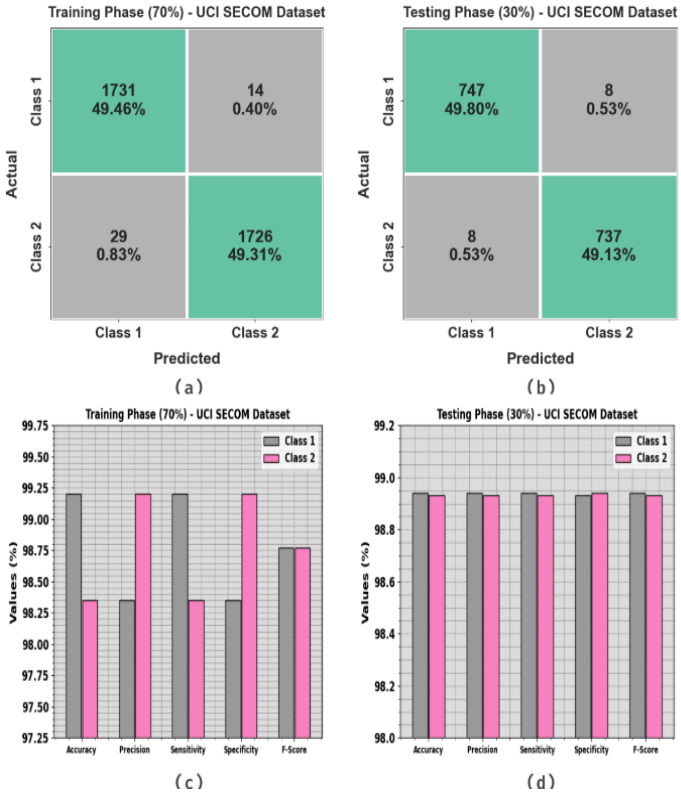

Figure 10. UCI_SECOM dataset (a-b) confusion matrices and (c-d) classifier results under 70%TRAST and 30%TESST

The detection outcomes of the COAFSS-DLCS method at the UCI_SECOM dataset is reported in Table 4 and Fig. 11. These obtained values pointed out that the COAFSS-DLCS method obtains effectual recognition rates. Under both 70% training and 30% testing splits, the COAFSS-DLCS approach specifies consistently high performance across all key evaluation metrics, indicating robust reliability and effectiveness in fraud detection.

Table 4: Detection output of the COAFSS-DLCS approach on the UCI_SECOM dataset

UCI_SECOM Dataset					
Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
70% of TRAST					
Class1	99.20	98.35	99.20	98.35	98.77
Class2	98.35	99.20	98.35	99.20	98.77
Average	98.77	98.77	98.77	98.77	98.77
30% of TESST					
Class1	98.94	98.94	98.94	98.93	98.94
Class2	98.93	98.93	98.93	98.94	98.93
Average	98.93	98.93	98.93	98.93	98.93

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

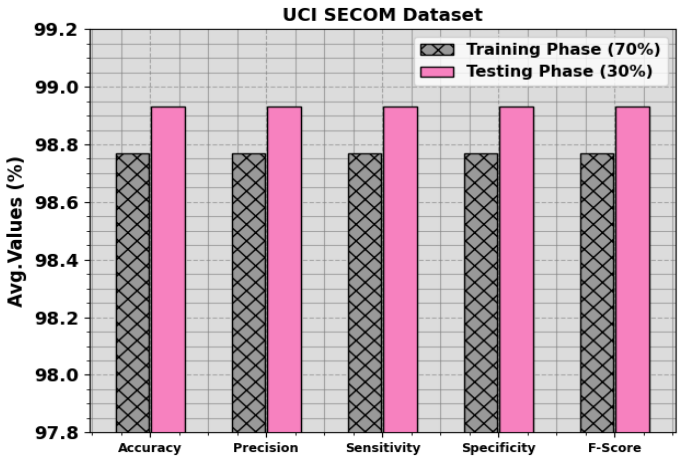

Figure 11. Average of the COAFSS-DLCS approach on the UCI_SECOM dataset

Fig. 12 illustrates the performance of the COAFSS-DLCS technique under the UCI_SECOM dataset using TRAA and VALA accuracy graphs. The upward trend in VALA accuracy highlights the robust learning behavior of the model and its capability in generalizing well to unseen data.

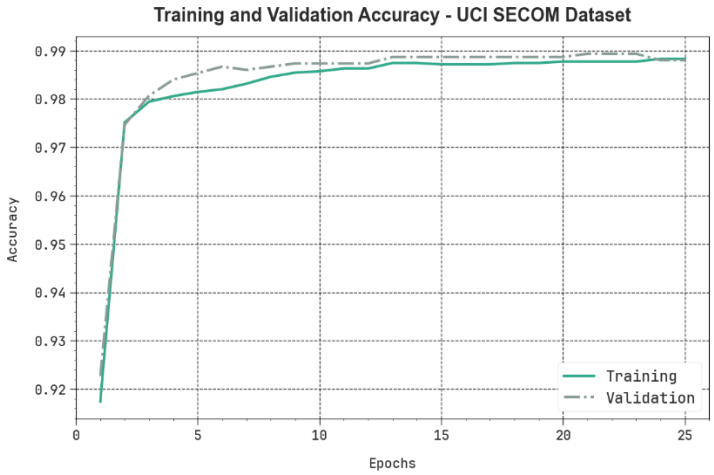

Figure 12. $Accu_y$ Curve of the COAFSS-DLCS approach under UCI_SECOM dataset

Fig. 13 indicates the TRLA and VALL outputs of the COAFSS-DLCS method under UCI_SECOM dataset. The COAFSS-DLCS model under the TRA data emphasizes its aptitude to capture patterns in both datasets.

Figure 13. Loss curve of the COAFSS-DLCS method at UCI_SECOM dataset

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

Fig. 14 portrays the PR curve of the COAFSS-DLCS method under the UCI_SECOM dataset over every class. It proves the enriched facilities of the COAFSS-DLCS approach in the 2 class's identification.

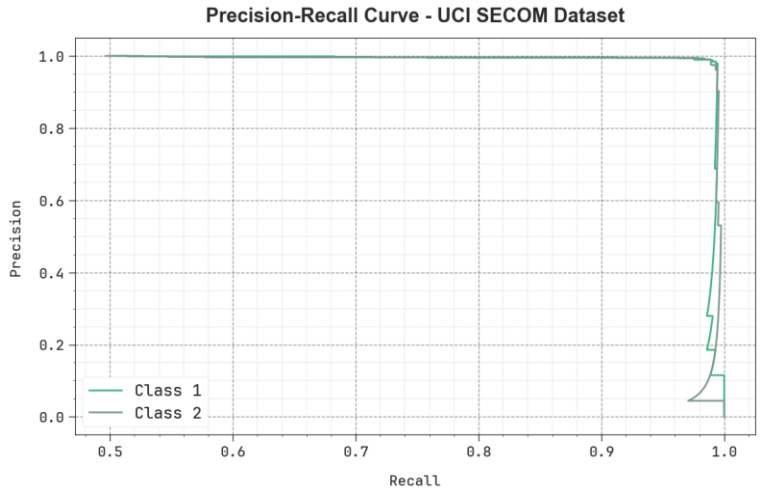

Figure 14. PR curve of the COAFSS-DLCS method at UCI_SECOM dataset

Fig. 15 specifies the ROC graph of the COAFSS-DLCS model under the UCI_SECOM dataset. This figure gives emphasis to the increased classifier outcomes of the COAFSS-DLCS approach across all classes, showing the efficacy in solving diverse classification concerns.

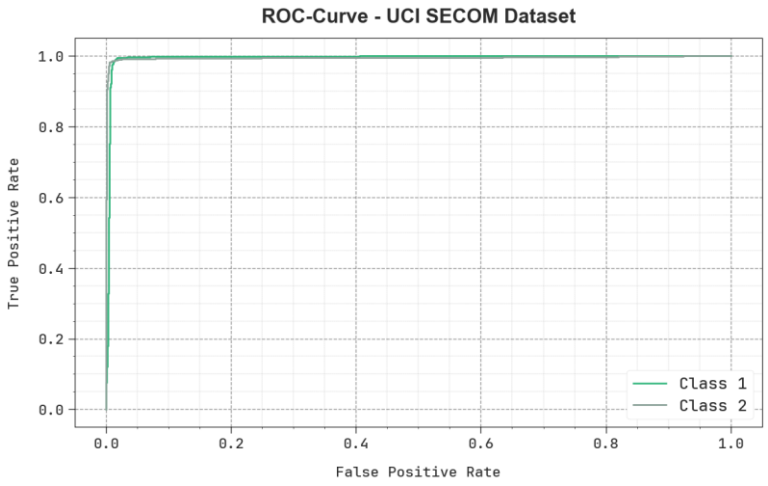

Figure 15. ROC curve of the COAFSS-DLCS method at UCI_SECOM dataset

A detailed comparison assessment of the COAFSS-DLCS method at the UCI_SECOM dataset is defined in Fig. 16. The table values presented that the Ensemble Model and DNN Layer techniques have informed poorer performance. Meanwhile, the PSO Ensemble Model, SSA-CRNN, and MFSDL-ADIIoT models gain moderately improved results. In addition, the GJODL-CADC model has achieved considerable performance. However, the COAFSS-DLCS method provides excellent performance with increased $prec_n$ of 98.93%, $reca_l$ of 98.93%, $accu_y$ of 98.93%, and F_{score} of 98.93%, respectively.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

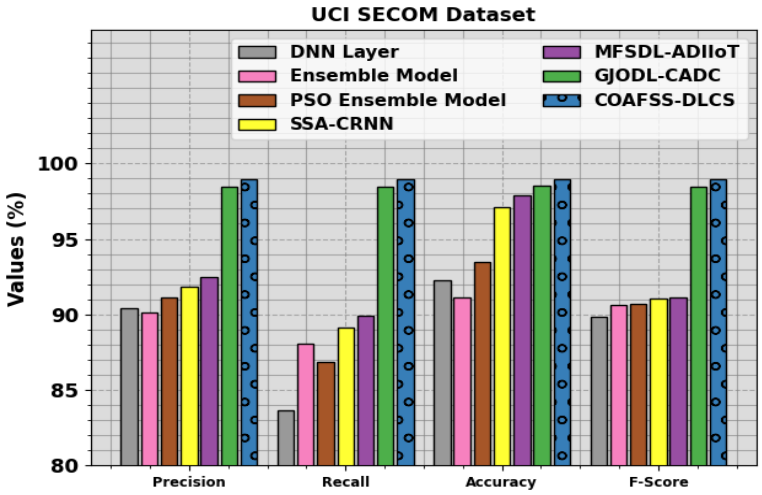

Figure 16. Comparative results of COAFSS-DLCS technique under UCI_SECOM dataset

Therefore, the COAFSS-DLCS approach is implemented for anomaly detection in the IIoT infrastructure.

5. Conclusion

In this article, the COAFSS-DLCS method is proposed. The aim is to classify the cyber threats involved in the IIoT network. It contains four major sub-processes namely data pre-processing, COA-based feature reduction, SLSTM-AE-based attack classification, AOA-based tuning. Initially, the min-max scalar is utilized to transform the input into valuable format. Moreover, the COAFSS approach is utilized for choosing an optimal feature reduction. Meanwhile, the intrusion classification is performed by employing the SLSTM-AE approach. Furthermore, the SLSTM-AE classifier is implemented for tuning process. A comprehensive empirical validation was carried out to validate the high efficiency of the COAFSS-DLCS method. The simulation outcomes inferred the ability of the COAFSS-DLCS technique compared to other models.

Acknowledgement: "The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/492/46"

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "Enhanced IDS with deep learning for IoT-based smart cities security," *Tsinghua Sci. Technol.*, vol. 29, no. 4, pp. 929-947, 2024, doi: 10.26599/TST.2023.9010033.

[2] K. M. Alalayah *et al.*, "Optimal deep learning based intruder identification in industrial Internet of Things environment," *Comput. Syst. Sci. Eng.*, vol. 46, no. 3, 2023.

[3] S. Abbas *et al.*, "Evaluating deep learning variants for cyber-attack detection and multi-class classification in IoT networks," *PeerJ Comput. Sci.*, vol. 10, p. e1793, 2024.

[4] Khacha, R. Saadouni, Y. Harbi, and Z. Aliouat, "Hybrid deep learning-based intrusion detection system for industrial Internet of Things," in *Proc. 5th Int. Symp. Inform. Appl. (ISIA)*, Nov. 2022, pp. 1-6.

[5] H. Gunjal, P. Patel, D. Ebrahimi, and F. Alzhouri, "A smart network intrusion detection system for cyber security of industrial IoT," *Authorea Preprints*, 2023.

[6] Rajak and R. Tripathi, "DL-SkLSTM approach for cyber security threats detection in 5G enabled IIoT," *Int. J. Inf. Technol.*, vol. 16, no. 1, pp. 13-20, 2024, doi: 10.1007/s41870-023-01651-7.

[7] P. Sharma *et al.*, "Deep learning-based intrusion detection system for Internet of Things networks for enhancing security against cyber attacks," in *Proc. Int. Conf. Electr. Electron. Eng.*, Singapore: Springer, Aug. 2023, pp. 685-699.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025

[8] V. Hemamalini *et al.*, "Artificial intelligence-blockchain-enabled-Internet of Things-based cloud applications for next-generation society," in *Automated Secure Computing for Next-Generation Systems*. Wiley, 2024, pp. 65-82.

[9] T. Gueye, Y. Wang, M. Rehman, R. T. Mushtaq, and S. Zahoor, "A novel method to detect cyber-attacks in IoT/IIoT devices on the modbus protocol using deep learning," *Cluster Comput.*, pp. 1-27, 2023.

[10] M. S. Al-Kahtani *et al.*, "Intrusion detection in the Internet of Things using fusion of GRU-LSTM deep learning model," *Intell. Autom. Soft Comput*, vol. 37, no. 2, 2023.

[11] L. A. Maghrabi *et al.*, "Enhancing cybersecurity in the Internet of Things environment using bald eagle search optimization with hybrid deep learning," *IEEE Access*, vol. 12, pp. 12345-12356, 2024.

[12] P. L. S. Jayalaxmi *et al.*, "DeBot: A deep learning-based model for bot detection in industrial Internet-of-Things," *Comput. Electr. Eng.*, vol. 102, p. 108214, 2022.

[13] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial Internet of Things," *Alexandria Eng. J.*, vol. 81, pp. 371-383, 2023.

[14] S. Latif *et al.*, "DTL-IDS: An optimized intrusion detection framework using deep transfer learning and genetic algorithm," *J. Netw. Comput. Appl.*, vol. 221, p. 103784, 2024.

[15] M. A. Rahman, M. S. Hossain, and A. S. M. Z. Rahman, "A comprehensive review of cybersecurity challenges in industrial IoT: Current trends and future directions," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 2342-2356, 2024, doi: 10.1109/JIOT.2023.3245678.

[16] Souri, M. Norouzi, and Y. Alsenani, "A new cloud-based cyber-attack detection architecture for hyper-automation process in industrial Internet of Things," *Cluster Comput.*, pp. 1-17, 2023.

[17] M. Elsisi *et al.*, "Robust indoor positioning of automated guided vehicles in Internet of Things networks with deep convolution neural network considering adversarial attacks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 1456-1468, 2024.

[18] J. Piran, H. E. Barkam, M. Imani, and F. Imani, "Hyperdimensional cognitive computing for lightweight cyberattack detection in industrial Internet of Things," in *Proc. Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, vol. 87356, p. V007T07A013, Aug. 2023.

[19] H. Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of min-max normalization and Z-score normalization in the K-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer," *Int. J. Inform. Inf. Syst.*, vol. 4, no. 1, pp. 13-20, 2021.

[20] M. Aslan and İ. Koç, "Modified Coot bird optimization algorithm for solving community detection problems in social networks," *Neural Comput. Appl.*, pp. 1-25, 2024.

[21] Vijayalakshmi and K. Ramar, "Multivariate congestion prediction using stacked LSTM autoencoder-based bidirectional LSTM model," *KSII Trans. Internet Inf. Syst.*, vol. 17, no. 1, pp. 112-130, 2023.

[22] Subbaiah *et al.*, "Efficient multimodal sentiment analysis in social media using hybrid optimal multi-scale residual attention network," *Artif. Intell. Rev.*, vol. 57, no. 2, p. 34, 2024.

[23] "UNSW-NB15 dataset," Kaggle. [Online]. Available: https://www.kaggle.com/mrwellsdavid/unsw-nb15. Accessed: Aug. 10, 2024.

[24] "UCI SEMCOM dataset," Kaggle. [Online]. Available: https://www.kaggle.com/paresh2047/uci-semcom. Accessed: Aug. 10, 2024.

[25] L. A. Maghrabi *et al.*, "Golden jackal optimization with a deep learning-based cybersecurity solution in industrial Internet of Things systems," *Electronics*, vol. 12, no. 19, p. 4091, 2023.

DOI: https://doi.org/10.54216/JCIM.170205
Received: April 10, 2025 Revised: June 26, 2025 Accepted: August 21, 2025