Enhancing Security in Cloned Nodes: An Intelligent Framework for Attack Detection and Mitigation using Deep Learning with Optimization Algorithm in Wireless Sensor Networks

P. Kalvikkarasi[1,*], K. Selvakumar[2]

[1]Research Scholar, Department of Information Technology, FEAT, Annamalai University, Chidambaram, India

[2]Professor, Department of Information Technology, FEAT, Annamalai University, Chidambaram, India

Emails: kalviphd@gmail.com; kskaucse@gmail.com

Abstract

Wireless Sensor Network (WSN) signifies a state-of-the-art technology that combines energy-effective sensors with wireless transmission services enabling prompt surveillance and data collecting from the nearby environments. Owing to the intrinsic features of WSNs, they face numerous challenges of security that range from resource-based attacks, like computational overload or energy depletion, to interception, eavesdropping, and tampering. With the hacked data, the attackers can replicate the same sensors and use clones in the corresponding WSNs. This kind of cloning of the sensors, which is comprised of the WSN, is called a clone attack. Since the replicated sensors formed by the attackers have parallel keys and information, therefore the clone attacks have become a great attack for WSN. To defend WSNs against cyberattacks, machine learning (ML) and deep learning (DL) were applied to classify malicious and normal traffic. This study designs an Attack Detection and Mitigation using Deep Learning with an Optimization Algorithm in Wireless Sensor Networks (ADMDL-OAWSN). The main objective of the ADMDL-OAWSN system is to improve security in cloned nodes for the cyberattack detection model. In the primary step, the data pre-processing employs the StandardScalar method to transform input data into a suitable format. Next, the proposed ADMDL-OAWSN model designs a crayfish optimization algorithm (COA) for the subset of the feature selection (FS) to pick the most related features from an input dataset. For the attack classification process, the convolutional neural network and bi-directional gated recurrent unit with attention mechanism (CNN-BiGRU-A) technique have been exploited. At last, the parameter tuning of the CNN-BiGRU-A is applied by the design of the secretary wolf bird optimization (SeWBO) algorithm. Extensive experiments have been conducted to validate the results of the ADMDL-OAWSN system. The simulation results revealed that the ADMDL-OAWSN system emphasized furtherance when compared to other recent systems

Keywords: Attack Detection; Cloned Nodes; Deep Learning; Optimization Algorithm; Wireless Sensor Networks

1. Introduction

Wireless sensor networks (WSNs) have drawn a fair number of study attention in the past decade. Several parts of such networks are already investigated and these kinds of systems are now well established for various applications starting from habitat monitoring to surveillance [1]. Security is a critical problem in WSN. Without data confidentiality, integrity, and availability numerous real-time WSN applications are in vain. Consequently, multiple surveys are aimed to provide security solutions for these systems [2]. Mitigation and Detection of threats against WSN is an attractive subject among investigators, specifically, deliberating the exclusive challenges of these systems, which are generally imposed by their resource constraints. WSNs are found applications in a broad array of domains comprising smart agriculture, healthcare, environmental monitoring, home and industrial automation etc., owing to their capability of monitoring the data collection, and physical environment and transfer it wirelessly [3]. Fig. 1 determines the infrastructure of WSN.

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

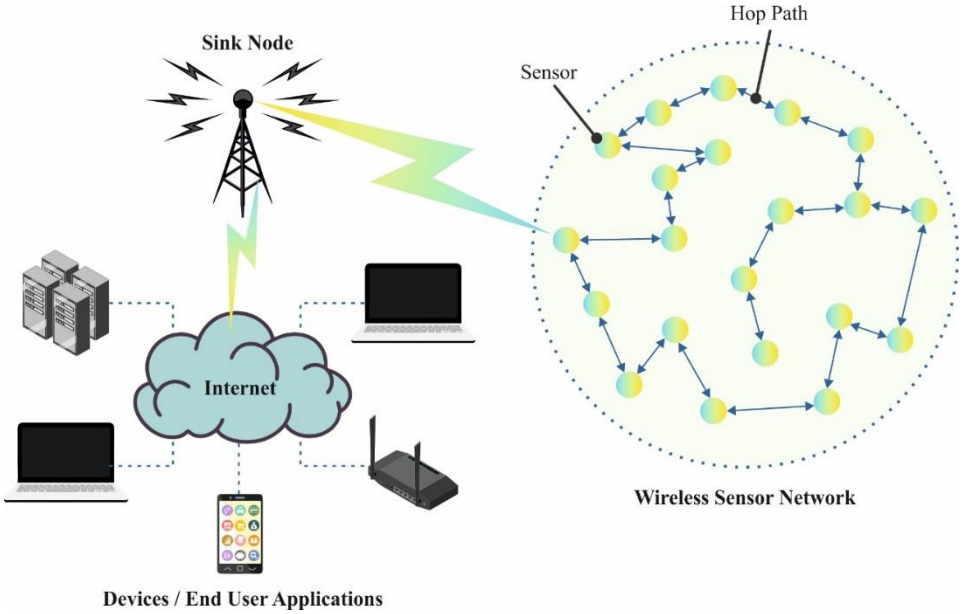

Figure 1. Structure of WSN

WSNs are frequently employed in hostile and harsh settings, which are unavailable and even risky parts to accomplish several monitoring challenges [4]. WSNs have viable solutions for a broad range of real-time challenges; nonetheless, a group of novel security challenges occurs in sensor networks owing to the fact that existing sensor nodes are absent hardware support for tamper-resistance and repeatedly employed in unattended settings where they are susceptible to take and compromise by an opponent's [5]. Adversaries that are capable of launching a set of diverse physical threats comprising node replication threat, denial of service (DoS) attack, radio or signal jamming, eavesdropping, Sybil attack, and node outage can utilize the unattended form of WSNs. Other threats namely wormholes, selective forwarding attacks, and sinkholes. In WSNs, clone recognition is a crucial security challenge where the challenger acquires a legal node, removes its credentials, and employs many clones of that node to control the network [6]. Clone Nodes can be impacted by several kinds of threats namely Blackhole, TDMA, Grayhole, and Flooding in the fact that cloned nodes can serve as entry points for these threats.

WSN systems are vulnerable to many crucial cyber threats owing to their insufficient security ability and restricted source nodes [7]. These cyber-threats have several objectives: altering, hacking, and stealing information the sensors are gathered [8]. WSNs are unique features that render traditional heavyweight security measures, comprising key management, and spread spectrum, inadequate owing to constrained resources, like data storage, packet buffering, and computational power [9]. A promising solution to safeguard WSNs against cyber threats is Deep Learning (DL) and Machine Learning (ML). Appropriately trained DL and ML methodologies will permit us to classify malicious and normal traffic [10]. ML models can examine huge databases and identify anomalous patterns or behavior indicative of threats in the real world. While DL models can remove intricate features and improve the accuracy of threat recognition methods.

This study designs an Attack Detection and Mitigation using Deep Learning with Optimization Algorithm in Wireless Sensor Networks (ADMDL-OAWSN) model. In the primary step, the data pre-processing employs the StandardScalar method to transform input data into a suitable format. Next, the proposed ADMDL-OAWSN model designs a crayfish optimization algorithm (COA) for the subset of the FS process to pick the most related features from an input dataset. For the attack classification process, the CNN and bi-directional GRU with attention mechanism (CNN-BiGRU-A) techniques have been exploited. At last, the parameter tuning of the CNN-BiGRU-A is applied by the design of the secretary wolf bird optimization (SeWBO) algorithm. The simulation results revealed that the ADMDL-OAWSN system emphasized furtherance when compared to other recent systems.

2. Related Works

Deepalakshmi and Kumanan [11] present Fake Clones for Adversaries Detection (FCAD) with effectual relay selection. The FCAD method generates the private and public keys depending on the Elliptic Curve Cryptography model. It determines the receiver and sender fake clone nodes to split up the adversaries effectively. In addition, the process of Sand Cat Swarm Optimizer (SCSO) selects a better relay node depending on energy, delay, and node mobility. Bhuvana et al. [12] projected an advanced transfer learning (TL) method for recognizing the clone threats depending on Neural Fuzzy intensive-sub spectral scaling feature selection (NFI-SSFS) to safeguard by

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

utilizing Co-operative Secure Optimal Link Stability Routing Allocation (CS-OLSR). The communication logs are gathered to consume the different feature levels of packet difference rate under memory and transmission defect fact with the sport of false injection impact rate (FIIR) and Time stamp communication behaviour rate (TSCBR). Then CS-OLSR is implemented to guarantee the secured routing depending on the recognized region of clone threat. Vatambeti et al. [13] project an ML-based clone node detection (ML-CND) model to recognize clone nodes in wireless systems. The objective is to recognize clones effectually enough to preclude cloning threats. Utilization of a lower-cost identity verification process to recognize clones in particular positions together around the world. Employing the node identity models, the most dependable broadcast path can be chosen. This process is meant that utilized for retrieving data from the system node.

In [14], the hybrid model for employing Compression sensing (CS) has been projected to reduce the transmission counts from sensor networks. Nevertheless, in earlier research, a clustering model utilizes hybrid CS for sensor networks, and an explanatory method was employed to inspect the connection between beam size and a large amount of broadcasts of hybrid CS technology. Where ANN models are applied to identify the clone nodes. In [15], a CND protocol named Stacked Ensemble Learning-Clone Attack Detection (SEL-CND) is projected. This recognizes the clone node of Mobile WSN. A cluster head (CH) and a random number of sensor nodes exist in each cluster. The Entropy Dove Swarm Optimizer (EDSO) model is presented for enhancing the execution in the system chooses CH. EDSO models are projected to depend on the foraging behaviors of doves. Subsequently, the SEL-CND mechanism is presented to identify the cloned node that exists in the WSN. The SEL-CND method is executed based on the GRU, Autoencoder (AE), and Bi-LSTM. To recognize clones either locally or globally through the process of economical recognition verification. Numan et al. [16] developed an innovative method named Hybrid Random Walk assisted Zone-based (HRWZ) for clone node recognition in static WSN.

In [17], provides a novel adaptive sea-horse optimized light gradient boosting machine (ASHO-LGBM) model for safeguarding the system against node identity replicas. The ASHO method is employed in the ASHO-LGBM structure to enhance the accuracy of light gradient boosting machine (LGBM) features. This process is aimed to utilize and apply for collecting data over an internet module. Hameed et al. [18] offer an effective methodology for recognizing clone node threats on mobile IoT systems, which utilizes semantical data of IoT gadgets, called contextual data to detect them safely.

3. Materials and Methods

In this study, we design a novel ADMDL-OAWSN system. The main aim of the ADMDL-OAWSN system is to improve security in cloned nodes for the cyber threat detection model. To perform that, the ADMDL-OAWSN has four varieties of phase's data normalization, dimensionality reduction, hybrid classification process, and parameter tuning using the SeWBO model. Fig. 2 exemplifies the entire procedure of the ADMDL-OAWSN methodology.

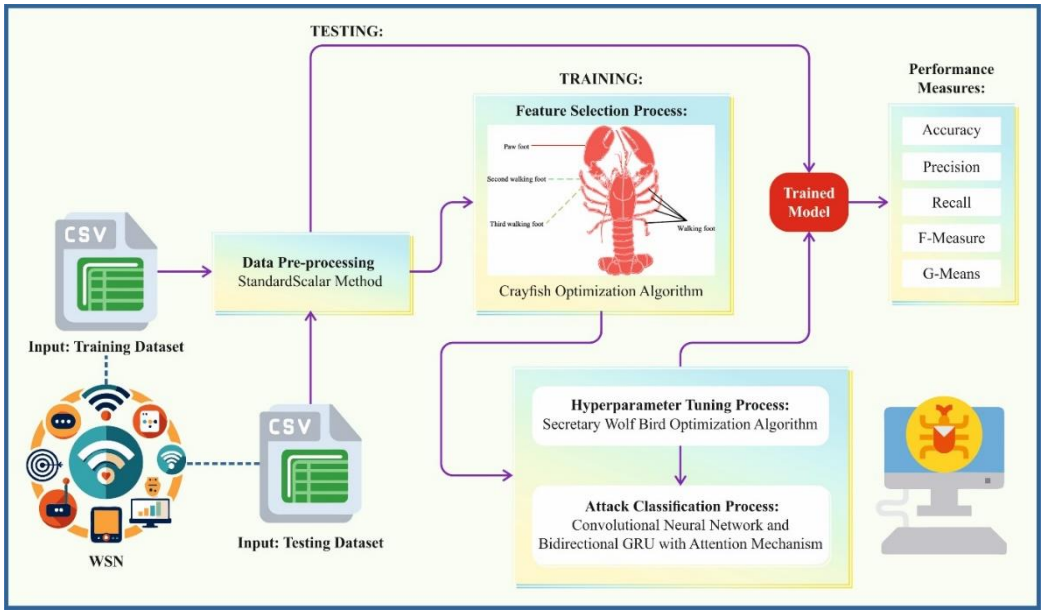

Figure 2. Overall process of ADMDL-OAWSN model

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

A. Stage I: Data Normalization

In the primary step, the data pre-processing employs the StandardScalar method to transform input data into a suitable format. The StandardScalar model rules to unit variance and eliminates the mean, guaranteeing constancy and removing bias, which might damage the solution of the method [19]. It normalizes data domains by scaling their standard deviation (SD) to 1 and mean to 0. By deducting the initial value from the transformed value and separating it via the SD, the transformed value is made utilizing Eq. (1).

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Whereas x characterizes the new value of the feature, z characterizes its changed value, σ embodies the standard deviation, and μ signifies the mean.

B. Stage II: Dimensionality Reduction

Next, the proposed ADMDL-OAWSN model designs COA for the subset of the FS process to pick the most related features from an input dataset. The COA is a bio-inspired optimizer model According to the crayfish's social behaviour [20]. This model imitators the foraging behaviour and social interactions of crayfish for solving optimizer issues. The COA starts with the candidate solution's primary population that is also enhanced iteratively over the sequence of steps stimulated by the behaviour of crayfish. Recently, the COA has achieved attention in the optimizer group owing to its efficiency and possible to solve real-time issues. Generally, the COA provides an innovative method for solving optimizer issues, and its original motivation from crayfish behaviour fixes it aside from conventional optimizer methods. The process of COA is explained in the succeeding subcategories.

Initialization

The COA beginnings with arbitrary initialization to yield possible solutions X by a detailed size of the population N and dimension dim. The position $X_{i,j}$ of individual i in size j^{th} is exhibited in Eq. (2).

$$X_{i,j} = lb_j + \left(ub_j - lb_j\right) \times rand \tag{2}$$

Whereas lb_j and ub_j represent the lower and upper limits of the j^{th} size.

Describing the temperature and number of crayfish

The temperature plays an important part in the dissimilar phases of crayfish, as specified in Eq. (3). Once the temperature surpasses 30, the crayfish searches for cooler places for its summer refuge. In the best temperature range of 15^oC to 30^oC, the crayfish starts its seeking movements. The foraging behaviour is characterized by the standard distribution because of its temperature sensitivity.

$$Temp = rand \times 15 + 20 \tag{3}$$

$$p = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma} \times exp\left(-\frac{(Temp - \mu)^2}{2\sigma^2}\right)\right) \tag{4}$$

Here, $temp$ epitomizes the temperature at which the crayfish is positioned. The μ variable indicates the higher temperature encountered by the crayfish. Moreover, variables such as σ and C_1' achieve the crayfish intake at changing temperatures.

Summer resort phase

When $temp$ surpasses 30^oC, the crayfish returns to its cave X_{shade} for summer break as outlined by Eq. (5).

$$X_{shade} = \frac{X_G + X_L}{2} \tag{5}$$

X_G means optimal location attained thus far, $and\ X_L$ refers to the present location of the population. The fight for cave takes place arbitrarily. If $rand < 0.5$, there is no fight, crayfish straightly capture ownership of the shelter as specified in Eq. (6).

$$X_{ij}^{t,+1} = X_{i,j}^t + C_2 \times rand \times \left(X_{shade} - X_{i,j}^t\right) \tag{6}$$

Now, t denotes the present location, $t + 1$ stands for the following location. The parameter C_2 is the reducing curve calculated by Eq. (7).

$$C_2 = 2 - \left(\frac{t}{T}\right) \tag{7}$$

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

The T value is the maximal iteration count.

Competition phase

Another crayfish is interested in a similar shelter if $temp > 30$ and $rand \geq 0.5$ and thus, they participate in contests to claim possession of it, as established by Eq. (8).

$$X_{ij}^{t,+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{shade} \tag{8}$$

Whereas, z refers to random crayfish and is gained by Eq. (9).

$$z = round\big(rand \times (N-1)\big) + 1 \tag{9}$$

Foraging phase

The crayfish starts searching for food after $temp$ decline below 30. Upon finding the nutrition, the crayfish measures its dimensions utilizing Eq. (10) and (11) to define either the place or size of the food item.

$$X_{food} = X_G \tag{10}$$

$$Q = C_3 \times \left(\frac{fitness_i}{fitness_{food}}\right) 1 \tag{11}$$

The crayfish splits its food into small pieces by utilizing its claws that mimic the procedure of searching for the most effective solution. The fitness function employed in the COA approach is intended to have a balance among the amount of chosen features in every solution (least) and the classification accuracy (highest) attained by employing these selected features, Eq. (12) signifies the FF for evaluating the solutions.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \tag{12}$$

Here, $\gamma_R(D)$ signifies the classifier rate of error of a given classifier. $|R|$ means a cardinality of the chosen subset and $|C|$ denotes the complete amount of features in the data, α, and β signifies dual parameters, which corresponds to the importance of classifier quality and sub-set length.

C. Stage III: Hybrid Classification Process

For the attack classification process, the CNN-BiGRU-A technique has been exploited. As an extensively applied network approach in the domain of DL, CNN is a feed-forward neural network with a depth configuration [21]. Traditional CNN normally contains fully connected (FC), input, pooling, convolution, and output layers. The input layer is responsible for receiving and inputting the new data into the CNN. In general, to enhance the computation time, the data should be pre-processed. It utilizes a convolution kernel to implement convolution processes on the data in a sliding window way until each data is handled. Moreover, the convolutional layer can efficiently decrease the feature mapping size and reduce the dimensions by fixing suitable padding and strides. The formulation for the convolution process is stated as shown:

$$y^n = \sum_{i=1}^{b^{n-1}} x_i^{n-1} \times w_i^n + a_i^n \tag{13}$$

Here, y^n characterizes the local region output value of layer n, b^{n-1} characterizes the bth channel in layer $n-1$, x_i^{n-1} refers to the output value of the ith channel of $n-1$ layer, $*$indicates 1D convolution processes, w_i^n symbolizes the weighted coefficient of $i\ th$ channel of n-layer, and a_i^n embodies the bias of ith channel of n-layer.

The convolution layer, as the most important module of the CNN, convolves the input data to remove local features. It chooses a convolution kernel to carry out the convolution process on the data over the sliding window model till each data is managed. In establishing the suitable padding and stride length, the feature mapping size was successfully decreased, and the objective of dimensionality reduction was attained.

Maximum pooling maintains significant local features by choosing the maximum value in the pooled area, whereas average pooling maintains the complete features data by computing the average value of the pooled area. Consequently, maximum pooling highlights significant features; however, average pooling retains a smoothness and a global view of the data.

The major function of the FC layer is to reconnect the features gained after pooling and convolution processes, thus decreasing the feature sizes. The output layer is situated after the FC layer and contains the responsibility for making the last classification forecasts and handling input values over a stated activation function to give the

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

essential outcomes. CNN contains great feature extraction capability. By utilizing its local view and weight-sharing features, it can successfully decrease computational time, learning parameters, and redundant information, finally streamlining the model's complexity.

The GRU is an enhanced form of the RNN, which targets to resolve the problem of vanishing gradient met by conventional RNNs after handling longer sequences. By presenting a gating mechanism, GRU enhances the data flow, allowing it to keep important data and remove redundant details more efficiently in comparison with normal RNN, GRU is normally at ease for training, mainly with longer sequence data. Nevertheless, GRU contains specific limitations. Its present output only relies on the preceding hidden layer (HL) and the present input, accomplishing it inspires to take upcoming information inside the sequence. To overwhelm this, the BIGRU combines a backward GRU architecture. In comparison with other variations of RNN, which integrates gating mechanisms and memory cells, recognized as LSTM, however, LSTM can successfully seize either short- or long-term dependences; BI-GRU improves model predictive ability by widely taking feature associations in time-series over concurrent treating of forward and reverse data.

The output layer H of BI-GRU is established by 3 sections: forward and reverse hidden information and input X. C represents the hidden information at time t, S signifies the neuron's activation value at time t, and $\leftarrow$ and $\rightarrow$ refers to backward propagation and forward propagation, correspondingly.

$$\begin{cases} \vec{C}^t = GRU\left(x_t, \vec{C}^{t-1}\right) \\ \overleftarrow{C}^t = GRU\left(x_t, \overleftarrow{C}^{t-1}\right) \\ h^t = \vec{w}^t \vec{c}^t + \overleftarrow{w}^t \overleftarrow{c}^t + b^t \end{cases} \tag{14}$$

Whereas $\vec{w}$ and $\overleftarrow{w}$ symbolize output weighting of the forward and reverse GRU, b^t characterizes the offset consistent with the HL.

The Attention Mechanism (AM) is a technology, which allocates dissimilar weights to numerous portions based on their significance after handling input data. By computing the relation between components inside the input sequence, the method can order more important data, thus increasing its performance. In detail, the AM enhances the sequence generation procedure by computing the attention weight and combining the HL of the encoder with the HL of the decoder. The BI-GRU method improves its localization prediction capability by concurrently treating forward and reverse information, allowing it to take tendencies that are more complete. Moreover, the AM additionally increases the model performance by particularly highlighting the most related portions of the data. At every time step, the AM allocates high weights to important features, which might specify important variations in the point of impact, guaranteeing that the model concentrates on the most significant data. The particular equation is as shown:

$$e_t = u \times \tanh(w \times h_t + b) \tag{15}$$

$$\alpha_t = \frac{\exp(e_t)}{\Sigma_{j=1}^{t} e_j} \tag{16}$$

$$s_t = \sum_{t=1}^{i} \alpha_t \times h_t \tag{17}$$

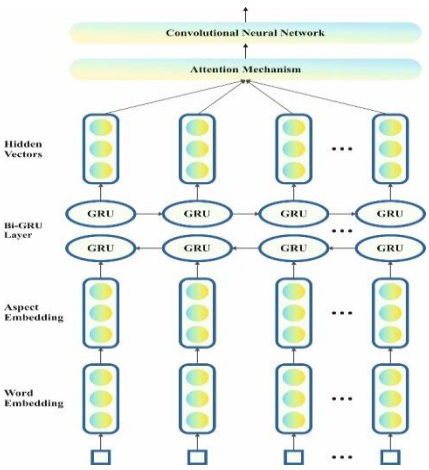

Figure 3. CNN-BiGRU-A Structure

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

Whereas e_t characterizes the attention probability distribution value established by BiGRU network layer output vector h_t at instant t. u and w and b represent weight and bias coefficient, α_t refers to the attention probability distribution of AM output to the BiGRU hidden layer, and s_t stands for attention layer output at time t. Fig. 3 portrays the structure of the CNN-BiGRU-A technique.

D. Stage IV: Parameter Tuning using the SeWBO Model

At last, the SeWBO algorithm applies the parameter tuning of the CNN-BiGRU-A. The proposed SeWBO model has been progressed by integrating the Secretary Bird Optimization Algorithm (SBOA) and the Wolf Bird Optimization (WBO) [22]. SBOA is nothing but a population-based meta-heuristic technique, which is made because of survival natures of the secretary birds. The hunting behavior creates the base of an exploration phase and the escaping tactic creates the idea behind the exploitation stage. Furthermore, SBOA improves efficacy and reduces the cost of computation. WBO is expressed by considering the relationship between the wolves and ravens. The addition of WBO in SBOA improves the flexibility of the SeWBO system and helps in evading premature convergence. The stages followed by SeWBO are explained below:

i) Initialization

The initial procedure in the execution of SeWBO is the random initialization of the location of a secretary bird in a searching space and is stated as,

$$C_{m,n} = lb_n + a \times (ub_n - lb_n), m = 1{,}2\ D, n = 1{,}2, Dim \tag{18}$$

Here, c_m defines the location of mth secretary bird, lb_n is the lower bound, ub_n means an upper bound, $a$and a refers to a randomly generated number between [0,1]. Besides, the optimization starts with a candidate solution C, which is arbitrarily formed within lower- and upper-bound constraints for a specific issue. Besides, the candidate solution C is mathematically stated below,

$$c = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} & \cdots & c_{1,Dim} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} & \cdots & c_{2,Dim} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} & \cdots & c_{m,Dim} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{D,1} & c_{D,2} & \cdots & c_{D,m} & \cdots & c_{D,Dim} \end{bmatrix}_{D \times Dim} \tag{19}$$

Here, the quantity of group members is represented as D, C signifies secretary bird group, c_m states mth secretary bird, $c_{m,n}$ means mth secretary bird in the nth dimension, and variable dimension is specified as Dim.

iii) Exploration Stage:

When serving on snakes, the hunting behavior has been separated into 3 stages searching, hunting, and consumption of prey. Based upon the organic statistics of hunting stages the complete hunting procedure is categorized into 3-time durations at every phase, with $< \frac{1}{3}I$, $\frac{1}{3}I < p < \frac{2}{3}I$, and $\frac{2}{3}I < p < I$, depending upon the above-mentioned stages with p. While, p signifies the current number of iterations, and I is the complete iteration count.

a) Searching for Prey:

The secretary bird's process is introduced with searching for its probable prey, mainly snakes. Differential evolution is employed, which uses the changes among the individuals to enhance the algorithm range by making global hunt abilities and novel solutions. Therefore, a differential mutation procedure is applied to avoid local goals. When searching for prey, the secretary bird's upgrading position is mathematically formulated below:

$$when\ p < \frac{1}{3}I, c_{m.n}^{newJ_1} = c_{m,n} + \left(c_{rand\_1} - c_{rand\_2}\right) \times L_1 \tag{20}$$

While, c_m^{newJ1} is a novel state at the initial stage and $c_{m,n}^{newJ1}$ refers to its value of nth dimension, L_1 states an arbitrarily formed array of dimension $1 \times Dim$ within the range of [0,· 1], and Dim represents a dimension of solution space.

b) Consumption of Prey:

A secretary bird is involved in a tactic for hunting after perceiving the snake. Thick keratin scales surround the bird's leg surface, which makes it difficult for the snake to set up in its body. At this stage, c_{best} concept and Brownian motion E^* have been utilized, and it is stated,

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

$$E^* = random(1, Dim) \quad (21)$$

While $random\ (1, Dim)$ signifies a randomly formed array with dimension $1 \times Dim$. When consuming the prey, the upgraded location of the secretary bird is demonstrated by,

$$while\ \frac{1}{3}I < p < \frac{2}{3}I, c_{m,n}^{newJ1} = c_{best} + exp\left[\left(\frac{p}{1}\right) \wedge 4\right] \times (E^* - 0.5) \times \left(c_{best} - c_{m,n}\right) \quad (22)$$

In Eq. (10), c_{best} refers to the current value of best.

c) Hunting of Prey:

Once a snake is tired, a secretary bird observes a chance to travel utilizing their high-power leg muscles for attack. Additionally, a leg-kicking tactic is initiated by a secretary bird, where it grows its leg and allows precise kicks with their sharp talons and objectives the snake's head. Therefore, a levy flight tactic is intended to improve the global search ability of an optimizer, which is achieved by improving the rate of convergence. Furthermore, a non-linear perturbation factor is progressed for creating the model more adaptive, active, and flexible by evading convergence of premature. A non-linear perturbation factor is selected as $\left(1 - \frac{p}{I}\right)\left(2 \times \frac{p}{1}\right)$. Besides, the secretary bird is upgrading location while hunting the prey is shown below:

$$while\ p > \frac{2}{3}I, c_{m,n}^{newJ1} = c_{best} + \left[\left(1 - \frac{p}{I}\right) \wedge \left(2 \times \frac{p}{I}\right)\right] \times c_{m,n} \times M \quad (23)$$

While weighted levy flight is denoted as M.

iv) Exploitation Stage:

The secretary bird's natural enemies are huge hunters such as jackals, eagles, foxes, and hawks, which take food or kill them. There are mostly dual types of evasion tactics to protect themselves. The first tactic is to run or fly away, at a higher speed in order to escape from danger. Besides, Camouflage is measured as the next strategy, in which the secretary bird utilizes the colors or structures in their surroundings to make it challenging for the hunters to recognize them. These dual conditions happen with equivalent likelihood such as N_1: Camouflage by environment and N_2: Run or fly away. A factor of dynamic perturbation is presented and denoted as $\left(1 - \frac{p}{I}\right)^2$. It helps the model to balance exploration and exploitation. Lastly, the secretary bird's location has been upgraded,

$$c_{m,n}^{newJ1} = \begin{cases} N_1 : c_{best} + (2 \times E^* - 1) \times c_{m,n}, & if\ q < q_j \\ N_2 : c_{m,n} + L_2 \times \left(c_{random} O \times c_{m,n}\right), & else \end{cases} \quad (24)$$

While, $q_j = 0.5$, E^* signifies Brownian motion, q denotes the randomly generated number, $and\ L_2$ implies the random range of normal distribution with dimension $(1 \times Dim)$. c_{random} represents a random candidate solution, and O specifies a random range between the range of [1, 2]. If a condition N_2 met the equation, the above-mentioned formulation becomes,

$$c_{m,n}(p+1) = c_{m,n}(p) + L_2 * \left[c_{random} - O^* c_{m,n}(p)\right] \quad (25)$$

$$c_{m,n}(p+1) = c_{m,n}(p) + L_2 * c_{random} - L_2 O^* c_{m,n}(p) \quad (26)$$

$$c_{m,n}(p+1) = c_{m,n}(p)[1 - L_2 O] + L_2^* c_{random} \quad (27)$$

The WBO is a nature-inspired optimizer technique, which is mainly employed for classifying the behavior of birds while hunting for food. Furthermore, the enclosure of WBO in SBOA improves the flexibility of the SeWBO system and helps in evading the convergence of premature. From WBO,

$$c_{m,n}(p+1) = c_{m,n}(p) + q_1 * c_{m,n}^J(p) - q_2^* pcp \quad (28)$$

$$c_{m,n}(p) = c_{m,n}(p+1) - q_1^* c_{m,n}^J(p) - q_2^* pcp \quad (29)$$

Replacing Eq. (29) in Eq. (25),

$$c_{m,n}(p+1) = \left[c_{m,n}(p+1) - q_1^* c_{m,n}^J(p) - q_2^* pcp\right][1 - L_2 O] + L_2^* c_{random} \quad (30)$$

$$c_{m,n}(p+1) - c_{m,n}(p+1)[1 - L_2 O] = L_2^* c_{random} - \left(q_1 \times c_{m,n}^J(p) + q_2^* pcp\right)[1 - L_2 O] \quad (31)$$

$$c_{m,n}(p+1)[1 - 1 + L_2 O] = L_2^* c_{random} - (q_1 \times c_{m,n}^J(p) + q_2^* pcp)[1 - L_2 O] \quad (32)$$

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

$$c_{m,n}(p+1) = \frac{1}{L_2 O}\left[L_2^* c_{random} - \left(q_1^* c_{m,n}^J(p) + q_2^* pcp\right)(1 - L_2 O)\right] \quad (33)$$

Eq. (33) means the upgraded formulation of the SeWBO model. Moreover, in the above-mentioned calculation, q_1 and q_2 describe dual randomly generated numbers within the range of [0,1], $c_{m,n}^J(p)$ specifies the starving prey, and pcp means a center point of prey.

In addition, the randomly formed selection is shown as,

$$O = round(1 + random\ (1,1)) \quad (34)$$

While $random(1,1)$ signifies a generated value at random among $[0,1]$.

v) Evaluation of Feasibility:

The members of the SeWBO position are upgraded utilizing the above-mentioned stages in each iteration. When the upgraded position is better than other sites, then the modernized location is taken as the novel updated location.

vi) Termination:

The above-mentioned stages are completed constantly till the best outcomes are attained. The novel position of SeWBO members provides the finest solution for a stated problem.

The SeWBO system originates an FF for attaining an enhanced performance of the classifier. It defines a positive numeral to signify the better solution of the candidate outcomes. Here, the classification rate of error minimization is measured as FF, as set in Eq. (35).

$$fitness(x_i) = ClassifierErrorRate(x_i)$$
$$= \frac{no.of\ misclassified\ instances}{Total\ no.of\ instances} * 100 \quad (35)$$

4. Performance Analysis

The experimental study of the ADMDL-OAWSN system is studied under the WSN-DS dataset [23]. The dataset contains 374661 instances under 5 attack types such as normal, Grayhole, Blackhole, TDM (Time Division Multiple Access), and Flooding as exposed in Table 1. There are 18 no. of features available but only 12 features were chosen.

Table 1: Details of the dataset

Attack Types	No. of Instances
Normal	340066
Grayhole	14596
Blackhole	10049
TDM	6638
Flooding	3312
Total Instances	374661

Fig. 4 represents the classifier outcomes of the ADMDL-OAWSN system below 80% TRAPHA and 20% TESPHA. Figs. 4a-4b reveals the confusion matrices with correct recognition of each class. Fig. 4c shows the PR values, demonstrating superior performance over every class. Followed by, Fig. 4d determines the values of ROC, establishing outcomes with better analysis of ROC for different classes.

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

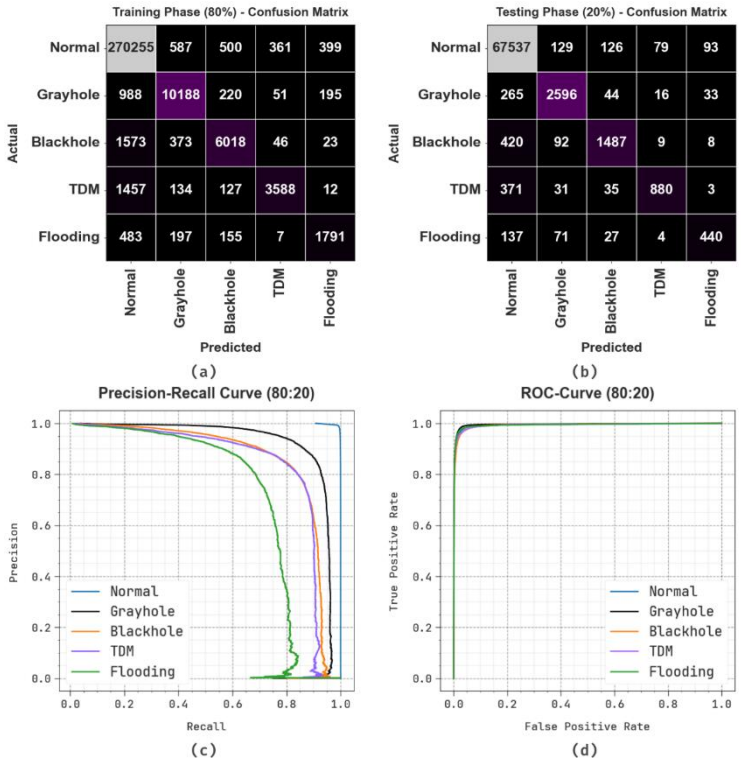

Figure 4. 80%TRAPHA and 20%TESPHA of (a-b) of confusion matrix, (c) PR curve, and (d) ROC curve

Table 2 and Fig. 5 show the attack detection of the ADMDL-OAWSN approach under 80%TRAPHA and 20%TESPHA. The outcome states that the ADMDL-OAWSN model accurately acknowledged the samples. With 80%TRAPHA, the ADMDL-OAWSN technique offers average $accu_y$, $prec_n$, $reca_l$, $F_{Measure}$, and G_{Means} of 98.95%, 87.08%, 79.45%, 82.88%, and 83.07%, correspondingly. Moreover, with 20%TSAPHA, the ADMDL-OAWSN methodology provides average $accu_y$, $prec_n$, $reca_l$, $F_{Measure}$, and G_{Means} of 98.94%, 87.81%, 78.50%, 82.63%, and 82.89%, respectively.

Table 2: Attack recognition of ADMDL-OAWSN system under 80% TRAPHA and 20%TESPHA

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F_{measure}$	G_{Means}
TRAPHA (80%)					
Normal	97.88	98.36	99.32	98.84	98.84
Grayhole	99.08	88.75	87.51	88.13	88.13
Blackhole	98.99	85.73	74.92	79.96	80.14
TDM	99.27	88.53	67.47	76.58	77.28
Flooding	99.51	74.01	68.02	70.89	70.95
Average	98.95	87.08	79.45	82.88	83.07
TESPHA (20%)					
Normal	97.84	98.26	99.37	98.81	98.82
Grayhole	99.09	88.93	87.88	88.40	88.41
Blackhole	98.98	86.50	73.76	79.63	79.88
TDM	99.27	89.07	66.67	76.26	77.06
Flooding	99.50	76.26	64.80	70.06	70.30
Average	98.94	87.81	78.50	82.63	82.89

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

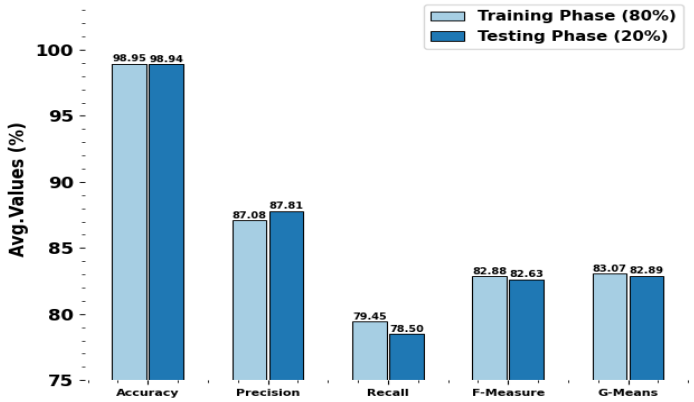

Figure 5. Average of ADMDL-OAWSN technique below 80% TRAPHA and 20%TESPHA

In Fig. 6, the training (TRAN) $accu_y$ and validation (VALN) $accu_y$ of the ADMDL-OAWSN system below 80:20 is demonstrated. The $accu_y$ values are calculated within the ranges of 0-25 epochs. The values of TRAN and VALN $accu_y$ display a growing trend of the ADMDL-OAWSN technique with maximum outcomes through several iterations. Furthermore, the TRAN and VALN $accu_y$remnants closer over the epochs, which indicates the worst overfitting and shows the maximum performance of the ADMDL-OAWSN system, which guarantees reliable forecasts on unseen samples.

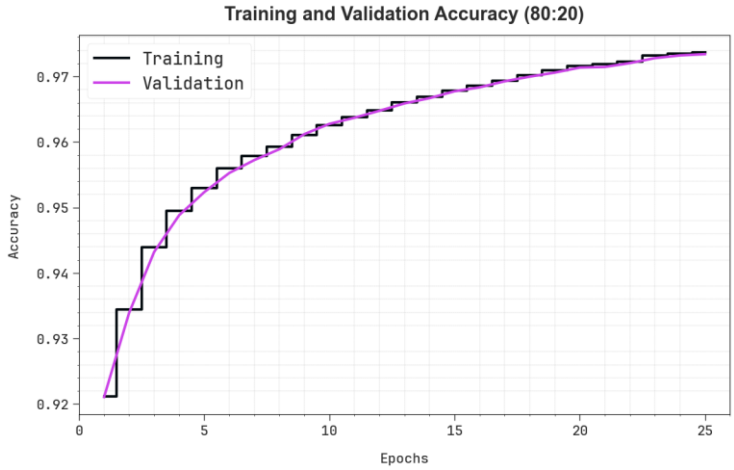

Figure 6. $Accu_y$ Analysis of ADMDL-OAWSN methodology under 80:20

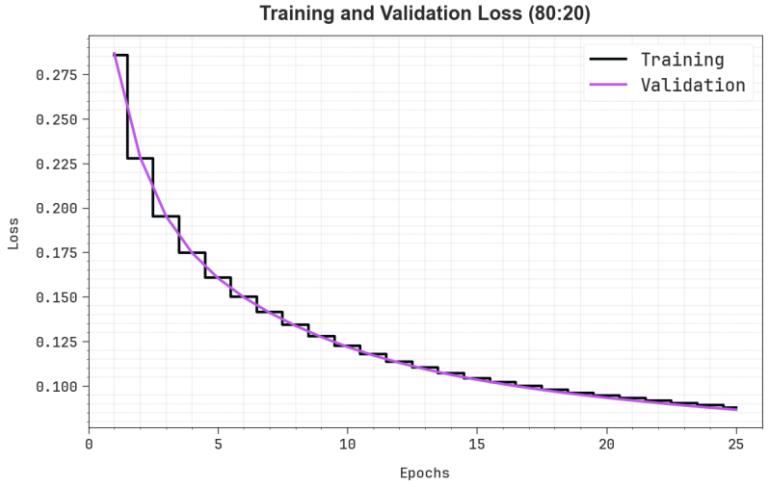

Figure 7. Loss graph of ADMDL-OAWSN model below 80:20

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

In Fig. 7, the TRAN loss (TRANLOS) and VALN loss (VALNLOS) graph of the ADMDL-OAWSN approach under 80:20 is exhibited. The loss values are calculated over the time of 0-25 epochs. It is indicated that the TRANLOS and VALNLOS analysis exemplifies a decreasing tendency, informing the ability of the ADMDL-OAWSN algorithm to balance a trade-off. The continuous decrease in values of loss guarantees the optimal performance of the ADMDL-OAWSN system and fine-tuning the prediction outcomes.

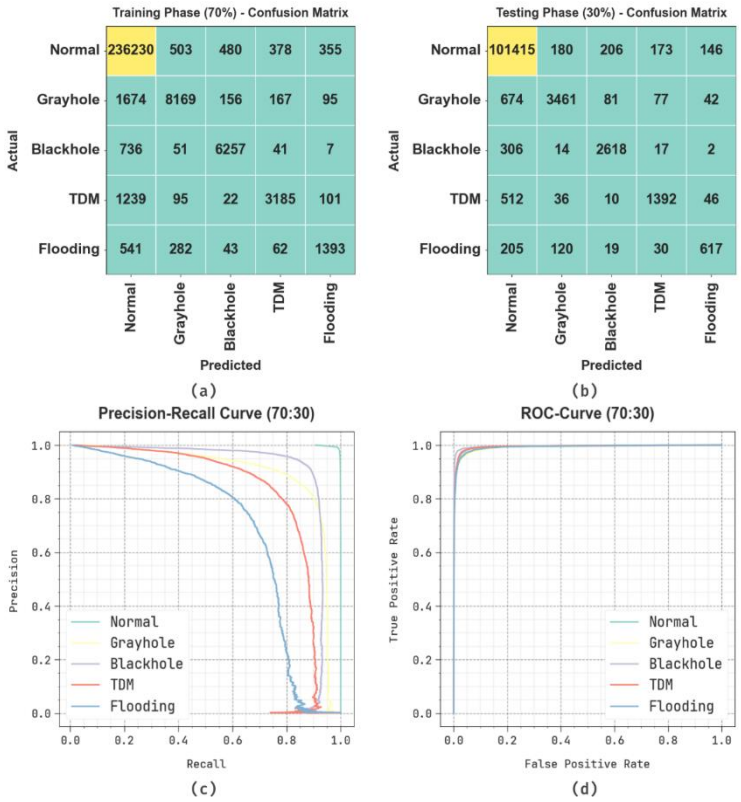

Figure 8. 70%TRAPHA and 30%TESPHA of (a-b) of confusion matrix, (c) curves of PR, and (d) ROC curve

Fig. 8 established the classifier results of ADMDL-OAWSN methodology below 70% TRAPHA and 30% TESPHA. Figs. 8a-8b shows the confusion matrices with the identification of each classes. Fig. 8c displays the analysis of PR, indicating superior outcomes over all class labels. Simultaneously, Fig. 8d proves the analysis of ROC, demonstrating accomplished outcomes with higher values of ROC for different classes.

The attack recognition of the ADMDL-OAWSN technique under 70%TRAPHA and 30%TESPHA is shown in Table 3 and Fig. 9. The outcomes imply that the ADMDL-OAWSN algorithm appropriately recognized the samples. With 70%TRAPHA, the ADMDL-OAWSN methodology offers an average $accu_y$, $prec_n$, $reca_l, F_{Measure}$, and $G_{Means}$of 98.93%, 86.49%, 79.15%, 82.52%, and 82.67%, correspondingly. Likewise, with 30%TRAPHA, the ADMDL-OAWSN method delivers average $accu_y$, $prec_n$, $reca_l, F_{Measure}$, and $G_{Means}$of 98.97%, 86.63%, 79.94%, 83.03%, and 83.16%, respectively.

Table 3: Attack detection of ADMDL-OAWSN technique below 70% TRAPHA and 30%TESPHA

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F_{measure}$	G_{Means}
TRAPHA (70%)					
Normal	97.75	98.26	99.28	98.77	98.77
Grayhole	98.85	89.77	79.61	84.39	84.54
Blackhole	99.41	89.93	88.23	89.07	89.07
TDM	99.20	83.09	68.61	75.16	75.51
Flooding	99.43	71.40	60.02	65.22	65.46

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

Average	98.93	86.49	79.15	82.52	82.67
TESPHA (30%)					
Normal	97.86	98.35	99.31	98.83	98.83
Grayhole	98.91	90.82	79.84	84.97	85.15
Blackhole	99.42	89.23	88.54	88.88	88.88
TDM	99.20	82.42	69.74	75.55	75.81
Flooding	99.46	72.33	62.26	66.92	67.11
Average	98.97	86.63	79.94	83.03	83.16

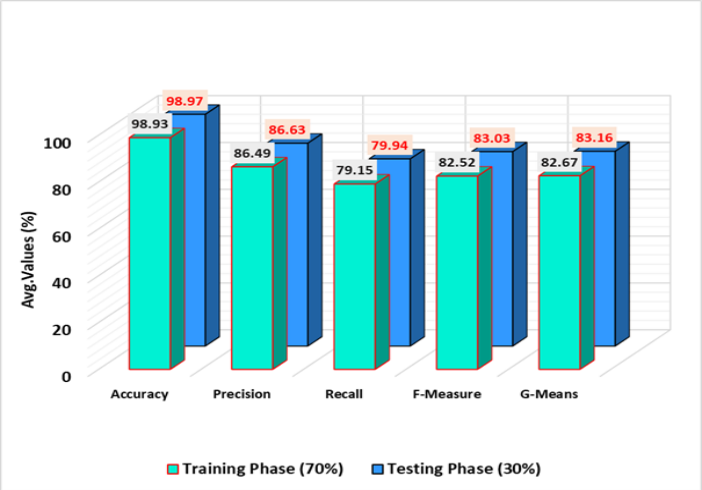

Figure 9. Average of ADMDL-OAWSN system under 70% TRAPHA and 30%TESPHA

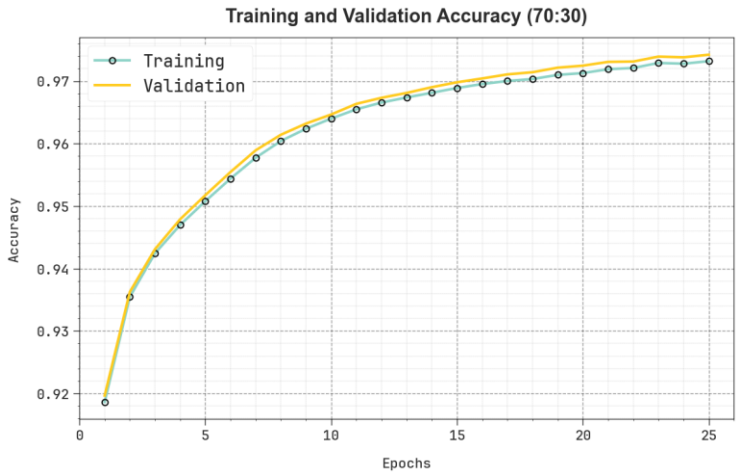

Figure 10. $Accu_y$ Curve of ADMDL-OAWSN model under 70:30

In Fig. 10, the TRAN $accu_y$ and VALN $accu_y$ analysis of the ADMDL-OAWSN approach below 70:30 is exemplified. The values of $accu_y$ are computed within the interval of 0-25 epochs. The outcome highlights that the values of TRAN and VALN $accu_y$ demonstrations a rising trend which notified the ability of the ADMDL-OAWSN technique with maximum performance over various iterations. Followed by, the TRAN and VALN $accu_y$ remnants closer over the epochs, which indicates lesser overfitting and displays the optimal efficiency of the ADMDL-OAWSN system, which guarantees consistent prediction on hidden samples.

In Fig. 11, the TRANLOS and VALNLOS graph of the ADMDL-OAWSN approach under 70:30 is shown. The values of loss are computed over the time of 0-25 epochs. It is signified that the values of TRANLOS and

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

VALNLOS illustrate a decreasing tendency, informing the capacity of the ADMDL-OAWSN algorithm in balancing a trade-off. The continuous decrease in values of loss guarantees the superior outcome of the ADMDL-OAWSN technique and tunes the forecast outcomes.

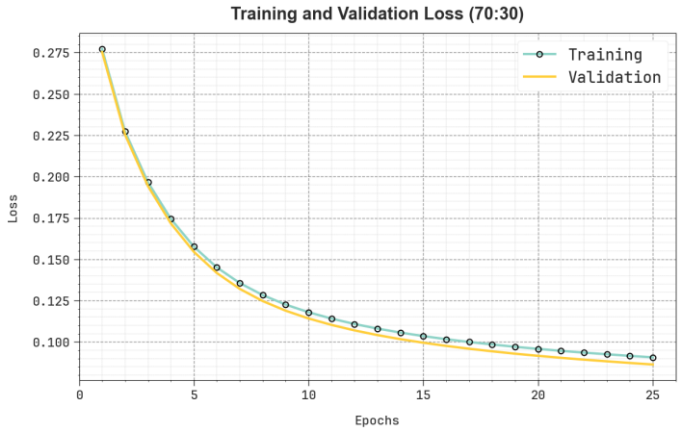

Figure 11. Loss graph of ADMDL-OAWSN technique below 70:30

Table 4 and Fig. 12 examine the comparative results of the ADMDL-OAWSN algorithm with the existing systems [24, 25]. The results highlight that the proposed IDCS-ELIBWO algorithm has accomplished higher performance with better $accu_y, prec_n$, $reca_l$, and $F_{Measure}$ of 98.97%, 86.63%, 79.94%, and 83.03%, correspondingly. Meanwhile, the existing systems such as J48, DT, AdaBoost, GB, XGBoost, KNN-AOA, and KNN-PSO have gained the worst performance.

Table 4: Comparative outcomes of ADMDL-OAWSN model with existing methodologies

Methods	$Accu_y$	$Prec_n$	$Reca_l$	$F_{measure}$
J48 Algorithm	91.34	82.71	76.10	78.73
Decision Tree	91.36	82.08	75.96	78.43
AdaBoost	95.79	85.49	69.31	76.21
Gradient Boosting	94.66	75.81	71.11	72.02
XGBoost	96.90	74.52	71.60	71.08
KNN-AOA	97.30	75.64	70.25	73.95
KNN-PSO	92.99	73.61	71.39	70.56
ADMDL-OAWSN	98.97	86.63	79.94	83.03

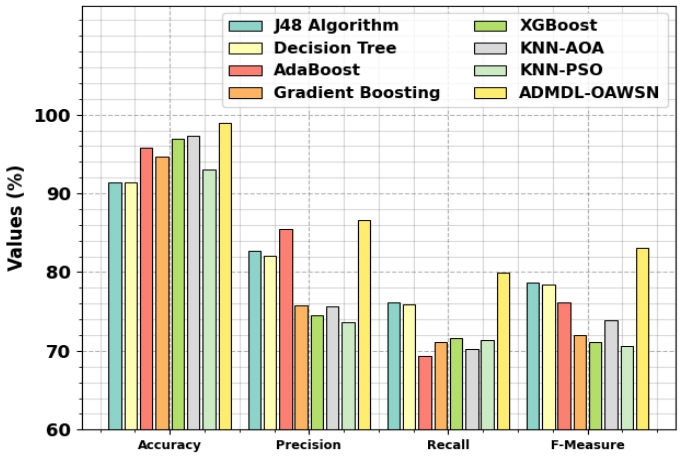

Figure 12. Comparative analysis of ADMDL-OAWSN model with existing methods

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

The processing time (PT) result of ADMDL-OAWSN system with existing approaches are displayed in Table 5 and Fig. 13. Based on PT, the ADMDL-OAWSN methodology offers worst PT of 6.05sec whereas the J48, DT, AdaBoost, GB, XGBoost, KNN-AOA, and KNN-PSO systems achieve greater PT values of 19.74sec, 8.78sec, 10.90sec, 7.79sec, 20.94sec, 15.40sec, and 22.24sec, respectively.

Table 5: PT result of ADMDL-OAWSN system with existing algorithms

Methods	Processing Time (sec)
J48 Algorithm	19.74
Decision Tree	8.78
AdaBoost	10.90
Gradient Boosting	7.79
XGBoost	20.94
KNN-AOA	15.40
KNN-PSO	22.24
ADMDL-OAWSN	6.05

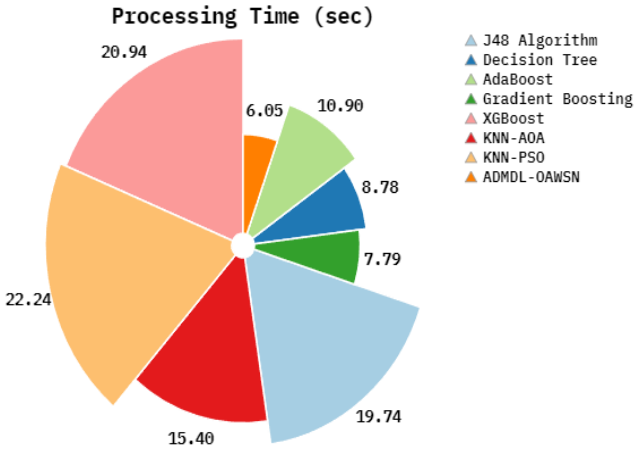

Figure 13. PT outcome of ADMDL-OAWSN system with existing approaches

5. Conclusion

In this paper, we proposed an ADMDL-OAWSN technique. The main aim of the ADMDL-OAWSN system is to improve security in cloned nodes for the cyber-attack detection model. To attain that, the ADMDL-OAWSN system has four types of stages data normalization, reduction of dimensionality, hybrid classification process, and parameter tuning using the SeWBO model. In the primary step, the data pre-processing employs the StandardScalar method to transform input data into a suitable format. Next, the proposed ADMDL-OAWSN model designs COA for the subset of the FS process to pick the most related features from an input dataset. For the attack classification process, the CNN-BiGRU-A technique has been exploited. At last, the hyperparameter range of the CNN-BiGRU-A system is applied by SeWBO algorithm. A huge range of experiments has been directed to validate the performance of the ADMDL-OAWSN system. The simulation outcomes revealed that the ADMDL-OAWSN system emphasized furtherance when equated to other recent systems.

Funding: "This research received no external funding"

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

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

References

[1] M. Premkumar and T. V. P. Sundararajan, "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks," *Microprocessors and Microsystems*, vol. 79, p. 103278, 2020.

[2] M. N. U. Islam, A. Fahmin, M. S. Hossain, and M. Atiquzzaman, "Denial-of-service attacks on wireless sensor network and defense techniques," *Wireless Personal Communications*, vol. 116, pp. 1993–2021, 2021.

[3] V. L. Thing, "IEEE 802.11 network anomaly detection and attack classification: A deep learning approach," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[4] E. Gelenbe et al., "IoT network attack detection and mitigation," in *Proc. 9th Mediterranean Conf. Embedded Computing (MECO)*, Jun. 2020, pp. 1–6.

[5] Y. E. Sagduyu, Y. Shi, and T. Erpek, "IoT network security from the perspective of adversarial deep learning," in *Proc. 16th Annu. IEEE Int. Conf. Sensing, Communication, and Networking (SECON)*, Jun. 2019, pp. 1–9.

[6] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine learning for wireless sensor networks security: An overview of challenges and issues," *Sensors*, vol. 22, no. 13, p. 4730, 2022.

[7] T. Kim, L. F. Vecchietti, K. Choi, S. Lee, and D. Har, "Machine learning for advanced wireless sensor networks: A review," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12379–12397, 2020.

[8] M. Aslam et al., "Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT," *Sensors*, vol. 22, no. 7, p. 2697, 2022.

[9] M. T. Kurniawan and S. Yazid, "Mitigation and detection strategy of DoS attack on wireless sensor network using blocking approach and intrusion detection system," in *Proc. Int. Conf. Electrical, Communication, and Computer Engineering (ICECCE)*, Jun. 2020, pp. 1–5.

[10] A. R. W. Sait, I. Pustokhina, and M. Ilayaraja, "Mitigating DDoS attacks in wireless sensor networks using heuristic feature selection with deep learning model," *Full Length Article*, no. 2, pp. 65–5, 2021.

[11] P. Deepalakshmi and T. Kumanan, "Fake clones for adversaries detection with efficient relay selection in MWSN," in *Proc. 2nd Int. Conf. Device Intelligence, Computing and Communication Technologies (DICCT)*, Mar. 2024, pp. 590–594.

[12] S. Bhuvana, S. K. Andrews, M. S. Josephine, and V. Jeyabalaraja, "Relative spectral feature analysis-based clone attack detection and enhance routing in wireless sensor networks using artificial neural networks," *J. Data Acquisition and Processing*, vol. 38, no. 3, p. 1770, 2023.

[13] R. Vatambeti et al., "Classification of HHO-based machine learning techniques for clone attack detection in WSN," *Int. J. Computer Network and Information Security*, vol. 15, pp. 1–15, 2023.

[14] P. E. David, "Automatic clone detection in wireless sensor networks using ANN," 2023.

[15] K. J. Nithya and K. Shyamala, "Entropy dove swarm optimization (EDSO) based cluster head selection and stacked ensemble learning-clone attack detection (SEL-CND) for wireless sensor network (WSN)," in *Proc. OPJU Int. Technology Conf. (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, Jun. 2024, pp. 1–13.

[16] M. Numan, F. Subhan, M. N. A. Khalid, W. Z. Khan, and H. Iida, "Clone node detection in static wireless sensor networks: A hybrid approach," *J. Network and Computer Applications*, vol. 232, p. 104018, 2024.

[17] S. C. V. Bhaskar et al., "Augmenting cybersecurity in WSN: AI-based clone attacks recognition framework," in *Proc. Asian Conf. Communication and Networks (ASIANComNet)*, Oct. 2024, pp. 1–6.

[18] K. Hameed et al., "A context-aware information-based clone node attack detection scheme in Internet of Things," *J. Network and Computer Applications*, vol. 197, p. 103271, 2022.

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025

[19] K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection model using deep learning optimization techniques," *Int. J. Data Science and Analytics*, pp. 1–23, 2025.

[20] S. Chauhan, G. Vashishtha, R. Zimroz, and R. Kumar, "A crayfish-optimized wavelet filter and its application to fault diagnosis," *arXiv preprint arXiv: 2502.08362*, 2025.

[21] J. Yu et al., "Impact localization system of CFRP structure based on EFPI sensors," *Sensors*, vol. 25, no. 4, p. 1091, 2025.

[22] S. Mekala and P. K. Singamsetty, "A hybrid approach for pancreatic cancer detection in CT scans using secretary wolf bird optimization and deep learning," *SSRN*, 2025. Available: [Online]. Available: https://www.ssrn.com/abstract=5134659

[23] Kaggle, "Wireless sensor network dataset," *Kaggle.com*, 2025. [Online]. Available: https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds

[24] H. Fares, A. D. Vibhute, Y. Mouniane, and H. Bouijij, "Intrusion detection in wireless sensor networks using machine learning," *Procedia Computer Science*, vol. 252, pp. 912–921, 2025.

[25] A. K. Alkhalifa et al., "Hybrid dung beetle optimization-based dimensionality reduction with deep learning-based cybersecurity solution on IoT environment," *Alexandria Engineering J.*, vol. 111, pp. 148–159, 2025.

DOI: https://doi.org/10.54216/FPA.190115
Received: November 20, 2024 Revised: January 23, 2025 Accepted: February 18, 2025