SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I12P111

Volume 12 Issue 12, 134-146, December 2025

Original Article

Automated Clone Detection in Wireless Sensor Networks Using Ensemble Learning Models with Hybrid Optimization-Based Significant Feature Selection Approach

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

[1,2]*Department of Information Technology, FEAT, Annamalai University, Chidambaram, Tamil Nadu, India.*

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

Received: 11 October 2025 Revised: 13 November 2025 Accepted: 12 December 2025 Published: 27 December 2025

Abstract - *Wireless Sensor Networks (WSN) consist of miniature sensor nodes that communicate among themselves via wireless channels, often in an unfriendly environment, and nodes can be carried and defeated. Thus, an enemy may also attack the clones by copying the nodes taken and broadening the breaching areas with the help of clones. Hence, to reduce the losses of clone nodes to the WSNs, it is crucial to detect them as soon as possible. Other types of clone detection systems have been proposed in the recent past for WSNs, bearing in mind the dissimilar types of network structures, such as deployment strategies and types of devices. The Deep Learning (DL) techniques, however, are used to identify and clone nodes in WSN. A Hybrid Optimization-Based Feature Learning is presented in this paper regarding Clone Detection Using Ensemble Learning Models (HOFLCD-ELM). The project seeks to create and assess an effective clone detection technique in wireless sensor networks to improve network security and integrity. The initial phase of data preprocessing is the min-max normalization approach, which transforms raw data into a usable format for modeling. In the feature subset selection procedure, the proposed HOFLCD-ELM model develops a hybrid optimization process in the form of Lyrebat Algorithm (LYBA) that integrates Lyrebird Optimization Algorithm (LOA) and Bat Algorithm (BA) in order to find the optimal features within a dataset. Subsequently, the system of Deep Belief Network (DBN) model, Convolutional Variational Autoencoder (CVAE) method, and Graph Convolutional Network (GCN) has been implemented to identify and classify clone attacks. Lastly, the optimization process of the Spider Wasp (SWO) model is used to acquire the parameter tuning process in enhancing the classification of the ensemble classifier. The experimental analysis of the HOFLCD-ELM model is done through a benchmark and a dataset. The results of the empirical study showed that the performance of the HOFLCD-ELM method was improved more than that of the current methods.*

Keywords - *Clone detection, Wireless Sensor Networks, Spider Wasp Optimization, Hybrid model, Ensemble deep learning.*

1. Introduction

The Wireless Sensor Networks (WSNs) and, in particular, their security issues, have found considerable momentum at present both industrially and academically. Since small sensor nodes in WSNs have limited capabilities in aspects of communication, processing, storage, and power, it is hard to enforce appropriate security measures and procedures of the WSNs [1]. Specifically, since WSNs are often deployed in unfriendly locations, sensor nodes are readily undermined and lost by attackers who can intercept confidential information in the lost sensor nodes [2]. After such a violation, the clone attacks can be launched by imitating the affected nodes and distributing them in the networks, such that the attacker can grow the struck areas through the use of the clones. Confidential information, such as encryption keys, stolen from the nodes that were attacked and stored in the clones, can authorize the attacker to retrieve the communication architecture in WSNs [3]. As an example, in a key management protocol of WSNs, the clones can be checked as important nodes in more than one area, such as disrupting data aggregation, sending wrong data, and discarding the packets as they choose. Hence, it is important to detect clone nodes in order to restrict their damage to WSNs [4]. Figure 1 is the overall organization of WSNs.

When such clones are not identified, the network becomes vulnerable to attackers and thus very vulnerable. As a result, clone attacks are very harmful. There is a need to have precise and practical clone attack detection formulations in order to mitigate their effects [5]. The primary challenge appears due to the fact that the duplicates also have all the authentication

information (ID, keys, codes, etc.) of the initial compromised node [6]. Therefore, they can sign out all checks and not be known to be counterfeit. Besides, an intelligent clone could be trying to evade identification by any means. In addition, clones can also collaborate to deceive the network manager into believing that they are actual [1]. Research findings indicate that a sensor node is prone to various risks, i.e., clone node attacks or node replication, by virtue of its nature, which includes the lack of non-tamper-responsive hardware, limited computing power, energy, and memory [7]. Many centralized, distributed, and network-based detection approaches were developed to prevent clone node attacks. Wireless Sensor Networks (WSN) clone nodes are identified using ML and DL [8]. These paradigms can examine network traffic, behaviour, and other characteristics of the nodes in order to detect clones. As it is further investigated and developed, such models will enhance the security and reliability of WSNs [9].

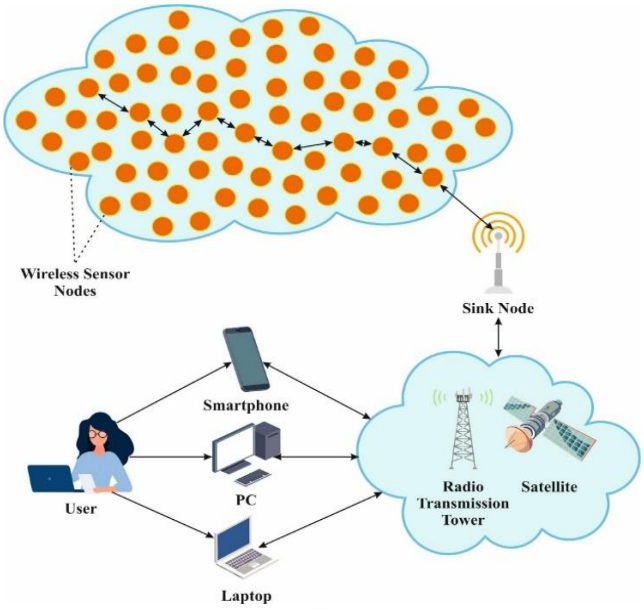


Fig. 1 General structure of Wireless Sensor Networks

A Hybrid Optimization-Based Feature Learning to Clone Detection is constructed in this paper via Ensemble Learning Models and referred to as HOFLCD-ELM. The key contributions of this paper are as illustrated below:

- An HOFLCD-ELM is a novel approach proposed to support and test an effective clone detection model within WSN in order to achieve network security and integrity.
- Normalization in the min-max approach is first applied in the data preprocessing stage.
- In the process of selecting a feature subset, the proposed HOFLCD-ELM model develops a hybrid optimization algorithm called Lyrebat Algorithm (LYBA) that was developed by combining Lyrebird Optimization Algorithm (LOA) and Bat Algorithm (BA).
- The Deep Belief Network (DBN) model, the Convolutional Variational Autoencoder (CVAE) method, and the Graph Convolutional Network (GCN) system have been implemented.
- Finally, the parameter tuning process is achieved using the Spider Wasp Optimization (SWO) model.

2. Literature Review

Nashaat et al. [10] introduced CloneXformer, a novel method to detect code clones. This method implements a collaborative methodology that employs several Large Language Models (LLMs) to understand code. This method utilizes a primary phase for preprocessing the input code, which assists the model in understanding and representing the code effectively. Later, these techniques are fine-tuned to recognize code clones with explainable outcomes, which clarify the types of clones. Dora et al. [11] presented an Intelligent Clone Detection and classification through Cat Swarm Optimizer alongside a DL approach for WSN. This approach's motive is to identify and classify clone nodes within the network accurately. Swilam et al. [12] introduced an improved AST, optimized by the presence of Condition-Type Edges that efficiently model logical connections in control structures. This new addition provides an in-depth semantic understanding of the code's decision-making, overcoming the flaws of traditional ASTs, which focus on syntactic relations. By incorporating this improved AST with Graph Neural Networks (GNNs), this methodology acquires strong feature representations that extract structural and semantic differences across programming languages.

A Novel Adaptive Sea-Horse Optimized Light Gradient Boosting Machine (ASHO-LGBM) technology by Bhaskar et al. [13] protected the network against node identity duplicates. ASHO-LGBM uses ASHO to improve LGBM feature accuracy. The node Intrusion Detection (ID) duplications are utilized in the selection of the most dependable communication way. In [14], a process known as Stacked Ensemble Learning-Clone Attack Detection (SEL-CND) has been proposed as a procedure for detecting clone attacks. This identifies the clone nodes of the Mobile WSN. The sensor network is segmented into groups. Clusters have a central node and an arbitrary number of sensor nodes. The Entropy Dove Swarm Optimizer (EDSO) selects the Cluster Head to enhance network performance. The EDSO model uses dove foraging. WSN clone nodes are identified via the SEL-CND module.

Vatambeti et al. [15] proposed an ML-based CND algorithm on WSN clone node detection. The objective is to detect clones to avert clone attacks accurately. The Optimized Extreme Learning Machine (OELM) and ELM kernels were utilized, optimized by the Horse Herd Metaheuristic Optimizer (HHO) approach. Bhuvana et al. proposed an upgraded transfer learning model using NFI-SSFS. [16], aims to ensure Cooperative Secure Optimal Link Stability Routing Allocation (CS-OLSR) and is contingent upon the detection of

clone attacks. The logs of communications are amalgamated in order to use the level of variance feature of the rates of packet discrepancy with the reliance on the memory and transmission errors, which focuses on the Time Stamp Communication Behavior Rate (TSCBR), and the False Injection Impact Rate (FIIR). Second, the CS-OLSR is applied to ensure secure routing in the area of clone attack.

3. Proposed System

The objective of the paper is to derive and analyze an effective clone detection model in the WSNs to increase the network security and integrity. To achieve that, the HOFLCD-ELM system has normalization of min-max, selection of features, an ensemble model, and parameter optimization. The total action of the HOFLCD-ELM system is shown in Figure 2.

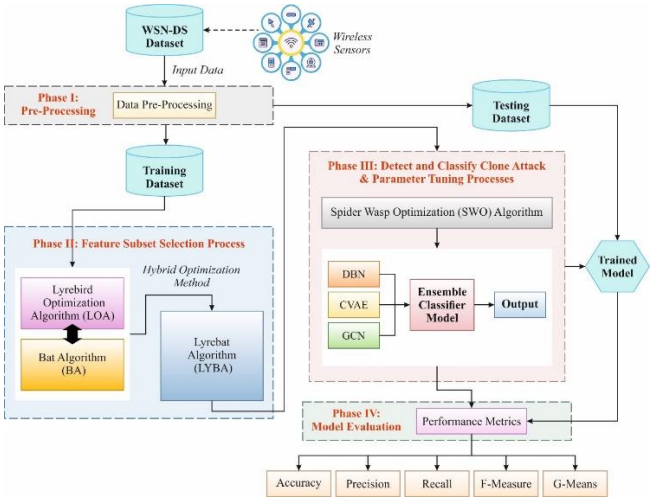


Fig. 2 Overall procedure of HOFLCD-ELM system

3.1. Data Preprocessing Model

Min-max normalization is the first data preparation approach used to prepare raw data for modeling. The Min-Max Normalization is a prominent preprocessing process that is used to normalize numerical data within a specific interval, which in most cases is [0,1] or the range [-1,1]. The model will ensure every feature plays a fair role during the training of a model, and not focus on features that are big [17]. Deep Learning (DL) techniques can generate better detection results and be trained on patterns by normalizing the database better. A general expression of Min-Max Normalization is as follows, given by Equation (1):

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} (new\_\max - new\_\min) + new\_\min \quad (1)$$

Whereas X denotes a new data point, $X_{\min}$ and $X_{\max}$ represent minimal and maximal values of the feature, new_min and new_max describe the preferred normalization range, X' Means normalized value. In the case where data comprises either positive or negative values, this model is changed to scale inside a range of [-1,1] as presented in Equation (2):

$$X' = 2 \times \frac{X - X_{\min}}{X_{\max} - X_{\min}} - 1 \quad (2)$$

This conversion ensures that the values remain around zero, which is also beneficial to DL methods because it stabilizes weight changes during training. It not only accelerates convergence but also improves the accuracy.

3.2. Feature Selection using Hybrid Optimization Method

When using the feature subset selection process, the suggested HOFLCD-ELM model develops a Hybrid Optimization Algorithm named LYBA. By adding BA's velocity module to LOA's hiding phase, the LYBA hybridization strikes a balance between exploration and exploitation [18].

The decision parameter values in the LYBA are determined by each participating lyrebird and are contingent upon the location inside the problem-solving area. A vector is a defined variable, and every lyrebird is a vector in mathematics. Equation (3) represents the model population as LYBA members. LOA members are randomly placed in the problem-solving domain using Equation (4).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \quad (3)$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \quad (4)$$

In this context, m represents the count of decision parameters, N indicates the total counts of lyrebirds, r is a random variable within the interval [0,1], and the upper and lower bounds of the decision variable are denoted by ub_d and lb_d , respectively. The LOA population matrix is represented by the symbol X, the i^{th} member of LYBA (promising solution) by X_i, and the d^{th} dimension in the search region by $x_{i,d}$. Equation (5) presents the vector representation of the evaluated values of the problem's objective function.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (5)$$

Each objective function value is represented by a vector F, where F_i is the i^{th} LYBA member. The option with the higher goal function value is best; the option with a lower objective function value is suboptimal. An exponential factor

is engaged in enhancing the lyrebird's updated stage position. This sample of the lyrebird indicates that the population update process consists of two phases: (i) concealment and (ii) evasion. When the lyrebird must choose between concealing itself and fleeing from peril, Equation (6) in the LYBA model simulates its cognitive deliberation. The placement of each LYBA member is altered in every iteration just for the execution of the first or second stage.

$$Updated\ process\ for\ X_i: \begin{cases} based\ on\ Phase\ 1, r_p \le 0.5 \\ based\ on\ Phase\ 2, else \end{cases} \quad (6)$$

In such a case, r_p Refers to a random number in the interval of [0,1].

3.2.1. Stage 1: Exploration (Escaping Strategy)

The LYBA stage makes use of a model based on the movement of the flight of a lyrebird to increase the density of population members within the search space. When LYBA is relocated to a more secure setting, it may demonstrate its capacity to execute a comprehensive global search and exploration process. It leads to substantial alterations in roles and the exploration of alternate locations within the realm of problem-solving.

The LYBA describes a safer area as a relative position of the most subjective member of the population that has the highest values of the objective function. Equation (7) has been used to find a list of safer areas for all the members of LOA.

$$SA_i = \{X_k, F_k < F_i \ \&\ k \in \{1,2,3,\dots,N\}\}, i = 1,2,\dots,N \quad (7)$$

The objective function value (F_k) of the kth row of the X matrix (X_k) is greater than the ith LYBA element ($F_k < F_i$). SA_i represents the safe areas for the i^{th} Lyrebird. The use of modeling by the lyrebird movement ended on this step. The new position of every LYBA member is calculated using Equation (8). The technique adapts its escape strategy to prey velocities using velocity factors. The model's flexibility lets it intelligently adjust to environmental or prey changes. Equation (9) states that the member associated with the objective will be moved if its value increases.

$$x_{i,j}^{P1} = x_{i,j} + v_i^t . r_{i,j} . \left(SSA_{i,j} - I_{i,j} . x_{i,j}\right) \quad (8)$$

$$v_i^t = v_i^{t-1} + \left(x_{i,j} - x_{i,j}^{best}\right) F_i \quad (9)$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} \le F_i \\ X_i, else \end{cases} \quad (10)$$

Whereas $x_{i,j}^{best}$ specifies the best choice and v_i^t Refers to the velocity of the prey. In this case, SSA_i characterizes the i^{th} safer place of the lyrebird; $SSA_{i,j}$; specifies its j^{th} size; the upgraded location is computed according to the recommended escaping tactic of LYBA, utilizing Equation (10); F_i^{P1} represents the objective function X_i^{P1}; $r_{i,j}$ represents random values from [0,1]; $I_{i,j}$ Represents randomly picked 1 or 2 numbers.

3.2.2. Stage 2: Exploitation (Hiding Strategy)

The positioning of people in the search range at this LYBA level is similar to the strategy of the lyrebird to retreat to a surrounding and safer nest. This is the tactic used by the lyrebird, which gradually changes its location as it carefully explores its immediate environment and walks around seeking shelter.

This demarcates LYBA's application in local search tasks. The original place of each member is ascertained by LYBA mimicking that of the lyrebird in flying to a favourite hiding place in the neighbourhood, as described in Equation (11). If Equation (12) is met, the associated member's objective function is substituted with the new location if it expands.

$$x_{i,j}^{P2} = x_{i,j} + \left(1 - 2r_{i,j}\right) . \frac{ub_j - lb_j}{t} \quad (11)$$

$$X_i = \begin{cases} X_i^{P2},\ F_i^{P2} \le F_i \\ X_i, else \end{cases} \quad (12)$$

The iteration number in this sample is t, and the random integers are denoted as follows. $r_{i,j}$ are drawn from the interval [0,1], $x_{i,j}^{P2}$ represents the j^{th} dimension, F_i^{P2} denotes the objective function value, and the new position of the i^{th} Lyrebird is determined using the proposed LYBA's concealing method. The Area of Fitness Measure (FF) concerning the classification accuracy and the desired number of features is measured. It reduces the set size of attributes and enhances the classifier's precision. The subsequent FF calculates individual solutions as shown in Equation (13):

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \quad (13)$$

In this instance, $ErrorRate$ is the rate of error when using the labeled features for classification. $ErrorRate$ pertains to the wrong (i.e., erroneous) percentage assigned to the classification counts made, and is defined as the product of (0,1) and SF is counts of features picked in the new database, F represents the total number of characteristics in the new database, whereas an is employed to govern the importance of quality and size of subgroups within the classifications.

3.3. Ensemble Classification Process

Then, the system of the DBN model, CVAE technique, and GCN system has been implemented to detect and classify clone attacks.

3.3.1. DBN Model

In order to effectively recognize the complex and dynamic patterns within raw data, without any extra structure, consider a DBN structure [19]. Based on the outstanding research of DBNs, it has acquired significance regarding its ability to discover composite, hierarchical representations of unlabeled data. DBNs can be used to model higher-dimensional distributions, and thus are largely applicable to the nonlinear and nonstationary character of data by stacking many layers of Restricted Boltzmann Machines (RBMs). Provided the raw data matrix $X \in \mathbb{R}^{n\times d}$ (while n denotes sample counts and d refers to feature counts), It converts this data into a novel data area H over numerous layers. Every DBN layer l, with parameters θ_l Uses a nonlinear transformation: $H^{(l)} = f_{\theta_l}\left(H^{(l-1)}\right)$, while $H^{(0)} = X$ and f_{θ_l} Characterizes the transformation by every RBM. An RBM models the combined distribution among the observed vector v and the Hidden Layer (HL) h utilizing a bipartite graph:

$$P(v,h;\theta) = \frac{1}{Z(\theta)}\exp(-E(v,h;\theta) \tag{14}$$

Whereas $E(v,h;\theta)$ refers to configuration energy (v,h):

$$E(v,h;\theta) = -\sum_{i,j} v_i\, w_{ij} h_j - \sum_i b_i\, v_i - \sum_j c_j\, h_j \tag{15}$$

With hidden biases c_j, weights w_{ij}, visible biases b_i, and the partition function $Z(\theta)$ that standardizes the distribution. The layers of DBN are trained using the contrastive divergence, gradually enhancing the feature representation as the states that are close to seizure appear, as compared to different ones. The DBN compresses lower-level data or higher-level trends, enhancing adaptability to new trends due to concept drift by showing the tourism industry data at various levels of abstraction. The success of the DBN in this model lies in its label-free nature, as it is an unsupervised algorithm that learns hierarchical feature representations directly from the data. The DBN leverages graphics and identifies motifs in the information dispersion, recognizing changes in the data distribution by modeling the joint likelihood distribution of concealed and visible units, without labeling samples.

3.3.2. CVAE Technique

Neural networks can be trained unsupervised to duplicate their input [20]. The AE's basic diagram has output and input. Commonly used for data compression. Autoencoders translate input data into feature space z using encoders. Feature space is the latent encoder space. This decoding challenge derives latent data representations and predicts the data for the input region.

Nevertheless, a single-layered autoencoder would be unable to eradicate the descriptive characteristics of raw data. It requires a sophisticated AE. This process was further complicated by obligating the latent representation to comply with some distributions, like Gaussian, Variational Autoencoders (VAEs). It resulted in the latent variable z shifting to a latent space that has a probability distribution and constant statistical measures.

This is utilized to get the variance, o, and mean, m, of the latent variable z, which entails encoding to extract the decoding inputs, employing the specified latent variable distribution, z. The VAE encoder will transform an input picture point into a distribution throughout the latent space. Nevertheless, rather than the mean value, sometimes this model can reconstitute the input signals. Hence, CVAEs are applied because they have an improved capacity in applying the encoding/decoding through using layers of Fully Connected (FC)-based and in the ability to source the time and locality relationship that occurs in the data.

The loss function of the CVAE is founded on two components. The initial component focuses on reducing the discrepancy between the input and output. In contrast, the subsequent component assesses the extent to which the latent space distribution deviates from the designated distribution. In this work, Mean Squared Error (MSE) is used to measure the dissimilarity between the reconstructed data and the input data to measure them. However, alternative tasks (such as the role of binary crossentropy) are also calculated, as shown:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(I^i - \bar{I}^i)^2 \tag{16}$$

Whereas$\bar{I}$And I represents reconstructed input and input, and n means data dimensionality. However, to compute the amount, the latent variable z approaches particular distributions; in such a case, a typical standard distribution, the divergence of the Kullback-Leibler (KL) D_{KL} It is applied that estimates the divergence among dual distributions and serves as a term of regularizer:

$$D_K = \int p\,(x)\log\left(\frac{p(x)}{q(x)}\right)dx \tag{17}$$

Whereas $p(x)$ and $q(x)$ are dual distributions. Therefore, the loss function L_{CVAE} should be subject to MSE and D_{KL} and described as shown:

$$L_{CVAE} = k \times MSE + E(D_{KL}) \tag{18}$$

Here, k means scaling factor, and E refers to expected value.

3.3.3. GCN System

Graph-Based Neural Networks (NNs) are also DL algorithms that have recently attracted particular interest in modeling linked data in the form of composite networks [21]. As opposed to regular NNs, GNNs take relational data as input

in the form of nodes and edges, not in the form of 1D strings. GCNs refer to NNs having tighter patterns compared to GNNs. GCNs are a method using a convolution operation on the input graph data in the form of arbitrarily defined filters, and subsequently involve a collection of operations to produce results. The best quality of GCNs relative to other graph-based methods is that they present a better insight into the spatial attributes based on the data in the graph architecture. During GCN techniques, a graph is described as $G = (V, E)$. During the graph description, V is described as the collection of nodes, and $E \subseteq VxV$ is well-defined as the collection of edges.

To work on the graph, node features are frequently stated by the feature matrix $X \in \mathbb{R}^{NxF}$. Whereas N denotes node counts and F denotes feature dimensions of every node. The edge information is characterized by the matrix of adjacency $A \in R^{NxN}$. The graph convolution process disseminates neighbourhood information through nodes by incorporating node features. Figure 3 illustrates the framework of the GCN system**.**

$$H^{(l+1)} = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right) \tag{19}$$

Whereas $H^{(l)}$ refers to node features in the lth layer, $\hat{A}$ Denote the normalized adjacency matrix and $W^{(l)}$ Denote learnable weights. The normalized adjacency matrix $\hat{A}$ Balance the result of node neighbourhoods, which offers mathematical stability, and is stated as shown.

$$\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2} \tag{20}$$

Here, $\tilde{A} = A + I$ denotes that self-connections of nodes are comprised by adding the unit matrix. $\tilde{D}$ specifies the degree matrix of $\tilde{A}$. Therefore, for the graph-level classification, node features are pooled and then classified:

$$z = Pool\left(H^{(L)}\right),\ \hat{y} = softmax(z) \tag{21}$$

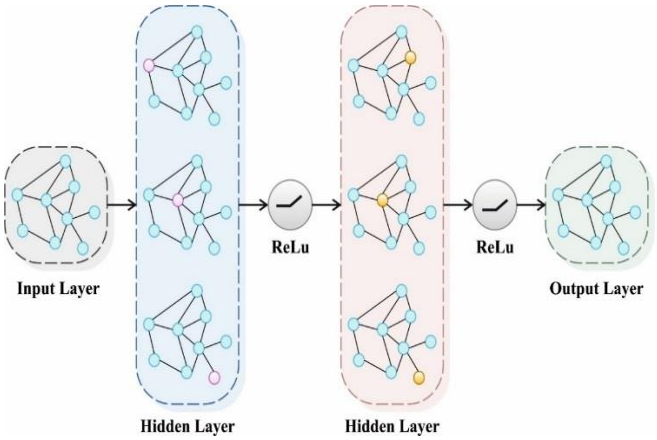


Fig. 3 Framework of GCN system

When running this updated rule, the features of each node are updated using a weighted average of the features of neighbouring nodes and a learnable weighted matrix. Degree normalization ensures the nodes that have many neighbours have fewer influences on the procedure that is updated. Similarly, the model learns to balance, and nodes that have many neighbours do not get over-taken.

3.4. SWO-based Parameter Tuning Model

Lastly, the parameter tuning process is realized by the SWO model towards enhancing the classification performance of ensemble classifiers. The optimizer strategy is the SWO; this strategy resembles the behavior of spider wasps [22]. They are recognized for seeking out spiders, injecting them with venom, and transporting them to their nests for their offspring to consume. The spider wasp present in the search area is a characteristic common to all candidates. It employs spiders and wasps as agents to mimic this method. In this given paper, parameter tuning has been solved by using SWO. Every person in this model defines an attractive solution in order to make the difficulty better. This model uses the Fitness Function (FF), which successively augments these solutions, which will be used to search for the best selections of features that can accurately predict. The optimization of the parameter gets carried out using the stages such as the initiation, evaluation of fitness, exploitation, exploration, and termination. Specific processes of this model are described as illustrated.

3.4.1. Initialization

In this case, the parameter population is arbitrarily determined. The optimization controls, including the number of people to use, the maximum iteration count to employ, the parameters of the SWO, etc., are further initialized to begin the optimizer process.

$$\kappa_{ij} = Lb_j + rand.\left(Ub_j - Lb_j\right) \tag{22}$$

Whereas κ_{ij} specifies the population, Ub_j and Lb_j Describes upper and lower search regions.

3.4.2. Fitness Evaluation

Accordingly, the fitness solution was projected for all sequences of parameters according to its goal function. During the presented setting, the primary goal of the SWO is to lower the loss function, as described in Equation (23).

$$Objectivefunction = \min\left(Loss_F = \frac{1}{tm}\sum_{i=1}^{tm}(A_p - P_p)^2\right) \tag{23}$$

The more the loss is received through the method, the greater the corresponding fitness of the particular set of parameters.

3.4.3. Exploration and Exploitation

Here, the model discovers the complete searching region and upgrades the values of parameters to fix its best range.

Now, the parameters such as the time constant $((\tau'))$Weights (W) and threshold potentials (ϕ) parameters are adjusted in the model to improve the detailed investigation. At first, characterize each wasp utilizing Equation (24).

$$P = (W, \tau', \varphi) \tag{24}$$

Formerly, compute the excellence of all wasps utilizing FF for assessing the prediction precision and complexity. Furthermore, the FF starts with the initialization procedure of every feature as f_1, f_2Formerly, the exploitation and exploration process was used at the location, all wasp utilizing Equation (25).

$$P_i = P_i(t) + r'.\left(B_{LocaI} P_i(t)\right) + \eta.R_e \tag{25}$$

Whereas, B_{local} It is characterized as a top local solution; the rate of learning is discovered as r', η means performance measure, and Ris random selection parameter process.

3.4.4. Termination

After the parameter improvement, the innovative solution was supported with the fitness solution. When the new tested fitness is increased, the new model chooses the better sequence of parameters to be used in training. The entire process of updating the parameters will advance until the highest iteration limit. This would mean that it gives the best value to the module in every iteration. The SWO model results in a feature function to increase classification efficacy. It describes a complex measure to assess the increased functionality of the possible corrections. The FF is the decrease of the error rate in the classification described in the following Equation (26):

$$fitness(x_i) = ClassifierErrorRate(x_i)$$

$$= \frac{no\ of\ misclassified\ samples}{Total\ no\ of\ samples} * 100 \tag{26}$$

4. Validation and Results

The validation of the HOFLCD-ELM system is experimentally confirmed in the database of WSN-DS [23]. There are 374661 instances of five types of attacks in this database, as indicated in Table 1. The TDM types are referred to as Time Division Multiple Access attack. It has a total of 18 features, but 13 have been selected.

Table 1. Details of the database

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

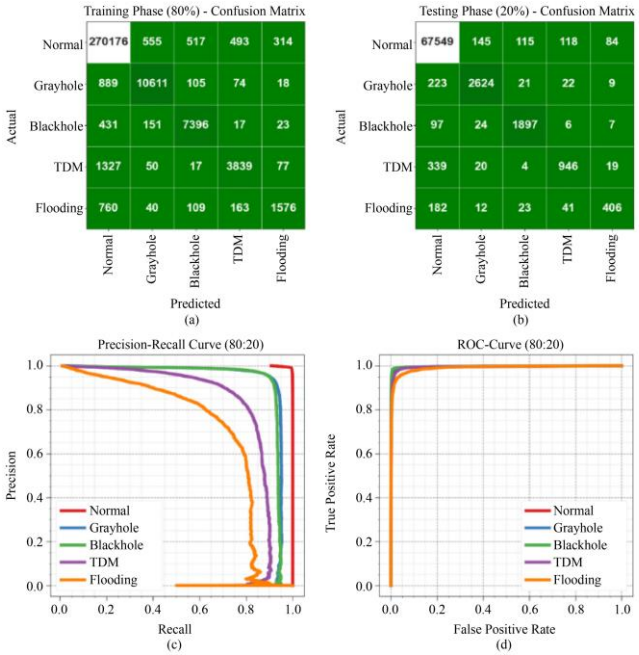


Fig. 4 80:20 of (a, b) Confusion Matrices and, (c, d) Curves of PR and ROC.

Figure 4 will explain the results of the HOFLCD-ELM method, which are of an 80:20 ratio. Figures 4(a), and 4(b) show the confusion matrix, as all the categories are detected and classified accurately. Figure 4(c) indicates that PR inspection is best in all classes. The ROC analysis is lastly presented in Figure 4(d), which demonstrates the achievement of success with better ROC values for each class.

Table 2. Clone attack detection of the HOFLCD-ELM model under 80:20

Class	Accuracy	Precision	Recall	F-Measure	G-Means
TRPHE (80%)					
Normal	98.24	98.75	99.31	99.03	99.03
Grayhole	99.37	93.02	90.72	91.85	91.86
Blackhole	99.54	90.82	92.24	91.52	91.53
TDM	99.26	83.71	72.30	77.59	77.80
Flooding	99.50	78.49	59.52	67.70	68.35
Average	**99.18**	**88.96**	**82.82**	**85.54**	**85.71**
TSPHE (20%)					
Normal	98.26	98.77	99.32	99.04	99.05
Grayhole	99.36	92.88	90.51	91.68	91.69
Blackhole	99.60	92.09	93.40	92.74	92.74
TDM	99.24	83.50	71.23	76.88	77.12
Flooding	99.50	77.33	61.14	68.29	68.76
Average	**99.19**	**88.91**	**83.12**	**85.73**	**85.87**

HOFLCD-ELM system clone attack detection at 80:20 is shown in Table 2 and Figure 5. Under 80% TRPHE, the HOFLCD-ELM model averages $accu_y$ of 99.18%, $prec_n$ of 88.96%, $reca_l$ of 82.82%, $F_{Measure}$ of 85.54%, and G_{Means}Of 85.71%. Likewise, at 20% TSPHE, the proposed HOFLCD-ELM model gets average $accu_y$ of 99.19%, $prec_n$ of 88.91%, $reca_l$ of 83.12%, $F_{Measure}$ of 85.73%, and G_{Means}Of 85.87%.

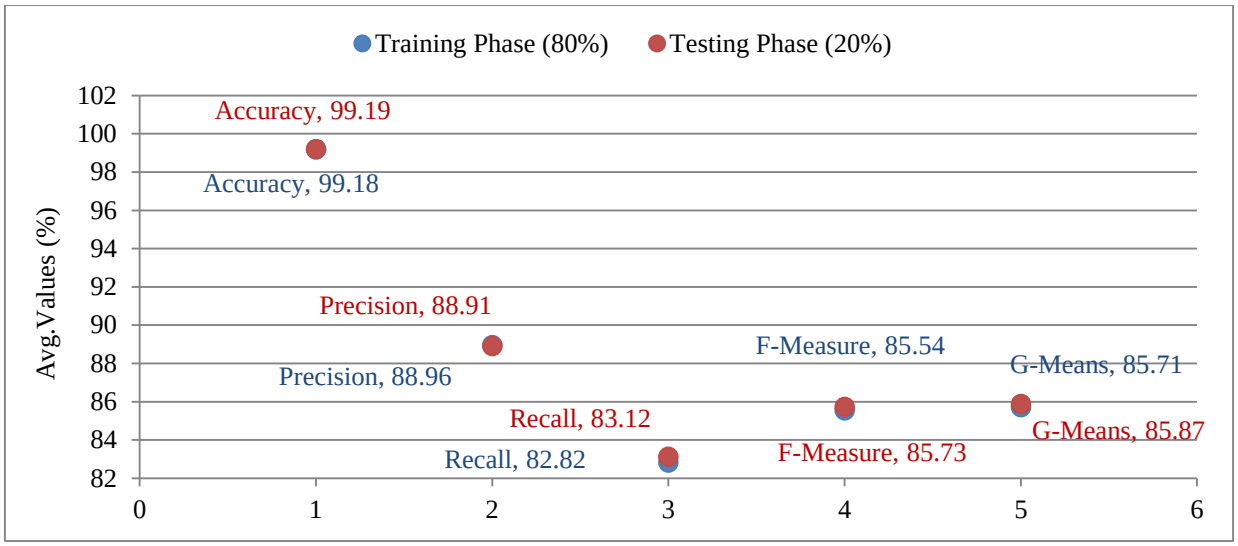


Fig. 5 Average values of the HOFLCD-ELM model under 80:20

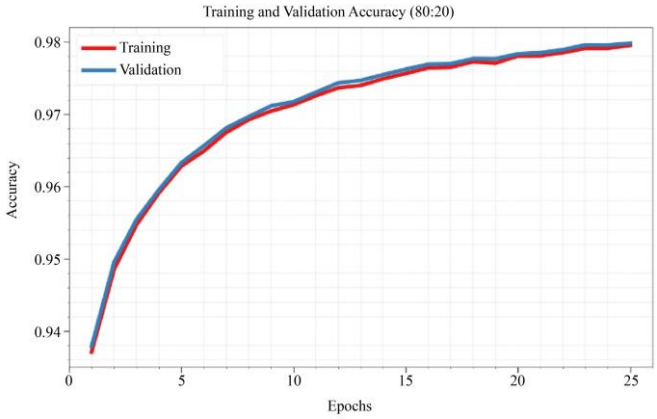


Fig. 6 $Accu_y$ curve of HOFLCD-ELM model under 80:20

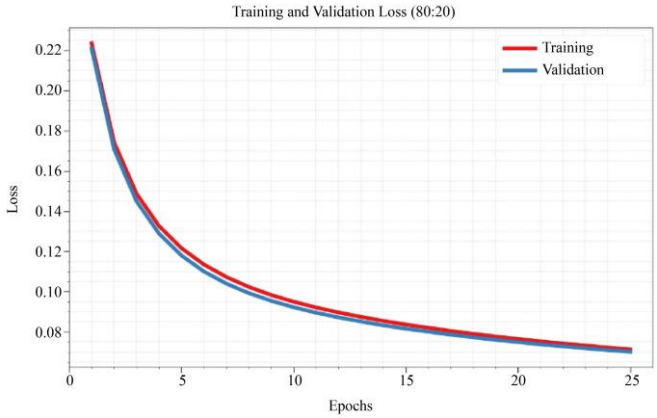


Fig. 7 Loss curve of HOFLCD-ELM model under 80:20

Figure 6 illustrates the Training (TRAIN) Accuracy ($accu_y$) and Validation (VALID) Accuracy ($accu_y$) of the HOFLCD-ELM approach using an 80:20 split over 25 epochs. Initially, both TRAIN and VALID accuracy exhibit rapid improvement, indicating that the data effectively captures patterns. This point in time demonstrates that successful generalization without overfitting, but only slightly above the training accuracy, shows that the VALID accuracy has been achieved. It shows maximum performance and minimal difference in performance between TRAIN and VALID with increasing training. This is regularized and generalized successfully when the two curves come close to each other in the process of training. This illustrates that the method has the best ability to identify and preserve positive attributes in visible and invisible data.

Figure 7 depicts the training and validation losses of the HOFLCD-ELM model, which was trained with an 80:20 split across 25 epochs. The model's initial input is constrained due to elevated TRAIN and VALID losses. The two losses progressively decrease with enhanced training, signifying that the model is successfully learning and refining its parameters. The model has not been overfitted and continues to generalize efficiently to fresh data, evidenced by the near-parallelism of the TRAIN and VALID loss curves during training.

Figure 8 shows the outcome of the classifier at 70:30 of the HOFLCD-ELM method. The Figures 8(a), and 8(b) show the confusion with the correct detection and classification of every class. The PR inspection, which provides the highest performance in each class, is provided in Figure 8(c). Finally,

the analysis of ROC is described in Figure 8(d), and effective outcomes were noted when the ROC values are higher, in the case of different classes.

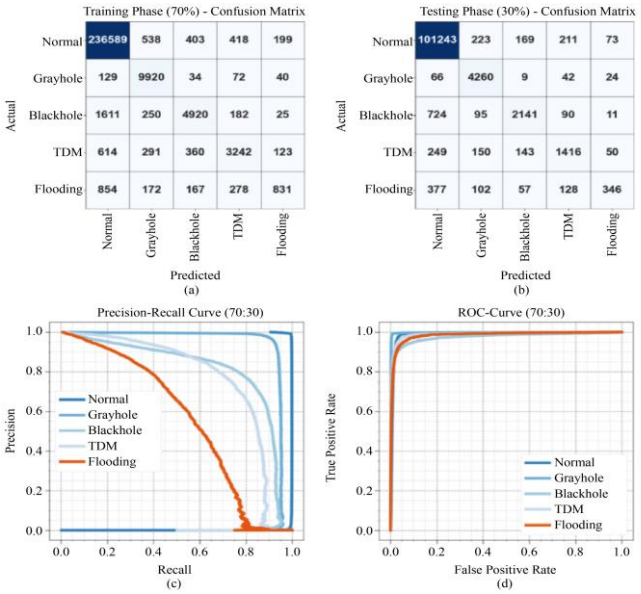


Fig. 8 70:30 of (a, b) Confusion matrices and, (c, d) Curves of PR and ROC.

Table 3 and Figure 9 show the HOFLCD-ELM system's clone attack detection at 70:30. Under 70% TRPHE, the proposed HOFLCD-ELM model gets an average $accu_y$ of 98.97%, $prec_n$ of 83.33%, $reca_l$ of 74.64%, $F_{Measure}$ of 77.80%, and G_{Means} Of 78.38%. Similarly, at 30% TSPHE, the proposed HOFLCD-ELM model obtains average $accu_y$ of 98.93%, $prec_n$ of 83.10%, $reca_l$ of 74.17%, $F_{Measure}$ of 77.29%, and G_{Means} Of 77.94%.

Table 3. Clone attack detection of HOFLCD-ELM model under 70:30

Class	Accuracy	Precision	Recall	F-Measure	G-Means
TRPHE (70%)					
Normal	98.18	98.66	99.35	99.00	99.00
Grayhole	99.42	88.80	97.30	92.86	92.95
Blackhole	98.84	83.62	70.41	76.44	76.73
TDM	99.11	77.34	70.02	73.50	73.59
Flooding	99.29	68.23	36.10	47.22	49.63
Average	**98.97**	**83.33**	**74.64**	**77.80**	**78.38**
TSPHE (30%)					
Normal	98.14	98.62	99.34	98.98	98.98
Grayhole	99.37	88.20	96.80	92.30	92.40
Blackhole	98.85	84.99	69.94	76.74	77.10
TDM	99.05	75.04	70.52	72.71	72.74
Flooding	99.27	68.65	34.26	45.71	48.50
Average	**98.93**	**83.10**	**74.17**	**77.29**	**77.94**

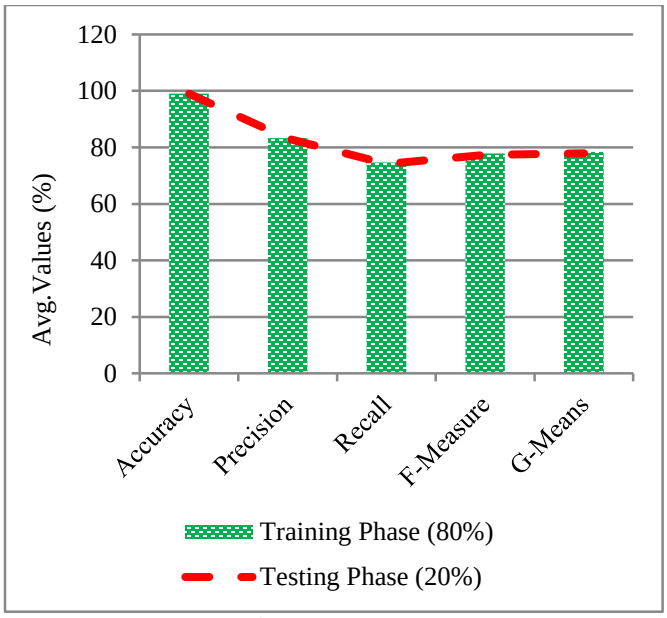


Fig. 9 Average values of the HOFLCD-ELM model under 70:30

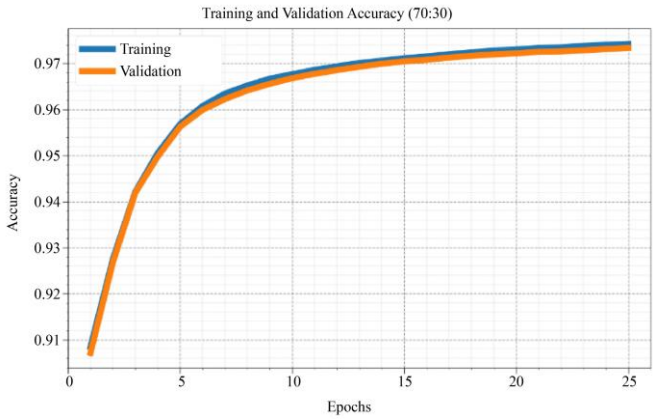


Fig. 10 $Accu_y$ curve of the HOFLCD-ELM model under 70:30

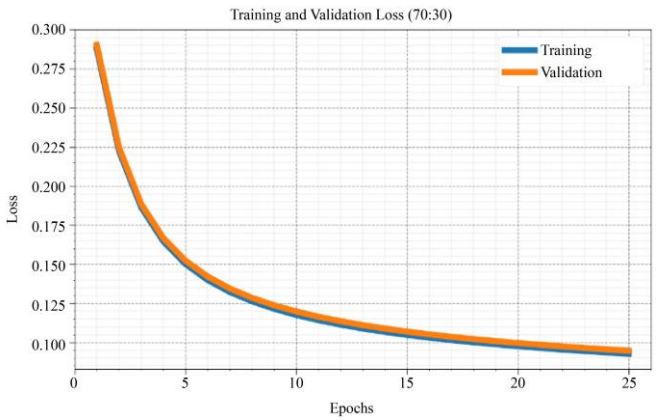
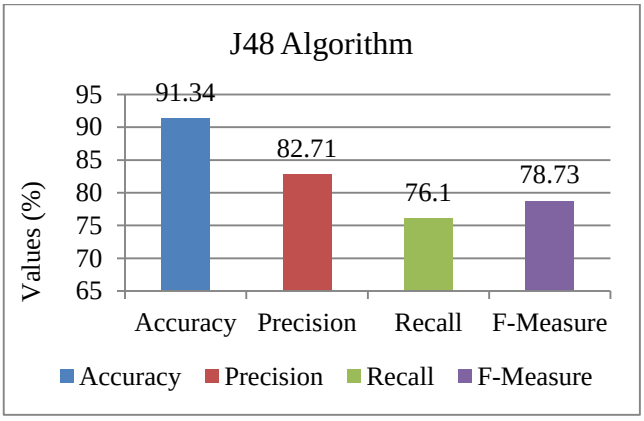


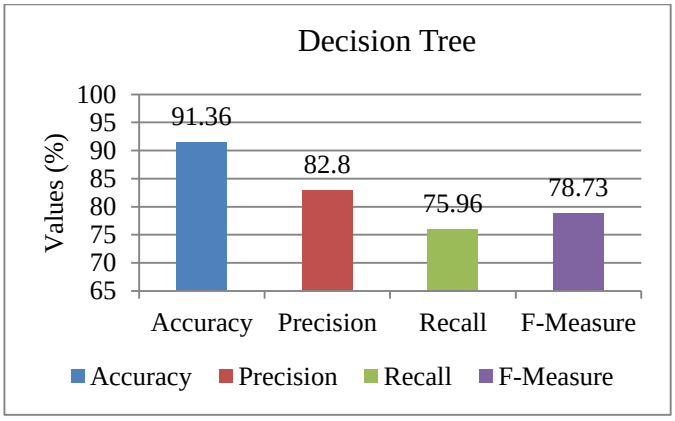
Fig. 11 Loss curve of HOFLCD-ELM model under 70:30

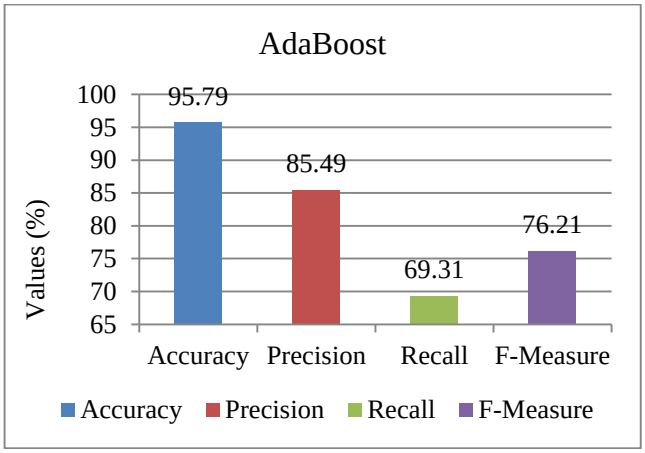
Figure 10 exemplifies the TRAIN $accu_y$ and VALID $accu_y$ Of an HOFLCD-ELM technique across 25 epochs at 70:30. Despite this, at the beginning, the trend is quite clear

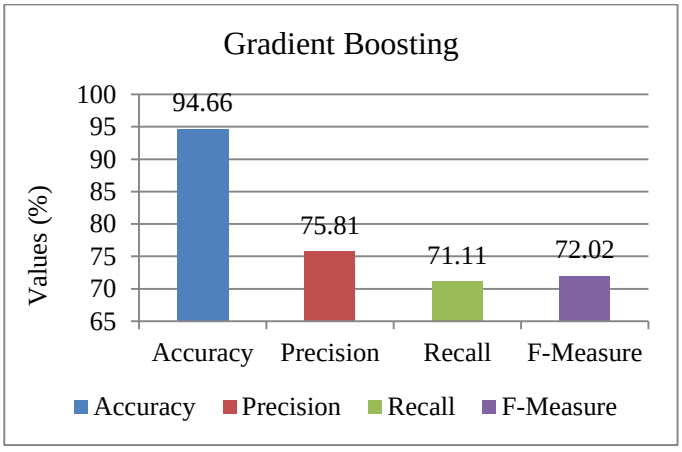
since both TRAIN and VALID accuracy are gaining momentum. This is a good start, since effective generalization with no overfitting has occurred, as the validation accuracy is just a little higher than the training accuracy. It demonstrates optimal performance in accordance with progression in training, with minimal disparity in performance between the TRAIN and Vald performances. The method is well regularized and generalized, as it can be seen that both curves are closely similar over the training process. This shows the extent to which the technique is able to remove and preserve crucial properties in both familiar and unfamiliar material.

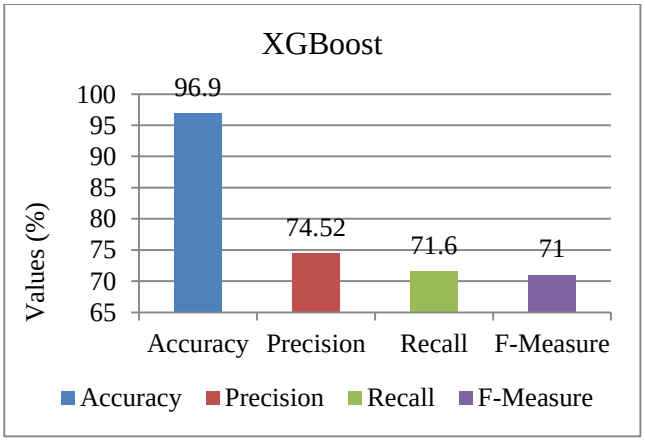
Figure 11 depicts the training and validation loss of the HOFLCD-ELM model, segmented at 25 epochs at a 70:30 ratio. The increased TRAIN and VALID losses suggest that the model's understanding of the input is limited. The two losses gradually diminish with the commencement of training, signifying that the model is effectively acquiring knowledge and refining its parameters. The model was not overfitted and was well generalised to new data by the fact that the TRAIN and VALID curves of loss were strongly related throughout training.

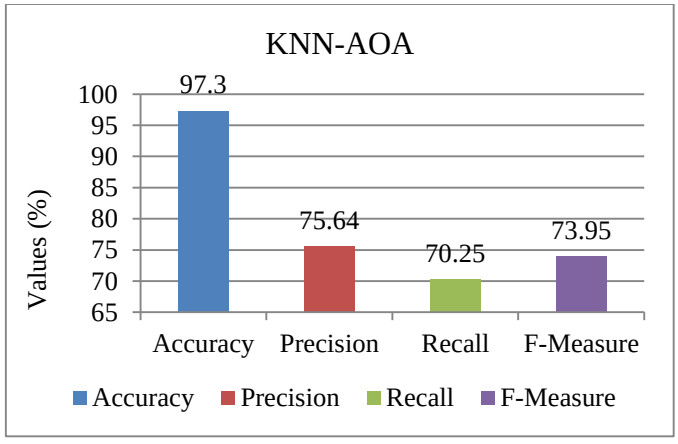












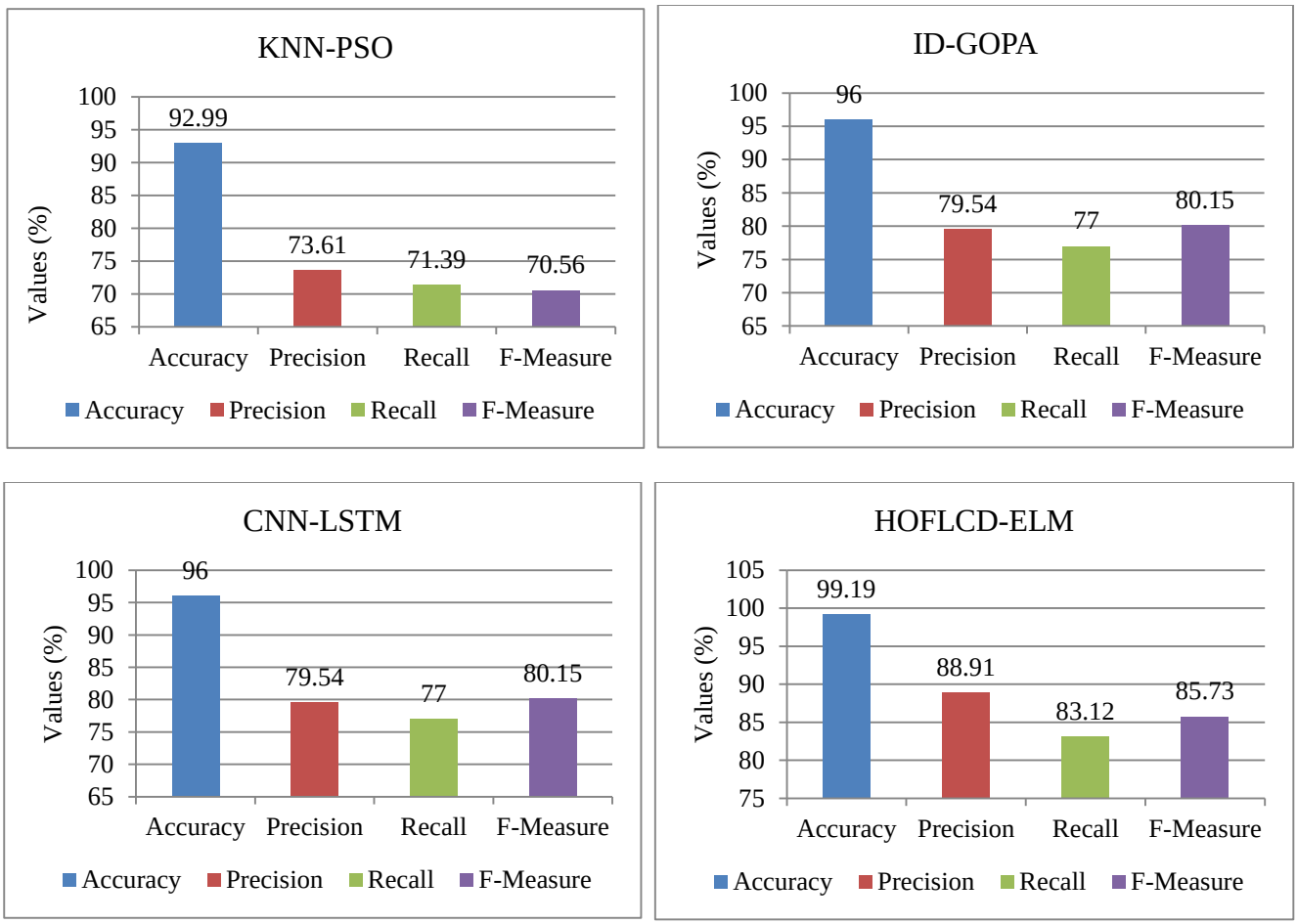


Fig. 12 Comparative analysis of HOFLCD-ELM with existing methods

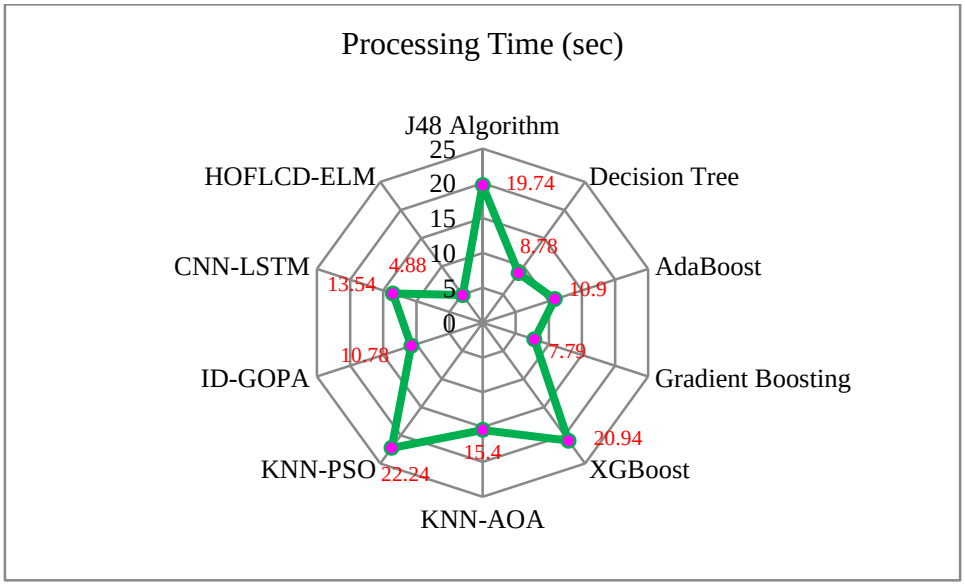


Fig. 13 PT outcome of HOFLCD-ELM with recent methods

Figure 12 illustrates the comparative examination of the HOFLCD-ELM methodology against existing techniques across multiple parameters [24, 25]. The results emphasized that the proposed HOFLCD-ELM model achieved the highest $accu_y, prec_n$, $reca_l$, and $F_{Measure}$ of 99.19%, 88.91%, 83.12%, and 85.73%, respectively. The present

methodologies, such as J48, Decision Tree, AdaBoost, Gradient Boosting, XGBoost, KNN-AOA, KNN-PSO, ID-GOPA, and CNN-LSTM, have shown worse performance under various metrics.

In Figure 13, the Processing Time (PT) of the HOFLCD-ELM method with current models is proven. Based on PT, the HOFLCD-ELM model offers a lower value of 4.88sec while the J48, Decision Tree, AdaBoost, Gradient Boosting, XGBoost, KNN-AOA, KNN-PSO, ID-GOPA, and CNN-LSTM methodologies got higher PT of 19.74sec, 8.78sec, 10.90sec, 7.79sec, 20.94sec, 15.40sec, 22.24sec, 10.78sec, and 13.54sec, respectively.

5. Conclusion

The study constructs and tests an effective clone detection model in WSNs to improve network security and integrity. The min-max normalization method is first used at the preprocessing stage of data to make raw data in order to be used in modeling. In this case, in the process of feature subset selection, the presented HOFLCD-ELM model develops the hybrid optimization model LYBA, incorporating the use of LOA and BA to decide on the selection of the optimal features in a dataset.

Subsequently, one DBN model coupled with the CVAE technique and GCN system to identify and label clone attacks has been implemented. Lastly, there is the parameter tuning method, where a better classification performance of ensemble classifiers is attained using the SWO model. The experimental analysis of the HOFLCD-ELM model is conducted based on a benchmark dataset. The empirical results showed the increased performance of the HOFLCD-ELM method as compared to the current methods.

References

[1] Muhammad Numan et al., "A Systematic Review on Clone Node Detection in Static Wireless Sensor Networks," *IEEE Access*, vol. 8, pp. 65450-65461, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Sachin Lalar, Shashi Bhushan, and Surender, "An Efficient Tree-based Clone Detection Scheme in Wireless Sensor Network," *Journal of Information and Optimization Sciences*, vol. 40, no. 5, pp. 1003-1023, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Harshita Patel et al., "A Review on Classification of Imbalanced Data for Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 4, pp. 1-15, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Periasamy Nancy et al., "Intrusion Detection using Dynamic Feature Selection and Fuzzy Temporal Decision Tree Classification for Wireless Sensor Networks," *IET Communications*, vol. 14, no. 5, pp. 888-895, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Amit Kumar Gautam, and Rakesh Kumar, "A Comprehensive Study on Key Management, Authentication and Trust Management Techniques in Wireless Sensor Networks," *SN Applied Sciences*, vol. 3, pp. 1-27, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Rahul Priyadarshi, Bharat Gupta, and Amulya Anurag, "Wireless Sensor Networks Deployment: A Result Oriented Analysis," *Wireless Personal Communications*, vol. 113, pp. 843-866, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Christian Miranda et al., "A Collaborative Security Framework for Software-Defined Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2602-2615, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mukaram Safaldin, Mohammed Otair, and Laith Abualigah, "Improved Binary Gray Wolf Optimizer and SVM for Intrusion Detection System in Wireless Sensor Networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1559-1576, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Rabie A. Ramadan, "An Improved Group Teaching Optimization based Localization Scheme for WSN," *International Journal of Wireless and Ad Hoc Communication*, vol. 3, no. 1, pp. 08-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mona Nashaat et al., "An Enhanced Transformer-Based Framework for Interpretable Code Clone Detection," *Journal of Systems and Software*, vol. 222, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jean Rosemond Dora, and Karol Nemoga, "Clone Node Detection Attacks and Mitigation Mechanisms in Static Wireless Sensor Networks," *Journal of Cybersecurity and Privacy*, vol. 1, no. 4, pp. 553-579, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zeina Swilam, Abeer Hamdy, and Andreas Pester, "Advanced Cross-Language Clone Detection Using Modified AST and Graph Neural Network," *2024 International Conference on Computer and Applications (ICCA)*, Cairo, Egypt, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Seelam Ch Vijaya Bhaskar et al., "Augmenting Cybersecurity in WSN: AI-Based Clone Attacks Recognition Framework," *2024 Asian Conference on Communication and Networks (ASIANComNet)*, Bangkok, Thailand, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] K. Jane 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)," *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, Raigarh, India, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ramesh Vatambeti et al., "Classification of HHO-based Machine Learning Techniques for Clone Attack Detection in WSN," *International Journal of Computer Network and Information Security*, vol. 15, no. 6, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Bhuvana et al., "Relative Spectral Feature Analysis-Based Clone Attack Detection and Enhance Routing in Wireless Sensor Networks Using Artificial Neural Networks," *Journal of Data Acquisition and Processing*, vol. 38, no. 3, pp. 1770-1791, 2023. [Google Scholar]

[17] Hadeel M. Saleh, Hend Marouane, and Ahmed Fakhfakh, "Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning," *IEEE Access*, vol. 12, pp. 3825-3836, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] N.S. Manohar Raji, "IDLRN-DBN: Segmentation-based Early Diagnosis of Rice Plant Disease Detection using Deep Belief Network," *KSII Transactions on Internet and Information Systems*, vol. 19, no. 5, pp. 1539-1563, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mostafa Mohammadpourfard, Chenhan Xiao, and Yang Weng, "Performance Guaranteed Deep Learning for Detection of Cyber-Attacks in Dynamic Smart Grids," *IEEE Transactions on Power Systems*, vol. 40, no. 6, pp. 4608-4620, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] D. Palumbo et al., "Damage Diagnostic Method by Artificial Intelligence Analysis of Shaking Table Data of a Typical Italian Building Prototype," *Journal of Instrumentation*, vol. 20, pp. 1-8, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Muhammed Ali Pala, "CNS-DDI: An Integrated Graph Neural Network Framework for Predicting Central Nervous System Related Drug-Drug Interactions," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 14, no. 2, pp. 907-929, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Sana Qaiyum et al., "Benchmarking Reinforcement Learning and Accurate Modeling of Ground Source Heat Pump Systems: Intelligent Strategy using Spiking Recurrent Neural Network Combined with Spider WASP Inspired Optimization Algorithm," *Results in Engineering*, vol. 27, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks, Kaggle. [Online]. Available: https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds

[24] Hajar Fares, "Intrusion Detection in Wireless Sensor Networks using Machine Learning," *Procedia Computer Science*, vol. 252, pp. 912-921, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Amal K. Alkhalifa et al., "Hybrid Dung Beetle Optimization based Dimensionality Reduction with Deep Learning based Cybersecurity Solution on IoT Environment," *Alexandria Engineering Journal*, vol. 111, pp. 148-159, 2025. [CrossRef] [Google Scholar] [Publisher Link]