Enhanced Chicken Swarm Optimization And Improved Convolutional Neural Network Algorithm For Attack Detection Over Iot Based Wireless Sensor Network

Mrs. B. Dhivya[1] , Dr. S. Thavamani[2]

[1]Research Scholar (Part-time) in Computer Science and Assistant Professor, Department of Computer Science, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore, Tamil Nadu, India.
[2]Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore, Tamil Nadu, India.

ABSTRACT
A useful, adaptable, and interoperable network of electronics, gadgets, and objects has been developed named as the Internet of Things (IoT). IoT has emerged from its early days and is regarded as the most significant technology in changing the Internet into an entirely connected future Internet. Recent developments in computing, networking, communications, software, and hardware technology are the primary factors influencing it Utilizing the potential of IoT in useful applications and services, IoT employs Wireless Sensor Networks (WSN) to remotely gather, exchange, and distribute data. But, the existing system has issues with various and serious security attacks. Also, it has problem with Attack Detection (AD)accuracy for the given dataset. To overcome the abovementioned problems in this research, Enhanced Chicken Swarm Optimization and Improved Convolutional (NN) Neural Network (ECSO-ICNN) algorithm is suggested. Some of the primary stages in this study are the system model (SM), NSL-KDD Data Collection (DC), Cluster Head (CH), Node Selection (NS), data pre-processing, and AD. The amount of Sensor Nodes (SN), sensor devices, SN, destinations, and Multipoint Relays (MPRs) with neighbor and CH nodes that are one-hop and two-hop are all included in the system model. Next, the ECSO method is employed for selecting the CH node. It generates best Fitness Values (FV) by means of higher accuracy and lower Energy Consumption (EC) for the given IoT based WSN. With 42 features and class labels, the NSL-KDD dataset is regarded as a class of attacks. Then, data pre-processing is done by using filtering and Feature Selection (FS) process which is used to handle duplication and redundant features effectively for the given NSL-KDD dataset. The ICNN algorithm, which effectively detects attacks, is the last method utilized for AD. In terms of f-measure, accuracy, recall, and precision, the simulation results show that the suggested ECSO-ICNN strategy performs better than the existing approaches for AD.

Keywords—Enhanced Chicken Swarm Optimization and Improved Convolutional Neural Network (ECSO-ICNN) algorithm, Internet of Things (IoT), CH node selection (CHNS), Wireless Sensor Networks (WSNs), Attack Detection (AD).

I. INTRODUCTION

IoT is a new technological innovation that has made it possible to gather, process, and share data for advanced applications [1]. Since IoT is becoming widely used near the edge of networks for actual apps like eHealth and smart cities, these innovative features have drawn the attention of urban developers and medical professionals. But the implementation of these smart services has been limited by an increase in quantity and complexity of unidentified cyberattacks. IoT security becomes complexity and challenging in apps due to the dispersion and heterogeneity. Furthermore, the unique service demands of the IoT: low

latency, resource constraints, distribution, scalability, and mobility, to mention a few that make AD fundamentally different from the methods currently in use. These requirements cannot be addressed by the centralized cloud. It follows that the security issues associated with IoT cannot be resolved by using clouds or standalone AD systems (ADS) [2].

Smart cities, healthcare, education, finance, energy, and transportation are among the sectors where IoT applications are becoming more sophisticated [3]. Thus, academia, sectors, and individuals attempts to provide security and safetyfor IoT devices and networks. To prevent a data catastrophe for IoT users, these factors should be the main focus. Cybercriminals, for instance, have the ability to remotely monitor a smart home system and intercept smart vehicle transmissions to pose a threat to public safety. Due to its significant vulnerability to IoT, the catastrophe event could have an impact on networks of sophisticated communication net, including websites, applications, social net, and botnets (or robot networks). Conversely, the IoT-based system may become completely or partially ineffective as a result of collaborating on a single communication route or component. In 2016, a cyberattack known as Dyn collected connected devices to be installed in smart cities and assembled them into botnets, or Zombie Armies, using software called Mirai. The diversity and complexity of attack vectors against the IoT system are also increasing, in addition to its vulnerabilities [4]. Consequently, WSN are regarded as a collection of resources that require SN to collect information from the background, process the format into a structured form, and then send the formatted data over a wireless channel to the specified terminal. The sensed data collected from several sensor types, including flow, pressure, magnitude, level, and temperature sensors, among others, will serve as the input source [5]. Compared to wired networks with robust architecture, wireless networks are vulnerable to attacks against attackers because of their vulnerable aspect.

In order to detect lights, electromagnetic signals, chemical or biological vapors, hostile existence, or boundary violations, WSN [6] provide crucial communication in combat zones or defense-oriented applications. Ensuring Energy-Efficient (EE) security in WSN, when nodes are moving is a challenging task. Because, in terms of protective aspects, controlling the location of roaming attackers and SN is crucial. There are various instances where attackers initiate attacks based on their objectives or without a clear motive. With the proper hardware and software, any wireless SN may sense a wireless channel and collect data being transmitted without authorization. Additionally, attackers may attempt to disrupt and alter the typical behavior of SN in order to disable WSN operations. As a result, the performance, throughput, and service of the SN could fall [7].

Data Aggregation (DA) is an extremely important approach in WSN, and by removing redundancy, it can help lower (EC) Energy Consumption. Collecting and combining the relevant data is the method of DA. One of the basic processing techniques for reducing EC is DA [8]. An effective way for preserving the scarce resources in WSN is by DA. Enhancing Network Lifetime (NL) is the primary objective of DA techniques through EE DC and DA. It is important to spread the load effectively when choosing a CH node in each cluster because CHs need additional energy to do their functions. The Base Station was received by CH after computing the aggregate for its cluster. The BS computes the total aggregation for the whole network using the data from each CH. Clustering is simply scalable, robust in the event of node failures, and has been demonstrated to significantly reduce EC.

There are numerous chances for companies as a result of the growth that connects worlds to consumers. In order to get around IoT network security, it additionally permitted hackers to investigate and employ various methods for attacking. Companies and customers may suffer significant financial and reputational losses as a result of security vulnerabilities on a network. Thus, maintaining the networks' security is crucial. The networks could be IoT-related or not. Spam filters, firewalls, and antimalware programs are just a few of the tools and solutions available to address network attacks of different types. Digital Forensics, Auditing, Log Analysis, Intrusion Detection Systems (IDS), Unified Threat Management (UTM), Intrusion Prevention Systems (IPS), and Access Control (AC) are a few instances of diverse tools and approaches (RM).

The IDS may out to be a vital and very helpful security solution for ensuring the security of the IoT network [9]. [10]. Identifying source IP, destination IP, and attack types, an IDS identifies network traffic attacks and notifies the network administrator.

AD over IoT-based WSN is the objective of this study. Although numerous studies and approaches have been presented, there has been little advancement in AD. The current methods for IoT-based WSN have drawbacks in terms of accuracy and precision. In this research, the ECSO-ICNN technique is presented for enhancing the overall performance of the IoT-WSN framework to solve the aforementioned

difficulties. The SM, CH NS using the ECSO method, NSL-KDD data collection, data pre-processing, and AD via the ICNN algorithm are the primary contributions of this study. The suggested technique uses efficient techniques for the IoT-WSN to deliver benefits in terms of cost-effectiveness, efficiency, and dependability.

The remaining portions of the study is arranged in this way: In Section 2, an in-depth review of the literature on CHS, AD, and the NSL-KDD dataset in IoT-based WSN is provided. Section 3 provides specifics on the suggested methodology for the ECSO-ICNN method. The experimental data and result os performance analysis are presented in Section 4. Section 5 provides a summary of the conclusions.

II. RELATED WORK

The Random Forest (RF) and Synthetic Minority Over-sampling Technique, a novel technique implemented for AD in an IoT network, and (RF-SMOTE) was presented by Karthik, M. Ganesh, and MB Mukesh Krishnan (2021) in [11]. The experimental analysis for IoT AD in this paper evaluates two extensively used datasets: Network-Based detection of IoT (N-BaIoT) and NSL-KDD. During the experimental stage, the RF-SMOTE framework improved accuracy for Binary Class (BC) by a minimum of 0.14% and a maximum of 14.25% on the NSL-KDD dataset. Furthermore, the model demonstrated an average accuracy improvement on the dataset for 4 classes ranging from 0.04% to 7.35%.

To identify different types of attacks on IoT networks, Keserwani et al. (2021) addressed IDS in [12]. Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) have been combined for obtaining pertinent IoT network features. To attain high AD accuracy, RF classifier gets the retrieved data. In order to evaluate the model using the Python programming framework, the experiments used the datasets KDDCup99, NSL–KDD, and CICIDS-2017. For MultiClass Classification (MCC), the GWO–PSO–RF NIDS framework has demonstrated improved accuracy. To demonstrate the model's efficacy, its accuracy has been compared to other comparable methods. The data imbalance problem is addressed by the work.

The development of visual tools for identifying security risks in IP-enabled WSNs is the main goal of Tsitsiroudi et al. (2016)'s [13]. A human-interactive visual anomaly detection system, EyeSim is a tool that can monitor and quickly notify users if wormhole (WH) links are present. It can also identify the malicious nodes (MN) that make up the WH link. Through the application of Dynamic Routing (DR) information in cognitive network (DA) Data Analysis, EyeSim has the ability to identify attackers. The accuracy of detection is used to evaluate EyeSim's effectiveness.

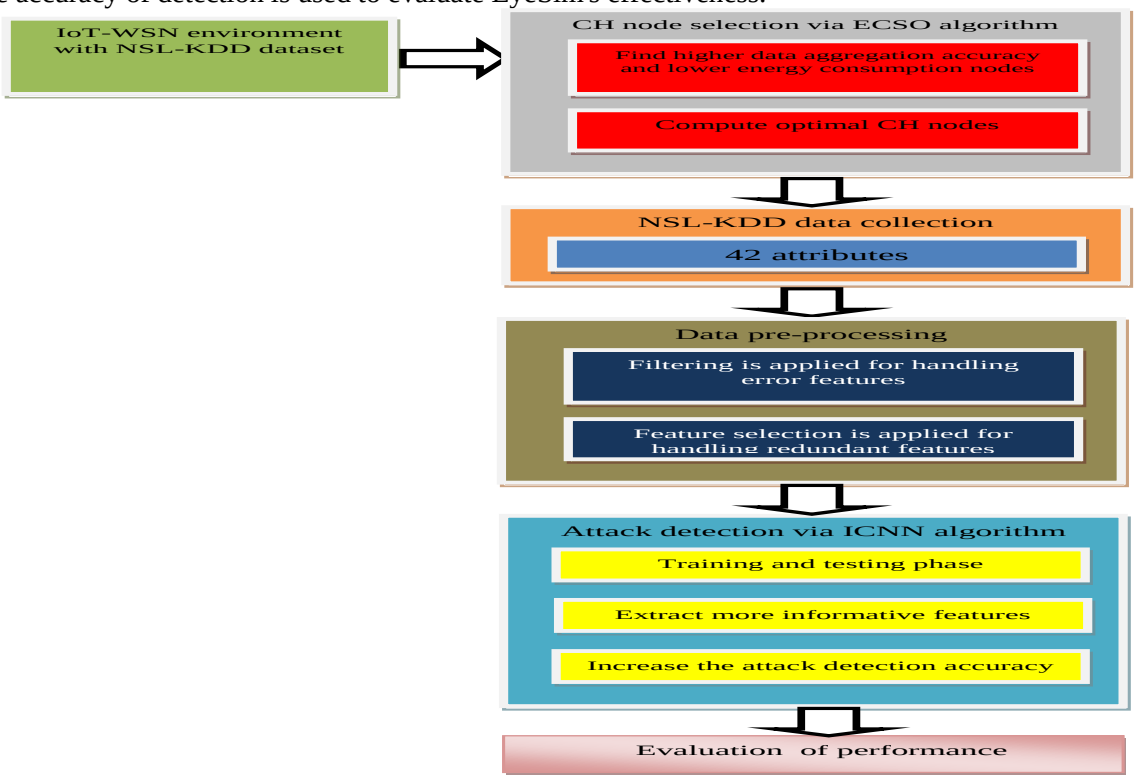

Fig 1. Overall block diagram of the suggested technique

The outcomes of the simulation demonstrate EyeSim's capacity to quickly and reliably identify several wormhole risks.

A hybrid optimization methodology was created, as it contains (Metaheuristic lion optimization (ML) and Firefly optimization (F) method) and it was recommended by Krishna et al. (2021) in [14]. The NSL-KDD and NBaIoT datasets are employed to eliminate noise and data that is missing from the input data by means of pre-processing. Then, Recursive Feature Elimination (RFE) was employed by Feature Extraction (FE). After the data separation procedure, the low rate attacks are selected by using this approach. A RF classifier is employed to classify the attacks after the features have been selected. In terms of attack classification, the hybrid ML-F approach performs better than the current Gradient Boost (GB) classifier approach. Recall, f-measure, accuracy, and precision were all attained via the hybrid ML-F approach.

According to Alqahtani et al. (2020) in [15], there is a practical and efficient way to identify IoT botnet attacks.

The Genetic-based extreme GB (GXGBoost) model with a Fisher-score-based FS technique was combined in this approach to identify the most pertinent features and bottle attacks. A representative filter-based FS technique called the Fisher score maximizes inter-class distance and minimizes intra-class distance, which helps to find significant features and eliminate unimportant ones.In contrast, GXGBoost is a highly efficient model that is employed in the classification of IoT botnet attacks. An open botnet dataset of IoT devices was used for multiple studies. Utilizing just three of the 115 data traffic parameters, the approach demonstrated a high detection rate and enhanced the total efficacy of the IoT botnet AD process, according to evaluation outcomes obtained through the holdout and 10-fold (CV) Cross-Validation approaches.

A lightweight and EE function-based DA solution for a cluster-based hierarchical WSN has been presented by Jan et al. (2020) in [16]. Both the node level and the CH level are served by this method. To improve the accuracy of the aggregation, DA is applied at the node level using Exponential Moving Average (EMA), and any outliers are detected using a threshold-based method. To transmit highly accurate DA to the BS, it used a modified Euclidean Distance (ED) function at the CH level. The simulation outcomes indicate that the method lowers communication, transmission, and EC at the nodes and CH while providing greatly optimized, DA to the BS.

In 2022, Taher et al. introduced the novel Tunicate Swarm Algorithm (TSA) in [17], which utilized a recurrent neural network (NN) and long-short-term memory (LSTM). Then, pre-processing the input data into a format is the initial step and it can be employed for achieving this. Additionally, a model based on LSTM RNN is employed to identify attacks in the IoT environment. In ANN models, there is a direct relationship among the complexity and count of parameters (P) and the frameworks performance. To prevent over- or under-fitting, it is essential to monitor the count of P in every model layer. Changing the number of levels in the data structure serves as a technique for preventing this from occurring. The LSTM-RNN model's Hyper-Parameter (HP) values are adjusted using the TSA to enhance the model's finding performance. TSA is utilized to resolve several issues that conventional optimization techniques were unable to resolve. It also lowered the algorithm's convergence time and increased performance. Benchmark datasets were used for a number of experiments. The TSA-LSTMRNN model outperformed similar models as demonstrated by its increased accuracy, recall, and precision, respectively.

III. PROPOSED METHODOLOGY

For the purpose of improving the total efficiency of the IoT-WSN system, the ECSO-ICNN algorithm has been suggested in this study. The system model, CH NS via the ECSO method, NSL-KDD data collection, data pre-processing, and AD via the ICNN method are the primary contributions of this study. Fig. 1 presents the suggested system's overall block diagram.

A. System model

An instance of an IoT application utilizing WSN for tracking the environment is presented in this section. It utilizes sensor devices, SN, and CH nodes to build an IoT based WSN model. Clusters are groups of SN within the network. All have few CH and a few Member Nodes (MN). Data sensing for attention is carried out by each MN in the bottom layer. Each group should have SN, with a single node having the ability to become the CH at any given moment. The end user receives the fused data that the sink node has gathered from the CHs of every group.

In order to collect, integrate, and forward data from MN, CH creates the routing backbone in the middle layer. Data from CHS is relayed to the server via the BS in the upper layer. The scalable and energy-efficient IoT is supported by this deployment. An effective way to preserve energy is to put IoT components above this framework. V is the collection of wireless links that connect the nodes, and N is the collection of all the nodes in the background, G(N,V) represents an IoT network. The relay nodes' communication radius is denoted by R, while the local nodes' communication radius is represented by r.

With regard to their efficiency and level of data, a certain group of attackers may be able to access the network through such nodes. It is assumed that there is no BS and that network is random. Every node is set up with functions that have a similar importance. In terms of their location, velocity, type of work (monitors, routers, idle), level of energy, etc., these nodes are selected and generated as random sets of assigned weights. It features MPRs with one-hop neighbor and two-hop neighbors, SN, and destinations. Forwarding SN, which are constantly changing from various transmitting sensors, are used by several receivers for receiving data.

B. CH node selection using Enhanced Chicken Swarm Optimization (ECSO) algorithm

An approach ECSO is employed in this study for selecting the CH nodes. Clustering is one of the effective process which are applied in the IoT based WSN to preserve the precious battery power of SN. Due to the great energy efficiency, clustering extends the life of WSNs. The following is a list of WSN clustering advantages: (i) very low EC (ii) To minimize the number of broadcast nodes linking BS, collected data or compressed data is transmitted directly between CH and BS. SN are grouped together into virtual groups called clusters during the clustering process, and nodes inside distinct clusters carry out distinct tasks. Organizing nodes into clusters according to certain standards and selecting the most effective node as the CH from every cluster is known as clustering.

The IoT-based WSN's ECSO technique is employed in this section the CH nodes and DA. In WSN, the CH-based DA plays a crucial role in lowering the EC of individual SN.

CSO, a novel bio-inspired algorithm (BIA), is suggested for use in optimization settings.In order to optimize problems, CSO extracted intelligence from the chicken swarm (CS) by mimicking the hierarchical order and behaviors of the CS, such as (R)roosters, hens (H), and chicks (C).

Then, IoT-based WSN performance was maximized by the determination of best node combination, ECSO is employed in CH NS.

The technique simulates both the distinct chickens' behaviors and the hierarchical structure of CS. A CS is hierarchically ordered into multiple groups, with one rooster and numerous hens and chicks in every group. The laws of motion that apply to different types of chickens varies. Chickens' social lives are significantly influenced by a hierarchical order. Stronger hens will dominate weaker ones in a group. The more sub missive H and R that has to be positioned at the outside of the cluster coexist with the more dominating H that stay near the head roosters. The CSO technique's nature is presented in Fig 2.

Fig 2. Nature of CSO procedure

Chickens Movements

Rooster Movement: Higher FV enable roosters to look for food in more locations.

Hen movement: In search of food, hens follow the roosters in their group. Even though the other chickens would suppress them, it continued to randomly steal the tasty food that they found. In food search competition, the more dominant C had power over the timid ones.

Chick movement: The chicks approach their mother in an attempt to get food.

The following recommendations, that sum up the behaviors of the chickens, form the basis of the mathematical framework of CSO employed in [18]:

1) There exists different groups within the CS. A dominant R, couple of H and C.

2) The roosters, who are the group leaders, have the highest FV among the chickens, while the chicks have the lowest FV. This indicates the hierarchy of the swarm. The hens would be the others.

3) In a group, the mother-child bond, swarm hierarchy, and dominance relationship won't change. Only many time steps are required for these status updates.

4) The number of R can be denoted as R_n , number of H can be denoted as H_n , number of C can be represented as C_n , and number of mother hens can be denoted as M_n, make up the n virtual C of the Swarm. Positions in a D-dimensional space are used to represent each individual which is given in equation (1)

$$x_{i,j}(i \in [1,\dots,N], j \in [1,\dots,D]). \quad (1)$$

When it pertains to food access, the roosters with higherFV are given preference over those with lower FV. The situation wherein R with greater FV have the ability for food searching in a greater variety of locations than those with lower FV can be used to simulate this instance, for simplicity. Since the error rate in the normal CSO is an issue, the Gaussian distribution is employed for the purpose of error rate reduction. Thus, the ECSO should be written by equation (2) and (3)

$$x_{i,j}^{t+1} = x_{i,j}^{t} * (1 + Randn(0,\sigma^2)) \quad (2)$$

$$\sigma^2 = \begin{cases} 1, & if\ f_i \le f_k \quad k\epsilon[1,N], k \ne i \\ exp\left(\frac{(f_k - f_i)}{|f_i|+\varepsilon}\right), & otherwise \end{cases} \quad (3)$$

The lowest constant in the system, ε, is employed to prevent zero-division error. A R index, denoted by k, is chosen at random from the group of R, and the corresponding x FV is represented by f. With standard deviation (SD) σ^2 and mean 0 of Gaussian distribution is called $Randn(0,\sigma^2)$.

The hens can get food by following the roosters in their group. Even though the other chickens would suppress them, they would also haphazardly steal the tasty food that they found. In food search competition, the more dominant C had power over the timid ones. The following equation (4) are (5) are the mathematical formulation for these phenomena.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S_1 * rand * \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + S_2 * rand * \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \quad (4)$$

$$S_1 = exp\left((f_i - f_{r1})/(abs(f_i) + \varepsilon)\right) \quad (5)$$

$$S_2 = exp\left((f_{r2} - f_i)\right) \quad (6)$$

Here, the uniform random number over [0, 1] is called $rand$. A random selection of the C (R or H) is made from the swarm $r1 \ne r2$, and its index is $r2\epsilon$[1,..N]. The rooster, or ith hen's group-mate, is represented by $r1\epsilon$[1,..N].

It follows that S2 <1< S1since $f_i > f_{r1}, f_i > f_{r2}$. Presuming S1=0, the ith hen would go in quest of food first, then the other chickens. The S2 value decreases and the distance among the two H positions increases with the size of the difference in FV among them. Consequently, there would be a decreased likelihood of the hens stealing food found by other hens.

There are group competitions, which is why the formulation method for S1 is different from that of S2. For the sake of simplicity, competitions between a group of chickens are used to simulate the FV of the C in relation to the R's FV.

When S2=0, in its own territory ith hen would look for food. The rooster's FV is distinct for that particular group [19]. Therefore, the closer S1 approaches to 1 and the narrower the distance is between the locations of the ith H and its group-mate R, the lower the FV of the ith hen. Therefore, there's a greater chance that the dominant hens will eat the food than the submissive ones.In order to find food, the C move towards mother. This is expressed as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL * \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \quad (7)$$

In addition the mother of the ith C position ($m\epsilon[1,N]$)is denoted by $x_{m,j}^{t}$. The C would follow its mother to search for food if $FL(FL\epsilon(0,2))$ were a parameter. With each chick's FL choosing a random number between 0 and 2, take into account their distinct variations.

To optimize the NL for a given order of observed locations, the optimization problem's objective is to effectively aggregate the data and select the best CH at each sink position [19]. With the use of the linear programming approach h, it offers a longer lifespan for networks. In this work, the ECSO method is used for effective DA and appropriate CHS. Here, aggregation accuracy and energy are taken into consideration as FV. In order to guarantee that CH is constantly available with the necessary energy level, CH will be dynamically updated for each specific time period. By doing this, errors in DA caused by dead nodes are prevented. To ensure an accurate aggregation result, a specific amount of cluster size maintenance is implemented.

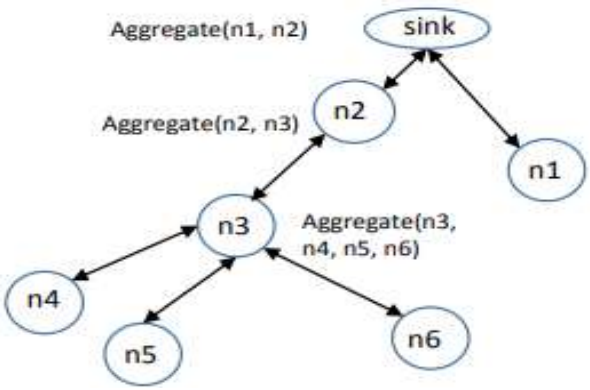

Fig 3. Example of data aggregation process

Macro clusters will split into micro clusters when there are more cluster members with more data to be combined.

Every micro cluster will therefore carry out DA, improving the aggregation process' accuracy. In Fig. 3, a DA process is demonstrated.

Node n3 contains aggregated data from nodes n4, n5, and n6 in this case. Following transmission to n2, this aggregated result is combined with n2 data. Ultimately, the sink node aggregates the data from n2 and n1. Fig. 4 demonstrates the WSN's CH-based DA.

Algorithm 1: ECSO

Input: a population of n chickens (IoT based WSN)
Objective: Best solution (higher DA accuracy and lower EC)
Output: Optimal CH NS

1. Initialize the parameters such as R_n, H_n C_n, and M_n
2. Calculate the N chickens' FV, t=0; (higher DA accuracy and lower EC)
3. While $t < Maximum$ iteration do
4. If t%G==0 then
5. Rank the node's FV and in the swarm , a hierarchal order is created
6. Split the swarm into various flocks and find the link among the C and MH in the flock
7. End
8. For i=1 to N nodes (higher DA accuracy and lower EC) do
9. If i==rooster then
10. Update the solution(S) via (2)
11. End if
12. If i==hen then
13. Update the S via (4)
14. End if
15. If i==chick
16. Update the S via (7)
17. End if
18. Calculate the novel S
19. If the new generated S is superior than its prior individual, update it
20. End
21. End
22. Return x_{best} (better CH node selection)

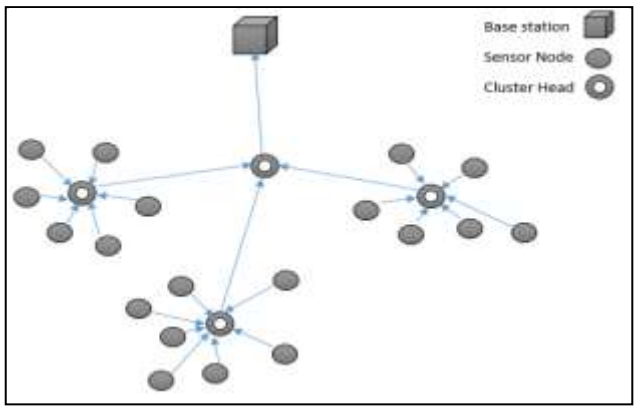

Fig 4. CH based DA in WSN

The ECSO technique is used in technique 1 to efficiently aggregate the number of data packets while selecting the optimum CH node. In Algorithm 1, the major node of the IoT-based WSN is improved by updating the chicken with the best FV. Determine the optimal values for every chicken (SN) and use equation (7) to identify the best hens (CH node) with the global best searching food process. The sink node receives the DA from the CH in this study. The suggested solution uses the ECSO method to identify node combinations that optimize IoT-based WSN performance while requiring the fewest possible hop counts. ECSO considers the parameters are such as lower energy consumption and higher data aggregation accuracy nodes which select the best CH node over the given IoT based WSN

C. NSL-KDD DC

The NSL-KDD cup dataset is taken into consideration in this study for IoT-based WSN AD. A training set and a testing set have previously been created from the sizable network traffic dataset NSL-KDD. Additionally, the performance of the suggested approach's detection utilizing semi-supervised ML approaches for attack classes having 42 features and class labels is evaluated using this dataset as a benchmark. 41 features are divided into 4 categories: content, host, traffic, and fundamental features. As presented in Table I, the dataset comprises 148,515 samples in total, of which 80% are training samples and 20% are testing samples. These samples correspond to four different attack classes. The dataset is divided into normal and pathological clusters in order to extract the vector features for training. After training, normal or abnormal clusters can be classified.

TABLE I. Different attack classes

Samples	Normal	DoS	Probes	U2R	R2L	Total
Training samples	67,343	45,927	11,656	52	995	125,974
Testing samples	9,711	7,458	2,421	200	2,654	22,544
Total number	77,054	53,385	14,077	252	3,649	148,518

The attack dataset is divided into four categories: DoS, probing, remote to local (R2L), and user to root (U2R).

There are 23 classifications of attack types in the dataset. The DoS attack prevents the authorized user from accessing the network and overloads the network service [20]. The U2R attack uses the legitimate user's passwords to sniff and cause the host system to become vulnerable. The network host's system is remotely compromised by the R2L. In violation of the security rule, the probe attack searches the network for data gathering and collection. While the others only have one link, there are many connections between DoS attacks and probing. Table II presents an overview of the 4 attack types found in the NSL-KDD benchmark dataset.

Table II. Attack classes description

Attack	Attack Description in the dataset
DoS	The attacker makes the network busy and denies the legitimate user access.

R2L	The intruder tries to gain access to the network or machine for a specific version of the FTP.
U2R	The attacker accesses the system's root and makes unauthorised attempts to the network.
Probe	It endeavours to assemble the data behind evading the security of the system

Additionally, an attack that corresponds to one of the five groups is identified on every record [20].

1. Normal: Anything that falls outside of the attack category is considered normal network traffic. In a binary classification, there are two labels: normal and anomalous.
2. Probe: As a means of getting data on a network and especially for getting beyond security measures, probing attacks include some form of surveillance. Take port scanning, for instance, ipsweep, mscan, nmap, portsweep, saint, and Satan are some of the attacks included in the dataset.
3. DoS: By overloading a network with pointless requests, an attacker can deplete its resources through DoS attacks. Pod, processtable, Neptune, back, land, mailbomb, smurf, teardrop, and upstorm are among the attacks included in the dataset.
4. U2R: Through the usage of vulnerabilities, the attacker with a regular user account is able to obtain full rights in these attacks. Buffer_overflow, load module, perl, rootkit, ps, sqlattack, and xterm are among the attacks found in the dataset.
5. R2L: These attacks involve a remote attacker gaining access to a local network machine in some way. ftp write, guess_password, imap, multihop, named, phf, send mail, snmpgetattack, snmpguess, warezmaster, worm, xlock, xsnoop, and http-tunnel are some of the attacks included in the dataset.

D. Data pre-processing

The level of quality of the datasets employed for training a ML model can impact the model's performance. The framework capacity to identify structures, generalize to novel information, and generate precise predictions can all be strongly impacted by the quality of the datasets. By eliminating errors and standardizing the setup, preprocessing the raw data can improve the efficacy of the ML model.

Filtering: Filtering data is one technique for cleaning it. Using a distorted signal pattern as input, this filter approximates a desired signal pattern accurately. Reducing the Mean Square Error (MSE) among the desired and predicted signal patterns is the main objective of this filtering technique.

Feature Selection: In order to improve prediction accuracy, minimize overfitting, training time, and complexity, and select useful and pertinent features for model learning, FS is crucial. Numerous techniques are used in the FS process. Equation (6) demonstrates the application of Spearman's Rank Correlation Coefficient (SRCC) calculation for a recursive FS method that subsequently dynamically selected features.

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2} \qquad (8)$$

Here, in equation (8) the mean values of x is symbolized as $\bar{x}$ and mean value of y is denoted as $\bar{y}$. The feature variables are denoted as x_i and y_i. ρ is the correlation coefficient

E. AD using ICNN algorithm over IoT based WSN

ICNN is used over IoT-based WSN for AD in this study. The ICNN training and testing phases use the selected features as input. A basic CNN consists of multiple Hidden Layers (HL), an Output Layer (OL), and an Input Layer (IL).

Convolutional Layer (CL), pooling layer, and Fully (FC) layers are typically included in a CNN's HL.

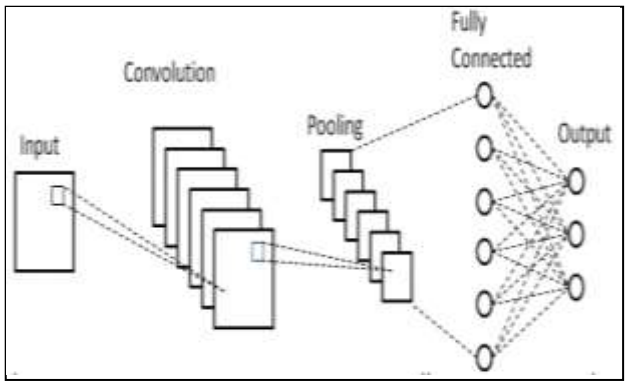

Fig 5. Basic CNN structure

CL sends the output to the following layer by applying a convolution process on the input. A single neuron's response to visual inputs is replicated by the convolution.

To merge the outputs of neuron clusters in one layer into a single neuron in the next layer, local or global pooling layers are used and is considered to be the particular feature of convolutional networks [21]. The average value for every cluster of neurons in the previous layer can be employed by mean pooling. Every neuron in one layer communicates with every other layer's neuron through FC layers. To obtain accurate outcomes, the weight values of the features in this suggested ICNN are optimized. The IL,CL, and FC layer make up the suggested ICNN. Analyzing high-dimensional data with this suggested strategy provides distinct advantages. Fig. 5 represents the fundamental structure of CNN.To effectively send the data to the following layer, the IL takes the incursion features from the training samples and integrates the data. Additionally, this layer specifies the initial parameters, including the size of the various filters and local receptive fields.

Using a convolution procedure, the intrusion (input) feature is processed by the convolution layer (Cx). The convolution calculation results from the preceding levels are combined to create FM (Feature Map), which is made up of multiple layers. Its primary functions are FE and network computational complexity reduction.

Each CL is followed by an (AF) Activation Function. An AF is a mapping function that creates a non-linear network structure by mapping an output to a set of inputs. The initial connection weights are based on all feature values that are supplied. After that, a new input pattern is used, and the result is calculated in equation (9) and (10)

$$y(n) = f(\sum_{i=1}^{i=N} w_i(n)x_i(n) \qquad (9)$$

$$\text{Where } f(x) = \begin{cases} +1 & if\ x \geq 0 \\ -1 & if\ x < 0 \end{cases} \qquad (10)$$

The iteration index can be denoted as n

Updated connection weights are based on

$$w_i(n+1) = w_i(n) + \eta(d(n) - y(n)x_i(n), \quad i = 1,2,..N \qquad (11)$$

The gain factor can be denoted as η in equation (11)

Next, apply SD in equation (12)

$$\sigma = \sqrt{\frac{1}{n} f_i(x_i - \overline{x)^2}} \qquad (12)$$

More precise AD outcomes are obtained by the ICNN network with these weighted features. The polynomial distribution function validates the outcomes of the study, that is conducted with the similar set of attack features. Because it minimizes the amount of connections among CL, the pooling layer lessens the computational load. Additionally, pooling layers enhance the receptive field of following CL and strengthen the translation invariance features. At the conclusion of the network's convolutional stream, one or more FC layers are often added, and errors are measured for training purposes using a loss function.

The suggested ICNN structure is shown in Fig. 6.

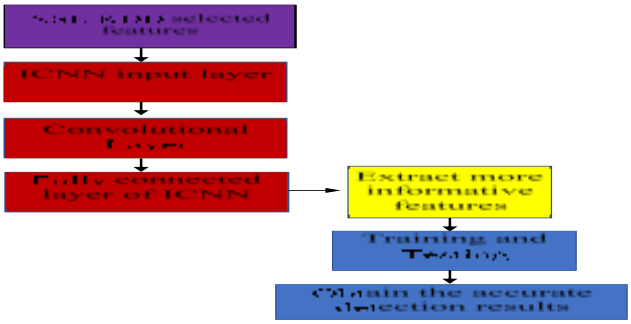

Fig 6. suggested ICNN structure for AD

FC layer: The size of the output FM gradually reduces after the feature passes through multiple CL. As a feature vector for this layer, each FM is made up of a single neuron. A classifier is FC to the vector. When a neuron is FC, all neurons in the previous layer are connected to all neurons in the subsequent layer. By classifying the attacks based on the best classifier values, the ICNN introduces an innovative feature.

Steps: ICNN for attack detection

1. Procedure NSL-KDD dataset
2. For all input feature, describe feature ∈ NSL-KDD dataset do
3. Convert the input into sub layers
4. Detect attack features
5. Extract additional beneficial and relevant features based on DoS, U2R, R2L and prob
6. For given NSL-KDD dataset, do training and testing procedure
7. Identify the attack features using IoT sensor devices in CH nodes over WSN
8. For every feature in the input dataset, copy the specified feature label.
9. Categorize more accurate attack results

For secure and energy-efficient routing in IoT-WSNs, an optimal CHS procedure is employed, which is based on ICNN. The lifespan of the network will constantly be maximized and the EC of individual SN will be reduced by an optimal group of CHs. Network adaptability is maintained in the context of dynamically introduced or updated features due to ICNN. The data of each feature is calculated, and the CL maps the feature with the highest value. It is best to detect and eliminate any malicious nodes throughout the CH election. It is employed to identify malicious nodes that are utilized in (DT) Data Transmissions to the BS.

IV. SIMULATION RESULT

In this work, NSL-KDD attack detection dataset is analyzed and implemented using Matlab. The performance metrics are such as accuracy, precision, recall and f-measure compared by using existing RNN, TSA-LSTMRNN and proposed ECSO-ICNN algorithms. Table III indicates the parameters and Table IV indicates the comparison values of current and suggested approaches.

TABLE III. PARAMETERS

Parameter	Setting	Parameter	Setting
Base Station	1	Topology	Hierarchical
Fixed size	1500m × 1500m	Number of attacks	2
Number of Nodes	200	Mobility Mode	Random
Protocol type	Routing	Number layers	10
cluster size	10	Max epochs	200
Attack type	wormhole	Data size	5000Kb

Number iterations	200	Simulation Time	5s

Accuracy

The accuracy of the framework is calculated in equation (13) by dividing the total number of true classification parameters ($T_p + T_n$) by the total number of classification parameters ($T_p + T_n + F_p + F_n$).

T_p *is True positive,*

Tn stands for True Negative

F_p stands for *False* positive

and F_n is False negative

$$\text{Accuracy} = \frac{T_p+T_n}{(T_p+T_n+F_p+F_n)} \qquad (13)$$

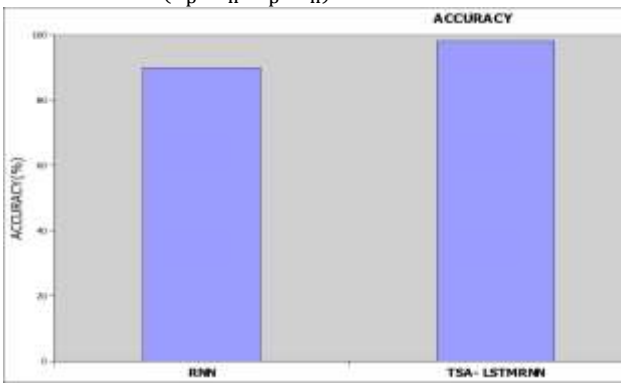

Fig 7. Accuracy

Fig. 7 shows the comparison measure of accuracy when evaluated with the current and suggested approaches. On the y-axis, accuracy value is marked, and on the other side as x-axis, the methods are given.

For the given NSL-KDD dataset, the suggested ECSO-ICNN technique delivers more accuracy than the current approaches, including RNN and ECSO-ICNN methods, which provide lesser accuracy. In order to improve the detection accuracy, pre-processing is used. The IoT-WSN performance is enhanced by the suggested ECSO-based CH NS. Thus, the outcome indicates that by selecting features optimally, the suggested ECSO-ICNN technique increases the accuracy of AD.

Precision

This formula is used to calculate the precision in equation (14):

$$\text{Precision} = \frac{T_p}{T_p+F_p} \qquad (14)$$

By the comparison of suggested and current methods, the precision is evaluated and it is presented in Fig. 8. The % of precision is plotted on the y axis whereas the methods are presented in x-axis.

For the given NSL-KDD dataset, the suggested ECSO-ICNN technique provides greater precision values while the current RNN, TSA-LSTMRNN technique, delivers lower precision. The suggested ECSO-ICNN technique outperforms IoT-based WSN in AD, based on outcomes.

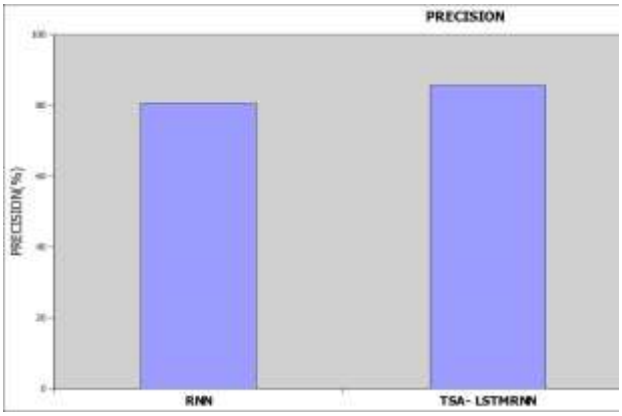

Fig 8. Precision

Recall

The recall value is computed in the following equation (15):

$$\text{Recall} = \frac{Tp}{Tp+Fn} \quad (15)$$

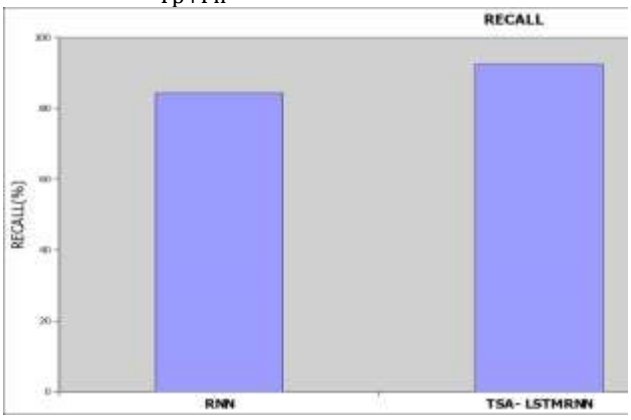

Fig 9. Recall

With the comparison of suggested and current methods, this assessed by recall, as Fig. 9 illustrates. On the x-axis, the methods are depicted, and on the y-axis, the recall value. The suggested ECSO-ICNN technique provides higher recall values on NSL-KDD dataset than the current RNN and TSA-LSTMRNN techniques. For the IoT-based WSN context, the outcome indicates that the ECSO-ICNN technique improves the AD accuracy.

F-measure

The weighted average of Precision and Recall is known as the F1 Score. FP and FN are taken into consideration in this measure which is given in equation (16)

$$\text{F-measure} = 2*\frac{(\text{Recall}*\text{Precision})}{(\text{Recall}+\text{Precision})} \quad (16)$$

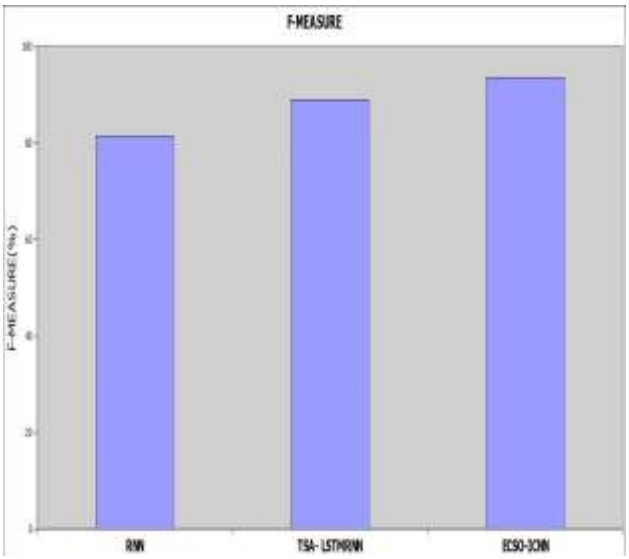

Fig 10. F-measure

The comparison of F-measure is done by employing the suggested and current methods and it is presented in Fig. 10. Here, x-axis denotes the methods whereas y-axis denotes the % of F-measure.

For the provided NSL-KDD dataset, the suggested ECSO-ICNN method delivers greater F-measure values by 93.48% compared to the current RNN and TSA-LSTMRNN methods, which produce lower F-measure values. For the IoT-based WSN environment, the outcome thus indicates that the ECSO-ICNN technique improves the AD accuracy.

TABLE IV COMPARISON TABLE OF CURRENT AND SUGGESTED APPROACHES

Methods/ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
RNN	90.02	80.56	84.22	81.39
TSA-LSTMRNN	98.1	85.59	92.34	88.96
ECSO-ICNN	98.8	91.24	95.73	93.48

V. CONCLUSION

To enhance the AD performance for the specified IoT-based WSN, the ECSO-ICNN technique has been suggested in this study. This work contains five main modules such as system model, CHS, pre-processing, NSL-KDD dataset collection and AD. Initially, system model is constructed using IoT and WSN setup. Then, CH NS is executed through utilizing ECSO procedure by best FV. After that, NSL-KDD dataset is collected with 42 features. To enhance the quality of the dataset, the goal of pre-processing is then to handle duplicate values. Finally, the attack detection is done by using ICNN algorithm which provides more accurate prediction performance. The suggested ECSO-ICNN model helps to improve the attack detection results in better way. From the outcomes of the experiment, it indicates that the recommended ECSO-ICNN procedure provides greater accuracy, precision, recall and F-measure than the current procedures. The study of FS based on optimization and an unsupervised ML framework that analyzes all unknown traffic will be included in future research.

REFERENCES

[1] A.A. Diro, and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761-768, May 2018, doi: 10.1016/j.future.2017.08.043.

[2] A.K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," Computer Communications, vol. 176, pp. 146-154, Aug. 2021, doi: 10.1016/j.comcom.2021.05.024.
[3] K.M. Sadique, R. Rahmani, and P. Johannesson, "Towards security on internet of things: applications and challenges in technology Procedia," Computer Science, vol. 141, pp. 199-206, Jan. 2018, doi: 10.1016/j.procs.2018.10.168.
[4] A. Mubarakali, K. Srinivasan, R. Mukhalid, S.C. Jaganathan, and N. Marina, "Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems," Computational Intelligence, vol. 36, no. 4, pp. 1580-1592, Feb. 2020, doi: 10.1111/coin.12293
[5] B.D. Deebak, and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," Ad Hoc Networks, vol. 97, pp. 102022, Feb. 2020, doi: 10.1016/j.adhoc.2019.102022.
[6] S. Abbasi-Daresari, and J. Abouei, "Toward cluster-based weighted compressive data aggregation in wireless sensor networks," Ad Hoc Networks, vol. 36, pp. 368-385, Jan. 2016, doi:10.1016/j.adhoc.2015.08.014.
[7] F. Khorasani, and H.R. Naji, "Energy efficient data aggregation in wireless sensor networks using neural networks," International Journal of Sensor Networks, vol. 24, no. 1, pp. 26-42, May 2017, doi: 10.1504/IJSNET.2017.084207.
[8] N.T. Nguyen, B.H. Liu, V.T. Pham, and Y.S. Luo, "On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data aggregation trees," Computer Networks, vol. 105, pp. 99-110, Aug. 2016, doi: 10.1016/j.comnet.2016.05.022.
[9] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," Internet of Things, vol. 3, pp. 82-89, Oct. 2018, doi: 10.1016/j.iot.2018.09.003.
[10] T. Sherasiya, and H. Upadhyay, "Intrusion detection system for internet of things," Int. J. Adv. Res. Innov. Ideas Educ.(IJARIIE), vol. 2, no. 3, pp. 2244-2249, Dec. 2018, doi: 10.1002/9781119892199.ch13.
[11] M.G. Karthik, and M.M. Krishnan, "Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks," Journal of Ambient Intelligence and Humanized Computing, pp. 1-11, Mar. 2021, doi: 10.1007/s12652-021-03082-3.
[12] P.K. Keserwani, M.C. Govil, E.S. Pilli, and P. Govil, "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model," Journal of Reliable Intelligent Environments, vol. 7, no. 1, pp. 3-21, Jan. 2021, doi: 10.1007/s40860-020-00126-x.
[13] N. Tsitsiroudi, P. Sarigiannidis, E. Karapistoli, and A.A. Economides, "EyeSim: A mobile application for visual-assisted wormhole attack detection in IoT-enabled WSNs," In 2016 9th IFIP Wireless and Mobile Networking Conference (WMNC), pp. 103-109, Jul. 2016, doi: 10.1109/WMNC.2016.7543976.
[14] E.P. Krishna, and A. Thangavelu, "Attack detection in IoT devices using hybrid metaheuristic lion optimization algorithm and firefly optimization algorithm," International Journal of System Assurance Engineering and Management, pp. 1-14, May 2021, doi: 10.1007/s13198-021-01150-7.
[15] M. Alqahtani, H. Mathkour, and M.M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," Sensors, vol. 20, no. 21, pp. 6336, Nov. 2020, doi: 10.3390/s20216336.
[16] S.R.U. Jan, R. Khan, and M.A. "An energy-efficient data aggregation approach for cluster-based wireless sensor networks," Annals of telecommunications, vol. 76, no. 5, pp. 321-329, Jan, 2021, doi: 10.1007/s12243-020-00823-x.
[17] F. Taher, M. Elhoseny, M.K. Hassan, and I.M. El-Hasnony, "A Novel Tunicate Swarm Algorithm With Hybrid Deep Learning Enabled Attack Detection for Secure IoT Environment," IEEE Access, vol. 10, pp. 127192-127204, Jan. 2022, doi: 10.1109/ACCESS.2022.3226879.
[18] K. Ahmed, A.E. Hassanien, and S. Bhattacharyya, "A novel chaotic chicken swarm optimization algorithm for feature selection," In 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 259-264, Nov. 2017, doi: 10.1109/ICRCICN.2017.8234517.
[19] D. Wu, S. Xu, and F. Kong, "Convergence analysis and improvement of the chicken swarm optimization algorithm," IEEE Access, vol. 4, pp. 9400-9412, Sep. 2016, doi: 10.1109/ACCESS.2016.2604738.
[20] S.A. Elsaid, and N.S. Albatati, "An optimized collaborative intrusion detection system for wireless sensor networks," Soft Computing, vol. 24, no. 16, pp. 12553-12567, Aug. 2020, doi: 10.1007/s00500-020-04695-0.
[21] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," IEEE Access, vol. 7, pp. 42210-42219, Mar. 2019, doi: 10.1109/ACCESS.2019.2904620.