International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I2P120

Volume 74 Issue 2, 280-292, February 2026

Original Article

Advanced Ensemble Deep Learning Model Integrated with Metaheuristic Optimisation for Secure and Reliable Intrusion Detection in Wireless Sensor Networks

M. Pradeepa[1], R. Ponnusamy[2]

[1,2]*Department of Computer and Information Science, Faculty of Science, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu, India.*

[1]*Corresponding Author : pradeepamca87@gmail.com*

Received: 25 August 2025 Revised: 23 January 2026 Accepted: 27 January 2026 Published: 14 February 2026

Abstract *- With the growth of reliance on the internet and the transfer of most businesses to present remote services, the problems in protecting the network and identifying attacks rapidly become more prominent, as the attack surface and cyberattacks improve in response. The current Wireless Sensor Networks (WSNs) intrusion detection methods that utilize Machine Learning (ML) techniques to detect previously known attacks use single layers of recognition, which means an expensive algorithm must be performed before identifying any suspicious action. Network Intrusion Detection Systems (IDS) present a type of service to the system, and it becomes unavoidable for some communication systems. ML methods are extensively applied in IDS; still, the performance of ML methods is less adequate while processing unbalanced attacks. This paper presents an Advanced Ensemble Deep Learning Model Integrated with Metaheuristic Optimization for Secure and Reliable Intrusion Detection (AEDL-MORID) methodology. The main objective of the AEDL-MORID methodology presents strong potential for real-time deployment in resource-constrained WSN environments, strengthening network resilience against sophisticated cyber threats. The AEDL-MORID method starts with data pre-processing techniques involving the handling of missing values and min-max normalization to ensure clean and consistent input for the learning models. For dimensionality reduction, the dung beetle optimization (DBO) method is utilized to detect the most informative features effectively. In addition, an ensemble classification method integrating Bidirectional Long Short-Term Memory (BiLSTM), Graph Convolutional Network (GCN), and Stacked Denoising Autoencoder (SDAE) is employed for attack detection. To further improve ensemble classification performance, the model parameters are fine-tuned using the Improved Crow Search Algorithm (ICSA) method. The experimentation of the AEDL-MORID model is conducted on the WSN-DS dataset. The experimental validation of the AEDL-MORID system indicated a better accuracy of 99.81% compared to recent techniques.*

Keywords *- Intrusion Detection, Dung Beetle Optimization, Wireless Sensor Network, Attack Detection, Resource Constrained, Deep Learning.*

1. Introduction

The WSN leverages numerous low-cost, wirelessly connected sensor nodes to enable a diverse range of applications. The nodes in WSNs are resource-constrained in terms of storing, communicating, and computing abilities [1]. Like other networks, WSNs are also prone to safety threats due to their wireless and distributed features [2]. The constrained battery power requires low computations to boost the network lifetime. Malicious actors can easily misuse those susceptible networks, gain access, and pose a crucial security threat in WSNs [3]. While WSNs present diverse critical applications, intrusion detection stands out as particularly vital. It allows the monitoring of sensitive areas like borders, remote locations, and infrastructure [4]. Normally, intrusions are malicious activities that breach security protocols to access systems and carry out unauthorized activities. As a reasonable complement to the firewall, IDS detects malicious behaviour and secures the network [5]. It is essential to develop an efficient IDS for the WSN. Several IDSs are proposed, where the data-mining-driven techniques are proven to be highly efficient. The development of advanced devices and network technologies creates extensive data, progressively reducing the IDS's detection rates [6]. Detection of intruders with higher detection precision has become intricate due to the network's constantly evolving nature and resource demands for processing extensive data from distributed environments. Also, IDSs are crucial for user authorization, authentication, and managing suspicious activities [7]. IDSs employ dual primary methods. Rule- and signature-based IDS. While these can precisely identify established attacks, they are ineffective

against zero-day attacks due to the shortage of pre-existing attack patterns. Anomaly-based IDSs identify intrusions by flagging deviations from normal resource utilization or network traffic patterns [8]. Though these systems can identify zero-day as well as known attacks, they are associated with an elevated rate of misclassification. Furthermore, by incorporating Deep Learning (DL), a subfield of ML, WSNs can attain a high level of security, defend sensitive data, and ensure the reliable operation of critical applications in the face of continual and advanced attacks [9]. Figure 1 depicts the general infrastructure of IDS in WSN.

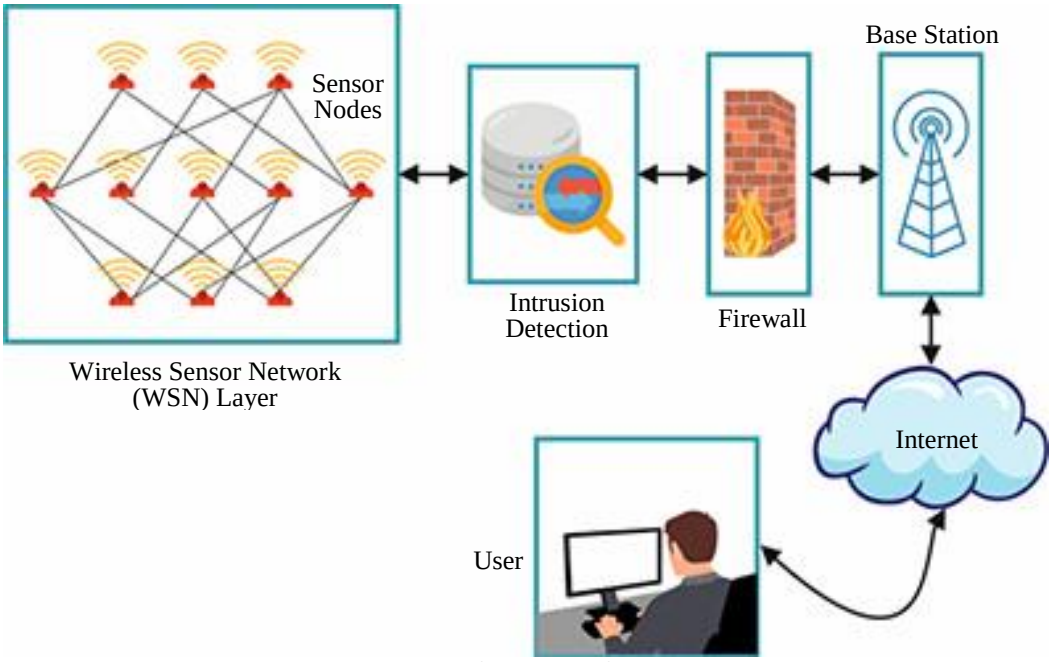

Fig. 1 General infrastructure of IDS in WSN

This paper presents an Advanced Ensemble Deep Learning Model Integrated with Metaheuristic Optimization for Secure and Reliable Intrusion Detection (AEDL-MORID) methodology.

- Initially, missing values are handled via min-max normalization to ensure clean and consistent input.
- The Dung Beetle Optimization (DBO) method is adopted for detecting the most informative features.
- Furthermore, an ensemble of Bidirectional Long Short-Term Memory (BiLSTM), Graph Convolutional Network (GCN), and Stacked Denoising Autoencoder (SDAE) is implemented for classification.
- The Improved Crow Search Algorithm (ICSA) method is used for fine-tuning.
- The novelty is in the integration of BiLSTM, GCN, SDAE, and metaheuristic techniques such as DBO and ICSA. The AEDL-MORID model efficiency under the WSN-DS dataset.

2. Literature Survey on Secure and Reliable Intrusion Detection in WSN

Srivastava and Prakash [10] presented a method for analyzing the addressing scheme of RBMs, a kind of Neural Network (NN). This method focuses on a two-part approach. The goal is attained using the Chaotic Ant Optimizer (CAO) model. The authors created a technique utilizing RBMs to define the optimum confidence level for all sensor nodes. This study presents an improved multimodal method employing DL to solve difficulties in ID and energy optimization using WSNs. Alhusseini et al. [11] proposed a Hybrid IDS (HyIDS) method by integrating Energy Valley Optimizer (EVO) for Feature Selection (FS) with ML classifiers.

Sakthimohan et al. [12] proposed a Secure DL-driven Energy-Efficient Routing (SDLEER) module for WSNs with an IDS for identifying attacks within the network. This module addresses the current application's disadvantages by integrating energy-efficient IDPS in a single network. In [13], a Hybrid Optimized DNN (HODNN) is developed utilizing DNNs to enhance its recognition precision. The source node identifies the shortest path to the destination, subsequently identifying malicious nodes and performing secure routing without them. An improved energy-efficient centralized clustering routing protocol finds the optimal route for routing data. Pande et al. [14] explored the HIDS and NIDS, which are the two primary kinds of IDS. The IDS functions by incessantly monitoring network traffic or specific hosts' activity, examining trends, and detecting suspicious or abnormal behaviour.

Sharma et al. [15] presented a novel ID technique that combines operational and developmental frameworks, concentrating particularly on WSNs. With the escalating number of attacks, defending SNs becomes progressively vital. Along with security violations, unauthorized access to systems by attackers presents a threat to critical resources. The research highlights the need for a unique ID technique and strong feature extraction and classification approaches.

Karthic and Kumar [16] suggested a novel IDS to present protection in statistical communications by detecting intruders on WSNs. Then, a novel FS method named improved conditional random field-based FS for selecting the most contributing features, and an optimized hybrid DNN (OHDNN) is proposed to classify. The HDNN is a hybridization of CNN and LSTM. Also, an adaptive golden eagle optimizer is utilized for parameter optimization. Kumar, Vijayan, and Karthik [17] developed an IDS system by utilizing ML and DL methods with advanced extraction for real-time smart manufacturing.

Shukla, Dwivedi, and Mishra [18] utilized Kernel Principal Component Analysis (KPCA), Lévy flight-driven FS, and an optimized Deep Neural Network with LSTM (DNN LSTM) model. Also, the lévy flight Grasshopper Optimizer Algorithm (GOA) is used for tuning.

Guru et al. [19] presented a transformer CNN BiGRU with an Artificial Bee Optimizer (ABO) model. Mahato and Dutta [20] integrated Grey Wolf Optimizer (GWO) based FS with the Light Gradient Boosting Machine (LightGBM) ensemble learning model.

Though the existing studies are efficient, they lack optimization and adaptation techniques. Real-time applicability of many models can be seen in resource-limited systems, as they require high computational needs.

Various studies show an imbalance and are less novel. Issues such as privacy and scalability are observed in diverse studies. Hence, a research gap exists in developing a lightweight, scalable, and privacy-aware IDS to incorporate FS and performance, via which effective accuracy can be achieved.

3. Methodological Frameworks

This study develops an AEDL-MORID method. The main aim of the AEDL-MORID method presents greater potential for real-time use in resource-constrained WSN, and enhances network resilience against refined cyber threats.

It has four different types of processes involved: pre-processing, feature reduction, ensemble classification, and tuning. Figure 2 illustrates the flow of the AEDL-MORID method.

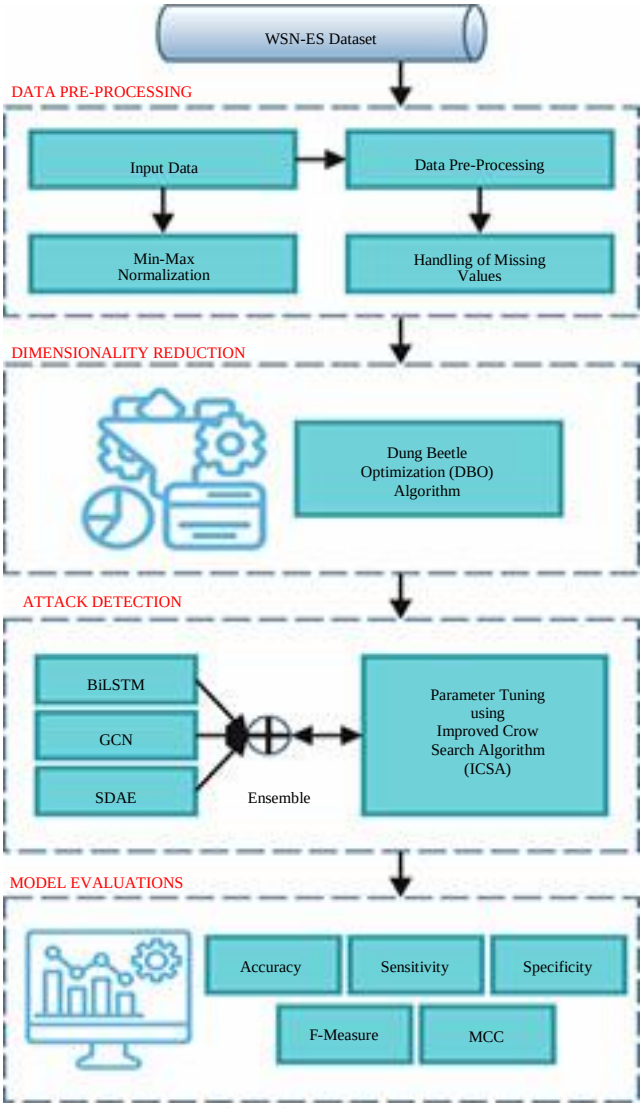

Fig. 2 Entire flow of the AEDL-MORID method

3.1. Data Pre-Processing Methods

To obtain that, the AEDL-MORID method starts with data pre-processing methods, including handling of missing values and min-max normalization to guarantee clean and constant input for the learning methods. The cleaning, converting, and arranging of primary data into an appropriate entity for the computation and modeling process is called data preprocessing [21]. It is an essential phase in the data search procedure that improves the normality and effectiveness of NN methods. There are two methods, such as handling missing values and min-max normalization models, which are utilized to pre-process the achieved data.

3.1.1. Handling Missing Values

It is a standard procedure to eliminate columns or rows that have null values. In the same way, rows that have various columns recognized as null are eliminated. Established the averages or mid-points by utilizing the mode, median, or mean

to fix values in the columns. After being associated with the accounting expert for the data applied to performance, this model is efficient.

3.1.2. Min-Max Normalization

A linear transformation is used on the new data by min-max normalization, scaling accounting features using matrices, and standardizing the data to the range -1. Over the application of min-max normalization, the relation among the new account information values is preserved.

$$X' = \frac{X - X_{\text{Min}}}{X_{\text{Max}} - X_{\text{Min}}} \quad (1)$$

The maximum values of data and low values of data are gained, and each value is substituted into Eq. (1). The variable X represents feature information. X_{Min} and X_{Max} characterize X's minimum and maximum values as integers. X' signifies the upgraded value of every input of data.

Min-max normalization and handling missing values are two data pre-processing methods that give a clean, well-scaled dataset. The wholeness of the data is guaranteed by presenting suitable statistical measures for missing values in the setting.

To increase the presented convergence of the model and precision in analysis and processing, min-max normalization transforms every transaction-related variable into a normalized range (0-1) in the developed performance.

3.2. DBO-based Dimensionality Reduction Process

For FS, the DBO model is used to identify the most informative attributes successfully. The DBO model has the merits of greater processing ability and is suited to manage optimization concerns with fewer than three parameters optimized [22].

3.2.1. Rolling Behaviour

$$y_i(p+1) = y_i(p) + \alpha_l k y_i(p-1) + b_l \Delta y$$

$$\Delta y = |y_i(p) - y^{worst}| \quad (2)$$

Now, $y(p)$ implies the position of i_{th} DB in p_{th} iteration; f depicts the existing iteration number; k signifies the defection co-efficient with value of (0,0.2]; b_l signifies constant with an interval of (0,1), α_1 denotes natural coefficients with a range of 1 or -1, 1 refers to non-deviation, and -1 embodies deviation from unique directions; Δy is employed to represent the modification in intensity of light and y^{worst} represents the global poor location of DB. The dancing behaviour and location of DB are upgraded based on Eq. (3).

$$y_j(p+1) = y_i(p) + \tan(\theta_s)\left|y_i(p) - y_j(p-1)\right| \quad (3)$$

Now, θ_s shows the angle of defection, while $\theta_s = 0, \pi/2,$ or π, the DB's site is not progressed.

$$Low_{b^*} = \text{maxi}(Y^* \times (1-R), Low_b)$$

$$Up\_b^* = \text{mini}(Y^* \times (1+R), Up\_b) \quad (4)$$

Here, Y^* depicts existing local best locations; Low_b^* and Up_b^* refer to lower as well as upper boundaries of egg-laying zone; $R = 1 - \frac{p}{P_{\max}}\ P_{\max}$ denotes higher iteration amounts; Up_b and Low_b, depict upper as well as lower boundaries, correspondingly. The position modification of the egg is defined by the succeeding Eq. (5):

$$B_i(p+1) = Y^* + p_1(B_i(p) - Low\_b^*) + b_2(B_i(p) - Up\_b^*) \quad (5)$$

Now, D_l indicates the dimension, b_1 and b_2 depict dual independent arbitrary vectors of dimension $1 \times D_l$, $and\ B(p)$ implies the location of i_{th} egg ball at p_{th} choice.

According to the searching behaviour of DB, the optimum searching region of DBs and the position of modifications in searching are determined.

3.2.2. Searching Behaviour

$$Low\_b^b = \text{maxi}(Y^b(1-R), Low\_b)$$

$$Up\_b^b = \text{mini}(Y^b(1+R), Up\_b) \quad (6)$$

$$y_i(p+1) = y_i(p) + C_1(y_i(p) - Low\_b^b) + C_2(y_i(p) - Up\_b^b) \quad (7)$$

Now, Up_b^b and Low_b^b refers to the higher and lower limits of the optimum searching region; Y^b depicts the global optimum position; C_1 stands for an arbitrary number succeeding the standard distribution; C_2 specifies an arbitrary vector in (0,1) and $y(p)$ implies i_{th}location of DB in iteration t.

3.2.3. Stealing Behaviour

$$Y_{i(p}+1) = Y^b + eg\left(\left|Y_{i(p)} - Y^*\right| + \left|Y_{i(p)} - Y^b\right|\right) \quad (8)$$

Now, g refers to an arbitrary vector of dimension 1xD that follows the normal distribution, e signifies a constant value, and $y(p)$ implies the place of the i_{th} thief DB at $p_{he\ tth}$ iteration.

In this method, the objectives are incorporated into a single objective equation where a current weight detects the relative significance of each objective.

The fitness function (FF) is accepted as integrating either the motives of FS.

$$Fitness(Y) = \alpha \cdot E(Y) + \beta * \left(1 - \frac{|R|}{|N|}\right) \quad (9)$$

Now, $Fitness(Y)$ exemplifies subset $Y's$ fitness value, $E(Y)$ indicates classifier error rates through applying the picked out attributes within the subset Y, $|R|$ and $|N|$ symbolize chosen feature amounts and the new feature amounts in the dataset consistently, α and β indicate weights of the reduction ratio and the classification error.

3.3. Ensemble Attack Detection Methods

In addition, an ensemble classification model incorporating BiLSTM, GCN, and SDAE is used for attack detection. The ensemble provides higher robustness and accuracy in detecting intrinsic and growing attacks compared to individual models.

3.3.1. Bi-LSTM Model

LSTM is a version of recurrent NN (RNN) broadly employed for single-variable time series data. Particularly intended to overcome the issue of vanishing gradient experienced by conventional RNNs [23].

Forget Gate

Forget gates regulate whether to hold or reject data from cell states. It utilizes the function of a sigmoid to create a value between zero and one that reflects how much past data must be retained.

$$f_t = \sigma\left(V_f[h_{t-1}z_t] + d_f\right) \quad (10)$$

Input Gate

The input gate is responsible for establishing the novel data to be stored in the cell state. A function of the sigmoid is also employed to classify the values that need to be upgraded, whereas tanh creates probable novel data.

$$i_t = \sigma(V_i[h_{t-1}, z_t] + d_i) \quad (11)$$

$$\overline{c_t} = \tanh(V_C[h_{t-1}, z_t] + d_C) \quad (12)$$

Here, c_t matches the novel candidate memory value; i_t depicts the output of the input gate. V_i and V_C refer to the weight that needs to be determined, d_i and d_C refer to biases, and $tanh$ represents the hyperbolic tangent function.

Update the Memory Cell

The value of the memory cell is upgraded under the forgetting, input gates, and the newly chosen candidate memory value.

$$C_t = f_t \times C_{t-1} + i_t \times \overline{C_t} \quad (13)$$

Now, C_t denotes the value of the novel memory cell, whereas C_{t-1} signifies the value of the memory cell from the preceding time step.

Output Gate

It regulates data transmission from the memory cell to the Hidden Layer (HL). It employs either the sigmoid function or the tanh function to generate the HL and the equivalent prediction.

$$O_t = \sigma(V_o[h_{t-1}, z_t] + d_o) \quad (14)$$

$$h_t = O_t \times tanh(C_t) \quad (15)$$

Now, l_t signifies the HL at the present step, O_t depicts the output gate's output, d_O denotes the bias, and W_O refers to the weight. The final predictions can endure additional processing depending on HL h_t, with the nature of this processing reliant on the particular task.

The LSTM effectually models long-term dependency in time-series data; it is specifically appropriate for sequential data. Nevertheless, this method incurs computation costs and requires meticulous tuning of hyperparameters.

A Bi-LSTM network is an extended version of the conventional LSTM framework, substantially improving its capability to understand patterns with sequential data by processing the sequence in either direction, from past to future and from future to past. In standard LSTM, the method depends on preceding time-steps for predictive analytics, which induces missing beneficial data from upcoming values in the sequence. The Bi-LSTM framework tackles this limitation by utilizing dual distinct LSTM layers, one of which processes the sequence of input in forward as well as backward direction. The outputs from either direction are then integrated, presenting a more comprehensive context at every time step. To employ either direction, Bi-LSTM can frequently acquire more intricate temporal patterns and improve predictive performance compared with unidirectional models.

GCN Method

Particularly, GCN signifies a convolutional NN (CNN) intended for graph-structured data [24]. Although traditional CNNs outshine in removing spatial aspects from Euclidean structures, various non-Euclidean frameworks occur in systems.

GCN presents an innovative method for processing this data and has found massive applications and relevant fields. The X and Y denote input and output signals, G indicates the graph, and the GCN processing approach is specified.

$$f(X, A) = Y \quad (16)$$

Now A signifies the graph's adjacency matrix, with components in matrix A depicting the connectivity relations among nodes in graph G.

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{l}W^{l}\right) \quad (17)$$

While $\tilde{A} = A + I$ signifies the adjacency matrix with additional self-connection, H^l and W^l signify the output and parameter layer values l; D implies diagonal degree matrices, and $\sigma(\cdot)$ denotes the activation function.

SDAE Technique

In addition, the SDAE model is utilized for the classification technique [25]. It is selected for its ability to learn and higher-level feature representation from incomplete or noisy input data. In training, SDAE influences the method to remove significant patterns, enhancing generalizability and decreasing overfitting. Its deeper, hierarchical framework allows it to acquire inherent spatial and context relations with scenes more effectually than shallow techniques.

To strengthen the AE and its utilization, add a random noise to produce enhanced data x'. The noise data acts as input for DAE. A novel model capably attains both degraded and original features resulting from noise, considerably increasing the sturdiness of AE and execution. This model effectually tackles the issue of overfitting. Two conventional models occur by adding noise in DAE: At present, the input data x undergoes random 0 based on a specific map function; subsequently, noise is directly integrated with the data. Consequently, the upgrades and transformation of the decoding and encoding maps are required.

$$h = S(Wx' + b_1) \quad (18)$$

$$z = S(W'h + b_2) \quad (19)$$

Here, b_1 and b_2 refers to the bias vector and the activation function, S means the sigmoid function, and W' and W depict the weight matrices.

The loss of DAE is specified:

$$L_{DAE} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{1}{2}\|\hat{x}_i - y_j\|^2\right) \quad (20)$$

Now n depicts the number of input instances, y_j denotes the j_{th} input instance, $\hat{x}_i$ represents the reconstructed output for the i_{th} instance, and $\|\cdot\|$ refers to the norm operation.

Numerous DAE elements were linked to advance the SDAE model to increase the removal of feature ability in particular DAE models.

3.4. Parameter Fine-Tuning Techniques

To further improve the ensemble classification outcomes, the parameters of the models are adjusted utilizing the ICSA [26]. This efficiently alters parameters and improves convergence speed and performance. It also averts local minima and efficiently balances exploration and exploitation, resulting in more accurate and reliable IDS outcomes. Crows are considered the best intellectual bird species. Crow hides food and recovers it if needed. Crows stay closer to each other, monitor and explore, whereas another crow has preserved their meal, and then picks it up once the owner has just gone.

The model includes four core rules, all of which originate from the behavioural models of crows. They have a propensity to collect in larger quantities. They have good memories and can recall accurately where the meal was concealed. They are well-known to stay together to snatch nutrition. They have the talent to observe their surroundings. The CSA is a new kind of swarm intelligence optimizer model that was created by demonstrating the intellectual activities that it performs while looking for and finding nutrition. The reality is that only two aspects need to be modified, which makes it direct and interesting for use in technical fields. The conventional CSA model presents a poor optimizer solution due to its lower solution diversity, poor exploitation and exploration, and poor optimizer outcomes.

The CSA mimics the crow's behaviour by storing extra food and retrieving it if needed. Based on the optimizer concept, they perform the search, the surroundings near it act as the searching region, and storing the place of diet in a completely arbitrary manner is a viable choice.

The CSA sticks to the succeeding principles that originate from the behaviour of crows: (1) They are gregarious species; (2) they can recollect the place of a hidden meal; (3) they will emulate one another and take food for one another; and (4) they try their best to stop others from robbing their meal.

Step 1: Set the parameters and problem statement of the algorithm.

N: Group size

Ft: Flight length

$Iter\_max$: Maximal iteration

AP: Awareness probability

Step 2: Set the crow memory and location.

The group consists of N crows, which are randomly distributed through a d-dimensional searching region, where d represents the overall promising channels. The primary row locations are characterized by Eq. (21):

$$crows = \begin{Bmatrix} z_1^1 & z_2^1 & \cdots & z_d^1 \\ z_1^2 & z_2^2 & \cdots & z_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ z_1^N & z_2^N & \cdots & z_d^N \end{Bmatrix} \quad (21)$$

It is assumed that they have secreted their meal in their initial locations since they are considered to have small involvement at this time. The crow's memories are explained below.

$$crows = \begin{Bmatrix} m_1^1 & m_2^1 & \cdots & m_d^1 \\ m_1^2 & m_2^2 & \cdots & m_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_1^N & m_2^N & \cdots & m_d^N \end{Bmatrix} \quad (22)$$

Step 3: Calculate the fitness of every crow.

By entering the decision variables' values into the goal function for all crows, the excellence of its place was computed. The goal function for channel selection reflects the covariance (CV) and entropy (EN) of the channels, which aids in choosing main channels with high data. Channel selection helps in reducing the computing cost of the detection method. Now, w_1 and w_2 are chosen so that $w_1 + w_2 = 1$.

$$fitness = w_1 * EN + w_2 * CV \quad (23)$$

Step 4: Create a new crow location.

To upgrade, they randomly choose group members, like Crow J, and emulate it to discover the place of the hidden meal. Here, the novel place of the crow is upgraded.

$$z^{i,iter+1} = \begin{cases} z^{i,iter} + r_i \times fl^{iter} \times \left(m^{j,iter} - z^{i,iter}\right) & for\ r_j \geq AP^{j,iter} \\ A\ random\ number & or\ else \end{cases} \quad (24)$$

The conventional CSA upgrades the at random population, resulting in weak solution convergence and diversity. Therefore, the ICSA presents two competitive learning systems to improve the solution's diversity: exploration-exploitation and convergence searching region. The LFEL tactic upgrades the top object utilizing the Lévy step to enhance the exploration area of the model. It deliberates the top 2 solutions (z_{best1} and z_{best2}) using the maximum fitness values, as provided in Eq. (25).

$$z_i^{LFEL} = z_{best} + (2 * r1 - 1) * levy(\beta) * (z_{best1} - z_{best2}) \quad (25)$$

Now, z_i^{LFEL} specifies the upgraded object gained utilizing the LFEL model, β signifies the index of distribution, and $r1$ represents a random index among $(0,1)$.

Still, it utilizes the RWM tactic to improve exploitation of the model. Their location is upgraded utilizing the RWM approach as shown in Eq. (26).

$$z_i^{RWM} = z_{worst} + r2 * (z_{best} - z_{worst}) \quad (26)$$

Now, z_i^{RWM} indicates the upgraded crow utilizing the RWM model, $r2$ means a number generated at random between 0 and 1, and z_{worst} represents solutions with poor fitness.

Step 5: Possibility inspection of novel crow locations.

The novel location in all crows was inspected for the possibility. They change location if a novel place is possible. Or else, they do not move to the new place and remain in their current place.

Step 6: Calculate the fitness value for different places.

Step 7: Upgrade the crow memory utilizing Eq. (27).

$$m^{i,iter+1} = \begin{cases} m^{i,iter+1} & if\ fitness\ of\ m^{i,iter+1} > fitness\ of\ m^{i,iter} \\ m^{i,iter} & otherwise \end{cases} \quad (27)$$

Step 8: Check the end condition. Steps 4-7 were reiterated till $iter\_max$ was attained. Once the end prerequisite is fulfilled, the best memory location concerning the objective function value is presented as an optimizer problem solution. The ICSA model initiates an FF to reach an improved classifier outcome. It establishes an optimistic number to characterize the greater outcome of candidate solutions. The reduction in classifier error rate is determined as the FF, presented in Eq. (28).

$$fitness(z_i) = ClassifierErrorRate(z_i) = \frac{No.of\ misclassified\ samples}{Overall\ sample\ counts} \times 100 \quad (28)$$

4. Experimental Validation

In this part, the AEDL-MORID model is inspected utilizing the WSN-DS dataset from the Kaggle repository [27]. The dataset has 374661 samples with four classes as outlined in Table 1. The overall features are 18, and the chosen attributes are 15. The model runs on Python 3.6.5 with an i5-8600k CPU, 4GB GPU, 16GB RAM, 250GB SSD, and 1TB HDD, with 0.01 learning rates, ReLU, 50 epochs, 0.5 dropout, and batch size 5. The class imbalance and overfitting are handled by using downsampling, data augmentation, and regularization techniques. Figure 3 presents the classifier outputs of the AEDL-MORID system on the test dataset. Figures 3(a)-3(b) display the confusion matrices on a 70%TRAPA and 30%TESPA. Figures 3(c) and 3(d) show the PR and ROC investigation, showing the maximal output for diverse classes.

Table 1. Dataset description

Classes	Samples
"Normal"	"34006"
"Blackhole"	"10049"
"Grayhole"	"14596"
"Flooding"	"3312"
"Scheduling Attacks"	"6638"
Overall Samples	**374661**

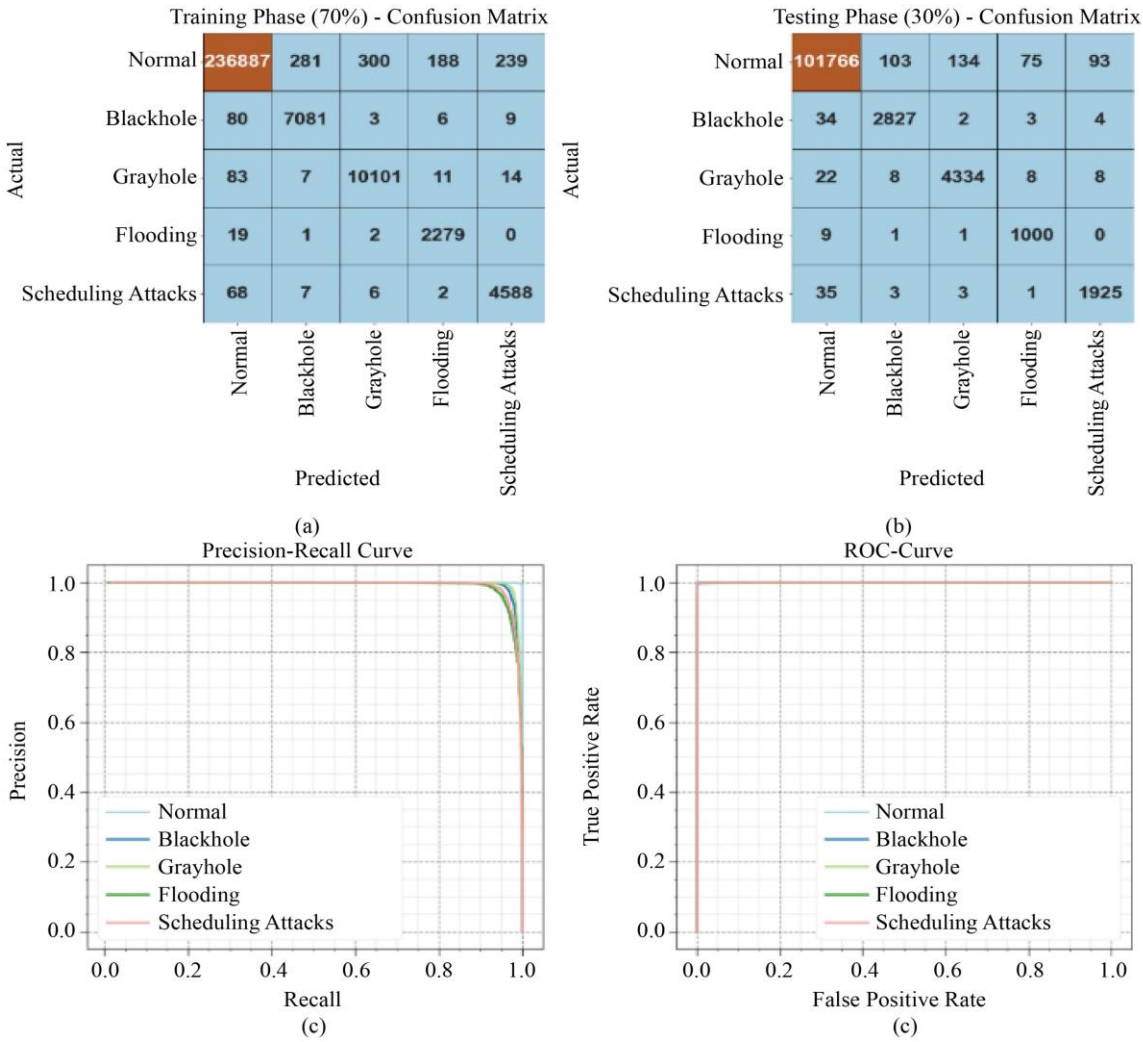

Fig. 3 (a-b) Confusion matrices, and (c-d) curves of PR and ROC.

Table 2. Classification results of the AEDL-MORID approach with TRAPA of 70% and TESPA of 30%

Classes	$Accur_y$	$Sensi_y$	$Speci_y$	$F_{Measure}$	MCC
TRAPA (70%)					
Normal	99.52	99.58	98.97	99.74	97.21
Blackhole	99.85	98.63	99.88	97.29	97.23
Grayhole	99.84	98.87	99.88	97.93	97.86
Flooding	99.91	99.04	99.92	95.22	95.24
Scheduling Attacks	99.87	98.22	99.90	96.38	96.33
Average	99.80	98.87	99.71	97.31	96.77
TESPA (30%)					
Normal	99.55	99.60	99.02	99.75	97.33
Blackhole	99.86	98.50	99.90	97.28	97.22
Grayhole	99.83	98.95	99.87	97.90	97.82
Flooding	99.91	98.91	99.92	95.33	95.35
Scheduling Attacks	99.87	97.86	99.90	96.32	96.27
Average	99.81	98.77	99.72	97.32	96.80

Table 2 and Figure 4 depict the overall classification outputs of the AEDL-MORID method on 70%TRAPA and 30%TESPA. The outputs depict that the AEDL-MORID method precisely recognized the sample attacks. On 70%TRAPA, the AEDL-MORID method presents $accur_y$, $sensi_y$, $speci_y$, $F_{measure}$, and MCC of 99.80%, 98.87%,

99.71%, 97.31%, and 96.77%, respectively. Additionally, on 30%TESPA, the AEDL-MORID model presents $accur_y$, $sensi_y$, $speci_y$, $F_{measure}$, and MCC of 99.81%, 98.77%, 99.72%, 97.32%, and 96.80%, respectively.

Figure 5 describes the training (TRAIN) $accur_y$ and validation (VALID) $accur_y$ of the AEDL-MORID framework. At the primary stage, either TRAIN or VALID $accur_y$ rises quickly, indicating successful learning of designs from the data. The VALID $accur_y$ somewhat outstrips the training $accur_y$, proposing excellent generalization without overfitting. As training develops, imitate the maximum and minimum performance gaps. The consistent overlap among curves indicates effective generalization and regularization, demonstrating the potential of the system in preserving key attributes from noticed and unnoticed data.

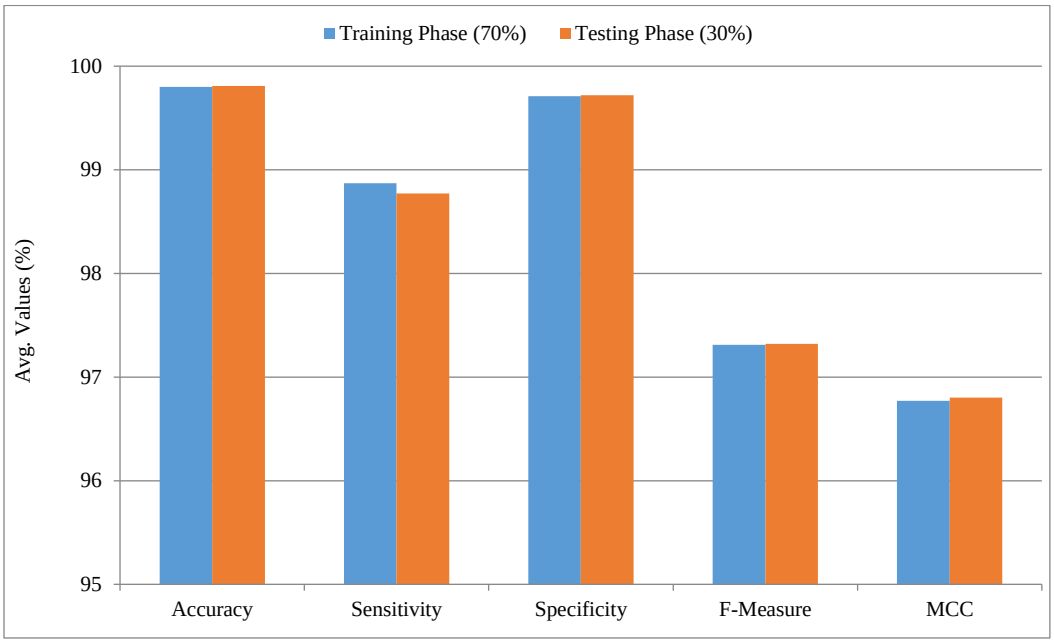

Fig. 4 Average of AEDL-MORID approach on 70%TRAPA and 30%TESPA

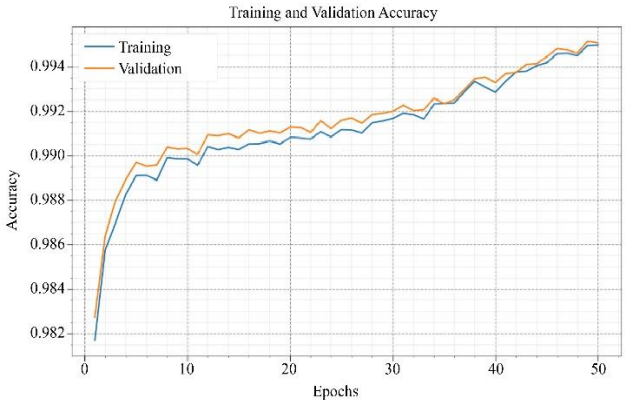

Fig. 5 $Accur_y$ curve of the AEDL-MORID approach

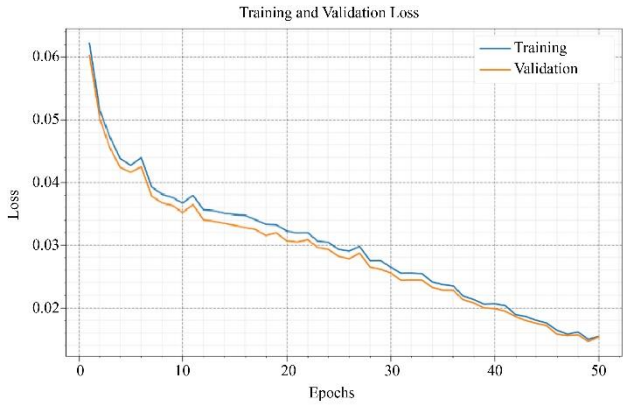

Fig. 6 Loss curve of AEDL-MORID approach

Figure 6 explains the TRAIN and VALID losses of AEDL-MORID. To begin with, either TRAIN or VALID losses are greater, signifying the model initiates with an incomplete data grasp.

As TRAIN progresses, both losses steadily decrease, depicting effective learning. The consistent overlap between both losses suggests good generalization and minimal overfitting.

To exhibit the improved outcome of the AEDL-MORID method, a short comparison analysis is presented in Table 3 and Figure 7 [28]. The outputs represented that the KNN-PSO method attained a lower outcome with $accur_y$ of 93.57%, $sensi_y$ of 96.43%, $speci_y$ of 95.63%, and $F_{measure}$ of 93.75%. Meanwhile, the GOA-GS-IDNN method achieved an $accur_y$ of 93.64%, $sensi_y$ of 96.49%, $speci_y$ of 95.68%, and $F_{measure}$ of 93.81%. Similarly, the GB, LSTM, AdaBoost,

XGBoost, KNN-AOA, DNN, GWO-LSTM, DNN+KAN, RKOA-AETD, Knowledge-Improved DNN, and BCOA-MLID methods outperformed moderate outcomes. Likewise, the FSBMOA-IDWSN approach depicts better performance with $accur_y$ of 99.67%, $sensi_y$ of 96.99%, $speci_y$ of 99.63%, and $F_{measure}$ of 94.75%. However, the AEDL-MORID approach determines a promising outcome with $accur_y$ of 99.81%, $sensi_y$ of 98.77%, $speci_y$ of 99.72%, and $F_{measure}$ of 97.32%.

Table 3. Comparison evaluation of the AEDL-MORID approach with existing models

Methods	$Accur_y$	$Sensi_y$	$Speci_y$	$F_{Measure}$
AEDL-MORID	99.81	98.77	99.72	97.32
FSBMOA-IDWSN	99.67	96.99	99.63	94.75
BCOA-MLID	99.47	96.31	99.22	94.11
RKOA-AETD	98.99	75.41	96.51	79.58
AdaBoost Model	96.30	96.56	95.76	90.90
Gradient Boosting	95.08	95.94	94.89	94.02
XGBoost Method	97.53	96.72	95.05	92.05
KNN-AOA	97.89	96.28	97.13	90.85
KNN-PSO	93.57	96.43	95.63	93.75
LSTM	95.13	96.01	94.95	94.08
DNN	97.58	96.78	95.11	92.12
GWO-LSTM	97.97	96.34	97.18	90.92
GOA-GS-IDNN	93.64	96.49	95.68	93.81
Knowledge-Improved DNN	99.20	99.18	99.15	99.12
DNN+KAN	98.75	98.60	98.70	98.65

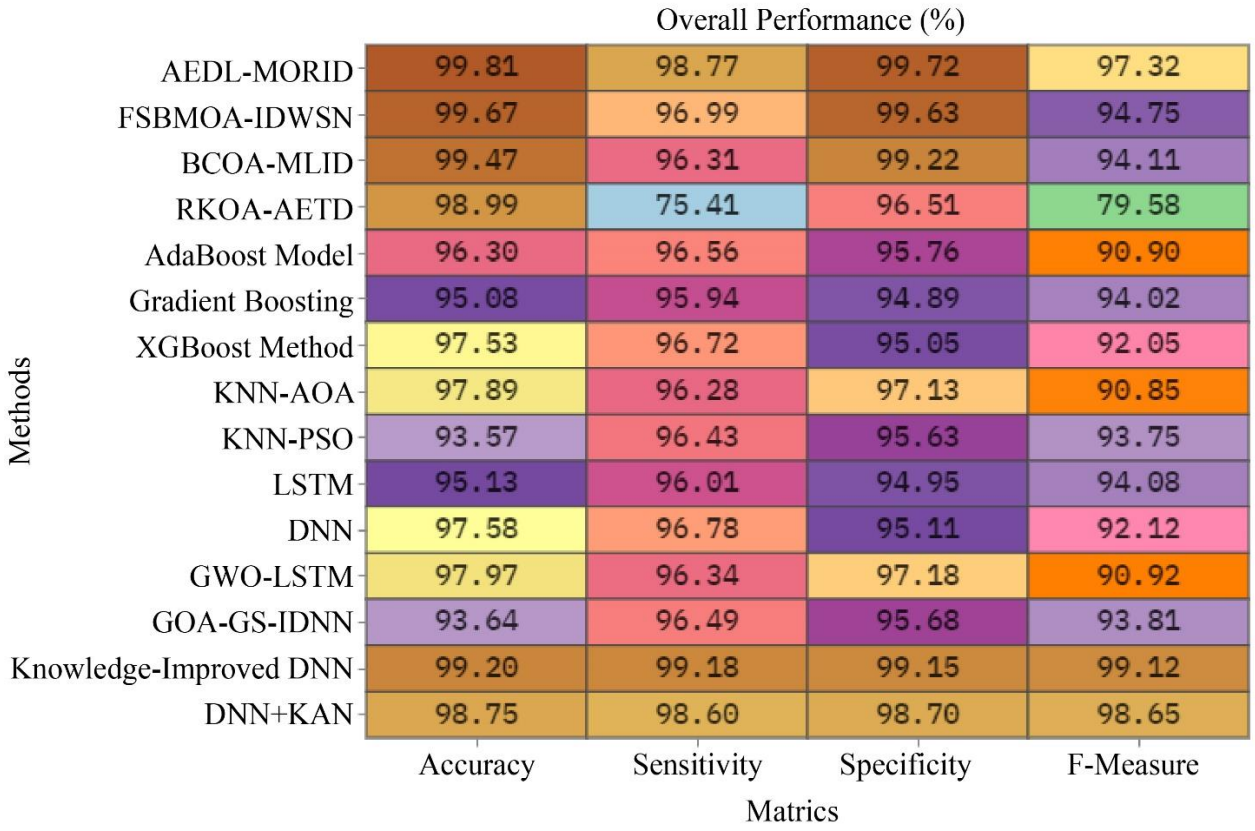

Fig. 7 Comparison evaluation of the AEDL-MORID approach with existing models

The processing time (PT) of the AEDL-MORID approach is compared to recent techniques in Table 4 and Figure 8. Outputs underline that KNN-PSO, AdaBoost, XGBoost, KNN-AOA, and GWO-LSTM methods have gained lower performance with enhanced PT of 29.58min, 25.58min, 24.55min, 21.53min, and 21.07min, correspondingly. In

addition to that, the GOA-GS-IDNN, RKOA-AETD, DNN, Gradient Boosting, and Knowledge-Improved DNN techniques have informed adjacent PT values of 20.00min, 19.78min, 19.00min, 18.75min, and 18.67min, respectively.

In the meantime, the ICFSCN-MHOA, DNN+KAN, BCOA-MLID, and LSTM techniques have managed to inform considerable PT of 9.34min, 16.09min, 16.26min, and 16.99min. However, the AEDL-MORID model displayed better performance with the least PT of 5.87 minutes.

Table 4. PT analysis of AEDL-MORID methodology with existing models

Methods	PT (min)
AEDL-MORID	05.87
ICFSCN-MHOA	09.34
BCOA-MLID	16.26
RKOA-AETD	19.78
AdaBoost Model	25.58
Gradient Boosting	18.75
XGBoost Method	24.55
KNN-AOA	21.53
KNN-PSO	29.58
LSTM	16.99
DNN	19.00
GWO-LSTM	21.07
GOA-GS-IDNN	20.00
Knowledge-Improved DNN	18.67
DNN+KAN	16.09

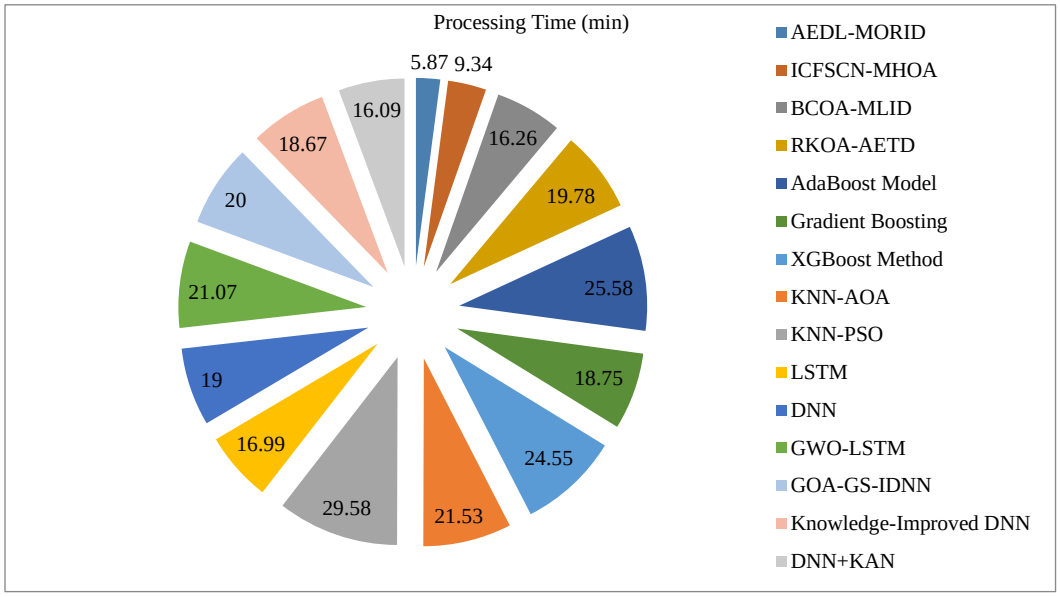

Fig. 8 PT analysis of AEDL-MORID methodology with existing models of 93.57%, $sensi_y$ of 96.43%, $speci_y$ of 95.63%, and $F_{measure}$

Table 5 illustrates the ablation study analysis of the AEDL-MORID model. The AEDL-MORID model illustrated an $accur_y$ of 99.81%, $sensi_y$ of 98.77%, $sensi_y$ of 99.72%, and $F_{measure}$ of 97.32%. By removing GCN and BiLSTM but keeping SDAE with DBO, FS, and ICSA tuning resulted in an $accur_y$ of 99.16%, $sensi_y$ of 97.99%, $speci_y$of 99.02%, and $F_{measure}$ of 96.80%. Using SDAE with DBO, FS without tuning, and without GCN and BiLSTM provided an $accur_y$ of 98.60%, $sensi_y$ of 97.43%, $speci_y$of 98.44%, and $F_{measure}$ of 96.26%. With GCN, DBO, and ICSA tuning and without BiLSTM and SDAE, depicted an accuracy of 97.86%, $sensi_y$of 96.69%, $speci_y$ of 97.77%, and $F_{measure}$ of 95.55%. Also, using GCN with DBO without tuning and without BiLSTM and SDAE provided an $accur_y$ of 97.26%, $sensi_y$ of 95.92%, $speci_y$of 97.13%, and $F_{measure}$of 94.86%.

BiLSTM with DBO and ICSA tuning but without GCN and SDAE depicted an $accur_y$ of 96.65%, $sensi_y$ of 95.38%, $speci_y$ of 96.42%, and $F_{measure}$of 94.23%, and BiLSTM with

DBO without tuning and without GCN and SDAE achieved an $accur_y$ of 96.14%, $sensi_y$ of 94.63%, $speci_y$ of 95.91%, and $F_{measure}$of 93.60%. Table 6 embodies the computational efficiency assessment of the AEDL-MORID model in terms of Floating-Point Operations (FLOPs), Graphics Processing Unit (GPU), and inference time [29]. The DCNN model required 100.130 M GLOPs, 2493 M GPU, and 5.07 ms inference time, while GIDS used 1.590 M GLOPs, 2381 M GPU, and 3.64 ms. NovelADS needed 36.460 M GLOPs, 3126 M GPU, and 2.69 ms, and iForest took 5.470 M GLOPs, 3221 M GPU, and 8.16 ms. AAIDS-STCANN achieved efficiency with 0.300 M GLOPs, 2533 M GPU, and 3.55 ms. The AEDL-MORID approach outperformed all with only 0.084 M GLOPs, 934 M GPU, and the fastest inference time of 1.05 ms, illustrating superior computational efficiency and suitability for real-time applications.

Table 5. Ablation study evaluation of the AEDL-MORID model

Methods	$Accur_y$	$Sensi_y$	$Speci_y$	$F_{Measure}$
AEDL-MORID (Ensemble classifier with DBO FS and ICSA tuning)	99.81	98.77	99.72	97.32
SDAE+DBO+ICSA (With FS and tuning without GCN and BiLSTM)	99.16	97.99	99.02	96.80
SDAE+DBO (With FS without tuning and GCN and BiLSTM)	98.60	97.43	98.44	96.26
GCN+DBO+ICSA (With FS and tuning without BiLSTM and SDAE)	97.86	96.69	97.77	95.55
GCN+DBO (With FS without tuning and BiLSTM and SDAE)	97.26	95.92	97.13	94.86
BiLSTM+DBO+ICSA (With FS and tuning without GCN and SDAE)	96.65	95.38	96.42	94.23
BiLSTM+DBO (With FS without tuning and GCN and SDAE)	96.14	94.63	95.91	93.60

Table 6. Evaluation of the AEDL-MORID model based on FLOPs, GPU, and inference time

Models	GLOPs (M)	GPU (M)	Inference Time (ms)
DCNN	100.130	2493	5.07
GIDS	1.590	2381	3.64
NovelADS	36.460	3126	2.69
iForest	5.470	3221	8.16
AAIDS-STCANN	0.300	2533	3.55
AEDL-MORID	0.084	934	1.05

5. Conclusion

In this article, the AEDL-MORID model has been presented. The aim of the AEDL-MORID system provides a more substantial potential for real-time utilization in resource-constrained WSN atmospheres to support network resilience against refined cyber threats. To obtain that, the AEDL-MORID framework employs min-max normalization for data pre-processing. For FS, the DBO model is used to detect the most informative attributes successfully. In addition, an ensemble classification model incorporating BiLSTM, GCN, and SDAE is used for attack detection. To further improve the ensemble classification outcomes, the parameters of the models are adjusted utilizing the ICSA. The AEDL-MORID model was evaluated on the WSN-DS dataset, attaining an improved accuracy of 99.81% over other approaches. The limitations include reliance on labelled datasets. The model also exhibits restricted adaptability and scalability in extremely large or diverse WSN environments. The research gap is in developing unsupervised or semi-supervised IDS, lightweight real-time frameworks, and adaptive mechanisms for growing attack patterns.

References

[1] V. Gowdhaman, and R. Dhanapal, "An Intrusion Detection System for Wireless Sensor Networks using Deep Neural Network," *Soft Computing*, vol. 26, no. 23, pp. 13059-13067, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Halima Sadia et al., "Intrusion Detection System for Wireless Sensor Networks: A Machine Learning based Approach," *IEEE Access*, vol. 12, pp. 52565-52582, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] K. Sedhuramalingam, and N. Saravanakumar, "A Novel Optimal Deep Learning Approach for Designing Intrusion Detection System in Wireless Sensor Networks," *Egyptian Informatics Journal*, vol. 27, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Fatima Al-Quayed, Zulfiqar Ahmad, and Mamoona Humayun, "A Situation based Predictive Approach for Cybersecurity Intrusion Detection and Prevention using Machine Learning and Deep Learning Algorithms in Wireless Sensor Networks of Industry 4.0," *IEEE Access*, vol. 12, pp. 34800-34819, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Safa Otoum, Burak Kantarci, and Hussein Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68-71, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Abhilash Singh et al., "A Deep Learning Approach to Predict the Number of K-Barriers for Intrusion Detection Over a Circular Region using Wireless Sensor Networks," *Expert Systems with Applications*, vol. 211, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bandar Almaslukh et al., "Deep Learning and Entity Embedding-based Intrusion Detection Model for Wireless Sensor Networks," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 1343-1360, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Liqun Yang et al., "Real-Time Intrusion Detection in Wireless Network: A Deep Learning-Based Intelligent Mechanism," *IEEE Access*, vol. 8, pp. 170128-170139, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Remah Alshinina, and Khaled Elleithy, "A Highly Accurate Deep Learning based Approach for Developing Wireless Sensor Network Middleware," *IEEE Access*, vol. 6, pp. 29885-29898, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jyoti Srivastava, and Jay Prakash, "Deep Learning-Enabled Energy Optimization and Intrusion Detection for Wireless Sensor Networks," *Opsearch*, vol. 62, no. 1, pp. 368-405, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Maryam Mahdi Alhusseini, Alireza Rouhi, and Mohammad-Reza Feizi-Derakhshi, "AI-Powered Hybrid Intrusion Detection Framework for Cloud Security using Novel Metaheuristic Optimization," *arXiv Preprint*, pp. 1-18, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[12] M. Sakthimohan, J. Deny, and G. Elizabeth Rani, "Secure Deep Learning-Based Energy Efficient Routing with Intrusion Detection System for Wireless Sensor Networks," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 46, no. 4, pp. 8587-8603, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ali Siddiq, and Yahya Jaber Ghazwani, "Hybrid Optimized Deep Neural Network-Based Intrusion Node Detection and Modified Energy Efficient Centralized Clustering Routing Protocol for Wireless Sensor Network," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 6303-6313, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Priyanka Pande, Harsh Mathur, and Lalit Kumar Gupta, "Machine Learning-Based Intrusion Detection System using Wireless Sensor Networks," *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Hanjabam Saratchandra Sharma, Arindam Sarkar, and Moirangthem Marjit Singh, "An Efficient Deep Learning-Based Solution for Network Intrusion Detection in Wireless Sensor Network," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 6, pp. 2423-2446, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Karthic, and S. Manoj Kumar, "Hybrid Optimized Deep Neural Network with Enhanced Conditional Random Field based Intrusion Detection on Wireless Sensor Network," *Neural Processing Letters*, vol. 55, no. 1, pp. 459-479, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] M. Nirmal Kumar, T. Vijayan, and B. Karthik, "Intrusion Detection Framework for Industrial Wireless Sensor Networks in Smart Manufacturing," *Pioneering AI and Data Technologies for Next-Gen Security, IoT, and Smart Ecosystems*, pp. 197-218, 2026. [Google Scholar]

[18] Alok Kumar Shukla, Shubhra Dwivedi, and Aishwarya Mishra, "An Effective Hybrid Deep Learning Metaheuristic Model for Robust IoT Intrusion Detection," *Discover Computing*, vol. 28, no. 1, pp. 1-31, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] R. Pavithra Guru, Thomas M. Chen, and Mithileysh Sathiyanarayanan, "ABO Optimized Hybrid Trans-CNN-Bi-GRU Approach for Intrusion Detection in IoT Networks: A Privacy-Preserving Solution," *Cluster Computing*, vol. 29, no. 1, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[20] Sandeep Mahato, and Subrata Dutta, "Ensemble based Meta-Heuristic Optimized Approach for Network Intrusion Detection using LightGBM," *Cluster Computing*, vol. 28, no. 12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] YaHui Lv, "Research on Improving the Efficiency and Accuracy of Accounting Data Processing based on Intelligent Financial Software," *International Journal of High Speed Electronics and Systems*, vol. 35, no. 2, pp. 1-24, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[22] Shuxun Li et al., "Composite Noise Reduction Method for Internal Leakage Acoustic Emission Signal of Safety Valve based on IWTD-IVMD Algorithm," *Sensors*, vol. 25, no. 15, pp. 1-32, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Bouchra Fatima Zohra Diaf et al., "A Novel Bidirectional Lstm-Based Approach for Wind Speed Forecasting: A Case Study of Oran and Adrar, Algeria," *SSRN Electronic Journal*, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Adson Silva, and Ricardo Farias, "AD-VAE: Adversarial Disentangling Variational Autoencoder," *Sensors*, vol. 25, no. 5, pp. 1-14, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Adil O. Khadidos, "Energy-Aware Unmanned Aerial Vehicle-Assisted Mobile Multimedia Communication via Metaheuristic Optimization with Stacked Deep Learning Models," *Alexandria Engineering Journal*, vol. 129, pp. 864-876, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[26] Maithili Shailesh Andhare et al., "A Novel Optimized Hybrid Deep Learning Framework for Mental Stress Detection using Electroencephalography," *Brain Sciences*, vol. 15, no. 8, pp. 1-27, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[27] Bassam Kasasbeh, Kaggle, 2026. [Online]. Available: https://www.kaggle.com/bassamkasasbeh1/datasets

[28] M. SriRaghavendra et al., "Knowledge Improved Hybrid DNN-KAN Framework for Intrusion Detection in Wireless Sensor Networks," *IEEE Access*, vol. 13, pp. 127558-127569, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[29] Donghyeon Kim, Hyungchul Im, and Seongsoo Lee, "Adaptive Autoencoder-based Intrusion Detection System with Single Threshold for CAN Networks," *Sensors*, vol. 25, no. 13, pp. 1-22, 2025. [CrossRef] [Google Scholar] [Publisher Link]