RESEARCH PAPER

Hybrid Convolutional Neural Network And Long Short Term Memory Algorithm For Efficient Attack Detection In Iot Based WSN

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

[1] 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.

Received: 2nd Mar, 2026 | **Revised:** 14th Mar, 2026 | **Accepted:** 4th Apr, 2026 | **Available Online:** 20th Apr, 2026

ABSTRACT

The modern Internet of Things (IoT) concept now includes Wireless Sensor Networks (WSNs) as an essential element. Attack detection is the main problem with WSN. and the existing system has problem with temporal feature learning. As a result, there is a considerable decline in total system performance. Hybrid Convolutional Neural Network – Long Short Term Memory (HCNN-LSTM) algorithm and Enhanced Butterfly Optimization (EBO) are proposed to address the aforementioned issues in this study. This work contains main steps are such as system model, Cluster Head (CH) node selection, preprocessing and attack detection. The system model encompasses the quantity of Sensor Nodes (SN), sensor devices, destinations, and Multipoint Relays (MPRs), along with one-hop and two-hop neighboring and CH nodes. The CH node is then selected using the EBO approach. For the specified IoT-based WSN, it produces the best fitness results via increased accuracy and reduced energy usage. The NSL-KDD dataset is considered 42 features and class descriptors represent this class of attacks. To properly manage duplication and redundant features for the specified NSL-KDD dataset, data pre-processing is then carried out utilizing filtering and feature selection procedures. Finally, attacks based on both spatial and temporal features are detected for the supplied dataset using the HCNN-LSTM technique. Concerning preciseness, recall, reliability, and f-measure, the results of the simulation show that the EBO and HCNN-LSTM technique that was developed is superior to traditional methods.

Keywords: Wireless Sensor Networks (WSNs), Internet of Things (IoT), attack detection, Enhanced Butterfly Optimization (EBO), Hybrid Convolutional Neural Network – Long Short Term Memory (HCNN-LSTM).

How to cite this article: Dhivya B, Thavamani S. Hybrid Convolutional Neural Network and Long Short Term Memory Algorithm for Efficient Attack Detection in IoT Based WSN. Int J Drug Deliv Technol. 2026;16(32s):903-917. DOI: 10.25258/ijddt.16.32s.100

Source of support: Nil.

Conflict of interest: The authors declare no conflict of interest.

I. INTRODUCTION

In a challenging environment with inconsistent communication paths, WSNs are made by several sensor nodes that are extremely decentralized. It is extremely challenging to implement IT network solutions in WSNs due to the complexity and interconnectedness of several devices and protocols, as well as the diversity of routing protocols and services offered [1] [2]. As a result, the majority of the security measures in place currently are either inadequate or incompatible. To reach a particular range of security levels, a variety of strategies and trends have been used, including trust management and encryption approaches that utilize lightweight ways to provide low-cost encryption for devices with limited power and power. Serious attacks against routing protocols are also possible. For example, malicious or misleading routing information might be introduced into a network, leading to delays or packet losses because of routing conflicts. Numerous techniques, including encryption and information correlation between several nodes, have been used to prevent routing attacks.

Cluster Head (CH) depending on their position and balancing remaining energies, CHs have various lifetimes [3]. Clustering has been shown to be one of the most effective methods for WSN energy conservation. In a hierarchical cluster-based WSN, CHs use more energy because of the extra strain caused by collecting and aggregating data from their member sensor nodes and transmitting the aggregated data to the base station. Consequently, selecting the right CHs is essential to preserving sensor node energy and extending WSN lifetime [4]. Furthermore, it may be challenging to access sensors after they are positioned in the monitored region. A sensor network that is meant to monitor the state of a big city's sewage system, for instance, could not be reachable for battery replacement, software upgrades, or maintenance. As a result,

developing energy-efficient methods at each layer has received particular attention.

In general-purpose computing, security is a well-established area where security mechanisms handle computer functions including intrusion detection, authentication, and secure transactions [5]. Power consumption is often given top concern when developing security systems since battery life limits the lifetime of a sensor node. Security becomes important when sensor networks are deployed in hostile environments because they are subject to various adverse assaults. Communication transactions should be verified and encrypted to ensure security. Considering its low power consumption and straightforward hardware requirements, symmetric key schemes are more suitable for WSN cryptography. However, most of them fall short of public key cryptography in terms of integrity, secrecy, and authentication [6]. Figure 1 depicts the IoT-based WSN architecture.

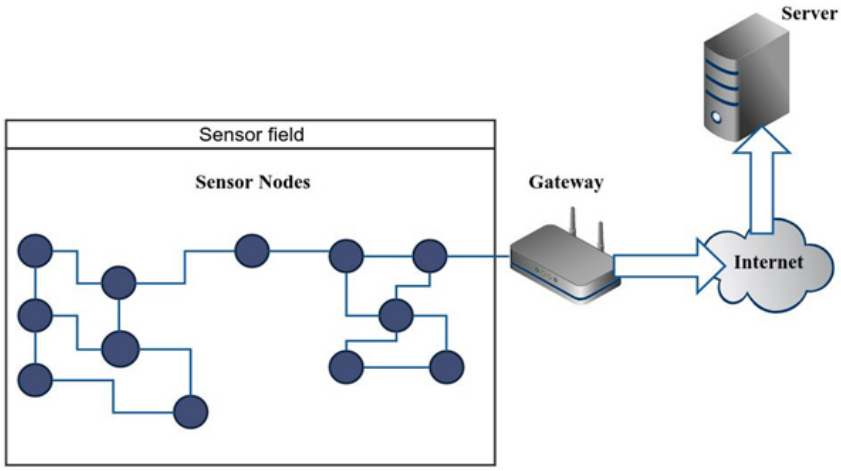

Fig. 1. IoT based WSN architecture

Smart object (thing) communication and integration have been envisioned as the IoT. A new context of future services and applications is brought forth by the primacy of IoT. IoT components include a variety of items connected to the Internet via wired and wireless networks, including sensors, mobile phones, and Radio Frequency Identification (RFID) tags. Smart objects have the ability to perceive, collect, and send data to meet the diverse needs of users. Additionally, in addition to providing dependable data forwarding, IoT-based WSNs are utilized such as air pollution, water condition monitoring, and smart cities, in both attending and unsecured situations and also in medical sensor networks. Enhancing energy effectiveness is an essential concern [3]. A cluster-based approach to WSN energy efficiency has already been proposed to several academics [7].

The nodes in clustering methods are divided into several regions, and the leader node is the cluster head. Data collection from member nodes, aggregation, and forwarding to the Base Station (BS) are the objectives of the cluster head. Either a single hop or multi-hop approach may be used to transmit data from cluster heads to BS. Clustering solutions can be classified as either probabilistic or non-probabilistic. Clusters are produced in a random sequence in probabilistic [8], which leads to an uneven distribution of loads and energy use. In contrast, the non-probabilistic approach selects cluster heads based on a number of parameters. The dynamic nature of sensor nodes means that although non-probabilistic approaches perform better than conventional probabilistic techniques, there are still unresolved issues with energy conservation and routing resilience for IoT based on WSN.

The selection of CH nodes is the study's primary objectives and spatial and temporal features based attack detection on IoT based WSN. The energy use and temporal feature-based detection accuracy of existing methods are disadvantages. In this study, the HCNN-LSTM model is used to enhance the overall system performance in IoT-based WSN to address the aforementioned problems. Pre-processing, attack detection, CH node selection, and system model development are the primary contributions of this study. Using efficient algorithms, the suggested approach improves performance in an IoT-based WSN setting.

This is the structure of the remainder of the paper: Section 2 provides an overview of some of the literature on attack detection and CH node selection. Section 3 provides specifics on the suggested technique for the HCNN-LSTM algorithm. Section 4 delineates the simulation results and provides an analysis of performance. Finally, Section 5 provides a summary of the results.

II. Related work

In [9], Haseeb et al (2019) intends to improve the network duration and data reliability by developing an energy-efficient and improved secure routing protocol (ESR) for intrusion prevention in IoT utilizing WSN. Initially, based on the intrinsic attributes of nodes, the suggested protocol generates several energy-efficient clusters. Second, the security and dependability of the sensory data between the cluster head and base station (BS) are attained using the (k,n) threshold-based Shamir secret sharing technique. The proposed security plan provides a basic method for managing incursions initiated by malicious nodes. In contrast to the current study on dynamic network topologies, the experimental findings utilizing the network simulator (NS-2) indicate that the proposed routing protocol enhanced network longevity by 37%, reduced average end-to-end delay by 24%, increased packet delivery ratio by 30%, and lowered average communication cost by 29%.

In [10], Deebak, B. D., & Al-Turjman, F. (2020) designed a secure routing and monitoring protocol using multi-variant tuples and the Two-Fish (TF) symmetric key approach, which prevented the perpetrators from entering the global sensor network. The Authentication and Encryption Model (ATE) serves as the foundation for the suggested methodology. The sensor guard nodes are selected using the Eligibility Weight Function (EWF), and they are concealed using a sophisticated symmetric key technique. The features of Ad hoc On-Demand Multipath Distance Vector (AOMDV) and Multipath Optimized Link State Routing (OLSR) are combined to provide a secure hybrid routing system. Compared the suggested method to the existing routing methods, the results demonstrate that it has a higher proportion of monitoring nodes.

In [11], Shahid et al (2021) the Energy Optimized Security against Wormhole Attack in IoT (ESWI) approach was designed to detect wormhole assaults while enhancing security and performance. This algorithm has been designed to minimize overhead and energy waste during operation by being straightforward and less complex. The simulation results of our approach produce advantageous results regarding packet delivery ratio and detection rate. Additionally, it results in a higher throughput, lower d-to-end latency, and lower energy usage.

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 [12]. 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 0.14% at the lowest and 14.25% at the highest 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%

In 2022, Taher et al. introduced the novel Tunicate Swarm Algorithm (TSA) in [13], 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 Recurrent Neural Network (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. Monitoring the P count in each model layer is essential to prevent overfitting or underfitting. Modifying the quantity of tiers 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. The TSA-LSTMRNN model outperformed similar models as demonstrated by its increased accuracy, recall, and precision, respectively

In [14], Majid, M.A (2022) proposed an adaptive clustering and routing technique that uses less energy to increase the WSN's lifetime. The basic LEACH protocol is improved with a new weight-based LEACH (LEACH-W) protocol that increases network lifetime and conserves energy. By using adaptive multi-hop algorithms and a uniform cluster distribution to shorten the data transmission distance between nodes, the method lowers energy consumption in WSN. The number of nodes for each CH, the distance between nodes, and the sensor nodes' remaining energy are all taken into consideration while selecting the CH nodes. Variations in network situations are addressed by the proposed method, which also takes into account variable settings in practical scenarios. The proposed approach uses load-balanced clustering to increase the WSN's lifetime. Based on simulation results, the proposed routing protocol and clustering algorithm perform better than the existing methods in many aspects. A balanced distribution of clusters, a longer duration, and more energy efficiency than conventional methods are the objectives of the method.

III. Proposed Methodology

In this research, spatial and temporal features based attack detection is done by using HCNN-LSTM algorithm over IoT based WSN. The system model, CH node selection via the EBO method, NSL-KDD data collection, data pre-processing,

and temporal features based attack detection via the HCNN-LSTM method are the primary contributions of this study. Fig. 2 presents the suggested system's overall block diagram.

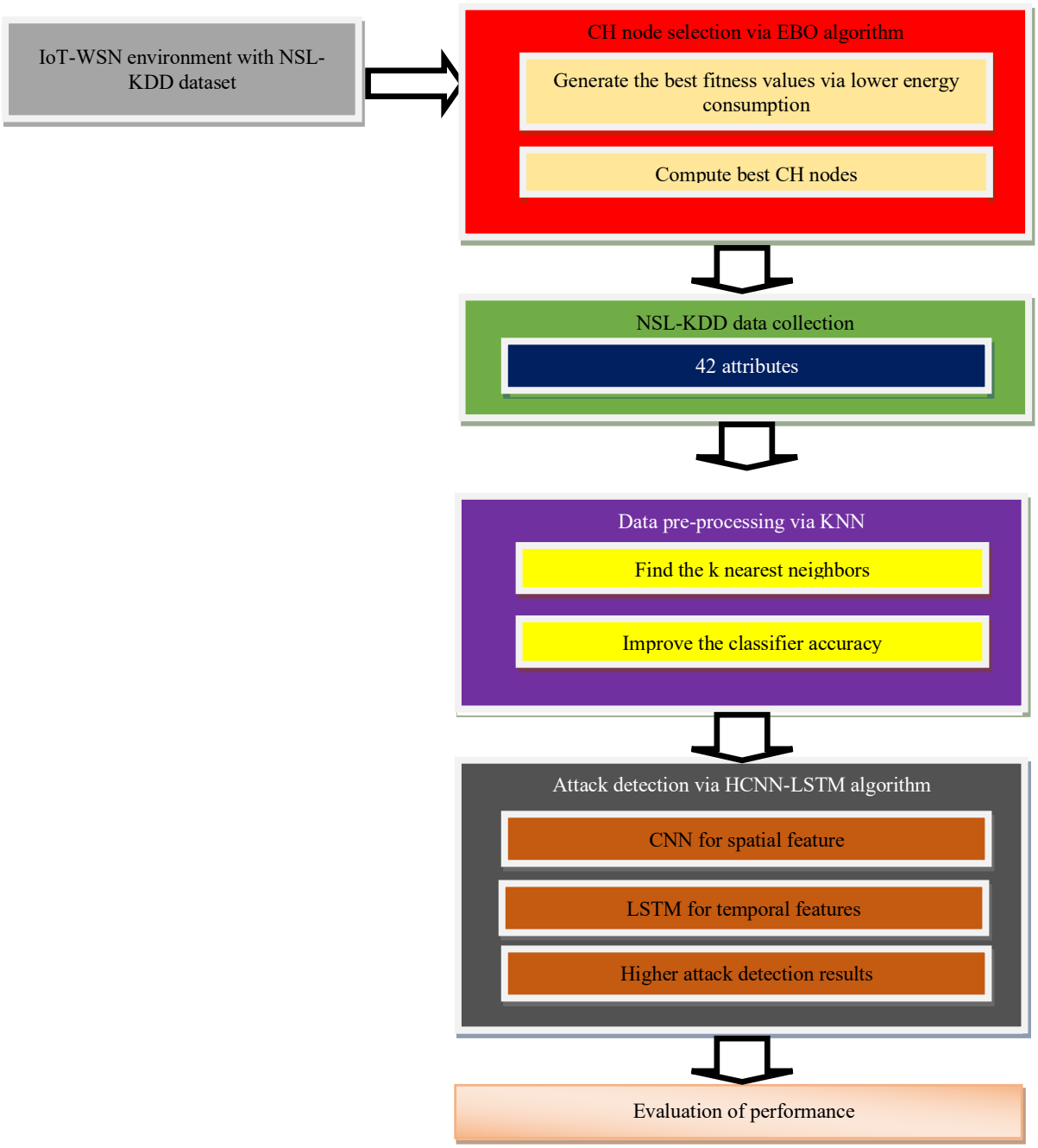

Fig. 2. Block schematic of the proposed system overall

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 sent over the BS in the upper layer to the server. 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. Cluster formation and Cluster Head (CH) node selection using Enhanced Butterfly Optimization (EBO) algorithm in IoT based WSN

The EBO algorithm is used in this study to identify the best CH over an IoT-based WSN. Energy and delay measurements form its basis. The EBO algorithm generates optimum solutions that increase the network's speed and energy efficiency. The best fitness values are generated via objective function which is to provide efficient packet transmission over IoT based WSN. Each node's energy level increases by the clustering methods, which also manage the network effect's massive scope and traffic. It is simple to identify faulty nodes in a cluster, and an appropriate algorithm can replace them automatically. The graphical approach G (N, E) is shown in the cluster network. E is the total energy required for sending the packet from one node to another, and N is the number of nodes in the cluster $i\ to\ j\ and\ i,j\ \in\ N$. The node E_{Ri} residual energy is used for determining CH energy. The packet's speed to the distance d_ij is equal to time needed to deliver every packet between I and J, and also the energy remaining E_{Ri} is computed by dividing the node's total energy by its consumed energy. Equation (1) provides the energy and the total number of packets P_n needed for transmitting one packet E_p.

$$E_{CH} = \frac{E_{Ri}}{\sum_{j=1}^{n}((d_{ij}+(E_p \times P_n))} \quad (1)$$

The data packet is widely monitored across the network as it moves among sensor nodes. The quantity of data that is moved between the source and destination in a second is specifically used to assess the network's data throughput.

In IoT-based WSNs, the efficiency of network operations is paramount. This efficiency is predominantly influenced by how well the network manages its energy resources and handles delays. Traditional methods of CH selection often fail to simultaneously optimize these critical metrics, leading to suboptimal network performance. This necessitates a robust optimization technique capable of handling multiple objectives. The natural foraging behaviors of butterflies served as the model for the EBO algorithm [15]. It utilizes sensory modality and fragrance (pheromone) intensity to search for optimal solutions in a multidimensional space. For CH selection in IoT-based WSNs, the EBO algorithm has been adapted to consider energy consumption and delay as key metrics for fitness evaluation.

In this work, CH is selected by EBO to select the optimal CH nodes over IoT based WSN. EBO is a novel algorithm inspired by nature that solves energy efficiency problems in hybrid networks by simulating the mating behavior of butterflies and food search (reduced energy usage with chosen sensor nodes). To identify the location of a nectar partner, butterflies use their sense of smell to select nodes in an optimum method. This is the primary basis for the suggested EBO algorithm. Butterflies have been proven to have a highly beneficial sense of identifying the source of fragrance, according to the results of scientific investigations (CH node selection).

The degree of intensity with which a butterfly produces fragrance is directly proportional to its fitness (CH node selection). In other words, the movement of a butterfly from one place to another, its fitness will change in a manner that is proportional to its movements. Three essential concepts sensory modality (*c*), stimulus intensity (*I*), and power exponent (*a*) for optimum node selection form the foundation of the EBO Algorithm's whole sensing and processing paradigm [16]. For the IoT-based WSN's sensor node selecting, *I* is connected with fitness (lower energy usage) in the EBO Algorithm. Equation (2) illustrates how a fragrance is created in the EBO Algorithm based on the physical strength of the stimulus.

$$f = cI^a \quad (2)$$

where In is the stimulus extent, f is the perceived quantity of C is the sensory approach, and c is the fragrance or the intensity with which other butterflies sense it, which is created by spending less energy. Therefore, a and c describe the range [0,1]. If, on the other together, and is a = 0, this means that none of the other butterflies are able to sense your fragrance. The parameter and is responsible for regulating the behavior of the algorithm. For the purpose of computing the rate at which the EBO algorithm converges, C is an extra crucial parameter that is also necessary. The previously mentioned concepts will be shown using a search algorithm, the aforementioned butterfly properties are simplified as follows:

1. The butterflies (nodes) to attract one another, they are all thought to release fragrance.

2. Each random or in the direction of the butterfly that is more fragrant, each butterfly will migrate.
3. In a butterfly, the stimulus intensity is influenced by or determined by the objective function's landscape.

EBO is divided into three stages: startup, iteration, and finalization. Each time an EBO is run, the initialization phase is completed first, followed by an iterative search for the best CH nodes and, eventually, the algorithm's termination when the most optimum selection solution is discovered. Energy consumption is determined in the EBO method and its solution space during the startup phase. In addition, the parameters utilized in EBO are given values. In the CH node selection search space, the positions of butterflies (sensor nodes) are produced at random based on their fitness and fragrance values. The algorithm begins the iteration phase after the startup step is complete. Each iteration involves moving every butterfly in the CH node selection solution space to a new position before determining how much energy each one uses. Initially, the system estimates the fitness values of each sensor node at different points across the solution space. Using equation (3), these butterflies will then produce fragrance at their locations. The node progresses toward the optimum nodes, or fittest solution (g∗), in the global search phase, which is represented by equation (3),

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i * ECE_W \quad (3)$$

In iteration number t, the solution vector x_i for ith node is denoted as x_i^t. The best selected node solution is represented in this instance by g^* among each of the current iteration's solutions. The representation for the ith butterfly's fragrance is f_i, and the random integer r∈[0,1] a representation of the local search phase is provided by equation (4).

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i * ECE_W \quad (4)$$

In the CH node selection solution space, the jth and kth butterflies are denoted by x_j^t and x_k^t, respectively. If r is a random integer and x_j^t and x_k^t are members of the same swarm, then equation (4) changes into a local random walk. To determine the most efficient selection of nodes from the hybrid network, it is possible for butterflies to look both regionally and globally for food and a mate. To transition between intense local search and common global search, EBO uses switch probability p. Continued repetition occurs until the halting requirements are not matched. The method delivers the optimal solution with the highest fitness at the end of the iteration phase. To find out the number nodes are optimal for the given network, node weight is additionally introduced to the EBO algorithm in equation (5). Using the best possible CH node selection within the specified hybrid network, the EBO algorithm aimed to improve energy usage.

By reducing the distance between two sample distributions, Cross Entropy (CE) may be used to determine the most effective parameters of a probability distribution and solve optimization problems. The CE approach is very resilient, highly adaptable, and has an exceptional global search ability.

$$CE = \frac{1}{N}\sum_{i=1}^{N} I_{s<r} \frac{f(x^i,v)}{g(x^i)} \quad (5)$$

with significance sampling density $g(x)$ and a random sample from $f(x;v)$ denoted by x^i. The distance between two sampling distributions is measured using the Kullback–Leibler divergence, referred to as the cross-entropy, to determine the optimum importance sampling density.

method 1 shows the general stages in the suggested EBO method. Algorithm 1 generates the initial population based on in the IoT-based WSN, the number of nodes (Step 1), and then uses the sensor modality c and power exponent a (Step 3) to determine the stimulus intensity I_i at x_i (Step 2). The decreased energy use is the basis of these issues. Following that, the stopping conditions are determined (Step 4), and the fragrance value is determined (Step 6) for every butterfly in the network. Then, in Step 8, determine which node in the population is the best, and in Step 10, produce a random number, r. If $r < p$, use equation (11), and if else, use equation (12) to move randomly in the direction of the best butterfly. Following that, update a value (Step 17) and assess each person in considering their new position (Step 18). Finally, use end while to terminate the process (Step 19). The suggested EBO algorithm's flowchart is seen in Figure 3.

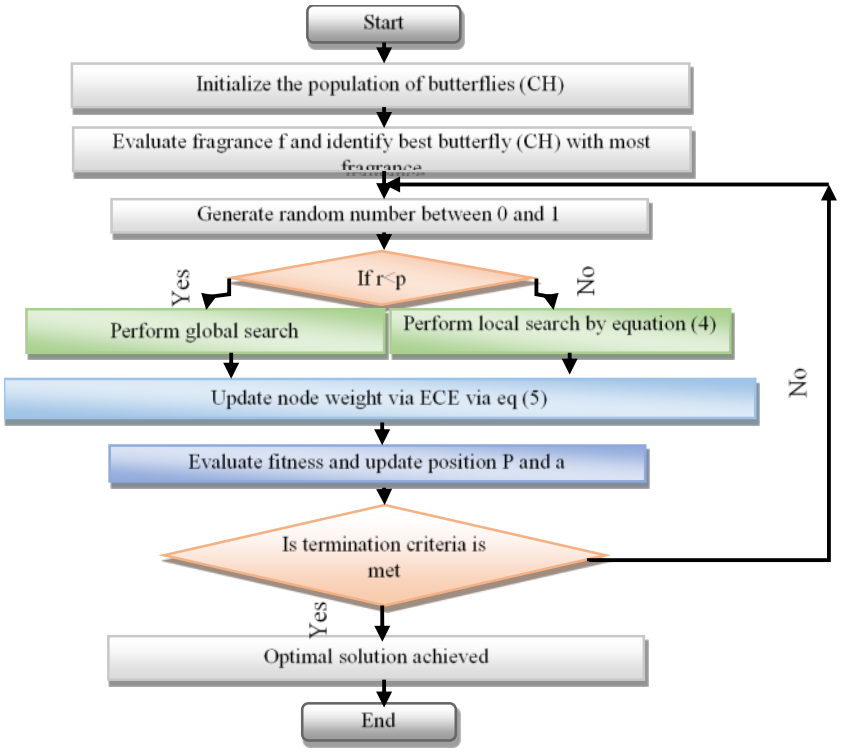

Fig. 3. Flowchart of EBO

Algorithm 1: EBO algorithm for optimal CH node selection

1. Construct the initially n butterfly population $x_i = (i = 1,2, \dots, n)$
2. Reduced energy expenditure $f(x_i)$ indicates the stimulus intensity I_i at x_i
3. Describe the switch probability (p), sensor modality (c), and power exponent (a).
4. If the stoppage requirements are not satisfied, do
5. Each butterfly *f* in the population does
6. Determine the fragrance for fusing using equation (4), then use equation (5) to generate weight via entropy.
7. End for
8. Select the greatest butterfly
9. In the population, each butterfly does
10. Generate random number r
11. If $r < p$ then
12. Equation (3) directs you toward the optimum butterfly (optimal CH nodes via reduced energy usage), while equation (4) uses entropy to produce weight
13. Else
14. Calculate at random using equation (4)
15. End if
16. End for
17. Update the value of a
18. Assess each individual (CH) based on their new role

C. NSL-KDD dataset

In this paper, we analyze the NSL-KDD cup dataset for IoT-based WSN AD. The large network traffic dataset NSL-KDD has already been used to construct a training set and a testing set. 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 1, 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. To extract vector characteristics for training, the dataset is separated into normal and pathological individuals. After training, normal or abnormal clusters can be classified.

TABLE 1. DIFFERENT ATTACK CLASSES

Samples	**Normal**	**DoS**	**Probes**	**U2R**	**R2L**	**Total**
Training samples	67,343	45,927	11,656	52	995	125,974
Testing	9,711	7,458	2,421	200	2,65	22,544

samples					4	
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. In saturating the network service, the denial-of-service attack prevents authorized users from connecting to the network. 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 infraction of the security rule, the probe attack travels through the network to be collected data. While the others only have one link, there are many connections between DoS attacks and probing. Table 2 presents an overview of the 4 attack types found in the NSL-KDD benchmark dataset.

TABLE 2. ATTACK CLASSES DESCRIPTION

Attack	The dataset's description of the attack
DoS	In maintaining the network busy, the attacker prevents authorized users from accessing it.
R2L	For a certain FTP version, the hacker attempts to enter the network or machine.
U2R	In accessing to the root of the system, the attacker attempts to enter the network without authorization.
Probe	Assembling the data required to disable the system's security is its objective.

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

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, among the assaults ipsweep, mscan, nmap, portsweep, and saint are among the tools in the set, and Satan.
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. Among the risks identified the dataset contains the following: xterm, load module, perl, rootkit, ps, buffer_overflow, and sqlattack.
5. R2L: A local network machine is effectively compromised by a remote attacker in these assaults. The dataset contains the following types of attacks: ftp write, guess_password, imap, multihop, named, phf, send mail, snmpgetattack, worm, xlock, xsnoop, and http-tunnel.

D. Using the K-Nearest Neighbor (KNN) method for pre-processing

In this work, pre-processing is done by using KNN algorithm which is focused to improve the classification accuracy. According to the nearest neighbor approach, a test tuple is learned by comparing it with comparable training tuples. With the use of training data, KNN is regarded as a supervised learning approach that classifies points for a particular category. Assume that (X_i, C_i) contains data points with i = 1, 2……., n. Ci denotes labels for X_i for each i, whereas X_i stands for feature values. There are n properties that characterize the training tuples. Each tuple is a representation of a point in a space that has n consecutive dimensions. A KNN classifier searches for the k training tuples that are most similar to an unknown tuple by exploring the pattern space when it is given the unknown tuple. These k training tuples that are unknown are the k "nearest neighbors" of the unknowable [17]. Equation (6) defines the Euclidean distance, which provides an instance of a distance metric that is used to measure closeness.

$$(X1, X2) = sqrt\ (sum\ ((xj - xij)\ \hat{}2) \quad (6)$$

where, x - new point

xi – point that exists across all of the input characteristics j.

Algorithm 2: KNN

Input: Original NSL-KDD dataset

Output: Pre-processed dataset

Start

{

For each input feature that is part of the NSL-KDD dataset, do

Use (6) to calculate the Euclidean distance

End for

Find the K training instances that are closest to the instance of the unknown class

Replace the missing and incorrect values with KNN values after filling them in

Based on distance, the examples are arranged nearest neighbor

Select the K instance values that occur most often

}

End

E. Attack detection using Hybrid CNN with LSTM (HCNN-LSTM)

In this work, CNN is used to map inputs and examine the spatial properties. CNNs usually consist of three layers: the Convolutional Layer (CL), the Pooling Layer, and the Fully Connected Layer (FC). CL sends the output to the following layer by subjecting the input to a convolution technique. A single neuron's response to visual inputs is replicated by the convolution.

Developing in the next layer, a single neuron by combining the outputs of neuron clusters in the previous layer, local or global pooling layers are used and is considered to be the particular feature of convolutional networks. Use of the average value for each cluster of neurons in the layer above may be accomplished via the use of mean pooling. Through FC layers, each neuron in one layer may communicate with every other layer's neuron. The characteristics' respective weight values include proposed ICNN are customized to provide reliable results. 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, they employ a new input pattern, and the result is calculated in equation (7) and (8):

$$y(n) = f(\sum_{i=1}^{i=N} w_i(n)x_i(n) \quad (7)$$

$$\text{Where } f(x) = \begin{cases} +1 & if\ x \geq 0 \\ -1 & if\ x < 0 \end{cases} \quad (8)$$

The iteration index can be denoted as n

Weights for updated connections are derived from

$$w_i(n+1) = w_i(n) + \eta(d(n) - y(n)x_i(n), \quad i = 1,2,..N \quad (9)$$

In equation (9), the gain factor can be denoted as η

Next, apply SD in equation (10).

$$\sigma = \sqrt{\frac{1}{n} f_i(x_i - \overline{x)^2}} \quad (10)$$

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.

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, each neuron in the layer above connects to each of the others in the layer below. By classifying the attacks based on the best classifier values, the ICNN introduces an innovative feature. This CNN can extract spatial characteristics by configuring a large number of different-sized kernels. Convoluted 1*1, 2*2, and 3*3 kernels are the most often used kernel types; among these, features are effectively learned by 2*2 and 3*3 kernels, whereas learning is accelerated by 1*1 kernels.

F. LSTM-based categorization and learning of temporal features

In this work, Time-series data's time domain features are extracted using LSTM. The LSTM model developed by Zhao et al. [18] has been adjusted for this position in accordance with the IDS model's needs. To derive both temporal and spatial features from network traffic data, it is hence a hybrid CNN method. The general LSTM processes inputs using the sigmoid activation function after receiving them through three gates. To learn the termination conditions and obtain the temporal characteristics, this LSTM presents a detector for parameterization boundaries that produces binary output values in each layer. Furthermore, dense connections are added so that the layer ℓ may generate a feature map combination by using as input the feature maps from every previous layer. This approach improves the LSTM model's spatial feature learning to improve intrusion detection classifiers. At layer ℓ time step, setting the boundaries detector to 1 activates the LSTM model. transmits the summary representation to the base layer (ℓ + 1), which is less dense for spatial learning. Depending on the statuses of their boundaries, the layers select one of three operations: either update, copy, or flush. In equations (11–13), the typical LSTM equations are first developed.

$$\text{Gates and candidate } \begin{matrix} i_t \\ f_t \\ u_t \\ o_t \end{matrix} = Wx_t + Uh_{t-1} + b \qquad (11)$$

$$\text{Cell state: } c_t = c_{t-1} \odot \sigma(f_t) + tanh(u_t) \odot \sigma(i_t) \qquad (12)$$

$$\text{Hidden state: } h_t = \sigma(o_t) \odot tanh(c_t) \qquad (13)$$

Here, h_{t-1} depends the concealed state that was present before, c_{t-1} represents the previous cell state and LSTM input x_t. The LSTM structure receives this set of three parameters as input. Input, forget, candidate activation, and output gates use i_t, f_t, u_t, and o_t. Weight matrix W, activation function matrix U, and bias b.

To these conventional routines, add the boundary detector variable (z_t). is given in equation (14)

$$\begin{matrix} i_t \\ f_t \\ [u_t] \\ o_t \\ z_t \end{matrix} = Wx_t + Uh^1{}_{t-1} + z_{t-1}Vh^2{}_{t-1}b \qquad (14)$$

Here zt−1 is the border variable that was previously ℓ = 1; xt becomes the bottom-up connection's input at the current time-step, ($h^1{}_{t-1}, c^1{}_{t-1}$) representing the cell states and hidden states in frequent relation to ℓ = 1, $h^2{}_{t-1}$ with V representing the activation boundary matrix and ℓ = 2 indicating top-down connection hiding state.

If the boundary zℓ t−1 is at the bottom layer but not present in the previous phase, the Update function updates the layer summary image. Since this situation doesn't occur often, update procedures are rarely used. Simply ($h^1{}_t, c^1{}_t$) ← ($h^1{}_{t-1}, c^1{}_{t-1}$) is carried out by the copy process. This implies that the top layer remains secure until the bottom layer summary data comes. The flush process consists of two subtasks: Eject to get to the top layer and discard the current state, and RESET to reinitialize the states at each new segment. In other words, if Eject is not executed, Reset prevents the bottom layer summary from being absorbed by the upper layer. Therefore, the connections of short-term memory and the long-term temporal and spatial characteristics may be learned using the LSTM model.

IV. Simulation Result

In this work, NSL-KDD attack detection dataset is analyzed and implemented using Matlab. Accuracy, precision, recall, and f-measure are some examples of performance measures compared by using existing RNN, TSA-LSTMRNN, ECSO-ICNN and proposed HCNN-LSTM algorithms. Table 2 indicates the parameters.

TABLE 2. PARAMETERS

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

Number iterations	200	Simulation Time	5s

Accuracy

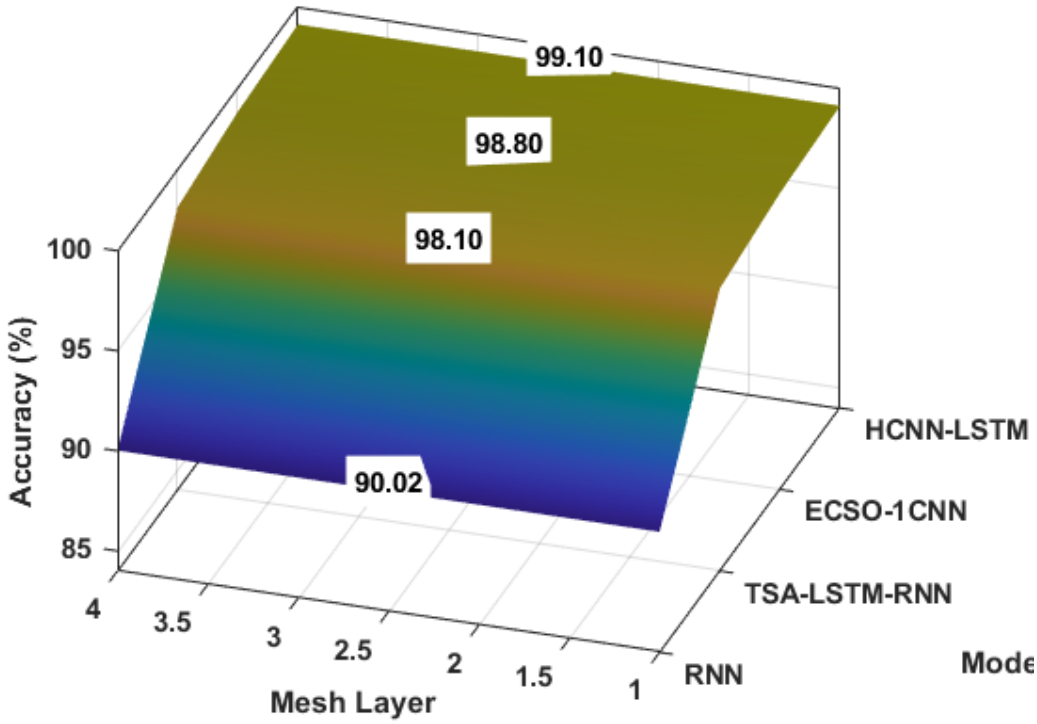

Fig. 4. Accuracy

Fig. 4 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 HCNN-LSTM technique delivers more accuracy than the previous RNN, TSA-LSTMRNN and ECSO-ICNN approaches. To enhance detecting accuracy, pre-processing and temporal feature learning is used. The IoT-WSN performance is enhanced by the suggested EBO-based CH node selection. Thus, the outcome indicates that by selecting features optimally, the suggested EBO based HCNN-LSTM technique increases the accuracy of attack detection.

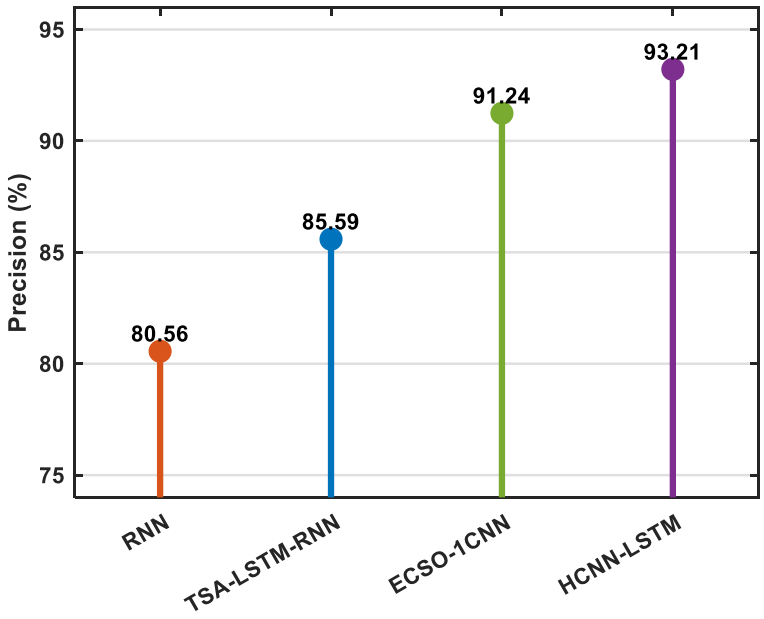

Fig. 5. Precision

Fig. 5 shows the comparison measure of precision when evaluated with the current and suggested approaches. On the y-axis, precision value is marked, and on the other side as x-axis, the methods are given. For the given NSL-KDD dataset, the suggested HCNN-LSTM technique delivers more precision than the previous RNN, TSA-LSTMRNN and ECSO-ICNN approaches. HCNN-LSTM is used for extracting the spatial–temporal aspects of network traffic data. The IoT-WSN performance is enhanced by the suggested EBO-based CH node selection. Thus, the outcome indicates that by selecting features optimally, the suggested EBO based HCNN-LSTM technique increases the accuracy of attack detections

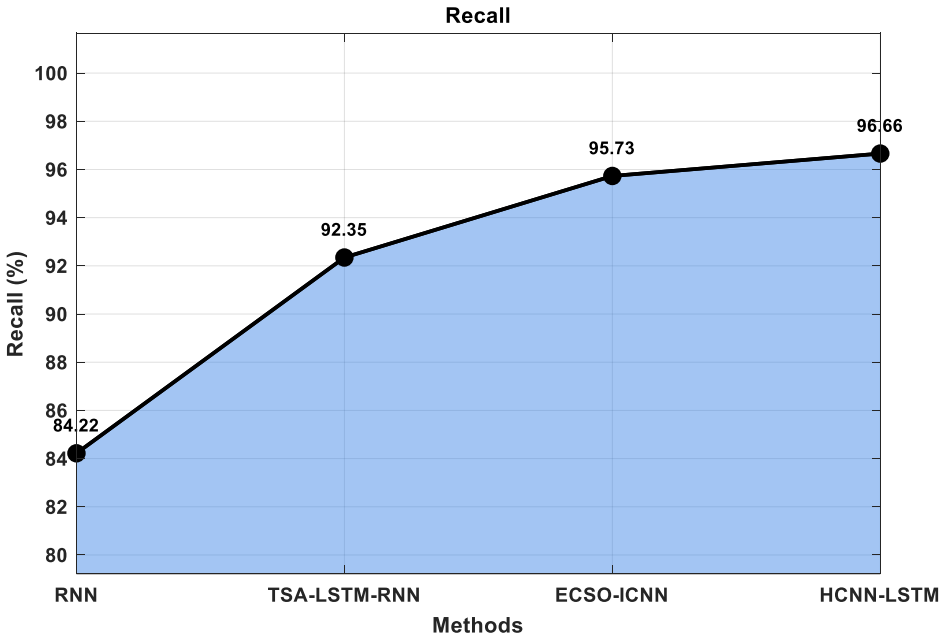

Fig. 6. Recall

Fig. 6 shows the comparison measure of recall when evaluated with the current and suggested approaches. On the y-axis, recall value is marked, and on the other side as x-axis, the methods are given. For the given NSL-KDD dataset, the suggested HCNN-LSTM technique delivers more recall than the previous RNN, TSA-LSTMRNN and ECSO-ICNN approaches. The IoT-WSN performance is enhanced by the suggested EBO-based CH node selection. Thus, the outcome indicates that by selecting features optimally, the suggested EBO based HCNN-LSTM technique increases the accuracy of attack detections.

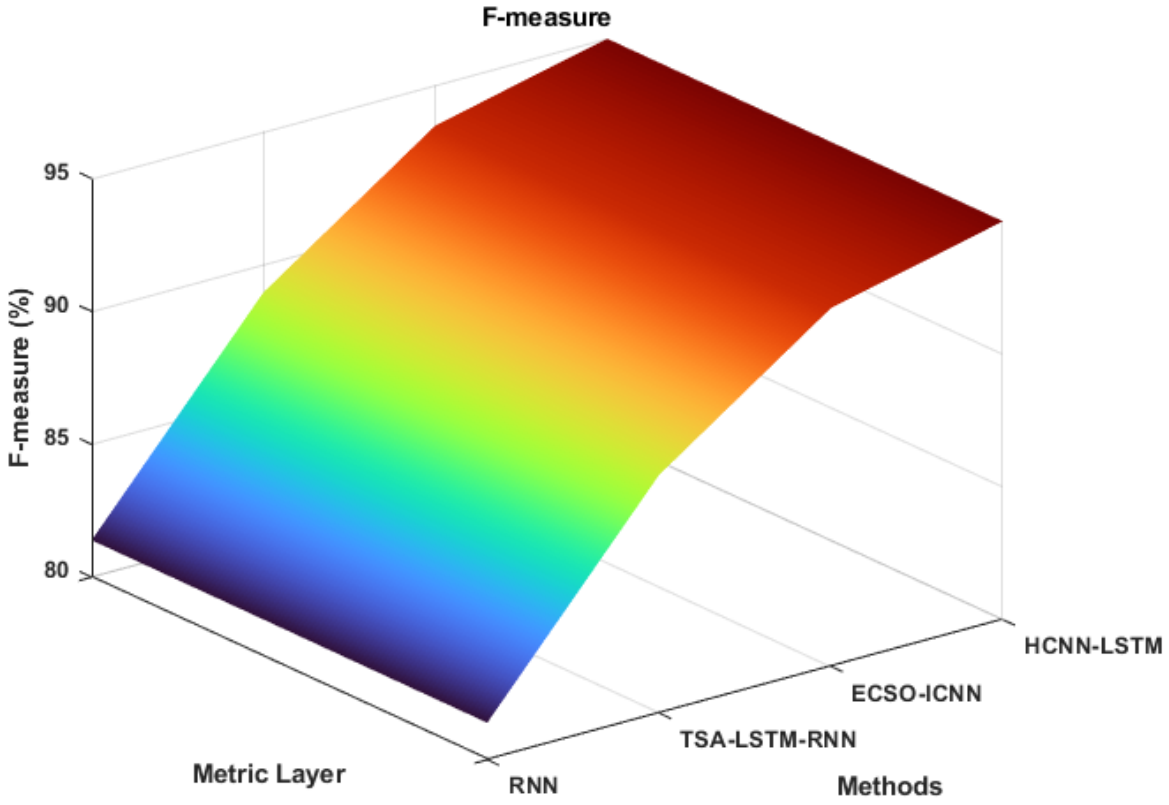

Fig. 7. F-measure

Fig. 7 shows the comparison measure of f-measure when evaluated with the current and suggested approaches. On the y-axis, f-measure value is marked, and on the other side as x-axis, the methods are given. For the given NSL-KDD dataset, the suggested HCNN-LSTM technique delivers more f-measure than the previous RNN, TSA-LSTMRNN and ECSO-ICNN approaches. Thus, the outcome indicates that by selecting features optimally, the suggested EBO based HCNN-LSTM technique increases the accuracy of attack detections.

V. Conclusion

To enhance the attack detection performance for the specified IoT-based WSN, the EBO-HCNN-LSTM technique has been suggested in this study. This work is divided into five major sections, including system model, CH node selection, pre-processing, NSL-KDD dataset collection and attack detection. Initially, system model is constructed using IoT and WSN setup. Then, CH node selection is executed through utilizing EBO procedure by best fitness values. After that, NSL-KDD dataset is collected with 42 features. To enhance the quality of the dataset, KNN algorithm is introduced. Finally, the more accurate attack detection performance is achieved by using the HCNN-LSTM method. It generates results based on spatial and temporal feature learning. While the HMLSTM classifies the network data and extracts the temporal characteristics, the suggested OCNN extracts the spatial features. From the outcomes of the experiment, it indicates that the recommended HCNN-LSTM 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. K. Haseeb, A. Almogren, N. Islam, I. Ud Din and Z. Jan, “An energy-efficient and secure routing protocol for

intrusion avoidance in IoT-based WSN," *Energies*, vol. 12, no. 21, pp. 4174, Nov. 2019, doi: 10.3390/en12214174.

2. X. Li, J. Peng, J. Niu, F. Wu, J. Liao and K. K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606-1615, Dec. 2017, doi: 10.1109/JIOT.2017.2787800.
3. T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand and A. H. Gandomi, "Residual energy-based cluster-head selection in WSNs for IoT application," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5132-5139, Feb. 2019, doi: 10.1109/JIOT.2019.2897119.
4. A. John, A. Rajput and K. V. Babu, "Energy saving cluster head selection in wireless sensor networks for internet of things applications," *In 2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0034-0038, Apr. 2017, doi: 10.1109/ICCSP.2017.8286486.
5. G. Kaur, P. Chanak and M. Bhattacharya, "Energy-efficient intelligent routing scheme for IoT-enabled WSNs," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11440-11449, Jan. 2021, doi: 10.1109/JIOT.2021.3051768.
6. R. Nagaraju, V. C, M. G, S. B. Goyal, C. Verma, C. O. Safirescu and T. C. Mihaltan, "Secure routing-based energy optimization for IOT application with heterogeneous wireless sensor networks," *Energies*, vol. 15, no. 13, pp. 4777, Jun. 2022, doi: 10.3390/en15134777.
7. M. Ilyas, Z. Ullah, F. A. Khan, M. H. Chaudary, M. S. A. Malik, Z. Zaheer and H. U. R. Durrani, "Trust-based energy-efficient routing protocol for Internet of things–based sensor networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 10, pp. 1550147720964358, Oct. 2020, doi: 10.1177/1550147720964358.
8. A. Shukla and S. Tripathi, "A multi-tier based clustering framework for scalable and energy efficient WSN-assisted IoT network," *Wireless Networks*, vol. 26, no. 5, pp. 3471-3493, Jul. 2020, doi: 10.1007/s11276-020-02277-4.
9. B. Suresh and G. S. C. Prasad, "An energy efficient secure routing scheme using LEACH protocol in WSN for IoT networks," *Measurement: Sensors*, vol. 30, pp. 100883, Dec. 2023, doi: 10.1016/j.measen.2023.100883.
10. 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.
11. H. Shahid, H. Ashraf, H. Javed, M. Humayun, N. Z. Jhanjhi and M. A. AlZain, "Energy optimised security against wormhole attack in iot-based wireless sensor networks," *Computers, Materials and Continua*, vol. 68, no. 2, pp. 1967-1981, Mar. 2021, doi: 10.32604/cmc.2021.015259.
12. 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.
13. 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, Dec. 2022, doi: 10.1109/ACCESS.2022.3226879.
14. M. A. Majid, "Energy-efficient adaptive clustering and routing protocol for expanding the life cycle of the IoT-based wireless sensor network," *In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC),* pp. 328-336, Mar. 2022, doi: 10.1109/ICCMC53470.2022.9753809.
15. J. Daniel, S. F. V. Francis and S. Velliangiri, "Cluster head selection in wireless sensor network using tunicate swarm butterfly optimization algorithm," *Wireless Networks*, vol. 27, no. 8, pp. 5245-5262, 10.1007/s11276-021-02812-x.
16. S. J. Pratha, V. Asanambigai and S. R. Mugunthan, "Hybrid Mutualism Mechanism-Inspired Butterfly and Flower Pollination Optimization

Algorithm for Lifetime Improving Energy-Efficient Cluster Head Selection in WSNs," *Wireless Personal Communications,* vol. 128, no. 3, pp. 1567-1601, Feb. 2023, doi: 10.1007/s11277-022-10010-x.

17. I. Triguero, D. García-Gil, J. Maillo, J. Luengo, S. García and F. Herrera, "Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 2, pp. e1289, Mar. 2019, doi: 10.1002/widm.1289.
18. Y. Zhao, Y. Shen and J. Yao, "Recurrent Neural Network for Text Classification with Hierarchical Multiscale Dense Connections," *In IJCAI*, pp. 5450-5456, Aug. 2019, doi: 10.24963/ijcai.2019/757.