Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.755224

# INTELLIGENT CLONE DETECTION AND CLASSIFICATION USING CAT SWARM OPTIMIZATION WITH DEEP LEARNING MODEL FOR WIRELESS SENSOR NETWORKS

**S.Bhuvana**
Research Scholar, Department of Computer Science and Engineering, Dr.MGR Educational and Research Institute, E-mail: bhupreethi@gmail.com

**Dr.S.Kevin Andrews**
Professor, Department of Computer Applications, Dr.MGR Educational and Research Institute, E-mail:kevin.mca@drmgrdu.ac.in

**Dr.M.S.Josephine**
Professor, Department of Computer Applications, Dr.MGR Educational and Research Institute, E-mail:Josephine.mca@drmgrdu.ac.in

Dr.V.Jeyabalaraja
Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, E-mail: jeyabalaraja@gmail.com

**Abstract**

The progress of wireless sensor networks (WSN) technology wasobtaininggreatly enhance count of attention from researchers in recent decades. Its huge count of sensor nodes (SNs) is most feature which generatesit useful to technology. The sensors areconnectedtogether to network model. These SNs can be usually exploited for various applications like target tracking, health monitoring, pressure monitoring, fire recognition, and so on. But, the disadvantage is that WSNs canfrequentlyutilize in hostile, critical environments but it could not control physical access. The adversary can capture the legitimate SNs, take them out and thengather any sensitive data like node ID, keys and accomplish a replication attack. This study presents an Intelligent Clone Detection and classification using Cat Swarm Optimization with Deep Learning (ICDC-CSODL)technique for WSN. The main goal of the ICDC-CSODL system lies in the accurate identification and classification of clone nodes in the network. To accomplish this, the presented ICDC-CSODL technique follows a two-stage process. Initially, the ICDC-CSODL system utilizes attention-basedbi-directional long short-term memory (ABiLSTM) approach for clone node detection. Next, in the latter stage, the CSO system is used to adjust the hyperparameter values of the ABiLSTM approach. The simulation results of the ICDC-CSODL technique are tested on a series of experiments. A widespread simulation results analysis illustrated the improvement of the ICDC-CSODL technique in terms of different measures.

**Keywords:** Wireless sensor networks; Clone node detection; Security; Machine learning; Cat swarm optimizer

## 1. Introduction

The existing innovations in information technology enabled the advancement of low-cost sensor nodes (SN) with communication and processing abilities in wireless sensor networks (WSN) [1]. The unique features of these low-cost SNs namely limited resources in the context of battery, processing, lack of tamper resistance, and hardware memory make them prone to node replication or clone node attack [2]. The WSN utilization in the harsh and remote atmosphere helps the attacker for capturing the legal node and abstract the saved credential data like ID that is easily replicated and re-programmed [3]. Hence, the attacker can control the entire network internally and perform similar functions with the legal nodes [4]. One dangerous attack on WSN is Clone node attack. During the clone attack, the attacker captures and targets a legal node, and abstracted the saved credentials utilizing certain specialized tools within one minute [5]. Next, the invader would isolate the obtained legal node in the network, adopts the clones, and hence has the capability and even stop the node withdrawal method [6]. Further, to lessen damages, clones should be identified in short period that is not a simple task owing to various elements like nodes with legal IDs, data, etc.

From the literature, it was found that due to the features of the WSN such as lack of tamper resistance hardware, limited processing, battery, and memory, the SNs are vulnerable to various attacks like node replication or clone node attack [7]. To counter clone node attacks, various methods like network-related detection methods, distributed-based detection, and centralized based recognition methods were devised [8]. Deep learning (DL) and Machine learning (ML) approaches are utilized for finding clone nodes in WSN. Such methods can examine the network behaviour, traffic, and other features of the nodes to find clones [9]. With further research and development, these algorithms have the capability to enhance the reliability and security of WSNs. This becomes the intention of researchers to devise enhanced detection protocols for clone attacks [10].

Large numbers of sensor devices are deployed in wireless sensor networks to monitor physical phenomena and collect data. However, attackers can compromise a network by introducing replica nodes that imitate legitimate nodes and disrupt its operation. The purpose of clone node detection is to identify fraudulently replicated nodes within a network. Various techniques, including statistical analysis, cryptographic methods, and anomaly detection algorithms, can be used to detect clone nodes. These methods identify clones by analysing the behaviour, communication patterns, or cryptographic signatures of the nodes. In order to detect anomalies that indicate the presence of cloned nodes, statistical analysis may entail examining signal intensity, packet arrival rates, or energy consumption.After the detection of clone nodes, the next stage is classification. Classification entails categorising the detected clone nodes according to their attributes or characteristics. The classification procedure can take into account variables such as the type of attack (e.g., node replication, node impersonation), the degree of similarity between clone and legitimate nodes, and the impact of clones on network performance. In wireless sensor networks, the purpose of clone node detection and classification is to improve network security and assure reliable data collection and communication. By identifying and classifying clone nodes, network administrators can mitigate the impact of the clones by isolating or removing them from the network, updating security protocols, or instituting intrusion detection systems. In wireless sensor networks,

efficient clone node detection and classification techniques are essential for maintaining data integrity, confidentiality, and availability, as well as ensuring the overall robustness and security of the network infrastructure.

This study presents an Intelligent Clone Detection and classification using Cat Swarm Optimization with Deep Learning (ICDC-CSODL) approach for WSN. The main objective of the ICDC-CSODL system lies in the accurate detection and classification of clone nodes in the network. To accomplish this, the presented ICDC-CSODL approach follows a two-stage process. Initially, the ICDC-CSODL technique utilizes attention-based bi-directional long short-term memory (ABiLSTM) algorithm for clone node detection. Next, in the latter stage, the CSO system is used to adjust the hyperparameter values of the ABiLSTM approach. The simulation results of the ICDC-CSODL technique are tested on a series of experiments.

## 2. Related Works

Ahmad [11] developed the pairing algorithm and a threat model for pairing all the SNs positioned in the neighbour. The efficiency of 3 ML approaches was compared to KNN, DT, and SVM with two openly accessible real-time datasetswith respect to testing time, attack detection accuracy, and trained time. In [12], the authors designed and implemented a prototype GeneDiff, a semantic-based representation binary clone recognition technique for cross-architecture. GeneDiff exploits a representation method dependent upon NLP for generating high-dimensional numerical vectors to all the functions based on using Valgrind intermediate representation (VEX) representation. GeneDiff is robust to different architectures and different compiler optimization options. Chen et al. [13] presented a new side-channel-based password cracking system, such as MAGLeak, to identify the victim's passwords by leveraging magnetometer, accelerometer, and gyroscope of IoT touch-screen smart devices. Particularly, the event-determined data gathered technique has been developed for ensuring that keystroke behaviors of the user can be accurately reflected. Furthermore, random forest model can be leveraged for the detection method, followed by the data preprocessing model.

Yadav et al. [14] classify various attacks on Android and IoT devices and mitigation approaches introduced by the researcher workers. This study provides a comparative outcome of malware detection model in the various platform of attacks. In [15], devised a new method for categorizing ASTs utilizing classical supervised-learning algorithm, but a feature learning model chose the representative syntax pattern for child subtrees of dissimilar syntax constructs. The presented method is used for the problem of labelling the expertise level of Java programmers.

In [22] the author proposed the method using the concept of a protocol for detecting clone nodes in a wireless sensor network using a Distributed Hash Table (DHT)-based approach. Clone nodes are nodes that maliciously impersonate legitimate nodes in the network, and detecting them is essential for ensuring the security and reliability of the network.The Distributed Hash Table (DHT): The protocol is based on the Chord DHT, which is used to organize nodes in a virtual ring structure and facilitate efficient key-based lookups. Chord assigns each node a coordinate based on the hash value of its MAC address.This Clone

detection mechanisms can generate false positives (identifying non-clones as clones) or false negatives (failing to identify actual clones). The accuracy of the detection process depends on various factors, including the quality of the hash function, the uniqueness of MAC addresses, and the effectiveness of the cache-based detection. Chord DHT itself may not be the most scalable option for large wireless sensor networks. As the network grows in size, the overhead associated with maintaining the Chord structure, finger tables, and routing can become significant. This can result in increased latency for message routing and maintenance operations.

In [23] the author proposed the proposed an FLCND- Fuzzy Logic Clone Node Detection model based on distributed clone spotting method. The researcher suggested the method for distributed clone node detection in models intensity packet delivery ratios and also diminishes packet loss, energy consumption, and end-to-end delays. Each mobile sensor node is uniquely identified by a node ID. It assumes the presence of replicated nodes in the network, each of sharing the same ID as an original node. Nodes communicate symmetrically within defined within information radius. The network employs a random waypoint motion model for deployment. Furthermore, the network is partitioned into distinct clusters, each with its designated cluster head. Sensor nodes are assigned to specific clusters. Cluster heads are responsible for maintaining various parameters such as speed, residual energy, delay, packet delivery ratio, and the integrity of suspected nodes' reported data.

This method involves additional communication and computation overhead, which can consume valuable energy resources in resource-constrained sensor nodes. This increased energy consumption can reduce the network's overall lifespan. Implementing a fuzzy logic-based system for clone node detection introduces complexity into the network's operation. Designing, configuring, and maintaining a fuzzy logic system can be challenging, especially in resource-constrained environments.

Vladimir Vapnik [24] the Vladimir Vapnik proposed an SVM classifier for fault detection in Wireless Sensor networks. The Support vector machines represents the class of Supervised Learning Algorithm which is utilized for the task of classification and regression analysis. They work by finding the optimal hyper plane (decision boundary) that best separates data points of different classes. In this context, SVM is applied to classify sensor data and detect faults. Vladimir Vapnik introduced the Support Vector Machines (SVM) algorithm, which is designed to solve binary classification problems by identifying an optimal hyper plane that effectively separates data points belonging to two distinct classes. SVM operates by maximizing the margin, which is the distance between this hyper plane and the nearest data points from each class, known as support vectors. This margin optimization process is governed by mathematical principles and constraints, aiming to achieve the best possible separation between the classes while minimizing classification errors. In essence, SVM seeks to establish a hyperplane represented as $f(x) = \langle w, x \rangle + b$, where w is the weight vector, x represents data points, and b is a bias term, to satisfy the condition $yi(wx_i + b) \geq 1$ for all data points, where $y_i$ denotes the class labels (-1 or +1). This rigorous approach ensures that the chosen hyper plane not only classifies the data accurately but also maintains a safe margin between the classes, contributing to its robustness and generalization ability.SVMs can be computationally expensive, particularly when dealing with large datasets. Training an SVM on a large dataset can take a long time and require a lot of memory. This makes SVMs less practical for big data scenarios.

When the number of features is close to the sample or when the data is noisy, SVMs can be prone to over fitting. Regularization techniques like adjusting the hyperparameter can help mitigate this risk.

## 3. The Proposed Strategy:

In this study, we have presented the ICDC-CSODL technique to improve security in the WSN. The main aim of the ICDC-CSODL approach lies in the accurate detection and classification of clone nodes in the network. To accomplish this, the presented ICDC-CSODL system follows 2-stage processes such as ABiLSTM-based clone node detection and CSO based hyperparameter tuning. Fig. 3.1 exemplifies the overall flow of ICDC-CSODL algorithm.

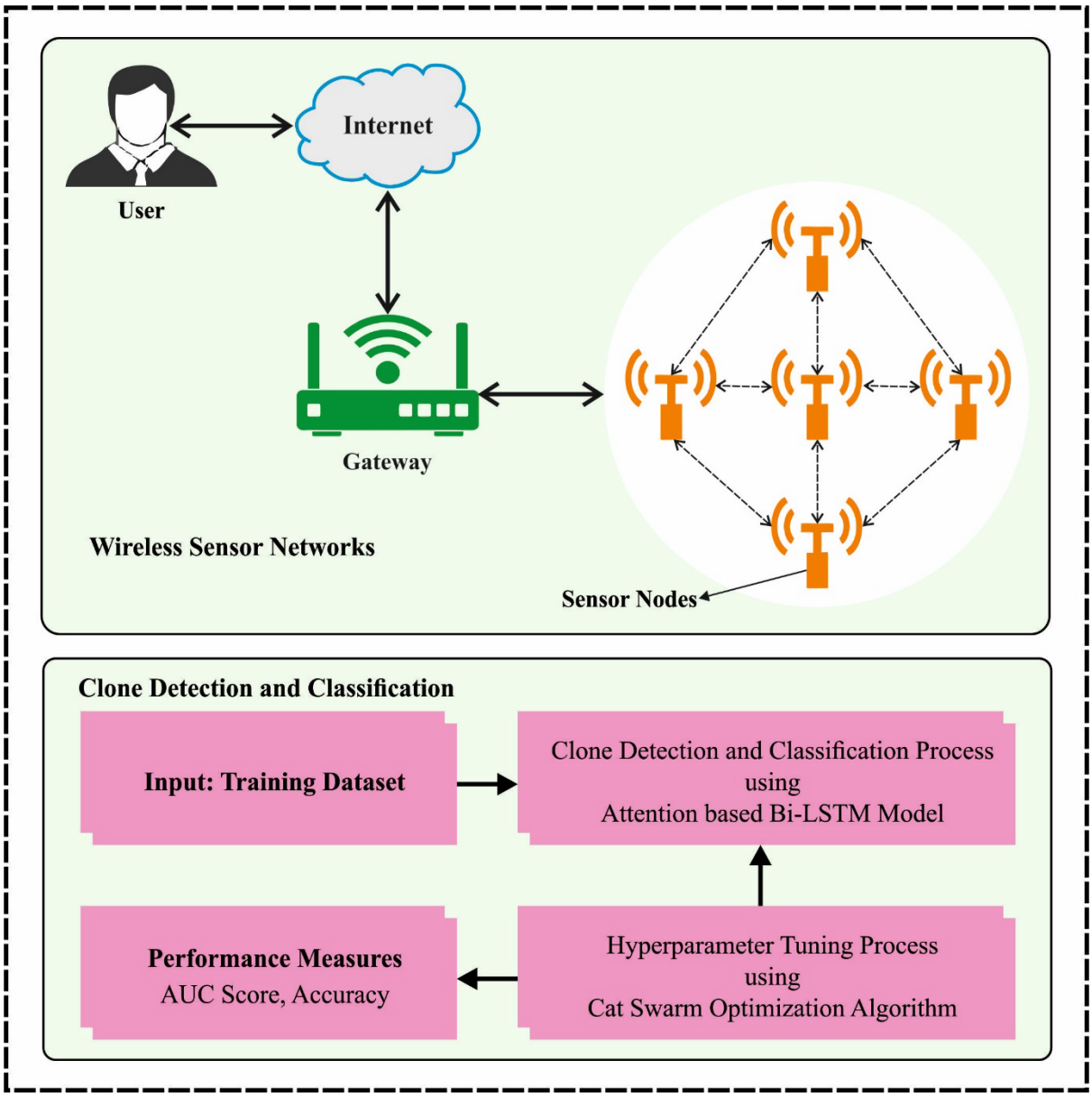


**Fig. 3.1.** Overall procedure of ICDC-CSODL approach

### 3.1. Clone Node Detection using ABiLSTM Model

The ICDC-CSODL technique primarily employed the ABiLSTM model for detecting clone nodes.

LSTM is a more commonly used DL algorithm. LSTM-NN has a complicated dynamic structure involving three gating units (input, forgetting, and output gates) [18]. The computation equation of LSTM-NN is given below:

$$\begin{cases} i_t = \sigma(W_{xi}xi + W_{hi}h_{i-1} + W_{ci}c_{i-1} + b_i) \\ f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t = f_t h_{t-1} + i_t \tanh\,(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ \mathrm{o}_t = \sigma(W_{xo}xt + W_{ho}h_{t-1} + W_{co}c_t + b_{\mathrm{o}}) \\ h_t = o_t tanh(c) \end{cases} \quad (1)$$

In Eq. (1), $i_t, f_t, c_t$, and $0_t$ denotes input, forgetting, updated cell state, and output gates, correspondingly;$h_t$ shows the output data; $x_t$ denotes input data; $\sigma$ and $\tanh$ denotes the sigmoid and hyperbolic tangent functions, correspondingly.$W$ indicates the weight coefficient; $b_i, b_f, b_c$, and $b_{\mathrm{o}}$ represents the offset, correspondingly;

LSTM has better prediction capability for non-linear time sequences;however, this model is a one-way broadcast method in which forecast value at later time has no effect on forecast value of existing time. In order to dam intelligent predictive module, it is widely assumed that two-way dynamic relationships among input as well as output at dissimilar times and utilize the novel monitoring value for reversingaccurate the predicted value for improving the predictive outcome. BiLSTM is an advanced two-way DL-NN enhanced by the LSTM and could accomplish best predictive outcomes than LSTM model.

The 2-LSTM layers in opposite direction create BiLSTM, one layer input the dataset in chronological sequence from start to finish, and the other layer input the dataset in reverse sequence from finish to start. This pair of hidden layers (HL) with opposite directions was lastly interconnected to resultant value. Thus, the forecast accuracy of BiLSTM is superior to typical LSTM or RNN. For the input, $x_i(i = 1,2,\cdots,n)$, the BiLSTM adopts the forward and reverses LSTM to implement forward and reverse recursion that takes $x_1$ and $x_n$ as the input dataset. Both output datasets can be integrated to attain the concluding output $y$:

$$\begin{cases} h_t = \alpha h_t^f + \beta h_t^b \\ y_t = \sigma(h_t) \end{cases} \quad (2)$$

In Eq. (2), $h_t^f$ and $h_t^b$ denotes the output of HL from the forward and reverse LSTM at $t$ time, correspondingly; $\alpha$ and $\beta$ denote the weight coefficient, correspondingly; and $\alpha + \beta = 1.\sigma$ denotes the sigmoid activate function and $\sigma(x) = \frac{1}{1+\exp(-x)}$.

The attention mechanism was utilized as a means to optimize the performance in signal and vision processing tasks by concentrating on feature segments of great significance. It is recently executed by the attentive NN model [19]. In ABiLSTM model, the attention module has been utilized across the dissimilar internal BiLSTM layers, along with over the BiLSTM

output layer. The forecast of resultant signalswasdeveloped by means of the conditional probability distribution of input signals.

$$p(y_i|x_0, \ldots, x_n, y_{i-1}) \tag{3}$$

However, this distribution is not feasible to calculate in real time applications:

$$p(y_i|x_0, \ldots, x_n, y_{i-1}) \approx g(y_i, h_i, C_i) \tag{4}$$

In Eq. (4),$h_i$ denotes the internal state of the BiLSTM, $g$ indicates BiLSTM, and $C_i$ shows the existing context, viz., vector holding data of which input is vital at the existing step. The context was developed in the input sequence$x$, and the existing state, $h_i$. Afterward, the BiLSTM has stepped with input series, and the attention module of network decides the attention that must be assumed by the annotation given at all the steps. Fig. 2 defines the framework of ABiLSTM. The transition function of the attentive NN is defined as follows:

$$e_t = v^T.\tanh(W_e.h_t + U_e.d_{t-1} + b) \tag{5}$$

In Eq. (5), $v, b, h_t, d_{t-1} \in \mathbb{R}^n$ and $W_e, U_e \in \mathbb{R}^{n*n}$ and $d$ indicates the input series. The attention score, $a^{t,t'}$, for every $t'$ is then calculated using the softmax function:

$$a^{t,t'} = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_t)} \tag{6}$$

The context vector, $C_t$, is calculated as the weighted summation of internal state, $\{h_1, h_T\}$:

$$C_t = \sum_{t'=1}^{T} a^{t,t'}.h_{t'} \tag{7}$$

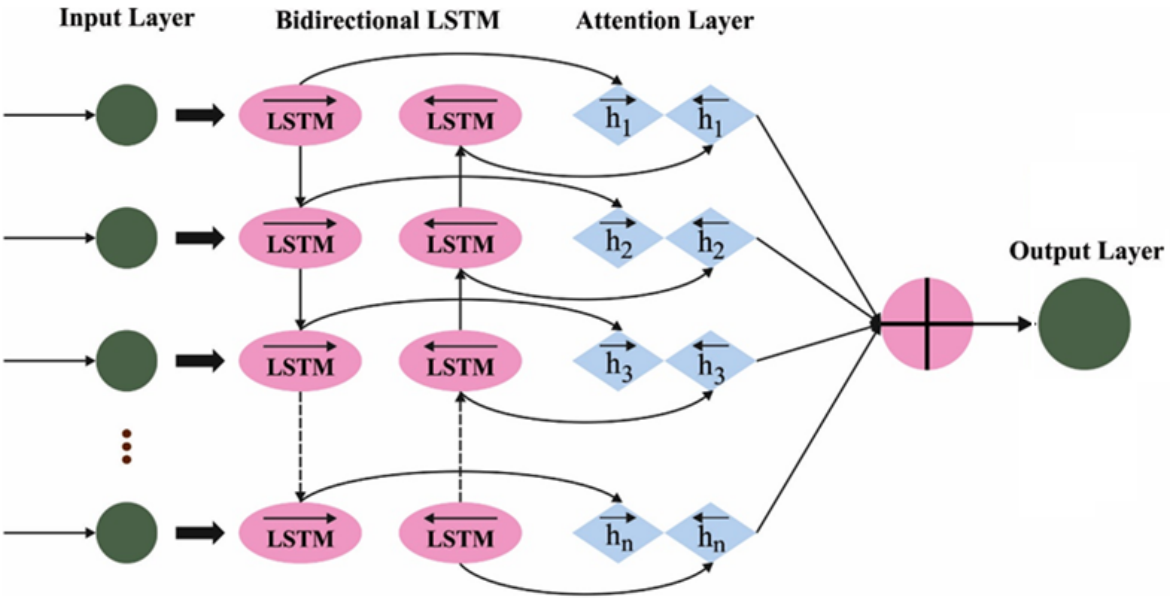


**Fig. 2.** Architecture of ABiLSTM

### 3.2. Hyperparameter Tuning using CSO Algorithm

Tuning of hyperparameters is essential for optimising the efficacy of machine learning models. For hyperparameter adjustment, the Cat Swarm Optimisation (CSO) algorithm is a metaheuristic optimisation technique. Define the hyperparameters to be tuned and the objective function used to quantify the model's performance. The CSO algorithm turn on the population of cats, with each cat representing a candidate solution, or a particular set of hyperparameter values. Each cat's objective function is evaluated, and the cats' positions are updated based on their previous positions and evaluation outcomes. This update rule is inspired by the behaviour of cats in the wild and aids in the enhancement of exploration and exploitation capabilities. The algorithm iterates until termination criteria, such as a limit number of iterations or convergence of the value of the objective function, are met. From the final population, the optimal solution representing the optimised hyperparameters is selected. The machine learning model is then trained with these hyperparameters, and its performance is assessed. By utilising the CSO algorithm for hyperparameter tuning, the search space is effectively investigated, resulting in enhanced model performance and enhanced generalisation when detecting clone nodes.

To adjust the hyperparameter values of the ABiLSTM model, the CSO algorithm is used. CSO algorithm is based on two primary behaviours of cats, which are hunting and resting [20]. Consequently, CSO includes seeking and tracing modes. Every cat represents a solution set that has a flag, its individual location, and a cost value. The location has been encompassed by M dimensions, but all the dimensions have its own velocity. Lastly, the flag is to show whether the cat is in tracing or seeking mode. In all the iterations, an optimum cat has been recognized that signifies the better solution.Seeking mode stimulates the resting performance of cats, which includes 4 important parameters namely counts of dimension to change (CDC), seeking memory pool (SMP), self-position consideration (SPC), and seeking range of the selected dimension (SRD).

In the seeking mode, SMP finds the count of copies of cat (candidate position) to be generated. Based on CDC and SPC, random copies are generated. CDC parameter is within $[0,1]$ interval and shows how many dimensions to be changed. For example, when CDC can be fixed to 0.8 and the amount of dimensions from the searching space was10, and next for every cat 8 dimensions would be changed and the rest remains constant. Finally, SPC is a Boolean valued parameter that shows whether the existing location of the cat would be chosen as one of the copies of SMP or not. The steps of these modes are given below:

i. Make $j$ copies of the existing location of cats, whereas$j = SMP$. If SPC is set to true, then $j = (SMP - 1)$ and keeps the existing location as candidate as one.
ii. CDC chooses some random dimensions to be modified for every copy. Next, randomly subtract or add SRD value from the existing position as follows:

$$X_{j,d} = (1 \pm rand * SRD) * X_{j,d} \quad (8)$$

In Eq. (8), $X_{j,d}$ denotes the location of cats; $j$ and $d$ shows the count of copies and dimensional for cat correspondingly;

iii. Evaluate the cost value for the candidate position.
iv. Utilizing the roulette wheelprocess, evaluate the choice probability of all the candidate points based on Eq. (9). Thus, the candidate point with best fitness cost has better possibilitiesthan choosing one. But if each fitness cost was unchanged, and next the choosing probability of all the candidate points would be 1.

$$Pi = \frac{|_{FS_i - FS_b}|}{FS_{\max} - FS_{\min}}, where\ 0 < i < j \tag{9}$$

If the objective is minimization then $FSb = FSmax$, or else $FSb = FSmin$.

Tracing mode stimulates the stalking actions of cats and the steps are given below:

1. Upgrade the velocity for each dimension using the following expression:

$$V_{k,d} = V_{k,d} + c_1 * rand\ * \left(X_{best,d} - X_{k,d}\right) \tag{10}$$

In Eq. (10), $V_{k,d}$ denotes the velocity for the $k^{th}$ cat at $d^{th}$dimension; $X_{best,d}$ shows the cat location with the better fitness cost; $X_{k,d}$ indicates the existing location of the $k^{th}$ cat at $d^{th}$ dimensional. $c_1$shows a constant and *rand* represents a single uniformly distributed random value within [0,1].

2. Set the novel velocity value to limit if it out-ranged the bounds of velocity.
3. Upgrade the location of $k^{th}$ cat base on Eq. (11):

$$X_{k,d} = X_{k,d} + V_{k,d} \tag{11}$$

In Eq. (11), $X_{k,d}$ denotes the location of $k^{th}$ cat in the $d^{th}$ dimension.

Using a Bi LSTM model and the Cat Swarm Optimisation (CSO) algorithm, the detection of clone nodes is a two-step process. The Bi LSTM model is initially trained to discover patterns and representations within the source code. The model predicts whether source code fragments are clone nodes based on their input. The Bi LSTM architecture is effective at preserving the sequential character of the code due to its capacity to capture both progressive and retrograde dependencies. Once the Bi LSTM model has been trained, the CSO algorithm is used to optimise the model's hyperparameters. These tuning parameters include learning rate, batch size, number of LSTM layers, number of hidden units, and dropout rate. The CSO algorithm initialises a population of candidate solutions, each of which represents a unique hyperparameter value combination. On a validation set, the objective function is the performance metric, such as accuracy or F1 score, of the Bi LSTM model.

The CSO algorithm revises the positions of the cats (candidate solutions) based on their previous positions and evaluations of objective functions. Adjusting the coordinates of the cats, the algorithm iteratively investigates the hyperparameter space, facilitating a balance between exploration and exploitation. During position updates, cats with higher objective function values are prioritised, imitating the behaviour of cats in nature. The CSO algorithm iterates until a termination criterion, such as a limit number of iterations or convergence of the objective

function, is met. The optimal set of hyperparameters for the Bi LSTM model is the finest solution found during the optimisation procedure. These optimised hyperparameters improve the model's ability to reliably detect clone nodes.

## 4. Parameter Settings

Clone node detection process results are discussed in Network Simulator (NS2) by means of parameters simulation and IEEE 802.11b is linked to the layer protocol of MAC. The simulation parameters shown in the Table 1. For this research work the maximum network size is considered as 100x100 with 100 nodes. We have also tuned the network environment can dynamically increase the number of nodes.

Table 1: Simulation Parameters

| S.No | Parameter | Value |
|---|---|---|
| 1 | Network size | Entire dataset |
| 2 | Number of nodes | Nodes in Dataset |
| 3 | Eaggregation | 5nJ/bit/signal |
| 4 | Packet size of normal node(pn) | 200 bits |
| 5 | Initial energy of each node | 1J |
| 6 | Broadcast range | 50 |

## 5. Results and Discussion

The suggested approach is verified using the Python programming language within an Anaconda environment. The evaluation employs a publicly accessible darknet dataset through cloning. To effectively identify clone attacks, a set of comparative parameters including classification accuracy, sensitivity, specificity, false ratio, and time complexity are utilized. The assessment leverages a confusion matrix. This section presents a detailed description of the outcomes and discussions stemming from the proposed techniques. The analysis encompasses a total of 30 services, with the corresponding parameters listed in
Table 2

**Table 2: Framework and values Evaluated**

| **Parameters** | **Values** |
|---|---|
| Simulation Tool | Anaconda, Jupyter notebook |
| Simulation language | Python |
| Name of the dataset | Clone Darknet dataset |
| No of users/ records | 100 records per epoch |
| Number of classes | High / low |

This section inspects the performance of the ICDC-CSODL systemon distinct measures. The outcomes are studied under clone attack.

**Table 3.**Classifier outcome of ICDC-CSODL system under clone attack

| Threshold | SVM Model | | FLCND | | DHT-DP | | ICDC-CSODL | |
|---|---|---|---|---|---|---|---|---|
| | AUC Score | Accuracy | AUC Score | Accuracy | AUC Score | Accuracy | AUC Score | Accuracy |
| 0.6 | 96.47 | 97.36 | 96.50 | 97.39 | 96.55 | 97.46 | 97.98 | 99.03 |
| 0.7 | 95.38 | 96.40 | 95.42 | 96.45 | 95.48 | 96.51 | 96.58 | 98.87 |
| 0.8 | 96.45 | 96.85 | 96.49 | 96.9 | 96.58 | 96.96 | 97.07 | 98.68 |
| 0.9 | 95.53 | 96.34 | 95.57 | 96.39 | 95.61 | 96.45 | 96.43 | 98.87 |

Firstly, Table 3 and Fig.5.1a represent the results under channel information data under clone attack. The results indicate that the ICDC-CSODL technique reaches increased AUC score value over the other models such as SVM, FLCND, DHT-DP.Similarly, Fig.5.2 b demonstrates the $accu_y$ results of the ICDC-CSODL technique with SVM model. The figure pointed out that the ICDC-CSODL technique reaches higher $accu_y$ values over SVM model. Next, Fig.5.3c illustrates the ROC results of the ICDC-CSODL technique and it is reported that the ICDC-CSODL technique achieved improved ROC values under all nodes.

The ROC it is used to assess the performance of a binary classification model, which can be applied to various classification tasks, including clone classification.

The ROC curve is constructed based on the True Positive Rate (Sensitivity) and the False Positive Rate.

1.True Detection Rate (Sensitivity, Recall)(TDR)

$$TDR = \frac{TP}{(TP + FN)} \tag{12}$$

2. False Acceptance Rate (FAR)

$$FAR = \frac{FP}{(FP + TN)} \tag{13}$$

SVM models achieve AUC scores ranging from 96.47% to 95.38% across a variety of thresholds, with accuracy levels that correspond to those scores falling somewhere between 97.36% and 96.40%. Based on these findings, it would appear that the SVM Model keeps up its reliable performance when it comes to identifying clone nodes. After that, FLCND, the algorithm, demonstrates performance that is comparable to that of the SVM Model. It attains AUC scores in the range of 96.50% to 95.42% and accuracy levels in the range of 97.39% to 96.45%. This demonstrates that FLCND is still a reliable method for clone node detection in wireless sensor networks (WSNs). The AUC scores for DHT-DP range from 96.55% to 95.48%, while the accuracy levels range from 97.46% to 96.51%. Both of these ranges are very encouraging. These findings show that DHT-DP continues to efficiently detect clone nodes in WSNs, presenting a feasible solution for the challenge that is being addressed here. ICDC-CSODL distinguishes out from the other algorithms that were studied due to its consistently

excellent performance. It achieves AUC scores in the range of 97.98% to 96.58% and accuracy levels in the range of 99.03% to 98.87%. According to these findings, ICDC-CSODL performs better than the other algorithms in terms of both the AUC score and the accuracy, further demonstrating its supremacy in the detection of clone nodes within WSNs.

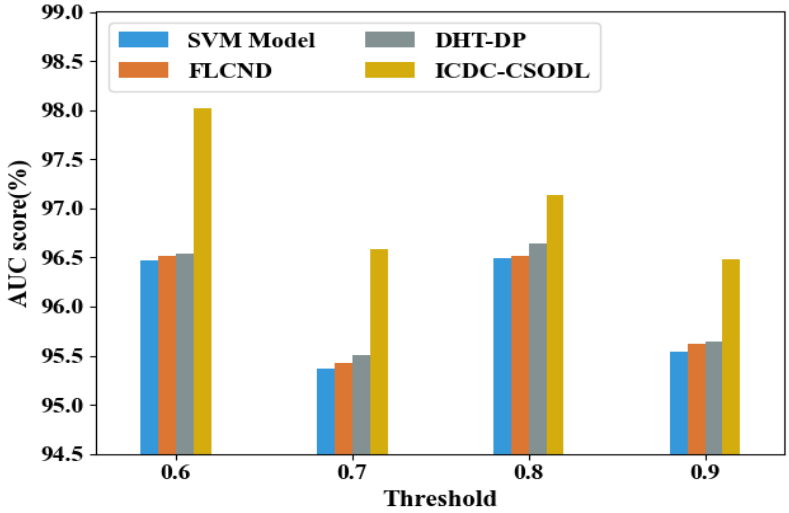


**Fig. 5.1.** a) Channel based AUC Analysis

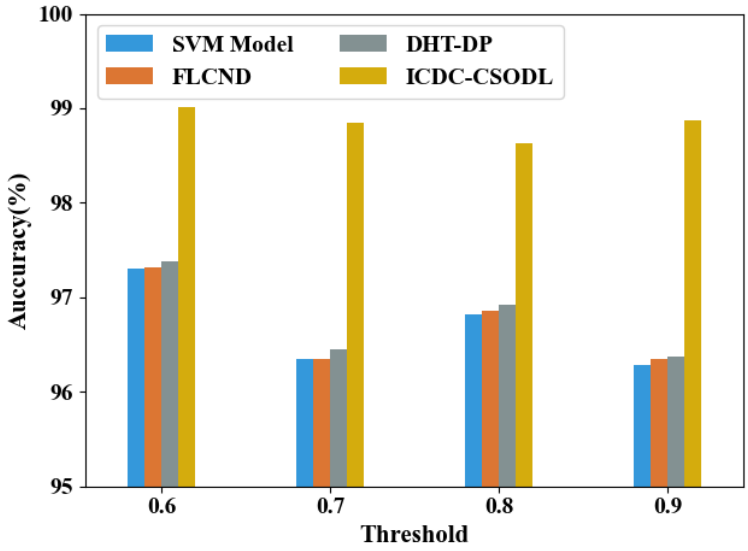


**Fig.5.2** b) Channel based Accuracy Analysis

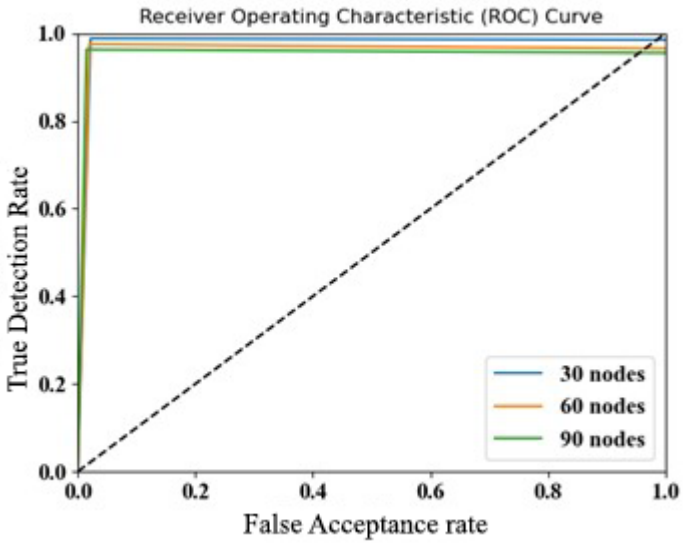


**Fig.5.3** c) ROC for detection clone attack under different nodes density

Firstly, Table 5 signify the outcome under ICDC-CSODL system on Clone attack detection . Fig. 5.4a displays the channel AUC score investigation of the ICDC-CSODL technique and SVM technique. The results of evaluating several clone node identification methods in terms of their AUC scores and their levels of accuracy at a number of different threshold values are presented in Table.3. SVM Model, FLCND, DHT-DP, and ICDC-CSODL are some of the methods that are evaluated and compared here. When the data are analysed, it is possible to see that all of the methods display relatively high AUC scores and accuracies. This demonstrates that the algorithms are effective at detecting clone nodes within wireless sensor networks (WSNs). Beginning with the SVM Model, it achieves AUC values ranging from 95.13% to 96.25% over a variety of thresholds, with accuracy levels that correspond to those scores falling somewhere between 98.10% and 98.47%. Based on these findings, it appears that the SVM Model is successful at identifying clone nodes on a consistent basis. Moving on to the FLCND method, the programme has performance that is comparable to that of the SVM Model. It attains AUC scores in the range of 95.16% to 96.29% and accuracy levels in the range of 98.15% to 98.50%. This lends credence to the notion that FLCND is a trustworthy method for clone node detection in WSNs. Additionally, DHT-DP demonstrates good outcomes, with AUC values ranging from 95.21% to 96.33% and accuracy levels ranging from 98.25% to 98.57%. According to these findings, DHT-DP is capable of effectively recognising clone nodes in WSNs, making it a viable solution for the task at hand. ICDC-CSODL distinguishes out from the other algorithms that were studied due to its consistently excellent performance. The AUC scores it obtains range from 98.49% to 96.86%, while the accuracy levels it reaches are between 99.43% and 97.55%. According to these findings, ICDC-CSODL performs better than the other algorithms in terms of both its AUC score and its accuracy, showing that it is superior when it comes to clone node detection within WSNs.

The outcome denotes that the ICDC-CSODL method reaches increased AUC score value over SVM approach. Similarly, Fig. 4b exhibits the $accu_y$ results of the ICDC-CSODL technique with SVM, FLCND, DHT-DP Models. The figure inferred that the ICDC-CSODL technique reaches higher $accu_y$ values over other compared system. Next, Fig. 4c exemplifies the ROC

results of the ICDC-CSODL technique and it is described that the ICDC-CSODL methodology attained greater ROC values under all nodes.

In Table 4, a detailed clone node attack detection result of the ICDC-CSODL technique is compared with the existing SVM model [21]. Fig. 5 illustrates the AUC score of SVM Model, it achieves AUC scores ranging from 97.15% to 96.06% across different thresholds, and Fig. 6 exemplifies the $accu_y$ investigation of clone node attack, the accuracy levels between 97.52% and 97.24%. These results suggest that the SVM Model maintains its consistent performance in detecting clone nodes. Then FLCND, the algorithm shows comparable performance to the SVM Model. It achieves AUC scores ranging from 97.57% to 96.09% and accuracy levels between 97.91% and 97.24%. This indicates that FLCND remains a reliable approach for clone node detection in WSNs.DHT-DP offers promising results with AUC scores ranging from 97.68% to 96.20% and accuracy levels between 98.02% and 97.37%. These findings imply that DHT-DP continues to effectively detect clone nodes in WSNs, offering a viable solution for this task.Among the evaluated algorithms, ICDC-CSODL stands out with consistently high performance. It achieves AUC scores ranging from 99.10% to 97.37% and accuracy levels between 99.71% and 97.58%. These results indicate that ICDC-CSODL outperforms the other algorithms in terms of both AUC score and accuracy, reaffirming its superiority in clone node detection within WSNs.

**Table 3 .**Comparative analysis of ICDC-CSODL system on Clone attack detection

| Threshold | SVM Model | | FLCND | | DHT-DP | | ICDC-CSODL | |
|---|---|---|---|---|---|---|---|---|
| | AUC Score | Accuracy | AUC Score | Accuracy | AUC Score | Accuracy | AUC Score | Accuracy |
| 0.1 | 96.06 | 97.24 | 96.09 | 97.27 | 96.20 | 97.37 | 97.37 | 99.10 |
| 0.2 | 97.15 | 97.52 | 97.19 | 97.57 | 97.30 | 97.68 | 97.84 | 98.35 |
| 0.3 | 96.63 | 97.86 | 96.67 | 97.91 | 96.78 | 98.02 | 97.58 | 99.71 |
| 0.4 | 96.58 | 97.98 | 96.62 | 98.03 | 96.73 | 98.14 | 98.66 | 99.52 |
| 0.5 | 96.55 | 97.89 | 96.6 | 97.94 | 96.72 | 98.05 | 97.32 | 99.00 |
| 0.6 | 96.20 | 96.19 | 96.24 | 96.24 | 96.35 | 96.35 | 97.97 | 98.33 |
| 0.7 | 97.41 | 97.32 | 97.47 | 97.37 | 97.60 | 97.48 | 98.11 | 97.95 |
| 0.8 | 96.06 | 97.24 | 96.1 | 97.29 | 96.21 | 97.4 | 97.37 | 99.10 |
| 0.9 | 97.15 | 97.52 | 97.18 | 97.55 | 97.29 | 97.65 | 97.84 | 98.35 |

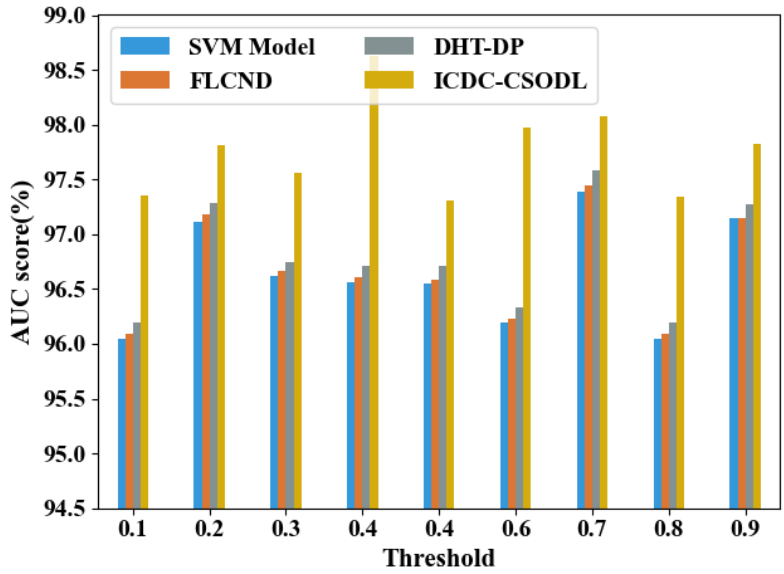


**Fig. 5. 4.** AUC of ICDC-CSODL system on clone attack detection

**6. Classification of accuracy**:

**(i)Data Preparation**

Convert the predicted labels (Ypred) and true labels (y_test) into pandas DataFrame objects.

**(ii) Noisy Labels Creation**

Get the values of the true labels (y_test) and calculate the number of values to flip (num_values) by taking 2.5% of the total number of labels.

Randomly select num_values indices from the true labels without replacement.

Create a modified label array (inverse_array) by copying the true labels and inverting the values at the selected indices.

**(iii)** Assessment of **Confusion Matrix:**

The confusion matrix is used to quantify and visualize the model's predictions compared to the actual ground truth

**(iv) Performance Metrics:**

a.The accuracy for each class (accurate) :

$$accurate = \frac{TP + TN}{TP + TN + FP + FN} * 100 \quad (14)$$

b. F1 score for each class (F1_score) by:

$$F1_{score} = 2 * \frac{TP}{2 * TP + FP + FN} * 100 \quad (15)$$

c.The precision for each class (precision) :

$$precision = \frac{TP}{TP + FP} * 100 \quad (16)$$

d.Recall for each class (recall):

$$recall = \frac{TP}{TP + FN} * 100 \quad (17)$$

e.False positive rate (FPR)

$$FPR = \frac{FP}{FP + TN} \tag{18}$$

f.False negative rate (FNR)

$$FN = \frac{FN}{FN + TP} \tag{19}$$

g.False discovery rate (FDR)

$$FDR = \frac{FN}{FN + TP} \tag{20}$$

h. true negative rate (TNR)

$$TNR = \frac{TN}{TN + FP} \tag{21}$$

i.The Misclassification Cost ($M_c$)

$$M_c = \frac{FP + FN}{accurate + 100} \tag{22}$$

**(v) Cohen's Kappa estimation:**

$$mrg_a = \frac{(TP+FN)(TP+FP)}{TP+FN+FP+TN} \tag{23}$$

$$mrg_b = \frac{(FP + TN)(FN + TN)}{TP + FN + FP + TN} \tag{24}$$

The expected agreement is found by

$$expec_{agree} = \left(\frac{mrg_a + mrg_b}{TP + FN + FP + TN}\right) \tag{25}$$

$$obs_{agree} = \frac{(TP+TN)}{(TP+FN+FP+TN)} \tag{26}$$

$$kappa = \frac{obs_{agree} - expec_{agree}}{1 - expec_{agree}} \tag{27}$$

$$acc = accuracy_{score(y_{test}, YPred)} \tag{28}$$

(vi)The overall accuracy of proposed ICDC-CSODL

$$accuracy = acc * 100 \tag{29}$$

accuracy→ Accuracy of Proposed ICDC-CSODL

(vii) **Accuracy of Proposed ICDC-CSODL**: 99.00017277744635

Accuracy of Proposed SVM: 97.50026894602499

Accuracy of FLCND: 85.0

Accuracy of DHT-DP: 80.0

## 7. Predicted Normal nodes and clone nodes

Fig 7.1 indicates average probability detection with Epochs compared with SVM, FLCND, DHT-DP, and ICDC-CSODL.The number of Epochs with average probability detection is increased in ICDC-CSODL methodology. Fig 7.2 indicates the Loss function with and without CSO and with CSO as the number of iterations increases. The loss with CSO is less compared with the loss without CSO.Fig 7.3 shows the graph compared with Model loss and Epoch

function suitable indication of training and test data set. Fig 7.5 indicates the predicted normal nodes with the clone nodes with its predicted label and true label as the confusion matrix . The number of normal nodes increases in its quadrant.

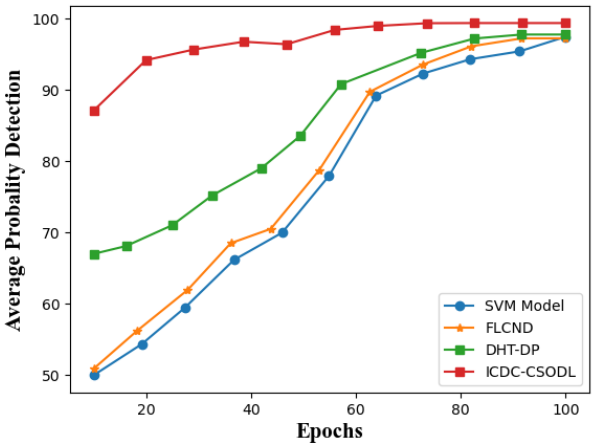


Fig.7.1 Average Probability detection

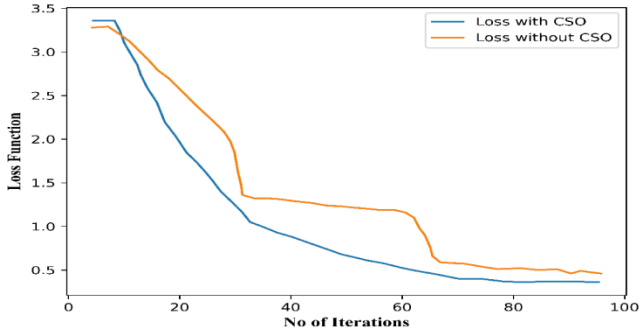


Fig.7.2 Optimization-Loss Function

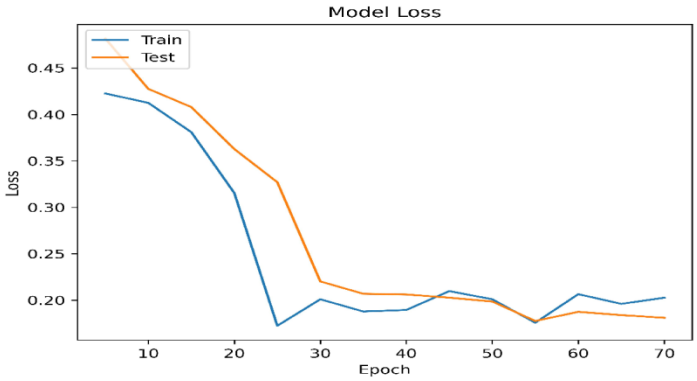


Fig.7.3 .Model Loss

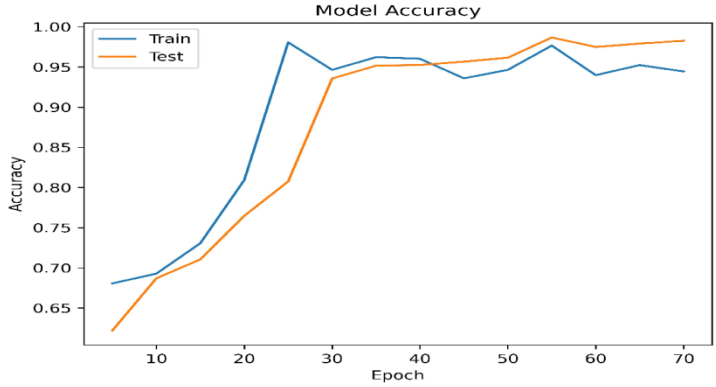

Fig.7.4 .Model Accuracy for Epoch

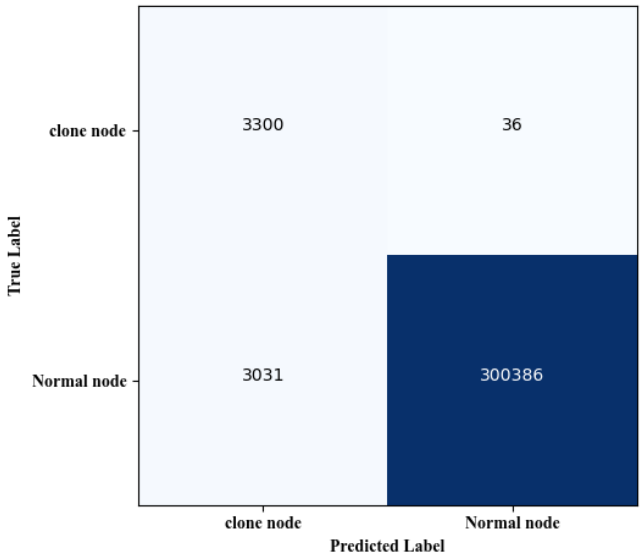


Fig.7.5 .Confusion matrix- ICDC-CSODL

## 5. Conclusion

In this study, we have presented the ICDC-CSODL technique to improve security in the WSN. The main aim of the ICDC-CSODL approach lies in the accurate detection and classification of clone nodes in the network. To accomplish this, the presented ICDC-CSODL system follows 2-stage processes such as ABiLSTM based clone node detection and CSO based hyperparameter tuning. At the primary stage, the ICDC-CSODL technique used the ABiLSTM model for clone node detection. Afterward, the CSO technique was utilized for adjusting the hyperparameter values of the ABiLSTM model. The simulation results of the ICDC-CSODL technique were tested in a series of experiments. A widespread simulation results analysis illustrated the improvement of the ICDC-CSODL technique in terms of different measures.

## References

[1] Xiao, X. and Zhao, M., 2022. Edge computing clone node recognition system based on machine learning. *Neural Computing and Applications*, pp.1-12.

[2] 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.

[3] Bajaj, K., Sharma, B. and Singh, R., 2022. Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor generated big data. *Complex & Intelligent Systems*, *8*(5), pp.3641-3658.

[4] Khosravi, H. and Rasoolzadegan, A., 2023. A Meta-Learning Approach for Software Refactoring. *arXiv preprint arXiv:2301.08061*.

[5] Piva, M., Maselli, G. and Restuccia, F., 2021, July. The tags are alright: Robust large-scale RFID clone detection through federated data-augmented radio fingerprinting. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic*

*Foundations, and Protocol Design for Mobile Networks and Mobile Computing* (pp. 41-50).

[6] Masood, M., Nawaz, M., Malik, K.M., Javed, A., Irtaza, A. and Malik, H., 2023. Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, *53*(4), pp.3974-4026.

[7] Vehabovic, A., Ghani, N., Bou-Harb, E., Crichigno, J. and Yayimli, A., 2022, June. Ransomware detection and classification strategies. In *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)* (pp. 316-324). IEEE.

[8] Bajao, N.A. and Sarucam, J.A., 2023. Threats Detection in the Internet of Things Using Convolutional neural networks, long short-term memory, and gated recurrent units. *Mesopotamian journal of cybersecurity*, *2023*, pp.22-29.

[9] Pajila, P.B., Julie, E.G. and Robinson, Y.H., 2022. FBDR-Fuzzy based DDoS attack Detection and Recovery mechanism for wireless sensor networks. *Wireless Personal Communications*, pp.1-31.

[10] Hanif, M., Ashraf, H., Jalil, Z., Jhanjhi, N.Z., Humayun, M., Saeed, S. and Almuhaideb, A.M., 2022. AI-Based Wormhole Attack Detection Techniques in Wireless Sensor Networks. *Electronics*, *11*(15), p.2324.

[11] Ahmad, U., 2022. A node pairing approach to secure the Internet of Things using machine learning. *Journal of Computational Science*, *62*, p.101718.

[12] Luo, Z., Wang, B., Tang, Y. and Xie, W., 2019. Semantic-based representation binary clone detection for cross-architectures in the internet of things. *Applied Sciences*, *9*(16), p.3283.

[13] Chen, D., Zhao, Z., Qin, X., Luo, Y., Cao, M., Xu, H. and Liu, A., 2020. MAGLeak: A learning-based side-channel attack for password recognition with multiple sensors in IIoT environment. *IEEE Transactions on Industrial Informatics*, *18*(1), pp.467-476.

[14] Yadav, C.S., Singh, J., Yadav, A., Pattanayak, H.S., Kumar, R., Khan, A.A., Haq, M.A., Alhussen, A. and Alharby, S., 2022. Malware Analysis in IoT& Android Systems with Defensive Mechanism. *Electronics*, *11*(15), p.2354.

[15] Ortin, F., Rodriguez-Prieto, O., Pascual, N. and Garcia, M., 2020. Heterogeneous tree structure classification to label Java programmers according to their expertise level. *Future Generation Computer Systems*, *105*, pp.380-394.

[16] Suragani, R., Nazarenko, E., Anagnostopoulos, N.A., Mexis, N. and Kavun, E.B., 2022, June. Identification and classification of corrupted PUF responses via machine learning. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (pp. 137-140). IEEE.

[17] Somayaji, S.R.K., Alazab, M., Manoj, M.K., Bucchiarone, A., Chowdhary, C.L. and Gadekallu, T.R., 2020, December. A framework for prediction and storage of battery life in IoT devices using DNN and blockchain. In *2020 IEEE Globecom Workshops (GC Wkshps* (pp. 1-6). IEEE.

[18] Song, J., Liu, Y. and Yang, J., 2023. Dam Safety Evaluation Method after Extreme Load Condition Based on Health Monitoring and Deep Learning. *Sensors*, *23*(9), p.4480.

[19] Alhnaity, B., Kollias, S., Leontidis, G., Jiang, S., Schamp, B. and Pearson, S., 2021. An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth. *Information Sciences*, *560*, pp.35-50.

[20] Ahmed, A.M., Rashid, T.A. and Saeed, S.A.M., 2021. Dynamic Cat Swarm Optimization algorithm for backboard wiring problem. *Neural Computing and Applications*, *33*(20), pp.13981-13997.

[21] Chen, S., Pang, Z., Wen, H., Yu, K., Zhang, T. and Lu, Y., 2020. Automated labeling and learning for physical layer authentication against clone node and sybil attacks in industrial wireless edge networks. *IEEE Transactions on Industrial Informatics*, *17*(3), pp.2041-2051.

[22] Neenu George,T.K.Parani.Detection of Node Clones in Wireless Sensor Network Using Detection Protocols. International Journal of Engineering Trends and Technology(IJETT)

[23] Sachin Lalar , Shashi Bhushan ,Surender Jangra ,Mehedi Masud , and Jehad F. Al-Amri. An Efficient Three-Phase Fuzzy Logic Clone Node DetectionModel. Hindawi, PP 2021, ID 9924478

[24] Salah Zidi, Tarek Moulahi, and Bechir Alaya.Fault detection in Wireless Sensor Networks through SVM classifier. In 2017 IEEE Sensors Journal, 2017.2771226