

An Intelligent Clone Detection Framework in Wireless Sensor Networks Using Hybrid Multi-Objective Optimization with Random Vector Functional Neural Network Architecture

P.Kalvikkarasi^{1,*}, Dr K.Selvakumar², Dr R.Padmaraj³

¹Research Scholar, Department of Information Technology, FEAT, Annamalai University, Chidambaram, Tamilnadu.

²Professor, Department of Information Technology, FEAT, Annamalai University, Chidambaram, Tamilnadu.

³ Assistant Professor, Department of Information Technology, FEAT, Annamalai University, Chidambaram, Tamilnadu.

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

HOW TO CITE:

P.Kalvikkarasi, K.Selvakumar
R.Padmaraj (2026). An Intelligent
Clone Detection Framework in
Wireless Sensor Networks Using
Hybrid Multi-Objective
Optimization with Random Vector
Functional Neural Network
Architecture
International Journal of Special
Education, 41(2), 1050 - 1070

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the
terms and conditions
of the Creative Commons Attribution
(CC BY)
license (<http://creativecommons.org/licenses/by/4.0/>).

ABSTRACT:

The progress of wireless sensor networks technology normally called WSNs, which was obtaining a considerably enlarged volume of attention from scholars for the past few years. Its vast amount of sensor nodes (SN) is the main feature that creates it advantageous to the technology. The SNs are normally employed for varied applications. In spite of this, the disadvantages of WSNs are frequently used in hostile and critical situations but they don't contain physical access. This realism makes them very susceptible to node replication or clone node threats. In recent times, deep learning (DL) has become a great tool for detecting clone attacks in WSNs, which are exposed to security attacks. This study presents a Randomization-Based Neural Network Architecture for Intelligent Clone Detection Using Hybrid Multi-objective Optimization (RNNICD-HMO) approach. The proposed RNNICD-HMO approach is to accurately detect cloned SNs and prevent security breaches in WSNs. Initially, data preprocessing is performed data cleaning and normalization, to ensure high-quality and consistent input for analysis. Next, the salp swarm algorithm (SSA) has been applied for feature selection to identify the most informative attributes. Subsequently, a random vector functional link (RVFL) network was utilized for efficient and reliable classification of cloned and legitimate nodes. To further enhance detection accuracy, a hybrid multi-objective ant colony system and simulated annealing approach (MACSSA) is applied for optimal parameter tuning of the classification model. The experimental study of the RNNICD-HMO system takes place utilizing dissimilar benchmark databases. The simulation outcomes stated an enhanced outcome of the RNNICD-HMO algorithm in terms of various measures.

Keywords: Wireless Sensor Networks; Clone Attacks; Salp Swarm Algorithm; Multi-objective Ant Colony System; Random Vector Functional Link

1. Introduction

Wireless Sensor Networks (WSNs) are a group of sensors with scarce resources that cooperate to attain a general aim. WSNs may be used in severe environments to accomplish either civil or military applications [1]. Fig. 1 represents the

general diagram of WSN. Because of their operating nature, they are frequently unattended, therefore disposed to various types of new threats. For example, an adversary can eavesdrop on each network interconnection; an adversary can take nodes, obtaining each data saved in those sensors are usually not expected to be

tamper-proof [2]. However, an adversary could replicate taken sensors and use them in the network for launching the type of malicious behaviour [3]. This threat is also known as the clone attack. Because a clone has valid data (cryptographic material as well as code) that can contribute to the network operations similarly to a non-compromised node, and so cloned nodes may launch an attack type [4]. These cloned nodes are then deliberately used in the network to perform malicious actions, like eavesdropping, denial-of-service (DoS) attacks, or data injection. Therefore, it is important to identify clone nodes quickly to minimize their damage to WSNs [5]. The easiest defensive measures against the clone threats are for protecting an adversary from obtaining confidential main materials from taken nodes by the feature of tamper-resistant hardware [6]. Hence, the hardware-driven defensive measures are too costly to be practical for a resource-limited sensor node (SN). Different types of software-driven clone identification systems have currently been presented for WSNs across multiple diverse kinds of network configurations, like device categories and implementation approaches [7]. The restriction of software-driven clone identification systems is certainly that they are not generic, sense that their effectiveness as well as efficiency can be based on their preconfigured network environments. So, an automated clone attack identification in WSN is critical [8].

Deep learning (DL) is an effective tool for identifying clone attacks in WSNs that are susceptible to security risks. DL offers an assuring alternative to address the restrictions of conventional methods by allowing adaptive

intrusion detection and being smarter [9]. Recurrent Neural Network (RNN), Deep Neural Network (DNN), and other DL techniques may examine intricate, higher-dimension network traffic information to detect irregular patterns suggestive of a clone attack [10]. This technique may study advanced attack activities without being based on predefined signatures or regulations, creation them efficient against new and difficult attacks. Through processing information in hidden layers, DL methods may obtain and choose key characteristics that signify evidence of a threat, enhancing detection precision and decreasing false alarms. Incorporating DL into WSN security structure enables the growth of more strong, scalable, and energy-effective clone identification methods [11].

This paper introduces a Randomization-Based Neural Network Architecture for Intelligent Clone Detection Using Hybrid Multi-objective Optimization (RNNICD-HMO) approach. The key contributions include the integration of data preprocessing to confirm that the input is of higher quality, the application of the salp swarm algorithm (SSA) for selecting the most beneficial attributes, and the use of a random vector functional link (RVFL) for accurate classification of cloned and genuine nodes. Moreover, the model integrates a hybrid multi-objective ant colony system and simulated annealing (MACSSA) algorithm for optimal parameter tuning, enhancing detection performance. Extensive experimentation on multiple benchmark datasets reveals the excellent performance of RNNICD-HMO over existing techniques across various evaluation measures.

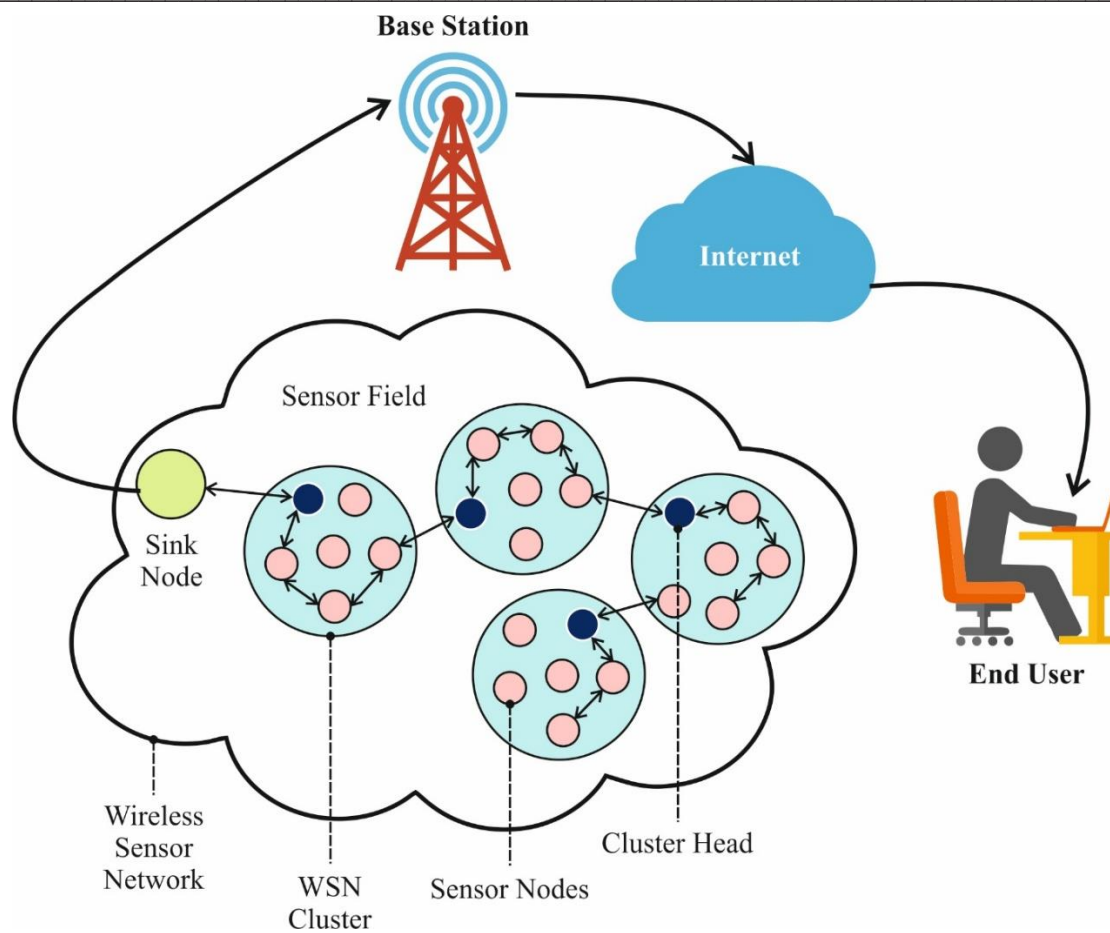


Fig. 1. General diagram of WSN

2. Review of Literature Works

This section briefly reviews prior related works on clone detection in WSN. Nashaat et al. [12] examined CloneXformer, a novel structure for code clone identification. The structure adopts a cooperative method that tackles many LLMs for code comprehension. The structure uses a preliminary stage for preprocessing the input code that assists the techniques in understanding and representing the code effectively. Therefore, it takes the syntactic level and the semantic level of the code, as it is based on a group of transformer-driven methods. After this technique is fine-grained for detecting code clones with explainable outcomes that explain the identified clone categories. Nithya and Shyamala [13] intended to obtain prolonged network lifetime as well as load balancing. The research makes an energy-effective sleep-aware protocol that enhances energy effectiveness, as well as selects a suitable CH depending on node energy. Dragonfly optimizer methods boost network

efficiency. A DL cloning node detector has been presented for detecting WSN clones. Clone detection is important to prevent clone attacks. In [14], a new authentication system is introduced to solve node cloning attacks, using a coordinate plane to represent geographical places. The method also takes extra functionalities, efficiently handling node revocations and additions while integrating collusion threat identification. The cloning node has similar data to the cloned one, and may act as if the cloning one for achieving the synchronized key, resulting in the leak of group interaction information.

Yahya and Kim [15] proposed CLCD-I, a DNN-driven method to detect cross-language code clones by employing InferCode. The intention of the techniques is dual: Initially, advancing them to a Siamese structure for the comparison process, and then capturing as input InferCode embedded in source code in various program languages. Therefore, software is progressively evolving in a multi-language setting in which code

is reprocessed through various programming languages. Zou et al. [16] suggested a scalable semantic code clone detection that depends on a semantically improved AST. Therefore, construct a social network-driven triads method for gathering the likeness characteristics among the dual approach by analyzing various kinds of triads within the network. After attaining each feature, utilize them for training an ML-driven code clone detection.

Zhang et al. [17] presented K-ASTRO, a lightweight Transformer technique that integrates semantic embeddings from LLMs with the structural characteristics of Abstract Syntax Trees (ASTs) for enhancing either effectiveness or precision in code susceptibility identification. The method proposes an AST-driven augmentation model stimulated by mutation testing, a structure-aware attention module that

integrates augmented AST characteristics, and a joint adaptation pipeline for unifying code syntax and semantics. Therefore, existing approaches are frequently based on resource-intensive techniques or graph-driven methods, restricting their availability and usefulness. Quradaa et al. [18] examined an innovative method for detecting semantic clones by utilizing the capacities of the Siamese bi-directional LSTM methods for learning semantic patterns as well as hidden syntactic in source code representations.

3. Methodological Framework

This manuscript presents the RNNICD-HMO model, a novel approach for accurate and intelligent clone detection in WSNs. Fig. 2 demonstrates the workflow of the presented RNNICD-HMO methodology. As depicted in Fig. 2, the RNNICD-HMO methodology proceeds through a sequence of phases.

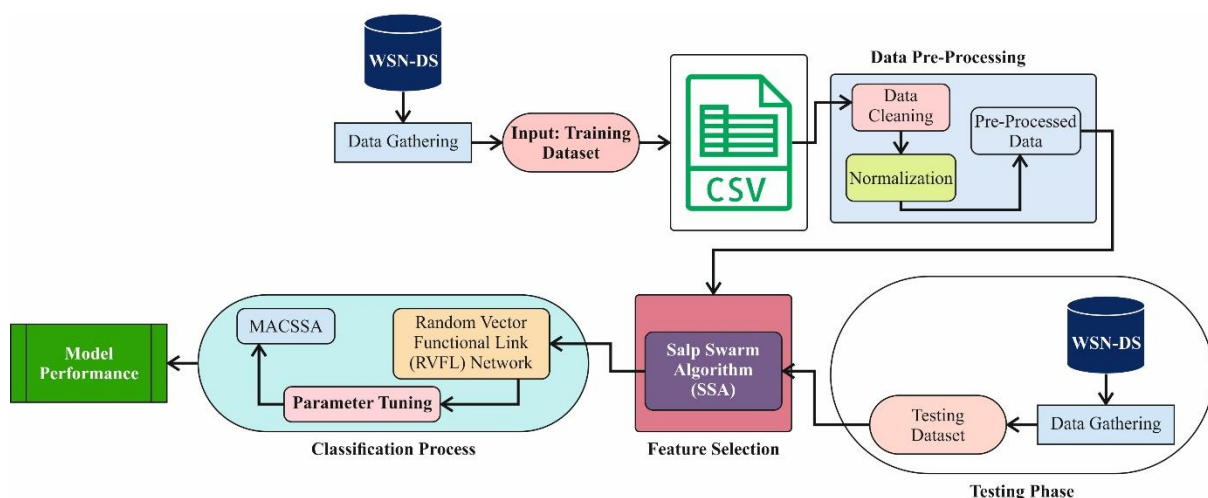


Fig. 2. Workflow of RNNICD-HMO approach

- Data cleaning and normalization are performed to ensure high-quality inputs for analysis.
- SSA is used to identify the most informative and relevant features.
- The RVFL method is utilized to accurately classify cloned and legitimate nodes.
- The hybrid MACSSA technique optimizes the classifier parameters to enhance detection accuracy.

3.1. Data Preprocessing

Data preprocessing approaches, including data cleaning and normalization, are applied to guarantee high-quality and consistent input for analysis. Data pre-processing is an essential step for ensuring the dependability and accuracy. It comprises various data-cleaning models to tackle errors and inconsistency [19]. Pre-processing also aids in eliminating noise, normalizing attributes, and handling missing values, enhancing the outcome of method and capability of generalizing unseen data. Preprocessing follows succeeding process.

- For data instances somewhere any

feature rate was missing, eliminate those instances.

- For data instances somewhere any feature rate is eradicated, those instances are eliminated.

For feature transformation, utilize the succeeding Eq. (1):

$$X \leftarrow \frac{\log_{10}(X + 1)}{10} \quad (1)$$

After transformation, every value of the features is scaled in the range [0,1]. Once the data is processed, instantly employ them for evaluation and training.

3.2. Proposed Salp Swarm Algorithm

The SSA is employed to select the most informative attributes contributing to precise detection of cloned nodes. SSA is a meta-heuristic optimizer method exhibited afterward the foraging and swarming behavior of salps, the foremost stages are from the leader and follower [20]. Generally, they are dual main stages of SSA, such as Population Initial and Position Update.

Population Initial Stage: The optimizer issue is considered to contain a D -dimensional space of search, and an initial locations of salp population were expressed as follows:

$$X_{i,j} = rand_{i,j} \cdot (UpB_{i,j} - LowB_{i,j}) + LowB_{i,j} \quad (2)$$

whereas $X_{i,j}$ means an initial location of slap, $i = 1, 2, \dots, NP$, $j = 1, 2, \dots, D$. NP denotes the no. of primary performances and D represents a dimensional. $rand_{i,j}$ denotes a randomly generated value in the interval of zero and one. $LowB_{i,j}$ and $UpB_{i,j}$ specify the lower and upper boundary values, respectively.

Position Update Stage: In an individual search procedure, the food resource F was handled as the target, which is an optimum intention. In the searching space, the location was upgraded as follows:

$$X_j^{leader} = \begin{cases} F_j + c_1 \cdot ((UpB_j - LowB_j)c_2 + LowB_j), & \text{if } c_3 > 0.5 \\ F_j - c_1 \cdot ((UpB_j - LowB_j)c_2 + LowB_j), & \text{if } c_3 \leq 0.5 \end{cases} \quad (3)$$

whereas X_j^{leader} indicates the leader position in j th dimension. F_j specifies food resource. c_1 , c_2 and c_3 mean a arbitrarily produced value in the range of $(0,1)$ as per to Gaussian law.

Afterward, the exploitation and exploration of follower balances by c_1 parameter, which was expressed below:

$$c_1 = 2 \cdot \exp - \left(\left(\frac{4 \cdot l}{T_{\max}} \right)^2 \right) \quad (4)$$

Whereas, l means the present iteration. $T_{\max}$ specifies the maximal no. of iteration. Particularly, the follower's location was upgraded by:

$$X_j^i = \frac{1}{2} (X_j^i + X_j^{i-1}) \quad (5)$$

Here, X_j^i means the location of i th follower from the dimension j if $i \geq 2$. In the proposed model, the fitness function (FF) is mainly aimed at holding a balance among the amount of selected attributes in every result (smallest) and classifier accuracy (highest) accomplished by deploying these chosen attributes. Eq. (6) indicates the FF for assessing the solution.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (6)$$

whereas $\gamma_R(D)$ indicates the classifier rate of error. $|R|$ represents a cardinality of preferred sub-set and $|C|$ signifies the total attributes count, α and β represent dual parameters that correspond to the significance of classifier quality and sub-set length. $\beta = 1 - \alpha$ and $\in [1,0]$

3.3. Clone Detection using RVFL

An RVFL network is utilized to effectively classify cloned as well as legitimate nodes. RVFL contains input and output layers as well as one hidden layer (HL) [21]. The layer of input and HL are linked to the layer of output through the weights of output layer. The weighted output layer is assessed to utilize analytical models, namely the pseudo-inverse or least-squares model.

By considering G as the matrix of HL, gained by arbitrarily projecting the matrix of input and implementing the non-linear activation function γ .

$$G = \gamma(VX + 1^t \otimes \zeta) \in \mathbb{R}^{M \times g} \quad (7)$$

Here $X \in \mathbb{R}^{P \times g}$ signifies arbitrarily initialized matrix weight, 1 is a vector of ones, and $\zeta \in \mathbb{R}^{1 \times g}$ indicates biased vector. Thus, G is specified:

$$G = \begin{pmatrix} \gamma(v_1 x_1 + \zeta^{(1)}) & \cdots & \gamma(v_1 x_g + \zeta^{(g)}) \\ \vdots & \vdots & \vdots \\ \gamma(v_M x_1 + \zeta^{(1)}) & \cdots & \gamma(v_M x_g + \zeta^{(g)}) \end{pmatrix}$$

Here $\zeta^{(j)}$ depicts j^{th} biased term of hidden node, $x_k \in \mathbb{R}^{P \times 1}$ signifies k^{th} column vector of weight attaching the k^{th} node of HL to every input node and $v_i \in \mathbb{R}^{1 \times P}$ denotes i^{th} row of inputs matrix V . The weight of resultant layer is evaluated.

$$[V \oplus G]O = \hat{Z} \quad (8)$$

Now $O \in \mathbb{R}^{(P+g) \times C}$ represents unknown weighted matrix that connects layer of resultant layer with HL and input layer. $\hat{Z}$ is predictable outcome. Q is evaluated:

$$Q = [V \oplus G]^{-1} Z \quad (9)$$

or,

$$Q = [V \oplus G]^{\dagger} Z \quad (10)$$

Here $[\cdot]^{-1}$ and $[\cdot]^{\dagger}$ signifies the inverse and pseudo-inverse, correspondingly. If $[V \oplus G]$ is non-singular, employ it to assess Q , then.

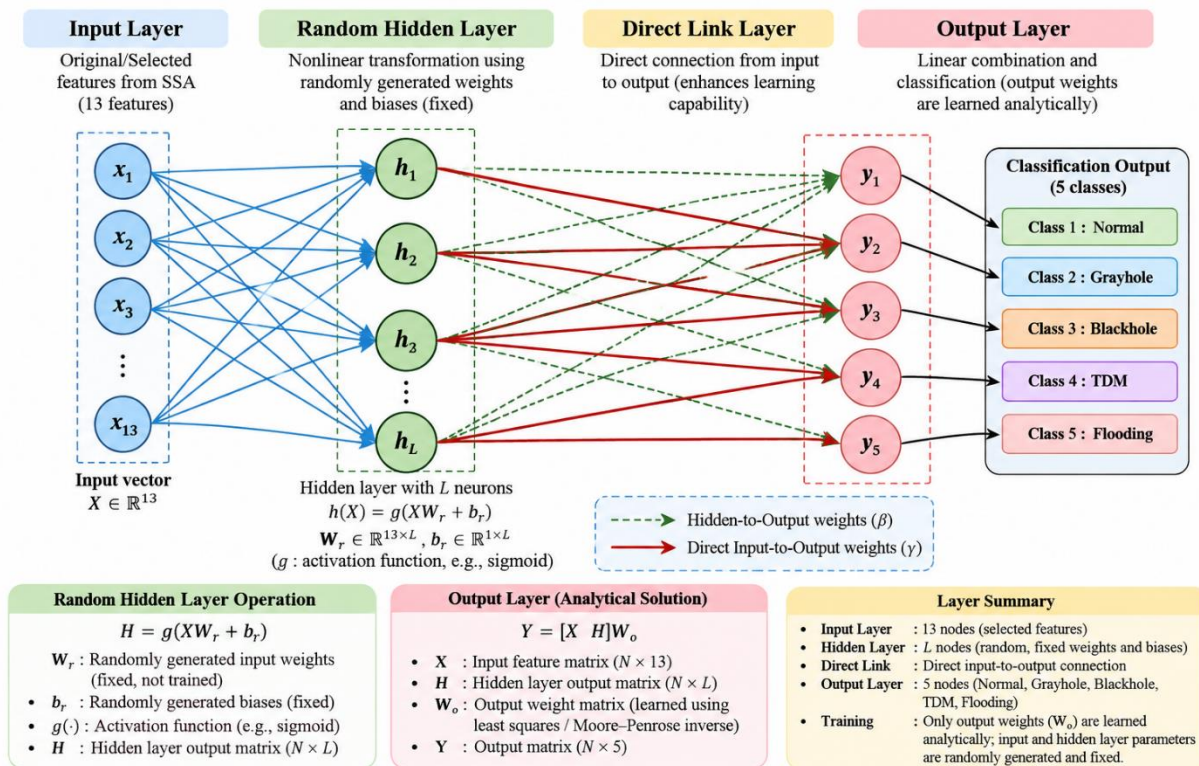


Fig. 3. Architecture of the Random Vector Functional Link (RVFL) Classifier for WSN Attack Detection

The design of inverse is a substantial challenge, specifically to entail huge matrices. It is computationally difficult and turns into a bottleneck for the RVFL-based model while managing high-order matrices. Eventually, the utilization of memory, complexity, impacts of performance, and extensibility connected to discovering the inverse of matrices are delay the outcome of RVFL-based approaches in real-time situations depicted by high dimensional data and an enormous sample size of samples.

3.4. Hybrid MACSSA-based Parameter Optimizer

A hybrid MACSSA methodology is leveraged for the optimal parameter fine-tuned of the classification model, leading to enhanced detection capabilities. Eventually, the parameter choice of the ensemble DL classifiers can be employed by the design of the hybrid MACSSA algorithm [22]. Currently, hybrid meta-heuristic methods have attained traction for utilizing the powers of numerous methods to resolve intricate optimizer problems. This study presents a hybrid

Heuristic Information: The heuristic information can be identified depending on three aspects:

multi-objective ant colony system integrated with the SA method to address the issue. In the ACO method, synthetic ant colonies are combined to address the difficulties optimizer. Ants navigate a system characterized by synthetic pheromones. The nest signifies the early states, and food indicates the concluding states. Every vehicle's path starts and finishes in a repository, equivalent to the food and nest. To inhibit local optima, the level of pheromone progressively disappears. Every ant searches the networks to create solutions.

Pheromone Configuration: This method uses dual separate pheromone trails for dual main functions that are upgraded distinctly after every iteration. The usage of various pheromone trails to tackle several multi-objective difficulties was discovered in numerous types of research, including inadequate region capability layout, reducing overall complete energy and time costs in single-machine preventive planning, and diverse-loaded school buses.

- The cost of traveling among customers i and j ;
- The time of traveling $time_{it}$;
- The quantity demanded $\tilde{d}_{jt}$ of customer j ,

For ant k set at customer i , dual distinct parts of heuristic data equivalent to the dual objective function of the difficulties are described as:

$$\eta_{ijt}^C = \frac{1}{c_{ijt}} \quad (11)$$

$$\eta_{ijt}^S = \frac{1}{time_{it} d_{jt}} \quad (12)$$

Quasi-Random Probability Rule: If ant k is in the position of customer i , the succeeding customer in the region of customer i in period t can be chosen depending on the succeeding rule:

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^{kt}} \left\{ \left[(\tau_{ijt}^C)^\alpha (\eta_{ijt}^C)^\beta \right]^\lambda \left[(\tau_{ijt}^S)^\alpha (\eta_{ijt}^S)^\beta \right]^{1-\lambda} \right\} & N_i^{kt} \neq \emptyset, q \leq q_0 \\ j^* & O.w. \end{cases} \quad (13)$$

When $0 \leq q_0 \leq 1$ represents arbitrary parameters and j^* denotes arbitrary variable:

$$p_{ij}^t = \frac{\left[(\tau_{ijt}^C)^\alpha (\eta_{ijt}^C)^\beta \right]^\lambda \left[(\tau_{ijt}^S)^\alpha (\eta_{ijt}^S)^\beta \right]^{1-\lambda}}{\sum_{i' \in N_i^{kt}} \left[(\tau_{ij't}^C)^\alpha (\eta_{ij't}^C)^\beta \right]^\lambda \left[(\tau_{ij't}^S)^\alpha (\eta_{ij't}^S)^\beta \right]^{1-\lambda}} \quad j \in N_i^{kt} \quad (14)$$

Whereas α , β , and λ are parameters. The λ value can be attained based on the next correlation:

$$\lambda = \begin{cases} 0 & k \leq a \\ \frac{k}{b-a} - \frac{a}{b-a} & a < k < b \\ 1 & k \geq b \end{cases} \quad (15)$$

Whereas a and b are dual parameters. Based on the λ explanation, around ants utilize just data regarding the objective function, whereas others utilize information regarding both main functions.

The decision rule to facilitate customer j after i within period t can be described as:

$$p_{(i,j)}^t = \begin{cases} \frac{\left[(\tau_{ijt}^C)^\lambda (\tau_{ijt}^S)^{1-\lambda} \right]^\alpha [\eta_{ijt}]^\beta}{\sum_{i' \in N_i} \left[(\tau_{ij't}^C)^\lambda (\tau_{ij't}^S)^{1-\lambda} \right]^\alpha [\eta_{ij't}]^\beta} & j \in N_i \\ 0 & O.w. \end{cases} \quad (16)$$

Whereas $\lambda \in (0,1)$ specifies the objective function comparative significance. Moreover, α and β are dual parameters, which specify the ant-related tendency to track the route utilizing the heuristic and pheromone information.

Pheromone Update: The pheromone level of every link can be upgraded over dual mechanisms. The evaporating rule decreases the pheromone of every chosen link based on the succeeding evaporating rates:

$$\tau_{ijt}^C \leftarrow (1 - \xi)\tau_{ijt}^C \quad (17)$$

$$\tau_{ijt}^S \leftarrow (1 - \xi)\tau_{ijt}^S \quad (18)$$

Whereas $0 < \xi < 1$ is a parameter. Afterwards, the choice of customer j after i in period t , its equivalent pheromone trail can be decreased by the ratio of $1 - \xi$, and its appeal can be decreased for following elections.

The pheromone levels on colony Pareto-optimum results links were upgraded in every iteration, as represented by:

$$\tau_{ijt}^C = \min \left\{ 1, \tau_{iji}^C \cdot \rho + \frac{Q}{C} \right\} \quad (19)$$

$$\tau_{ijt}^S = \min \left\{ 1, \tau_{iji}^S \cdot \rho + \frac{Q}{S} \right\} \quad (20)$$

Where C denotes the cost amount of Pareto-optimum solutions and S denotes the amount of $time_{it} * d_{it}$ whereas customer i can be situated in the Pareto-optimum solutions. The MACSSA grows a fitness function (FF) to achieve improved classifier results. It detects a positive integer to suggest the greater outcomes of candidate outcomes. During this work, the decrease in the classifier value of error has been measured as the FF.

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{Misclassified\ sample\ counts}{Total\ sample\ counts} * 100 \end{aligned} \quad (21)$$

4. Results and Discussions

The performance analysis of the RNNICD-HMO approach is examined under WSN-DS database [23]. This dataset holds 37,4661 cases with five attack kinds, like Blackhole, TDM

(Time Division Multiple Access attack), Flooding, Grayhole, and Normal. Table 1 provides complete details of the dataset. The dataset consists of 18 features in total, of which 13 features were selected for the analysis.

Table 1 Dataset details

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

Fig. 4 exhibits the classifier outcome of RNNICD-HMO system under 80%:20%. Figs. 3a, 3b display the confusion matrices with correct detection of every class label. Fig. 3c presents the PR curve, demonstrating better result through every class. However, Fig. 3d shows the ROC analysis, representing effective outcomes with great value of ROC for diverse class labels.

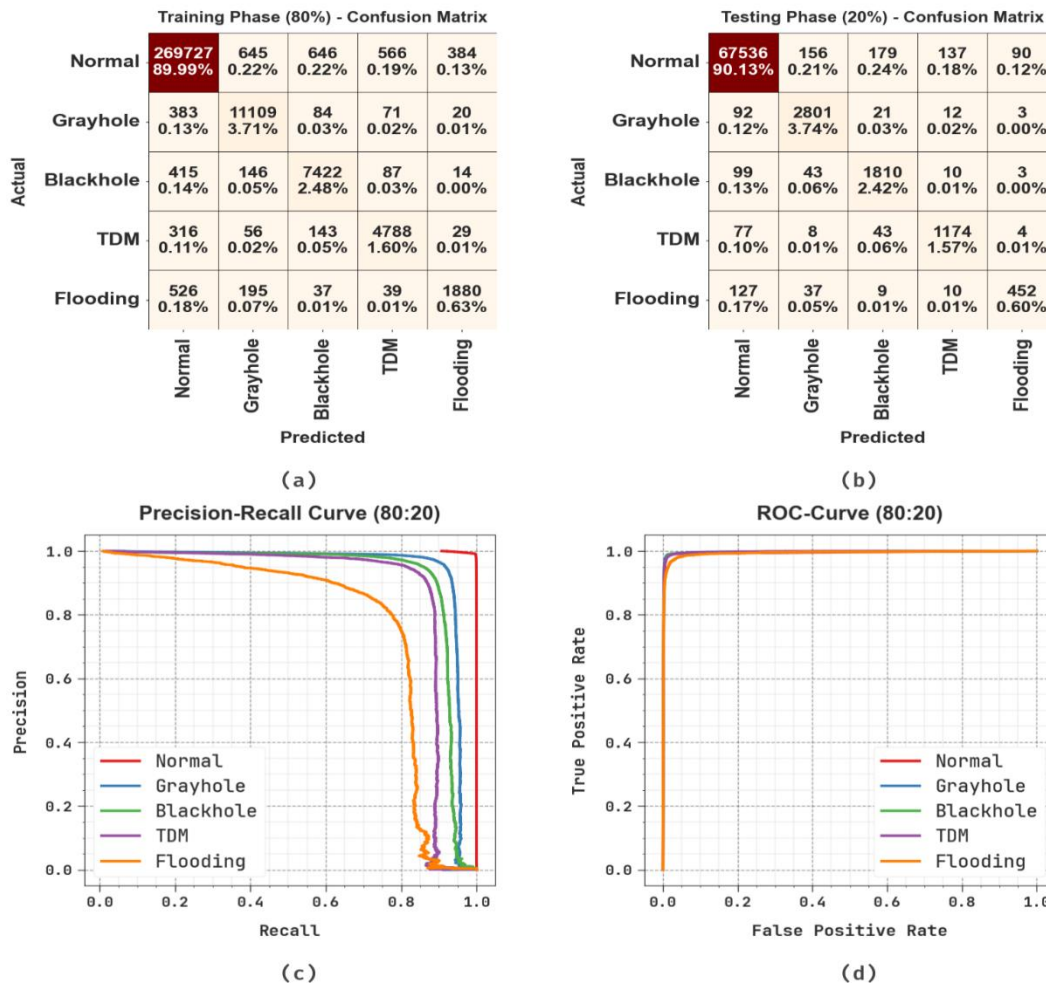


Fig. 4. 80% and 20% of (a, b) confusion matrices and (c, d) Curves of PR and ROC

Table 2 and Fig. 5 describe an attack detection of RNNICD-HMO technique under 80%:20%. On 80% training, the proposed RNNICD-HMO model reaches average $accur_y$, $preci_n$, $recal_l$, $F_{Measure}$, and G_{Means} of 99.36%, 89.39%, 89.25%, 89.22%, and 89.27%, respectively. Also, on 20% testing, the proposed RNNICD-HMO method accomplishes average $accur_y$, $preci_n$, $recal_l$, $F_{Measure}$, and G_{Means} of 99.38%, 89.70%, 89.60%, 89.55%, and 89.60%, respectively.

Table 2 Attack detection of RNNICD-HMO model under 80%:20%

Class	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$	G_{Means}
Training Phase (80%)					
Normal	98.71	99.40	99.18	99.29	99.29
Grayhole	99.47	91.42	95.22	93.28	93.30

Blackhole	99.48	89.08	91.81	90.42	90.43
TDM	99.56	86.25	89.80	87.99	88.01
Flooding	99.58	80.79	70.23	75.14	75.32
Average	99.36	89.39	89.25	89.22	89.27
Testing Phase (20%)					
Normal	98.72	99.42	99.17	99.30	99.30
Grayhole	99.50	91.99	95.63	93.77	93.79
Blackhole	99.46	87.78	92.11	89.89	89.92
TDM	99.60	87.42	89.89	88.64	88.65
Flooding	99.62	81.88	71.18	76.16	76.35
Average	99.38	89.70	89.60	89.55	89.60

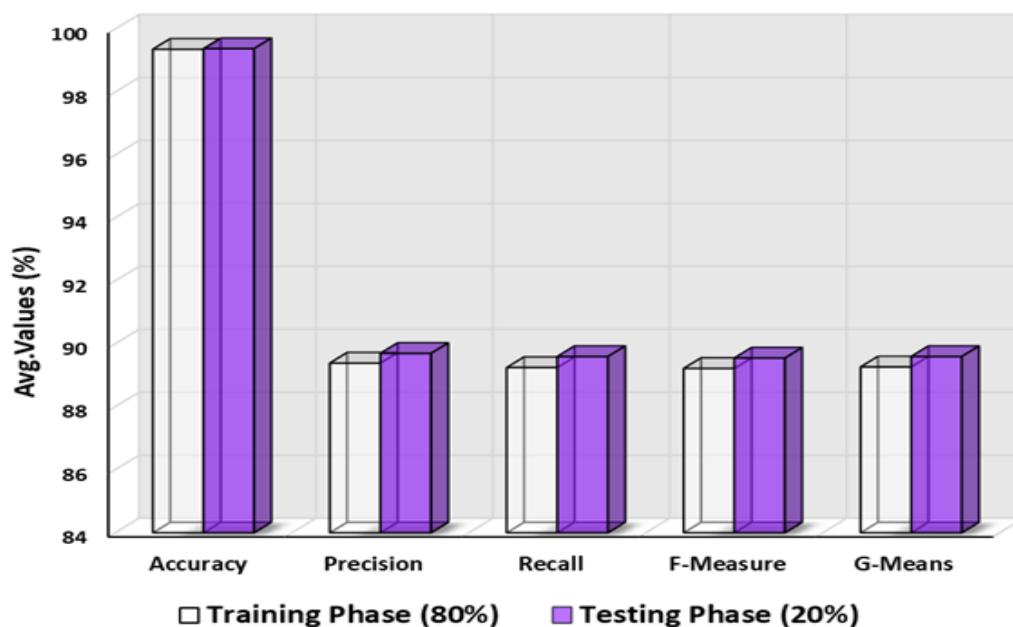


Fig. 5. Average values of RNNICD-HMO approach with 80%:20%

Fig. 6 shows the training (TRAN) and validation (VALD) accuracy of RNNICD-HMO model on 80:20 split over 25 epochs. Both curves progressively rise and slowly converge, which specifies that the model is learning effectually. The VALD accuracy persistently remains slightly greater than the TRAN accuracy, implying that the method is not overfitting and generalizes well to unseen data. The fluctuations in accuracy are expected owing to the difficulty of the task, but the whole upward tendency proves stronger performance and model stability.

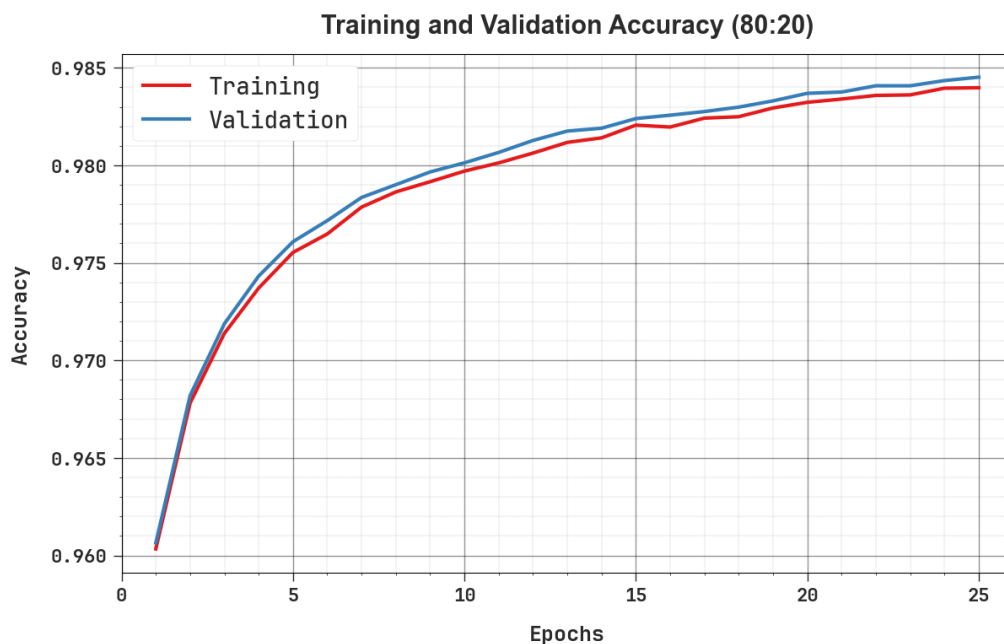


Fig. 6. Accuracy curve of RNNICD-HMO algorithm on 80:20 split

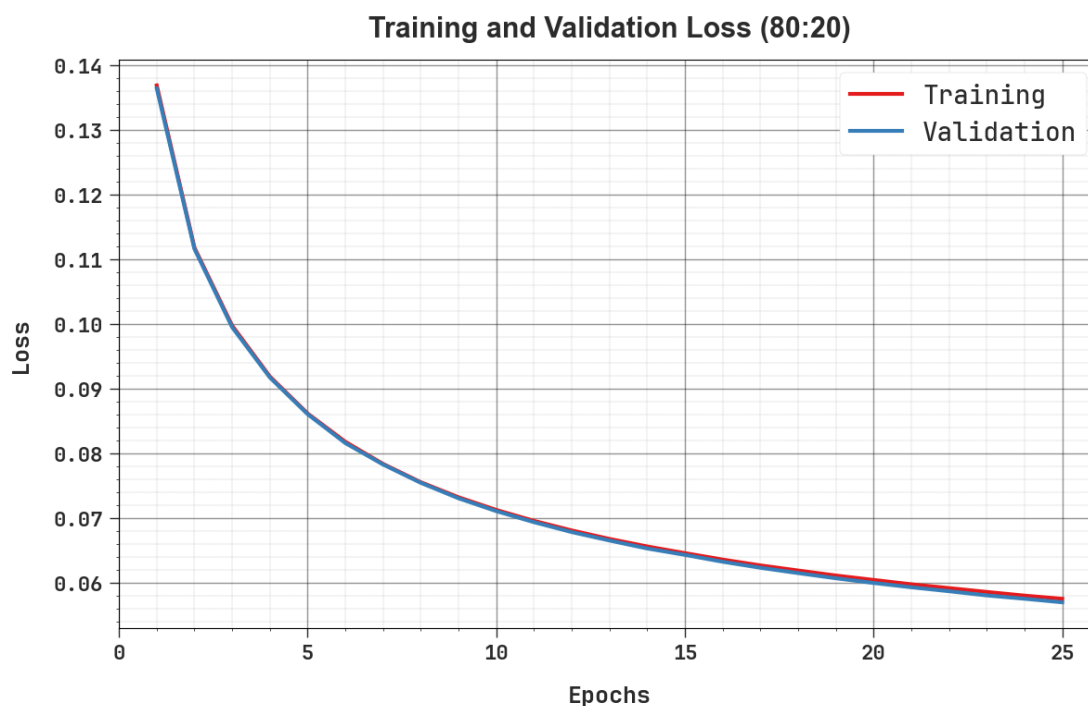


Fig. 7. Loss curve of RNNICD-HMO algorithm on 80%:20% split

Fig. 7 exemplifies the TRAN and VALD loss of RNNICD-HMO technique on 80:20 split over 25 epochs. Both curves display a constant downward tendency, signifying that the approach efficiently minimizes error throughout learning. The VALD loss stays slightly lower than the training loss through most epochs, implying good generalized and no signs of overfitting. Even though some fluctuations are monitored are becoming progressively constant and steady.

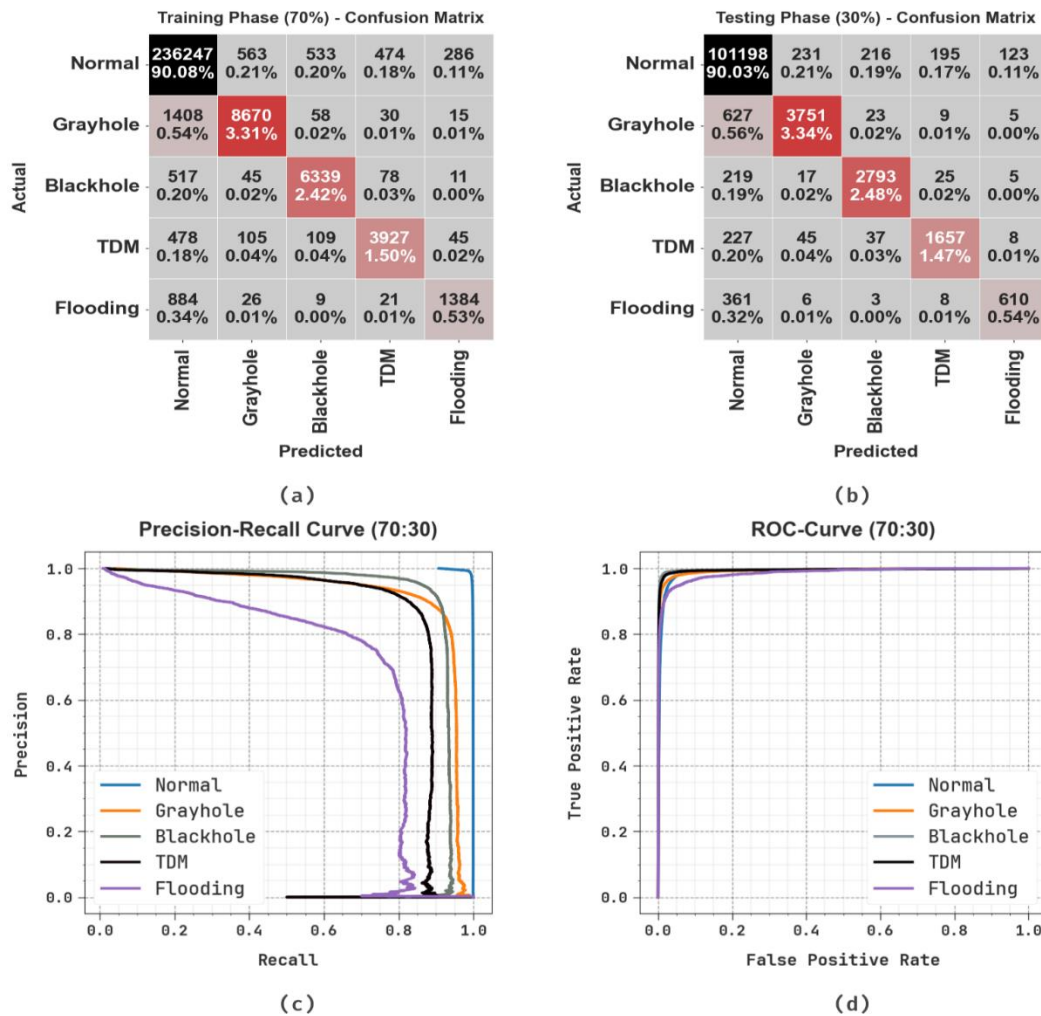


Fig. 8. 70% and 30% of (a, b) confusion matrices and (c, d) Curves of PR and ROC

Fig. 8 signifies the classifier outcomes of RNNICD-HMO model under 80%:20%. Figs. 7a, 7b display the confusion matrices with correct recognition of 5 classes. Fig. 7c shows the PR analysis, signifying superior result through each class. At last, Fig. 7d exemplifies the ROC curve, indicating effectual outcomes with great value of ROC for diverse class labels.

Table 3 and Fig. 9 define an attack detection of the RNNICD-HMO technique under 70%:30%. On 70% training, the proposed RNNICD-HMO model reaches average $accu_r$, $preci_n$, $recal_l$, $F_{Measure}$, and G_{Means} of 99.13%, 89.38%, 83.76%, 86.25%, and 86.41%, respectively. Likewise, on 30% testing, the proposed RNNICD-HMO model obtains average $accu_r$, $preci_n$, $recal_l$, $F_{Measure}$, and G_{Means} of 99.15%, 90.17%, 84.24%, 86.90%, and 87.05%, respectively.

Table 3 Attack detection of RNNICD-HMO model under 70%:30%

Class	$Accu_r$	$Preci_n$	$Recal_l$	$F_{Measure}$	G_{Means}
Training Phase (70%)					
Normal	98.04	98.63	99.22	98.92	98.92
Grayhole	99.14	92.15	85.16	88.51	88.58

Blackhole	99.48	89.94	90.69	90.31	90.31
TDM	99.49	86.69	84.20	85.43	85.43
Flooding	99.51	79.49	59.55	68.09	68.80
Average	99.13	89.38	83.76	86.25	86.41
Testing Phase (30%)					
Normal	98.04	98.60	99.25	98.93	98.93
Grayhole	99.14	92.62	84.96	88.62	88.71
Blackhole	99.52	90.92	91.30	91.11	91.11
TDM	99.51	87.49	83.94	85.68	85.70
Flooding	99.54	81.23	61.74	70.16	70.82
Average	99.15	90.17	84.24	86.90	87.05

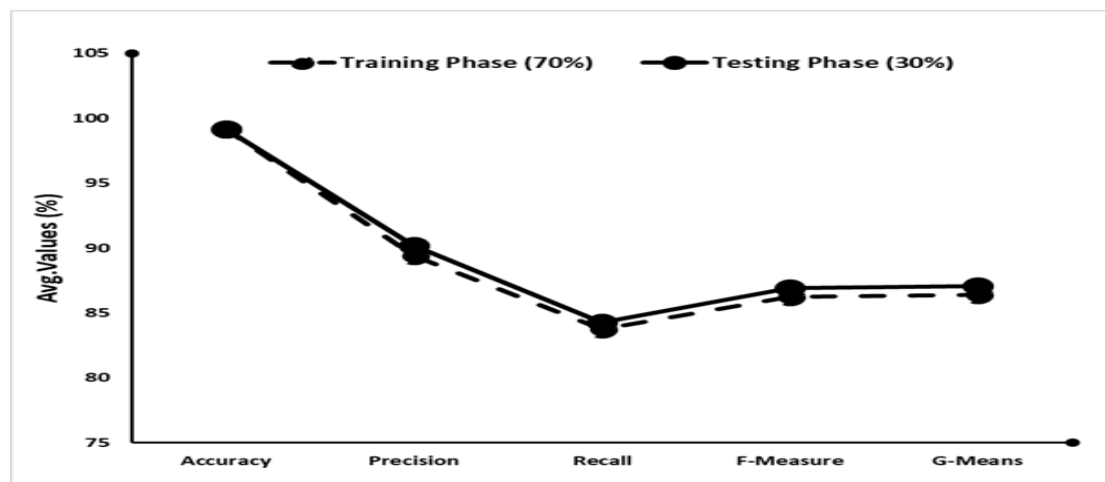


Fig. 9. Average values of RNNICD-HMO approach with 70% and 30%

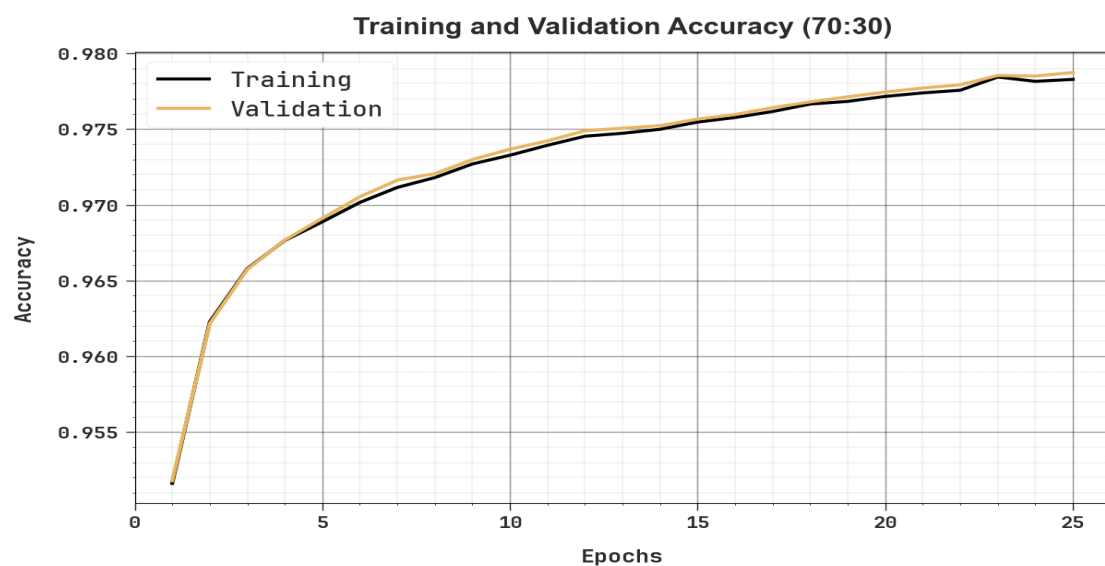


Fig. 10. Accuracy curve of RNNICD-HMO algorithm on 70:30 split

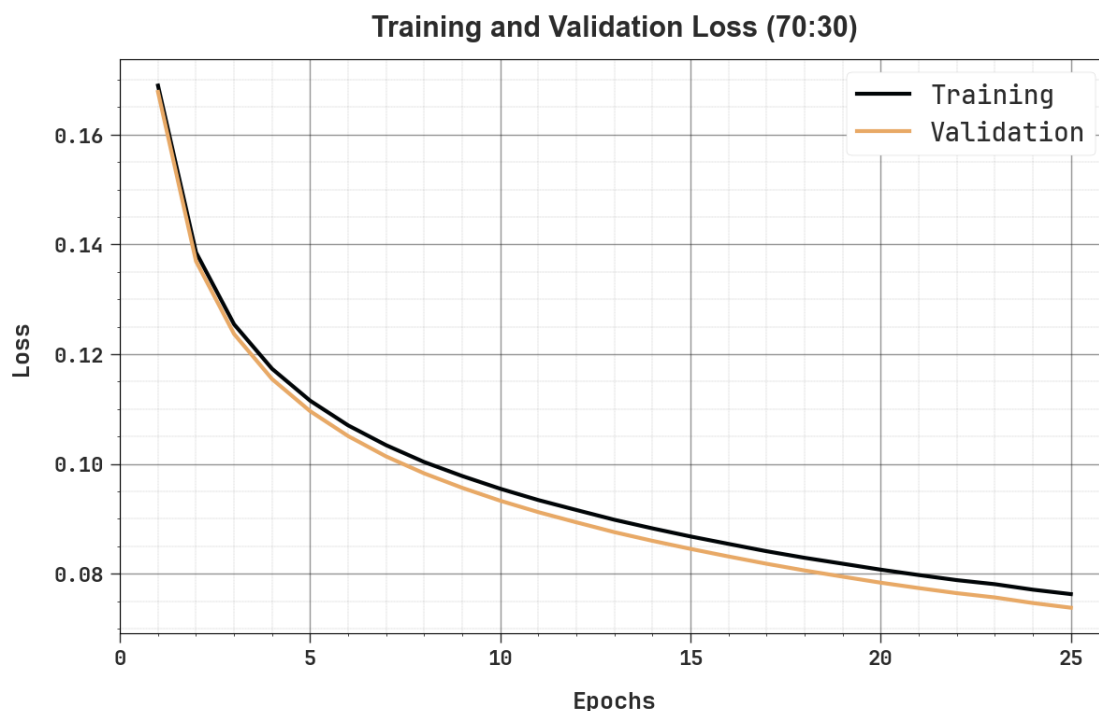


Fig. 11. Loss curve of RNNICD-HMO algorithm on 70:30 split

Fig. 10 displays the TRAN and VALD accuracy of RNNICD-HMO model on a 70:30 split over 25 epochs. Both curves gradually rise and slowly converge, which shows that the model is learning effectually. The VALD accuracy constantly remains slightly superior to the TRAN accuracy, implying that the method is not overfitting and generalizes well to unseen data. The fluctuations in accuracy are expected because of the intricacy of the task, but the whole upward tendency proves stronger performance and steadiness of the model.

Fig. 11 exemplifies the TRAN and VALD loss of the RNNICD-HMO method on 70:30 split over 25 epochs. Both curves display a constant downward tendency, demonstrating that the model efficiently minimizes error in learning. The VALD loss stays somewhat inferior to the training loss through most epochs, implying good generalized performance and no signs of overfitting. Though some fluctuations are monitored become steadier and more consistent.

Table 4 presents the epoch-wise training and validation performance of the RNNICD-HMO model over 25 epochs. The results demonstrate a progressive improvement in both training and validation accuracy, accompanied by a corresponding reduction in training and validation loss, indicating effective learning and convergence of the proposed model. The gradual stabilization of the accuracy and loss curves toward the later epochs suggests that the model approaches a stable learning state without significant overfitting, while the close correspondence between the training and validation trends indicates good generalization capability on unseen data.

Table 4. Epoch-wise Training and Validation Performance of RNNICD-HMO

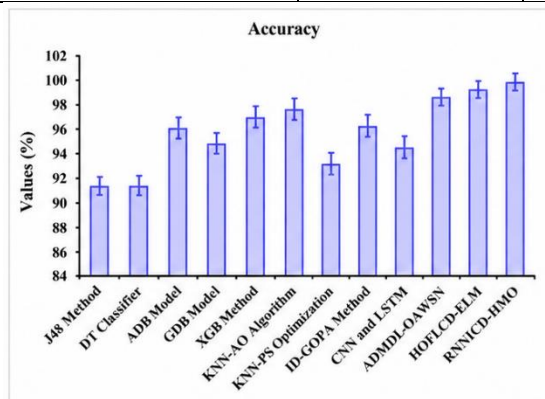
Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
-------	-----------------------	-------------------------	---------------	-----------------

1	88.20	88.65	0.48	0.46
2	90.10	90.35	0.41	0.40
3	91.60	91.85	0.36	0.35
4	92.70	93.00	0.32	0.31
5	93.50	93.80	0.29	0.28
6	94.20	94.50	0.26	0.25
7	94.80	95.10	0.24	0.23
8	95.30	95.60	0.22	0.21
9	95.80	96.10	0.20	0.19
10	96.20	96.50	0.18	0.17
11	96.60	96.90	0.17	0.16
12	96.90	97.20	0.15	0.14
13	97.20	97.50	0.14	0.13
14	97.50	97.80	0.13	0.12
15	97.80	98.05	0.12	0.11
16	98.05	98.30	0.11	0.10
17	98.25	98.50	0.10	0.09
18	98.45	98.70	0.09	0.08
19	98.65	98.85	0.08	0.075
20	98.80	99.00	0.07	0.065
21	98.95	99.10	0.06	0.055
22	99.05	99.20	0.052	0.048
23	99.15	99.28	0.045	0.041
24	99.25	99.35	0.039	0.035
25	99.36	99.38	0.035	0.031

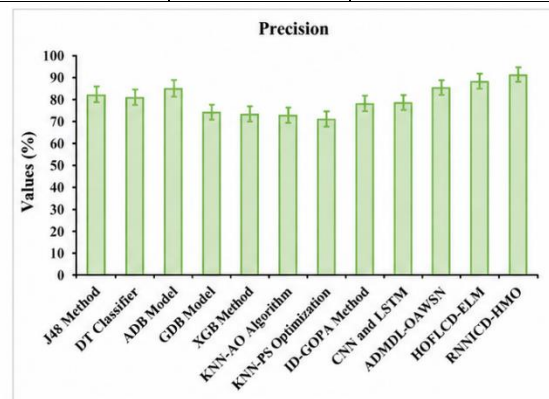
Table 5 and Fig. 12 depict the comparison outcomes of the RNNICD-HMO method with existing models under several measures [24-26]. The results emphasized that the J48, DT, and KNN-PS Optimization models have shown the worst performance. In the meantime, ADB, GDB, XGB, ID-GOPA, and CNN and LSTM methods have gotten slightly near results. Furthermore, the KNN-AO Algorithm has obtained reasonable performance with $accuracy$, $precis_n$, $recall_l$, and $F_{Measure}$ of 97.30%, 75.64%, 70.25%, and 73.95%, correspondingly. In comparison with the previously proposed ADMDL-OAWSN and HOFLCD-ELM methods, the ADMDL-OAWSN method obtained an accuracy, precision, recall, and F-Measure of 98.94%, 87.81%, 78.50%, and 82.63%, respectively, whereas the HOFLCD-ELM method achieved 99.19%, 88.91%, 83.12%, and 85.73%, respectively. Therefore, the proposed RNNICD-HMO method stated the greatest performance with highest $accuracy$ of 99.36, $precis_n$ of 89.39%, $recall_l$ of 89.25%, and $F_{Measure}$ of 89.22%.

Table 4 Comparative analysis of RNNICD-HMO algorithm with existing approaches

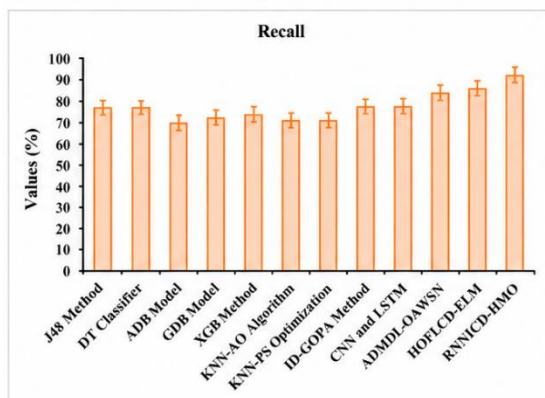
Methods	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
J48 Method	91.34	82.71	76.10	78.73
DT Classifier	91.36	82.08	75.96	78.43
ADB Model	95.79	85.49	69.31	76.21
GDB Model	94.66	75.81	71.11	72.02
XGB Method	96.90	74.52	71.60	71.08
KNN-AO Algorithm	97.30	75.64	70.25	73.95
KNN-PS Optimization	92.99	73.61	71.39	70.56
ID-GOPA Method	96.00	79.54	77.00	80.15
CNN and LSTM	94.00	80.36	80.50	81.99
ADMDL-OAWSN	98.94	87.81	78.50	82.63
HOFLCD-ELM	99.19	88.91	83.12	85.73
RNNICD-HMO	99.38	89.70	89.60	89.55



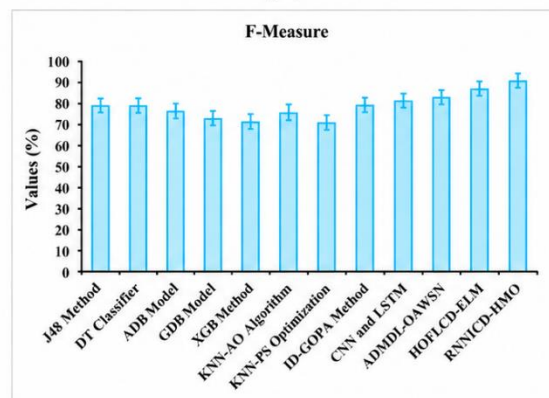
(a)



(b)



(c)



(d)

Fig. 12. Comparative analysis of RNNICD-HMO model (a) $Accur_y$ (b) $Preci_n$ (c) $Recal_l$ and (d) $F_{Measure}$

Table 5 and Fig. 13 depict the processing time (PT) outcome of the RNNICD-HMO model with existing methods and the previously proposed ADMDL-OAWSN and HOFLCD-ELM approaches. The table value states that the proposed RNNICD-HMO model has obtained lesser PT of 03.31 s, whereas the J48, DT, ADB, GDB, XGB, KNN-AO, KNN-PS Optimization, ID-GOPA, and CNN and LSTM techniques have achieved highest PT of 19.74 s, 8.78 s, 10.90 s, 7.79 s, 20.94 s, 15.40 s, 22.24 s, 10.78 s, and 13.54 s, respectively. In comparison, the previously proposed ADMDL-OAWSN and HOFLCD-ELM models have obtained PT values of 6.05 s and 4.88 s, respectively. Thus, compared to existing approaches as well as the previously developed methods, the presented RNNICD-HMO model is notably effective for clone detection in WSNs.

Table 5 PT outcome of RNNICD-HMO approach with existing techniques

Methods	Processing Time (sec)
J48 Method	19.74
DT Classifier	8.78
ADB Model	10.90
GDB Model	7.79
XGB Method	20.94
KNN-AO Algorithm	15.40
KNN-PS Optimization	22.24
ID-GOPA Method	10.78
CNN and LSTM	13.54
ADMDL-OAWSN	6.05
HOFLCD-ELM	4.88
RNNICD-HMO	03.31

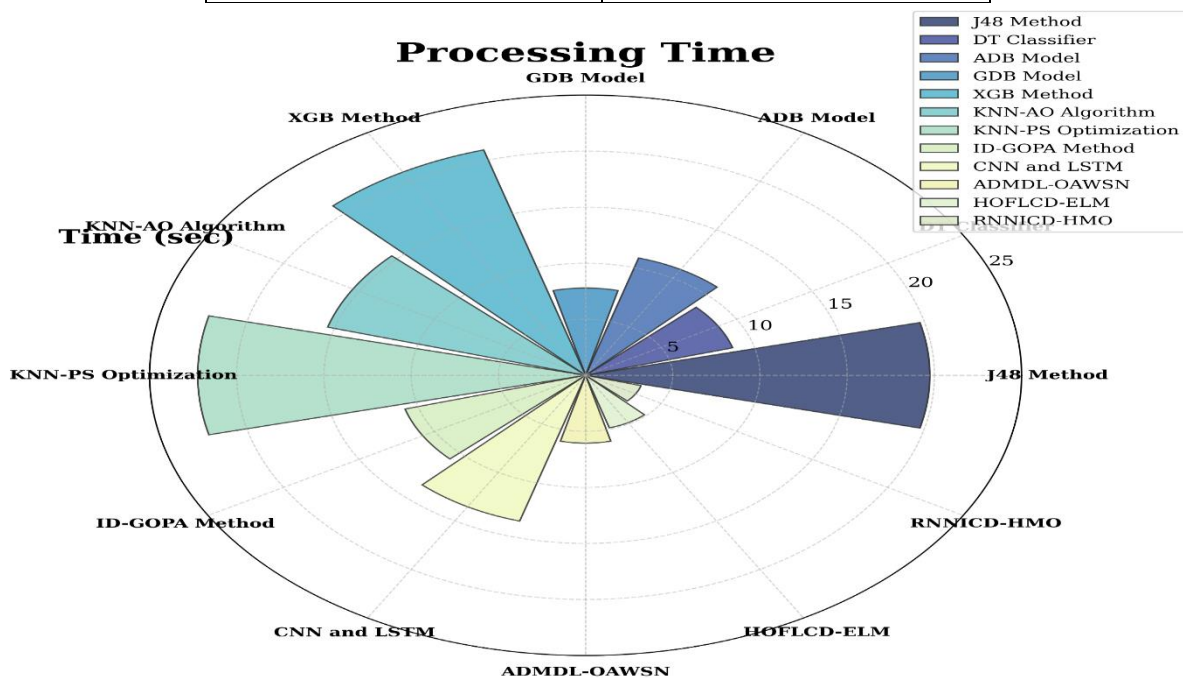


Fig. 13. PT outcome of RNNICD-HMO system with existing techniques

Table 6. Computational Efficiency Improvement of RNNICD-HMO

Comparison	Computational Time (s)	RNNICD-HMO Computational Time (s)	Reduction
J48	19.74	3.31	83.23%
KNN-PS	22.24	3.31	85.12%
XGB	20.94	3.31	84.20%
ADMDL-OAWSN	6.05	3.31	45.29%
HOFLCD-ELM	4.88	3.31	32.17%

Table 6 presents the computational efficiency improvement of the proposed RNNICD-HMO model in comparison with selected existing approaches and the previously developed ADMDL-OAWSN and HOFLCD-ELM models. The comparison is performed using computation time in seconds, where a lower computation time indicates faster execution and improved computational efficiency. The results clearly demonstrate that RNNICD-HMO requires the lowest computation time among all the compared methods, with a value of 3.31 s. This reduction in computation time is significant for Wireless Sensor Network (WSN) applications, where computational resources are generally constrained and rapid attack detection is desirable. Among the conventional approaches, KNN-PS requires the highest computation time of 22.24 s, followed by XGB with 20.94 s and J48 with 19.74 s. When compared with these methods, RNNICD-HMO reduces the computation time by 85.12%, 84.20%, and 83.23%, respectively. This indicates that the proposed method can perform the detection task substantially faster than the selected conventional and optimization-based approaches. The comparatively higher computation time of these methods can be associated with the additional computational operations involved in their respective classification and optimization procedures.

A significant improvement is also observed when RNNICD-HMO is compared with the previously proposed methods in this research. ADMDL-OAWSN requires 6.05 s, whereas HOFLCD-ELM requires 4.88 s. In comparison, RNNICD-

HMO completes the computation in only 3.31 s, representing reductions of 45.29% and 32.17%, respectively. This progressive reduction in computation time across the three research works demonstrates the computational-efficiency-oriented evolution of the proposed frameworks. Work 1 employs a CNN-BiGRU-Attention architecture with optimization, while Work 2 incorporates DBN, CVAE, GCN, and optimization. In contrast, Work 3 employs the RVFL classifier, together with SSA-based feature selection and MACSSA-based parameter optimization, providing a comparatively lightweight classification framework.

The lower computation time of RNNICD-HMO can also be related to the reduction of the feature space from 18 original features to 13 selected features, which reduces the dimensionality of the input supplied to the subsequent classification process. Furthermore, the RVFL architecture uses randomly generated hidden-layer parameters and analytical determination of output weights, avoiding the extensive iterative training associated with deeper architectures. Consequently, although SSA and MACSSA introduce optimization operations, the overall framework maintains lower computational requirements.

5. Conclusion

This manuscript presented an RNNICD-HMO model to precisely detect cloned sensor nodes and prevent security breaches in WSNs. To perform that, the proposed RNNICD-HMO approach initially applies data preprocessing methods such as data cleaning and normalization

to guarantee high-quality and consistent input for analysis. Afterwards, the SSA is implemented for the dimensionality reduction process to identify the most informative attributes. Then, an RVFL network is employed for effective and reliable classification of cloned and legitimate nodes. To further improve detection accuracy, a hybrid

MACSSA approach is utilized for optimum parameter tuning of classification model. The experimental study of the RNNICD-HMO approach ensues utilizing various benchmark databases. The simulation results indicated an improved outcome of the RNNICD-HMO approach concerning diverse measures.

References

- Sathya, V., David, P.E., Rajakumaran, M., and Hariprasath, S., 2023. Automatic Clone Detection in Wireless Sensor Networks using ANN.
- Numan, M., Subhan, F., Khan, W.Z., Hakak, S., Haider, S., Reddy, G.T., Jolfaei, A. and Alazab, M., 2020. A systematic review on clone node detection in static wireless sensor networks. *IEEE Access*, 8, pp.65450-65461.
- Shaukat, H.R., Hashim, F., Shaukat, M.A. and Ali Alezabi, K., 2020. Hybrid multi-level detection and mitigation of clone attacks in mobile wireless sensor network (MWSN). *Sensors*, 20(8), p.2283.
- Dora, J.R. and Nemoga, K., 2021. Clone node detection attacks and mitigation mechanisms in static wireless sensor networks. *Journal of Cybersecurity and Privacy*, 1(4), pp.553-579.
- Morales-Molina, C.D., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L.K., Perez-Meana, H., Olivares-Mercado, J., Portillo-Portillo, J., Sanchez, V. and Garcia-Villalba, L.J., 2021. A dense neural network approach for detecting clone id attacks on the rpl protocol of the iot. *Sensors*, 21(9), p.3173.
- Alansari, Z., Anuar, N.B., Kamsin, A. and Belgaum, M.R., 2022. A systematic review of routing attacks detection in wireless sensor networks. *PeerJ Computer Science*, 8, p.e1135.
- Sujihelen, L., Boddu, R., Murugaveni, S., Arnika, M., Haldorai, A., Reddy, P.C.S., Feng, S. and Qin, J., 2022. Node replication attack detection in distributed wireless sensor networks. *Wireless Communications and Mobile Computing*, 2022(1), p.7252791.
- Islam, M.N.U., Fahmin, A., Hossain, M.S. and Atiquzzaman, M., 2021. Denial-of-service attacks on wireless sensor network and defense techniques. *Wireless Personal Communications*, 116(3), pp.1993-2021.
- Salmi, S. and Oughdir, L., 2023. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. *Journal of Big Data*, 10(1), p.17.
- Ramasamy, L.K., KP, F.K., Imoize, A.L., Ogbemor, J.O., Kadry, S. and Rho, S., 2021. Blockchain-based wireless sensor networks for malicious node detection: A survey. *IEEE Access*, 9, pp.128765-128785.
- Bowman, B. and Huang, H.H., 2020, September. VGRAPH: A robust vulnerable code clone detection system using code property triplets. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 53-69). IEEE.
- Nashaat, M., Amin, R., Eid, A.H. and Abdel-Kader, R.F., 2025. An enhanced transformer-based framework for interpretable code clone detection. *Journal of Systems and Software*, 222, p.112347.
- Nithya, K.J. and Shyamala, K., 2025. Entropy dragonfly optimisation-based cluster head selection and deep learning clone node detection for wireless sensor network. *International Journal of Information and Computer Security*, 26(4), pp.394-421.
- Liu, W., Shi, J., Wang, H., Chen, T., Han, Z. and Li, Q., 2025. Coordinate plane based authentication method for detecting clone node in wireless sensor networks. *Journal of Information Security and Applications*, 93, p.104148.
- Yahya, M.A. and Kim, D.K., 2023. CLCD-I: cross-language clone detection by using deep learning with infercode. *Computers*, 12(1), p.12.
- Zou, D., Feng, S., Wu, Y., Suo, W. and Jin, H., 2023, November. Tritor: Detecting Semantic Code Clones by Building Social Network-Based Triads Model. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 771-783).

-
- Zhang, Y., Yang, J., Dong, H., Wang, Q., Shao, H., Leach, K. and Huang, Y., 2022. ASTRO: An AST-assisted approach for generalizable neural clone detection. arXiv preprint arXiv:2208.08067.
- Quradaa, F.H.A., Shahzad, S., Saeed, R. and Al-Hakimi, M.H., 2025. Enhancing Semantic Clone Detection with Siamese Bi-LSTM for Multi-Level Source Code Representations. *Arabian Journal for Science and Engineering*, pp.1-26.
- Shieh, C.S., Nguyen, T.L., Nguyen, T.T. and Horng, M.F., 2025. Unknown DDoS Attack Detection with Sliced Iterative Normalizing Flows Technique. *Computers, Materials & Continua*, 82(3).
- Li, Q. and Zhou, Y., 2025. An Enhanced Knowledge Salp Swarm Algorithm for Solving the Numerical Optimization and Seed Classification Tasks. *Biomimetics*, 10(9), p.638.
- Sajid, M., Quadir, A., Tanveer, M. and Alzheimer's Disease Neuroimaging Initiative, 2025. GB-RVFL: Fusion of randomized neural network and granular ball computing. *Pattern Recognition*, 159, p.111142.
- Niksirat, M., Saffarian, M., Tayyebi, J., Deaconu, A.M. and Spridon, D.E., 2024. Fuzzy Multi-Objective, Multi-Period Integrated Routing–Scheduling Problem to Distribute Relief to Disaster Areas: A Hybrid Ant Colony Optimization Approach. *Mathematics*, 12(18), p.2844
- <https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds>
- Fares, H., Vibhute, A.D., Mouniane, Y. and Bouijij, H., 2025. Intrusion Detection in Wireless Sensor Networks using Machine Learning. *Procedia Computer Science*, 252, pp.912-921.
- Alkhalifa, A.K., Alruwais, N., Mansouri, W., Arasi, M.A., Alliheedi, M., Alallah, F.S., Khadidos, A.O. and Alshareef, A., 2025. Hybrid dung beetle optimization based dimensionality reduction with deep learning based cybersecurity solution on IoT environment. *Alexandria Engineering Journal*, 111, pp.148-159.
- Saleh, H.M., Marouane, H. and Fakhfakh, A., 2024. Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning. *IEEE Access*, 12, pp.3825-3836.