SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Congestion Clustering Energy Aware Routing Algorithm (Cceara) and Blockchain-Based Deep Learning Ensemble Model in Wireless Sensor Networks

Mrs. C. Saranya[1], Dr. L. Sudha[2]

[1]Research Scholar, Department of Computer Science, VET Institute of Arts and Science (Co-education) College, Erode, Tamil Nadu, India.
[2]Associate Professor, Department of Computer Science, VET Institute of Arts and Science (Co-education) College, Erode, Tamil Nadu, India

ABSTRACT
Wireless Sensor Networks (WSNs) have emerged as a prominent research focus in computer science because of their extensive use in both military and civilian domains. The deployment of these networks, particularly in unattended or remote environments, introduces numerous security challenges. Depending on the origin and behavior of the adversary, these threats are generally categorized into internal attacks and external attacks. In this paper, proposed a new effective malicious node detection system called the Deep Learning Ensemble Model (DLEM). DLEM model combines the Deep Belief Network (DBN), Bidirectional Gated Recurrent Unit (BiGRU) and Bidirectional Long Short Term Memory (BiLSTM) classifier incorporating results with aggregation procedure that will increase network security and adaptation depending on the characteristics of incoming traffic. Congestion Clustering Energy Aware Routing Algorithm (CCEARA) is introduced for unlimited-lifetime WSN by combining the sensor nodes' distance and traffic load. Double Deep Q-neural Network (DDQN) is used to manage congestion control under a WSN model. Enhanced Wild Horse Optimization (EWHO) is introduced for Cluster Head Selection (CHS) during clustering process. DBN employs stacked Restricted Boltzmann Machines (RBMs) to learn hierarchical, abstract representations from raw traffic inputs by pre-training without supervision and fine-tuning. The bidirectional nature of the BiGRU model allows it to simultaneously analyze interdependence by considering previous and potential states by examining traffic sequences forward in addition to backward. BiLSTM classifier is a designed for attack detection where contextual data from time steps in the past and future improves detection accuracy. In addition, DLEM algorithms are taught and validated on the LEACH protocol, and Malicious Nodes (MNs) are eliminated from the network using Real-Time Message Content Validation (RMCV). The suggested protocol may increase network lifetime, Quality of Service (QoS) metrics, and detecting metrics, according to simulated results.

INDEX TERMS: Wireless Sensor Network (WSN), Deep Learning Ensemble Model (DLEM), Deep Belief Network (DBN), Bidirectional Gated Recurrent Unit (BiGRU) and Bidirectional Long Short Term Memory (BiLSTM), Double Deep Q -neural network (DDQN), Dynamic Weight Deep Q-Network (DWDQN), Enhanced Wild Horse Optimization (EWHO), Restricted Boltzmann Machines (RBMs), and Congestion Clustering Energy Aware Routing Algorithm (CCEARA).

1. Introduction

Wireless Sensor Networks (WSNs) have developed into an exciting and growing field of study because of their possible uses in environmental condition monitoring, supporting emergency operations, and enabling security surveillance in both supervised and autonomous missions [1–2]. Several small sensor nodes constitute a WSN that have communication, computing, and sensing capability. Due to features such as flexible deployment, autonomous self-organization, and the ability to provide real-time data collection, WSNs are widely adopted in applications including smart agriculture, medical monitoring, and environmental observation [3]. Nevertheless, since the majority of sensor nodes function on batteries, replacing or recharging these energy sources is highly difficult and often impractical. Consequently, creating routing protocols that are energy-efficient is crucial to ensure optimal utilization of the constrained energy resources in WSNs [4]. Numerous studies have investigated data transmission strategies aimed at distributing energy consumption evenly across the network, as well as data compression techniques designed to minimize the energy expenditure associated with communication [5]. When transmitting data of equal size, energy consumption increases significantly with communication distance, especially if the data gathering network is positioned distant from the base station. As a result, extensive transmission demands can reduce the performance experienced at the sink nodes, causing delays in queuing caused by limited available bandwidth [6]. Wireless sensors offer several benefits, including straightforward deployment, autonomous identification, self-diagnostic capability, and time synchronization that enables coordination with neighboring sensors to establish dynamic, self-organizing networks. However, despite their broad range of applications, WSNs encounter significant constraints such

as limited transmission bandwidth, limited memory, limited processor control, and limited energy supply. These limitations can adversely affect network performance and reduce overall network lifetime [7].
These include homogeneous and heterogeneous WSNs are possible. The hardware and capabilities of a homogeneous network, each sensor node is identical. Heterogeneous networks, on the other hand, have nodes with variations in processing power and storage capacity, reflecting computational diversity. The presence of both communication link heterogeneity and computational heterogeneity might result in higher energy use, consequently reducing the network's total lifetime. Several important factors, including as data aggregation, clustering, routing, standardization, defect detection, work scheduling, and event monitoring, must be properly considered while designing WSNs. Among these, clustering is widely adopted for managing network topology. This approach partitions sensor nodes into groups, referred to as clusters, according to predefined criteria such as ensuring Quality of Service (QoS), efficient utilization of resources, and balanced distribution of network load. The research community has extensively explored clustering as a means to organize sensor nodes to improve network scalability. This method improves the sensor network's operating lifetime by reducing radio transmission frequency. Consequently, clustering enables adaptive management of various application requirements, including improved network performance, decreased latency, increased dependability, improved fault tolerance, and decreased energy consumption [8].
Clustering provides an energy-efficient way to connect sensor nodes in a hierarchical network design. However, packet collisions, increased network congestion, and excessive energy consumption may develop when all nodes provide data straight to the sink node in an ad hoc network. A popular cluster-based methodology that seeks to lower energy consumption is the Low Energy Adaptive Clustering Hierarchy (LEACH). through an effective cluster head selection mechanism [9]. Through the division of the network into smaller groups, the protocol addresses key challenges through improved utilization of available resources. Within each cluster, a cluster head (CH) is selected to serve as an intermediary between sensor nodes and the sink node, which helps shorten the communication distance. To balance energy use and reduce overhead, the role of the cluster head is periodically reassigned at defined time intervals.
In a clustered network, the CH performs data compression on information collected from each of its constituent nodes and above an aggregated packet to the Base Station (BS), thereby reducing the volume of transmitted data. The CH manages overall cluster operations, collects and manages the routing and transmission of sensed data to the sink node from other Sensor Nodes (SNs) in the cluster. To preserve energy and increase WSN life, the CH alternates between sensor nodes in each cluster. The data transmission's energy efficiency to the BS is improved by selecting the CH using an optimum method, and the network lifetime is increased by periodically reassigning this role to distribute energy consumption more equally [10].
The CH establishes a Time Division Multiple Access (TDMA) schedule to coordinate data transmission and avoid message collisions. However, conventional clustering approaches face significant challenges in achieving scalability, reliability, and fault tolerance, while also ensuring efficient data delivery, reduced energy usage, adequate coverage of the sensing area, and prolonged network lifetime. Clustering in WSNs is considered a NP-hard optimization problem, making it difficult to solve effectively using conventional methods [11]. Consequently, advanced techniques derived from Computational Intelligence (CI) and Machine Learning (ML) have worked to obtain more accurate and efficient solutions for clustering optimization.
In WSNs, data traffic patterns are strongly influenced by the specific application and often exhibit periodic or continuous behavior. Excessive upstream traffic can overwhelm sensor nodes, exceeding their processing capabilities and leading to higher latency, packet loss, and frequent retransmissions. Degradation in network performance reduces application dependability. Network congestion may result from resource constraints, node workload, communication channels, and upstream traffic patterns. Congestion degrades overall network performance and increases the rate of packet loss. Moreover, the heterogeneous characteristics of WSNs make them vulnerable to a variety of security attacks [12]. Blockchain is an emerging technology that helps resolve concerns related to data security and dependence on third parties. Although widely known for its role in cryptocurrencies, blockchain is also applied in several domains such as healthcare, manufacturing, and smart city systems. It operates through a distributed ledger where new records are appended only after validation and agreement are achieved using a consensus mechanism.
This paper presents the Congestion Clustering Energy Aware Routing Algorithm (CCEARA), a protocol designed to identify malicious nodes and mitigate congestion by employing optimized cluster-based routing. In addition, a Double Deep Q-Network (DDQN) approach is incorporated, taking into account both the individual node's traffic and the surrounding local network traffic conditions. For CHS, an EWHO method based on residual energy, traffic load, centrality, node degree, inter-node distance, CH-to-BS

distance, and weighting time is used. MNs are identified and eliminated using RMCV combined using deep learning models like BiGRU, BiLSTM, and DBN, trained and evaluated under the conventional protocol.

2. Literature Review

Priya et al., [13] an improved malicious node detection framework, termed Multimodal Contrastive Domain Sharing Generative Adversarial Networks for Wireless Sensor Networks (MND-MCDSGAN-WSN), integrated with blockchain-based distributed storage, is proposed. Initially, sensor measurements are obtained from the WSN dataset. The acquired data are then preprocessed using a Multi-Observation Fusion Kalman Filter (MOFKF), which eliminates noise and enhances data consistency prior to further analysis. The preprocessed data are provided as input to the MCDSGAN framework, which divides network activity into attack types such as Flooding, TDMA, Normal, Blackhole, and Grayhole. In parallel, the MCDSGAN model's parameters are optimized using the Adaptive Marine Predator Optimization Algorithm (AMPOA), which improves the model's generalization capacity and stability. A Fair Proof-of-Reputation (FPoR) consensus process is used to ensure the involvement of reliable nodes, and the confirmed node information is reliably retained using the Interplanetary File System (IPFS) and to maintain tamper-resistant updates within the blockchain network. This integrated framework improves detection reliability, strengthens trust management, and provides a scalable security mechanism suitable for resource-constrained wireless sensor networks. The suggested MND-MCDSGAN-WSN method performs better than the current methods.
Yang et al., [14] a reputation-based malicious node detection scheme, termed MNDREL, was developed using an enhanced LEACH routing protocol for WSNs. Initially, cluster-head nodes are chosen according to the improved LEACH protocol. Subsequently, the remaining sensor nodes organize themselves into clusters by associating with the selected cluster heads and establish packet transmission routes accordingly. Each node appends its identification number along with its reputation assessment value before the packet is transmitted to the sink node. The sink node then parses the received packets to extract node identifiers and compares them with the corresponding source node numbers to generate a list of suspected nodes. Subsequently, by calculating the ratio of each node's suspicion value to its trust value and comparing the result to a predetermined threshold, malicious nodes are found. Two advanced approaches, especially the High-Reliability Trust Evaluation Model (HRTM) and the Fuzzy Logic-Based Multi-Attribute Trust Model (FMATM), were implemented and evaluated under identical experimental conditions. Simulation research indicates that the MNDREL model maintains a reduced false alarm rate while achieving improved efficiency in detecting opposed nodes in wireless sensor networks.
Gunjal and Kulkarni [15] introduced a hybrid Deep Learning-based architecture that combines Long Short-Term Memory (LSTM) with Deep Convolutional Neural Networks (DCNN) using weighted score level fusion for MND in WSN. The suggested model with balanced dataset using proposed Synthetic Minority Over-Sampling Technique-k nearest neighbors (SMOTE-k) algorithm correctly classifies nodes based on behavioral patterns and network dynamics by utilizing the feature extraction powers of DCNN in conjunction with the temporal sequence analysis of LSTM. Data balancing approaches were used to improve the model's performance and mitigate the impact of imbalanced datasets. A dependable and efficient method for improving security in WSNs is the hybrid DCNN-LSTM model, achieving superior accuracy is higher.
Nouman et al., [16] histogram Gradient Boost (HGB), a machine learning-based classifier, is used by base stations to categorize network nodes as either harmful or acceptable. A node's registration is later removed if it is found to be malicious to prevent it from participating in the network going forward. The IPFS records a node's data if it is found to be authentic. The data in this system is divided into sections, and a cryptographic hash is generated for each data segment. These hashes are subsequently recorded on the blockchain. Furthermore, the Verifiable Byzantine Fault Tolerance (VBFT) consensus protocol is adopted in place of Proof of Work (PoW) to achieve agreement among nodes and to verify transactions. The dataset for the WSN-DS wireless sensor network is used in extensive simulation experiments. The original dataset and a class-balanced version of the dataset are both used to evaluate the suggested method. In addition, the performance of the Histogram Gradient Boost (HGB) classifier is compared with several established methods, incorporating the Ridge classifier, Extreme Gradient Boost (XGB), Gradient Boost (GB), Adaptive Boost (AdaBoost), and Linear Discriminant Analysis (LDA). The comparison is carried out using assessment measures, including micro-F1 score, macro-F1 score, recall, accuracy, and precision.
Zawaideh and Salamah [17] an efficient Weighted Trust–based Malicious Node Detection (WT-MND) method is proposed to detect malicious nodes in a clustered wireless sensor network. The approach realistically considers node behavior by incorporating both false-positive and false-negative occurrences. Simulation results demonstrate that the WT-MND scheme effectively detects and isolates nodes exhibiting malicious behavior in a timely manner. The proposed scheme’s effectiveness is achieved through an adaptive trust-update mechanism, it provides a distinct trust-update parameter for each node contingent upon its observed conduct and makes it less difficult for nodes that suddenly malfunction to recover. The

higher DR and lower misdetection ratio (MDR) of the suggested strategy show that it is more effective and adaptable than existing approaches in the literature, which remain relatively stable even as network size increases. Additionally, the approach maintains its efficiency without introducing significant power consumption overhead.

Morsi et al., [18] a trusted mobile node, a hybrid Cluster Head network (ESMCH) is suggested as an efficient and secure method for detecting malicious nodes. Two algorithms are introduced by the ESMCH framework: The Balanced Load Sub-Cluster Head Selection method and the Intelligent Secured Fuzzy Clustering (ISFC) algorithm. These algorithms are intended to decrease reduced overall energy consumption and the distance between nodes. The ESMCH model is designed to secure aggregated data using an Enhanced Elliptic Curve Cryptography (EECC) mechanism. Simulations of the proposed model are conducted using Network Simulator 2 (NS2). Compared to other approaches, including Double Cluster Head-based Secure Aggregation (DCSDA), Elliptic Curve Cryptography-based Digital Signature Algorithm (ECCDSA), and Energy-Efficient Heterogeneous Ring Clustering (E2HRC), the ESMCH model demonstrates superior performance and security. It effectively enhances data protection, improves less energy and extends the network's lifetime. Energy consumption (EC) and routing overhead (RO) are two more network performance criteria that are considered during the evaluation.

Guo et al., [19] developed the Malicious Node Determination Algorithm (MNDA) and the Cluster Head Selection Algorithm (CHSA) for WSN. Initially, an adaptive clustering technique is employed to organize sensing nodes and select optimal cluster leaders, reducing communication overhead and balancing energy usage. Subsequently, the CHSA optimization framework combines multiple swarm-based metaheuristic strategies to identify anomalous behavior patterns in sensor nodes based on trust metrics, packet forwarding behavior, residual energy, and communication consistency. Comprehensive simulations show that, in comparison to traditional detection methods, the suggested CHSA-MNDA model provides enhanced Packet Delivery Ratio (PDR), decreased FPR, increased detection accuracy, and extended Network Lifetime. The results confirm that integrating hybrid swarm intelligence with clustering mechanisms provides an effective and scalable solution for enhancing WSN security against malicious node attacks.

Khashan [20] presented a Block Chain-Machine Learning (BC-ML) used CNN to effectively identify MN in WSN. Employing a hybrid lightweight encryption strategy that increases the security of data transfer between nodes by combining symmetric and asymmetric ciphers. Additionally, using conventional data transaction techniques, a new proof-of-authority methodology based on node digital certificates is suggested. This consensus method improves scalability and energy economy by reducing the computational overhead that comes with big datasets in traditional proof-of-work techniques. Convolutional neural networks (CNNs) and adaptive synthetic sampling are used by the model to examine network properties and node activity to correct dataset imbalances.

Obaid et al., [21] investigated a Cascaded Hermite Laguerre Neural Network (CHLNNet) for MND in WSN. The proposed approach integrates deep learning and optimization techniques with an emphasis on attack detection. Initially, according to the system model, network nodes are randomly placed. Subsequently, the transmission probability is evaluated by considering factors such as the node's geographic position, each node's residual energy and the degree to which cluster-head (CH) and cluster-member (CM) nodes are connected. The Dempster–Shafer Theory (DST) combined with Multi-Dimensional Trust (MDT) metrics is employed to identify the optimal cluster paths (CP) for efficient data transmission, considering residual energy and node-to-node distance as primary constraints. Consequently, the proposed CHLNNet approach attains a high malicious detection rate (MDR), reduced average latency (AL), increased throughput, lower end-to-end delay (E2E), and minimized RO.

Das and Dwivedi [22] malicious node identification and CH selection are done using the Large-Scale Energy-Aware Trust Optimization (LS-EATO) framework. Initially, to select CHs based on parameters including reliability, energy level, distance, and node density, a Harmonic Search Genetic Algorithm (HGA) is used. This method prevents malicious nodes from being selected as cluster heads by integrating trust values. Malicious nodes are subsequently identified using the Energy-Aware Intra- and Inter-Cluster Trust (EAIICT) estimation model. The model computes trust using a combination of distributed and centralized methods to detect malicious nodes. Furthermore, by incorporating residual energy (RE) into the trust calculation, the approach ensures that nodes with low energy are not mistakenly selected as trustworthy nodes. Additionally, MATLAB is used to simulate and compare the suggested LS-EATO approach with existing approaches to demonstrate its effectiveness. By examining parameters including communication overhead, detection accuracy, detection rate (DR), and energy consumption (EC) over a range of network sizes, the experimental assessment demonstrates the dependability of the suggested approach.

Shanmathi et al., [23] an enhanced approach combining Convolutional Neural Networks with Fuzzy Logic (CNN-FL) is proposed for detecting and classifying trustworthy and malicious nodes in WSNs. This method combines a routing strategy enhanced by a Neuro Genetic Optimizer (NGO) with a roulette wheel selection mechanism that is based on the LEACH routing protocol. In addition to ensuring secure data transfer and

significantly reducing latency and energy consumption, the CNN-FL-based routing technique increases network longevity. The suggested approach performs better than current protocols, according to simulation results, A-Star Search-based Neuro-Genetic Secure Routing Algorithm (ASNGSRA), Fuzzy Rule and Cluster-based Secured Routing with Outlier Detection (FRCSROD), and Dual-Domain Multi-Scale Convolutional Neural Network (DMCNN) are a number of such. Numerous criteria, including PDR, energy efficiency, accuracy, precision, recall, F1-score, and latency, show improvements.

Kumar et al., [24] an Improved Deep Convolutional Neural Network (IDCNN), Extended K-Means (EKM), and t-Distribution-based Satin Bowerbird Optimization (t-DSBO) were investigated to detect malicious nodes in wireless sensor networks utilizing the characteristics of each sensor node. In addition to enhancing network security, the approach enables transmission of data in an energy-efficient method utilizing the sensors' remaining energy to select cluster heads. The IDCNN detects malicious nodes and places them in a malicious list. The trusted nodes are clustered by EKM during the energy-efficient data transmission phase, and t-DSBO selects a cluster leader for each cluster depending on leftover energy. The cluster uses the CH to transmit data to the base station. If the CH is out of energy, the t-DSBO algorithm selects another CH. The approach efficiently detects malicious nodes while enabling energy-efficient data transmission, as demonstrated through experimental comparisons with existing methods.

3. Proposed Methodology

This paper presents the CCEARA protocol for detecting attacker nodes and reducing congestion through optimal routing. A DDQN model assesses node load and identifies packet congestion via priority queues. Deep learning is employed to identify and eliminate malicious nodes, while EWHO is used for a fitness function-based cluster-head selection. Routing performance is evaluated using QoS and detection metrics. Figure 1 illustrates the system's flow diagram.

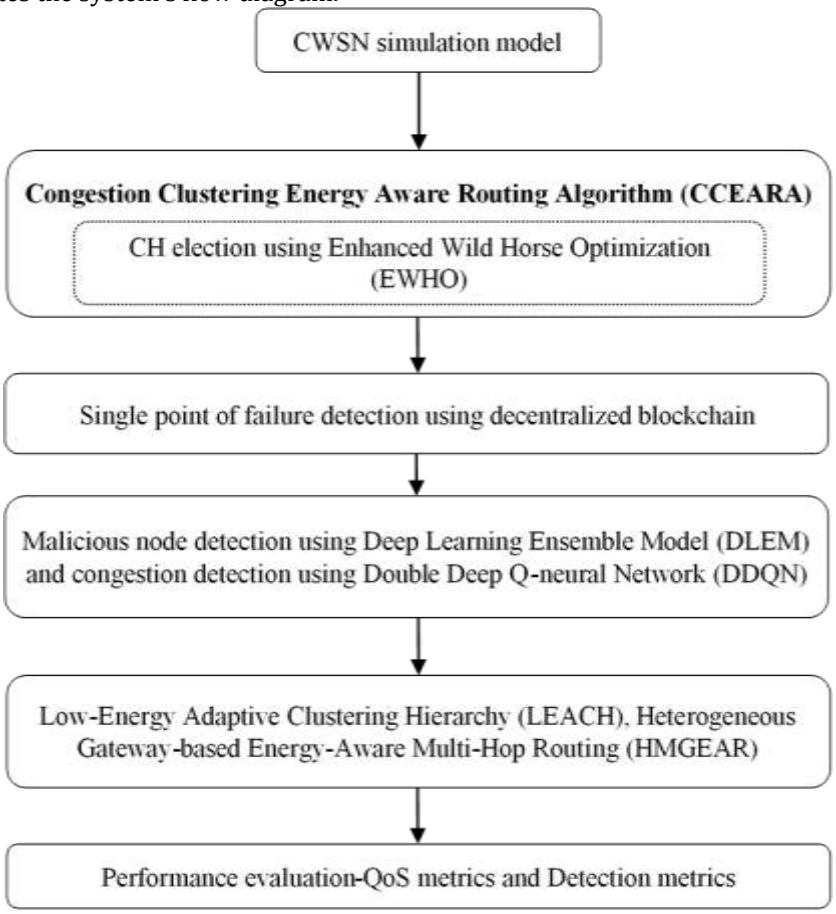

FIGURE 1. PROPOSED SYSTEM'S OVERALL FLOW DIAGRAM

3.1. Network Model

A CWSN with N sensor nodes equally distributed within a circle with diameter M is used in the network model. CHs, BSs, and sensor nodes compose the network, which is in the core and has infinite energy.

3.2. Energy Consumption Model

The multipath fading channel model is used to examine data transmission energy utilization from CHs to the BS; the free space model is disregarded because of the transmission distance. Equation (1) expresses the energy needed to send (e) bits across a distance (d), which is represented as $E_T(e, d)$:

$$E_T(e,d) = \begin{cases} e \times (E_{elec} + \varepsilon_{fs} d^2), d < d_0 \\ e \times (E_{elec} + \varepsilon_{mp} d^4), d \geq d_0 \end{cases} \quad (1)$$

$E_R(e)$ shows the amount of energy required to get the data from the e bit. Equation (2) is the expression.

$$E_R(e) = e \times E_{elec} \quad (2)$$

Equations (1) and (2) employ d to represent the transmitter-receiver distance and E_{elec} to denote the energy required by the transmitter or receiving circuit each bit. Use the free space model in Equation (2) for $d < d_0$, with ε_{fs} serving as the bit's energy factor. If not, ε_{mp} serves in the multi-path faded channel model as the energy factor per bit. Furthermore, d_0 serves as the threshold for distance. The following formula can be derived as long as it is entered as an individual variable into the multipath fading channel model and model in order to create equation (3):

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \quad (3)$$

As a result, Equation (4) defines the energy required to transmit e-bit data is calculated as follows:

$$E_{non-CH} = e.E_{elec} + e.\varepsilon_{fs} d^2_{to\ CH} \quad (4)$$

The separation between the CH and the cluster member node is shown by $d^2_{to\ CH}$ in Equation (4). The cluster's member nodes provide information to the CH, combines it with information it senses from its surroundings, and then provides the BS with the combined data. The CH receives information in the form of e bits. Equation (5) is used to compute the energy used in the process.

$$E_{CH} = \left(\frac{n}{k} - 1\right) e.E_{elec} + \frac{n}{k}.e.E_{DA} + e.E_{elec} + \begin{cases} e\varepsilon_{fs} d^2_{to\ BS}, d_{to\ BS} < d_0 \\ e\varepsilon_{mp} d^4_{to\ BS}, d_{to\ BS} \geq d_0 \end{cases} \quad (5)$$

E_{DA} is the quantity of energy the CH expends on each data point analysis, $d_{to\ BS}$ In equation (5), divide k clusters and maintain n nodes in the monitored region.

3.3. Clustering Model

As WSN clusters grow, inter-cluster communication traffic increases, while intra-cluster traffic rises as clusters decrease. Effective communications and managing energy require determining the optimum cluster number. The following describes the main phases of the clustering protocol [26]:

- Using the network optimal cluster number calculation formula, calculate the required number of clusters (k).
- The necessary k clusters can be constructed through the balanced energy use optimization of the AGglomerative NESting (AGNES) approach.
- Implement CH selection process in each cluster, then advance to EWHO energy balancing approach and huge cluster node selection before and after initial node failure.
- Provided energy and data updates.

To balance node energy demand and reduce total energy consumption, the network evaluates node death after each data transmission round (if a node dies, the process returns to Step 1; otherwise, it proceeds to Step 3). The protocol's preparation phase is represented by steps 1, 2, and 3, and its stabilization phase by step 4.

In the model, some cluster heads (CHs) may be located at distances greater than d_0 from the base station (BS), integrating the concepts of various route fading channels and open spaces to calculate inter-cluster energy consumption. The cluster's overall energy consumption inside the network over a single round is given by [26]:

$$E_{SUM} = kE_{CH} + nE_{non-CH} \quad (6)$$

The AGNES algorithm is a hierarchical clustering method. Initially, each object is treated as an individual cluster. The algorithm then repeatedly merges the two clusters that are closest together until the termination condition is achieved when the number of clusters decreases to (k). The final (k) clusters produced by the algorithm represent the target clusters. The merging criterion is based on the distance between clusters, typically measured in three ways: (1) maximum distance, (2) minimum distance, or (3) average distance. A cluster formation factor may be determined by taking the average and variance of the distances between two clusters throughout the clustering process [26].

A. Fitness Function

The EWHO fitness function select optimal CH from the network's sensor nodes. To avoid selecting a CH-dead node, the fitness function uses residual energy. Additionally, higher centrality is prioritized to minimize transmission distances among cluster members. The mathematical formulation and definitions of the fitness functions are presented below [26]:

CH Residual energy: BS receives data from networked sensor nodes Since previous actions take more energy, the best CH is the node with the most energy. Equation (7) describes f_1 residual energy [26]:

$$f_1 = \sum_{i=1}^{m} \frac{1}{E_{CH_i}} \tag{7}$$

where E_{CH_i} represents the ith CH residual energy.

Distance among sensor nodes: The transmission path distance is the primary determinant of the node's energy dissipation. Minimal transmission distance from base station reduces energy for selected node. Distance between sensors and CH (f_2) is in Equation (8) [26]:

$$f_2 = \sum_{j=1}^{m} \sum_{i=1}^{I_j} D\left(s_i, \frac{CH_j}{I_j}\right) \tag{8}$$

where $D\left(s_i, \frac{CH_j}{I_j}\right)$ indicated the distance between the sensor i and CH_j, where I_j is the quantity of the sensor node that corresponds to CH [26].

CH and BS Distance: For instance, data transmission will be energy-intensive when BS and CH reside. Thus, increased energy use may be connected to a rapid decrease in CH. To transmit information, the node most nearby to the base station is selected. The CH and BS's optimum range (f_3) is shown in equation (9). [26]:

$$f_3 = \sum_{i=1}^{m} D(CH_j, BS) \tag{9}$$

where BS and CH_j are divided by $D(CH_j, BS)$.

Node degree: Each CH has a certain amount of sensor nodes. According to energy loss, the CHs with fewer sensors are selected for a shorter amount of time. Equation (10) expresses node (f_4) degree [26]:

$$f_4 = \sum_{i=1}^{m} I_i \tag{10}$$

I_i is a measure of the number of CH_i sensors.

Node centrality: Equation (11) represents node centrality (f_5), a node's distance from its neighbors [26]:

$$f_5 = \sum_{i=1}^{m} \frac{\sqrt{\frac{\left(\sum_{j\in n}^{m} D^2(i,j)\right)}{n(i)}}}{L} \tag{11}$$

where L is the network dimension and $n(i)$ is the number of CH_i nearby nodes. This situation combines numerous objectives into a single function.

Traffic Load: The trust value surpasses the traffic demand for nodes trust threshold is determined to minimise energy overheads. Additionally, CT_{Min} and CT_{Max} thresholds that lie within the queue's buffer length range are defined to identify network congestion. Network congestion may be divided into three categories based on the established threshold values:

Less: The network is deemed to be congested if the length of the queue is below the limit CT_{Min}.

Medium: While a network's queue length is between CT_{Min} and CT_{Max}.

High: The congestion level is deemed to be excessive if the length of the queue above the limit CT_{Max}. Equation (12) represents the source node s, and let T_s^k k be the "kth" node's buffer size.

$$f_5 = CI_k = f\left(T_s^k\right), CI_{k'} = 1 - CI_k \tag{12}$$

Distance: The transmitter, the possible next node, and the sink are all measured. The values d_1 and d_2 reflect the separations between the anticipated future nodes and the present moment source and the sink, respectively. The equation (13) addresses the matching distance from sender d_1^M and comparable range from sink d_2^M:

$$d_1^M = 1 - d_1, d_2^M = 1 - d_2 \tag{13}$$

Therefore, Equation (14) may be used to compute node distance for matching d_1^M and d_2^M:

$$f_6 = T_d = \frac{\alpha . d_1^M + \beta . d_1^M}{\alpha + \beta} \tag{14}$$

where the d_1^M and d_2^M adjustment parameters are represented by α and β.

Waiting Time of the Nodes: Equation (15) determines if a node may be selected as CH by calculating the waiting time at each cluster node:

$$f_7 = WT = WT_{max} \times \omega\left(1 - \frac{RE}{IE}\right) \times \sigma\left(\left|Avg\left(S_i^k\right) - S_n\right|\right) \tag{15}$$

$\text{Avg}(S_i^k)$ is the transmission range's average nearby node speed, WT_{max} indicates the maximum time that has been set in advance, S_n node speed and residual energy are represented by and RE, respectively. Weighted values are S1, S2, S3, S4, S5, S6, and S7. The single objectives function is shown in equation (16) [26]:

$$F = S1f_1 + S2f_2 + S3f_3 + S4f_4 + S5f_5 + S6f_6 + S7f_7 \quad (16)$$

where, $\sum_{r=1}^{7} Sr = 1\ Sr \in (0, 1)$, Sr values are, respectively, 0.25, 0.20, 0.15, 0.15, and 0.15 [26]. The energy-based fitness evaluation indicates that nodes located at greater distances tend to obtain a higher overall fitness value. In such cases, nodes with high fitness values have a reduced probability of being selected as CHs. In contrast, nodes possessing higher residual energy and a greater number of neighbouring nodes generally produce lower overall fitness values. Nodes that exhibit lower overall fitness values have a high chance of being selected as CHs. These nodes then perform data aggregation to eliminate redundant information. By combining the collected data before transmission, the network reduces unnecessary communication with the BS, thereby conserving energy. The data are gathered and the cluster head (CH) is determined, each node assigns its overall fitness according to its energy level. The probability of a node being selected as CH is more likely if its fitness value is smaller since it uses less energy. The nodes within its connection range receive the data packets from the selected CH.

B. CH Selection using Enhanced Wild Horse Optimization (EWHO)

The WSN framework uses the EWHO algorithm, a metaheuristic optimization method based on the collective behavior of wild horses in the wild, to identify a perfect CH. The WHO mechanism models the natural tendency of horses to leave their original groups and join other groups before reaching adulthood, which helps prevent mating among close relatives such as siblings or daughters. This behavior is utilized to achieve effective CH selection. Stallions and mares remain in the same group and interact with one another while grazing, which supports the optimal selection of CHs. As foals grow older, they leave their original groups and join other groups, eventually forming their own families, a behavior that contributes to effective CH selection.

Wild Horse Optimization (WHO)

Horses are typically organized into two social structures: services domestically and internationally. Foals, stallions, and mares are among the many ages of individuals that constitute their herds (Figure 2). When participating in activities like grazing, stallions and mares often stay in the same group. To develop new family groups and lessen the possibility of inbreeding between stallions and their siblings, the foals progressively leave their natal groups as they become older and join other herds.

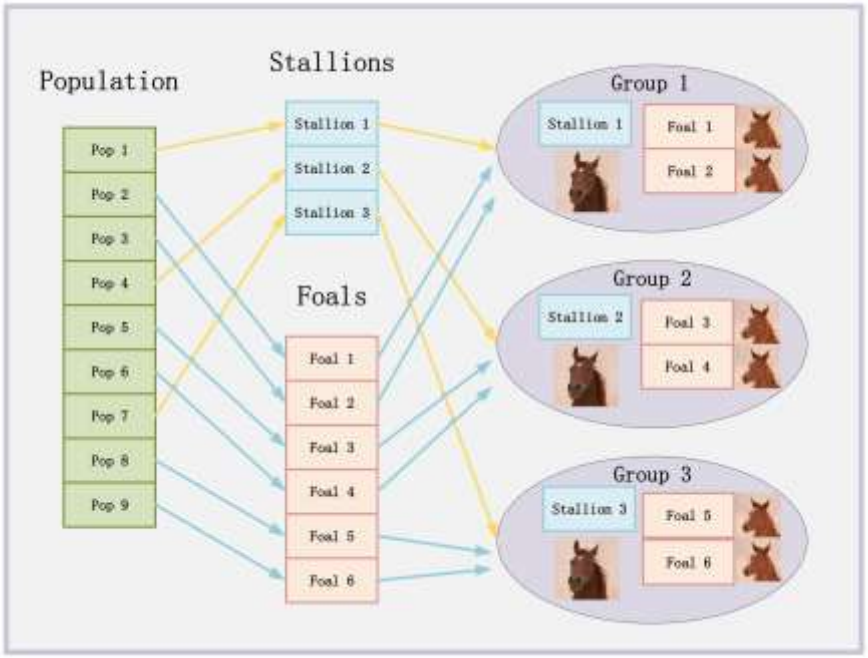

FIGURE 2. GROUPS FORMATION FROM ORIGINAL POPULATION

The WHO algorithm is derived from the natural social behaviors of horses, including grazing, dominance, hierarchical leadership, and mating patterns. It is implemented through five main steps, which are described below.

1. Sensor nodes parameter and random initialization: The global population of the node is the initial initialized, containing Np packets, each of which has Ns nodes. If there are N individuals and G groups, there are G leaders and N-G non-leaders. PS and G/N, is the definition of the percentage of stallions [26].

2. Grazing Behavior: Imagine a horse in the center of the graze field to recreate the grazing phase for the most optimal CH selection. Other individuals can migrate by using equation (17). [27],

$$X^{i}_{G,j} = 2Z\cos(2\pi rZ) \times \left(Sta_{G,j} - X^{i}_{G,j}\right) + Sta_{G,j} \quad (17)$$

where $X^{i}_{G,j}$ and $St_{G,j}$ are optimum CH selection sets for equation (18) determines R, which is an adaptive parameter, and Z, which is a random number between -2 and 2 [27],

$$P = \overrightarrow{R_1} < TR, IDX = (P == 0), Z = R_2 \Theta IDX + \overrightarrow{R_3}\Theta(\sim IDX) \quad (18)$$

where P is a vector with dimensions equal that of the CH election issue and which contains the numbers 0 and 1, $\overrightarrow{R_1}$ and $\overrightarrow{R_3}$ and R_2 are random numbers and vectors [0,1]. Equation (19) determines TR, a linearly decreasing parameter [27],

$$TR = 1 - \frac{t}{T} \quad (19)$$

where the current iteration is denoted by t and the maximum iteration by T.

3. Horse Mating Behavior: A distinctive characteristic of horses is that foals leave their original groups before reaching puberty and mating, which helps avoid inbreeding and supports optimal cluster head (CH) selection. This mating behavior among horses can be modeled using Equation (20) [27],

$$X^{p}_{G,k} = Crossover\left(X^{q}_{G,j}, X^{z}_{G,j}\right), i \neq j \neq k, q = z = end \quad (20)$$
$$Crossover = Mean$$

$X^{p}_{G,k}$ represents horse p's position in group k, generated by horse q's position in group i and horse z's position in group j. The crossover probability is set to PC by the basic WHO.

4. Group Leadership: Group leaders (stallions, Sta) guide the members of their group toward an appropriate location, such as a waterhole. They may also compete with other stallions for access to the water source, where the dominant group gains priority in using it for optimal CH selection. This behavior is modeled using Equation (21) [27],

$$\overline{Sta}_{G,j} = \begin{cases} 2Zcos(2\pi RZ) \times \left(WH - Sta_{G,j}\right) + WH \; if \; rand > 0.5 \\ 2Zcos(2\pi RZ) \times \left(WH - Sta_{G,j}\right) - WH \; if \; rand \leq 0.5 \end{cases} \quad (21)$$

In the jth group, $\overline{St}_{G,j}$ represents the candidate position, $Sta_{G,j}$ represents the current leader position, and WH represents the waterhole position.

5. Exchange and Selection of Leaders: Initially, leaders are randomly chosen as CHs. Subsequently, the optimal leaders (best CHs) are determined according to their fitness values. The exchange of positions between leaders and other individuals is modelled using Equation (22), as shown below. [27],

$$St_{G,j} = \begin{cases} X^{i}_{G,j}, if \; f\left(X^{i}_{G,j}\right) < f\left(St_{G,j}\right) \\ St_{G,i}, iff\left(X^{i}_{G,j}\right) \geq f\left(St_{G,j}\right) \end{cases} \quad (22)$$

where $f\left(X^{i}_{G,j}\right)$ and $f\left(St_{G,j}\right)$ are the stallion's and foal's respective fitness values.

Although WHO demonstrates competitive performance compared with several algorithms, it has limitations such as weak exploitation ability and developing in local optima. As a swarm-based metaheuristic inspired by horse social behaviours such grazing, dominance, leadership hierarchy, and mating patterns, the EWHO algorithm was proposed to solve these difficulties. [27].

Enhanced Wild Horse Optimization (EWHO)

The three methods increase the fundamental algorithm's optimization performance. Foals and stallions are first given a Random Running Strategy (RSS), reflecting the natural chasing and running behavior of wild horses. Second, an Inertia Weight Strategy (IWS) is introduced in exploration and exploitation in a balanced manner using the waterhole method. Finally, the Waterhole Mechanism (WHM) is implemented for stallions to further enhance the quality of the solutions.

1. Random Running Strategy (RRS)

Horses that are wild like chasing and racing. This is why the RSS is recommended for both stallions and foals. Equation (23) is used to update position:

$$X^{i}_{G,j} or \; \overline{Sta}_{G,j} = lb + (ub - lb) \times rand \quad (23)$$

where lb is lower border and ub is upper limit.

2. Dynamic Inertia Weight Strategy (DIWS)

The most effective global solution may be found with the use of the dynamic weight method. Therefore, in the first formula of Equation (21), the waterhole receives a dynamic inertia weight assist the stallions select a superior one. The following is the calculation of the weight and modified Equation (24):

$$w = \begin{cases} wmin + (wmax - w\min) \times \dfrac{f(t)i - f(t)min}{f(t)avg - f(t)min}, if\ f(t)i \le f(t)avg \\ w\ max, if f(t)i > f(t)avg \end{cases} \quad (24)$$

$$\overline{Sta}_{G,j} = 2Z\cos(2\pi RZ) \times \left(WH - Sta_{G,j}\right) + w \times WH \quad (25)$$

where $f(t)min$ is the population's limited fitness value, $f(t)avg$ is all stallions' average fitness, The upper and lower bounds are denoted by $w\min$ and $wmax$, while $f(t)i$ is the benefit of fitness for the current stallion at t[th] iteration.

3. Waterhole Mechanism (WHM)

In some cases, two stallions may discover the same waterhole at the same time, which can lead to competition each other for access. Enhance solution quality, the Waterhole Mechanism (WHM) is introduced Replace Equation (21)'s second expression. The position update is then performed using Equation (26), as given below.

$$\overline{Sta}_{G,j} = WH - Z \times (Sta_{G,j} \times Q_1 - Sta_{G,j} \times Q_2) \quad (26)$$

where the random integers between -1 and 1 are Q_1 and Q_2.

Equation (27) substitutes Equation (21) in WHO by combining DIWS and WHM:

$$\overline{Sta}_{G,j} = \begin{cases} 2Z\cos(2\pi RZ) \times \left(WH - Sta_{G,j}\right) + w \times WH, if\ R_3 > 0.5 \\ WH - Z \times (Sta_{G,j} \times Q_1 - Sta_{G,j} \times Q_2), if\ R_3 \le 0.5 \end{cases} \quad (27)$$

Algorithm 1 shows the EWHO in the WSN model with the most effective CH selection.

Algorithm 1. Pseudocode of EWHO

Start

1. Input EWHO parameters: $PC = 0.13, PS = 0.2, w\min = 0.01, wmax = 0.99$, PR $= 0.1$
2. Select N and T for the population and maximum iterations.
3. Randomize the horse population.
4. Select stallions and form foal groupings.
5. **While ($t \le T$)**
6. Utilizing Equation (19), determine *TR*.
7. Utilizing Equation (20), determine *Z*.
8. **For the number of stallions**
9. **For the number of foals**
10. **If** rand > PC
11. Equation (17) updates foal position.
12. **Else If** rand > PR
13. Equation (20) updates foal position.
14. **Else**
15. Equation (23) updates foal position.
16. **End**
17. **End for**
18. **If** rand > PR
19. **If** rand > 0.5
20. Utilize Equation (25) to ascertain the stallion's candidate position.
21. **Else**
22. Utilize Equation (24) to ascertain the stallion's candidate position.
23. **End if**
24. **Else**
25. Utilizing Equation (23), determine the candidate position for the stallion.
26. **End if**
27. **If** the stallion is in more resilient candidate position.
28. Use the candidate position to take the stallion's place.
29. **End if**
30. **End for**
31. Equation (22) is used to change the positions of foals and stallions.
32. t = t + 1
33. **End While**
34. Provide the best option that EWHO to identify.

End

3.4. Priority-Based Congestion Control using Double Deep Q-neural Network (DDQN)

Applications, routing, queuing, and neighbors comprise priority-based congestion management. Control and data packets are divided by the application module. Data traffic is further identified according to QoS requirements, and packets are grouped into four types: (i) dependable, (ii) typical, (iii) slow, and (iv)

crucial. To let each node to locally inform its neighbors about routing decisions, control packets are transmitted to nearby nodes. The queuing module assigns priority levels to packets before they are transmitted for channel access. Meanwhile, the module of the neighboring node maintains a record of forwarded packets and exchanges information regarding node status and quality with adjacent nodes. The routing module selects the most suitable paths by considering the attributes of neighboring nodes that indicate node quality. The choice of nodes and CHs is determined using a clustering approach along with a fitness function. The queuing module classifies packet priorities into two categories: queues with (i) Request Traffic (RET) and (ii) Non-Request Traffic (NRET). Packets that are susceptible to delays are put in the RET queue with a higher priority. The NRET buffer contains other types of data packets. Dividing incoming packets into RET and NRET queues helps reduce latency by ensuring timely handling of critical data within a specified time limit. In addition, the queuing model employs a DDQN to determine the priority of data packets. Figure 3 shows the traffic categorization that the application module performs.

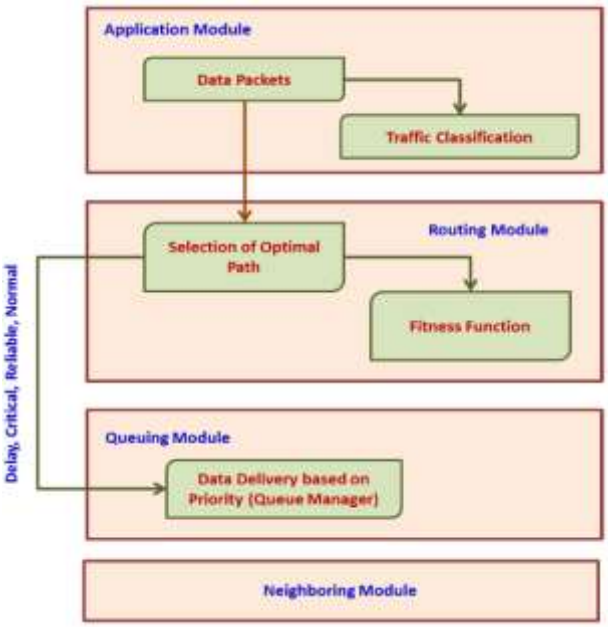

FIGURE 3. TRAFFIC CLASSIFICATION

Each packet transported over the communication mediums is prioritized by traffic type in priority-based data transfer. A multi-queue priority strategy lets nodes manage various traffic class queues, where each class is given a distinct level of importance within the service policy. Each queuing model uses specific threshold values for incoming packets according to the priority queues. These thresholds help manage network congestion efficiently and increase the network's overall efficiency.

Considering QoS requirements and data delivery, Q-learning and deep learning techniques are used in the DDQN reinforcement learning approach. DDQN is a reinforcement learning architecture that allows the agent to determine the most effective ways to increase long-term cumulative rewards based on classification accuracy or clustering performance [28]. It extends the Q-learning concept, which learns the optimal action-value function by value iteration, with a neural approximation and a double-estimator mechanism to minimize overestimation bias. The action-value of reinforcement learning is equivalent to the anticipated discounted total reward and can be represented as in equation (28):

$$Q(s_t, a_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t, a_t\right] \tag{28}$$

where $E(G)$ denotes the expected return, r_{t+k+1} is the discount factor, and $\gamma \in [0,1]$ is step $t+k+1's$ reward. Equation (29) of Bellman optimality is satisfied the most efficient action-value function $Q*(s,a)$ produces:

$$Q(s_t, a_t) = \mathbb{E}\left[r_t(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) \mid s_t, a_t\right] \tag{29}$$

Finding the most effective path of action is the objective of all value-based reinforced learning systems $\pi^* = \arg\max_a Q(s,a)$ the predicted cumulative reward is maximized. In standard Q-learning, the value iteration update rule is defined as equation (30):

$$Q_{I+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)\right) \tag{30}$$

The neural network is usually used to parameterize the policy π where $\alpha_t(s_t, a_t)$ is the learning rate at step t. In small-scale discrete state-action spaces, this tabular form performs efficiently; however, for continuous or large-scale environments such as big-data feature spaces, the method becomes infeasible. Therefore, a the neural network parameters for data delivery by θ is introduced to estimate the Q-function, which can be optimized by reducing the loss function that as equation (31):

$$L(\theta) = \mathbb{E}\left[\left(Y_t - Q(s_t, a_t; \theta)\right)^2\right] \tag{31}$$

where $Y_t = r_t + \gamma \max_a Q(s_{t+1}, a'; \theta)$ denotes the target value. By performing gradient descent on Equation (32), the update rule for the parameter θ is:

$$\theta_{t+1} = \theta_t + \alpha\left(Y_t - Q(s_t, a_t; \theta_t)\right)\nabla_{\theta_t} Q(s_t, a_t; \theta_t) \tag{32}$$

The objective function of DQN is defined as equation (33):

$$Y_t^{DQN} = r_t + \gamma \max_a Q(s_{t+1}, a'; \theta^-), \tag{33}$$

In this case, θ stands for the target network's parameters. However, DQN tends to overestimate Q-values because the same network is used for the selection and assessment of activities. Despite the effectiveness of the DQN to approximate the Q-value function with deep learning, this shared form can frequently lead to overestimation bias. To mitigate this problem, the DDQN presents two distinct networks; one that chooses actions and the other that estimates their resulting values to achieve more robust learning and precise value estimation [29]. To overcome overestimation, Double Q-Learning present's two estimators $Q_0(s, a)$ and $Q_1(s, a)$ that decouple action selection and evaluation. The update rule is in equation (34):

$$Q_i^{t+1}(s_t, a_t) = Q_i^t(s_t, a_t) + \alpha_i(s_t, a_t)[r_t + \gamma Q_{1-i}^t(s_{t+1}, a^*) \tag{34}$$
$$\text{where } a^* = \arg\max_a Q_i^t(s_{t+1}, a), i = 0,1$$

In DDQN, this principle is extended to deep networks. Action selection is done by parameters θ for the target network in addition to the online network with parameters evaluates the selected action θ^-. Thus, to estimate the desired objective function in DDQN, DQN repeatedly modifies the neural network parameters during practical training, as shown in equation (35).:

$$Y_t^{DDQN} = r_t + \gamma Q\left(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta); \theta^-\right), \tag{35}$$

In Figure 4, the flow of the DDQN is illustrated: The next action is selected by one network (the "online" or assessment network), and the activity's worth is assessed by a different target network, thereby reducing overestimation and improving stability of learning.

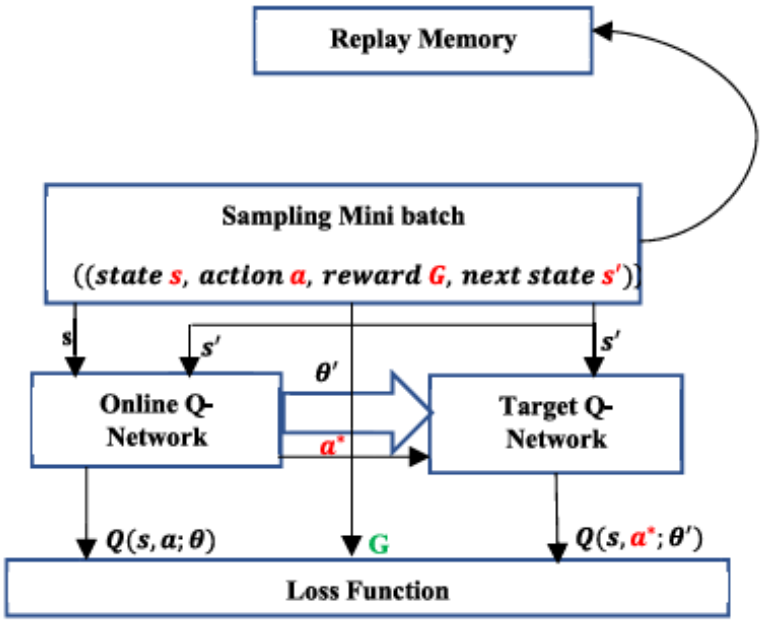

FIGURE 4. MODEL OF DOUBLE DEEP Q NETWORK (DDQN)

The suggested structure, state inputs s_t to the DDQN model. Each action a_t represents a possible classification or clustering operation based on the selected features, while the reward r_t corresponds to the performance gain. The decoupling of the selection and evaluation networks makes the learning process stable, convergent, and unbiased by overestimation. Across several episodes, the DDQN progressively

updates the evaluation network parameters θ, whereas θ is updated with the intended network parameters softly after each N step.

3.5. THREE-LAYERED ARCHITECTURE

The design has three levels, as seen in Figure 5. The first and third levels correspond to the CWSN, which includes sensor nodes, CHs, and the BS; the RMCV and DLEM network layer enabled by blockchain is the network layer. The DLEM model integrates DBN, BiLSTM, and BiGRU, and is trained and evaluated using a dataset generated by the LEACH protocol. In the CWSN, nodes carry out routing through the LEACH and HMGEAR protocols. However, certain nodes may behave maliciously during the routing process. To reduce such activities, The RMCV scheme and the trained model are used to identify malicious nodes (MNs). Routing nodes are categorized using DLEM and RMCV. Following the classification stage, legitimate nodes (LNs) are registered using the Proof of Authority (PoA) mechanism. Following malicious node detection (MND), the LEACH and HMGEAR protocols are then used for routing.

BLOCKCHAIN NETWORK: The blockchain-enabled RMCV and DBM mechanisms are used to identify MNs in the WSN. These mechanisms function at the BS and CHs. Following MND, smart contracts register LNs on the blockchain. The PoA consensus procedure is used when LNs register on the blockchain network.

CWSN: The CWSN's N sensor nodes route assisting with LEACH/HMGEAR. CHs process environmental data from sensor nodes. Before transmitting the data to the BS, the CHs evaluate and compile it. In WSNs, certain nodes may behave maliciously by transmitting false information, disrupting data traffic, or intentionally dropping packets. To address this issue, malicious node detection is carried out using an architecture driven by RMCV and DLEM. The RMCV method evaluates the trust level of nodes, while node classification is performed using DBN, BiLSTM, and BiGRU models. MNs in WSNs are found using the RMCV technique. Victim nodes in the network may receive adversarial messages transmitted by malicious nodes. These messages are assessed according to trustworthiness and integrity metrics, which are determined using three criteria: message conflict, similarity, and path. The network is removed of the malicious nodes once they are identified. Subsequently, routing takes place only among LNs through the LEACH and HMGEAR protocols.

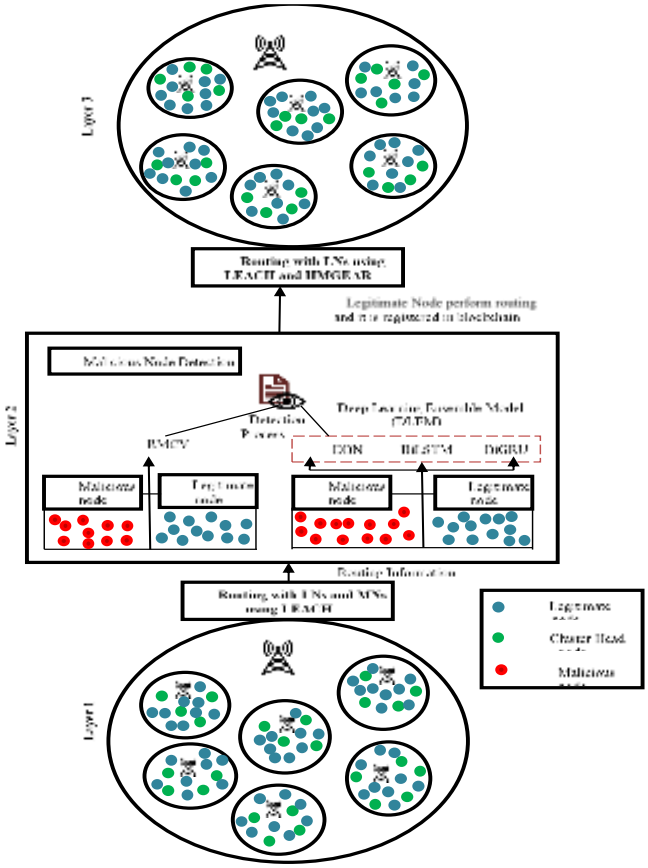

FIGURE 5. PROPOSED 3-LAYERED SYSTEM MODEL

Deep Belief Network (DBN)

A DBN is a network architecture formed by stacking multiple RBMs. When DBN is used for MND, training the network is a prerequisite. The two main stages of training are backward fine-tuning and pre-training. Each RBM layer is initially trained individually and unsupervisedly during the pre-training phase to

preserve important feature information. The input is then the RBM's output feature vector for a BP network that is created at the DBN's last layer. The entity relationship classifier is then put through supervised training. However, rather than optimizing the weights for the whole DBN, each RBM layer only does so for feature mapping in its own layer. As a result, the BP network allows the RBM layers to be changed and enhances the DBN model in its entirety by propagating the error signals from the top layer to the lower levels. The weights of a deep BP network may be initialized by training the RBM network. Figure 6 illustrates the DBN-based detection paradigm.

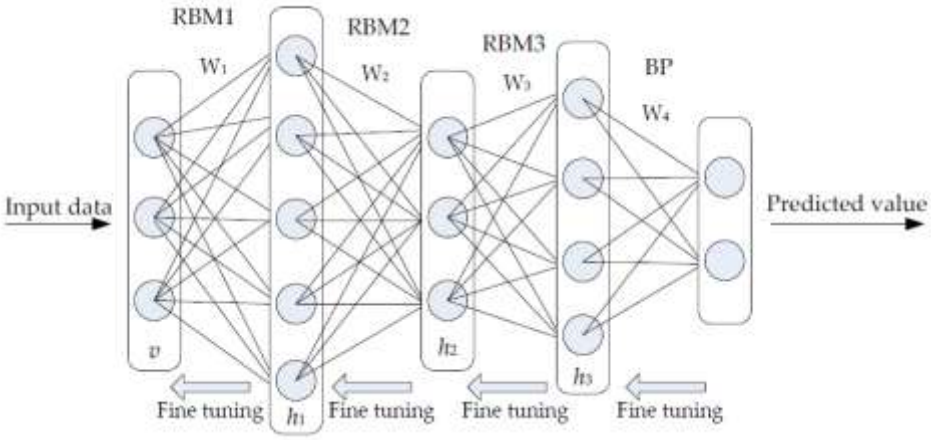

FIGURE 6. A DEEP BELIEF NETWORK-BASED DETECTION MODEL

A network that is RBM is one that contains layers that are both visible and hidden [30]. Figure 6 displays an RBM network topology with three layers: v, which connects MND data, h, which extracts effective features from input data, and W, it represents the visible and hidden layers' connection weight. The energy-based RBM model [31] uses v_i to describe visible-layer neuron i state and bias a_i, neuron j's state in the representation for the hidden layer is h_j, with corresponding bias b_j, and neuron i and j's connection weight is w_{ij}. Equation (36) represent the energy of the RBM:

$$E(v,H|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij h_j} \tag{36}$$

In the equation (36), $\theta = (w_{ij}, a_i, b_j)$ is RBM parameter, n and m are visible layer and hidden layer's corresponding neuron numbers. Equation (37) may be used to obtain the distribution of (v, h) joint probability:

$$p(v,h|\theta) = \frac{1}{Z(\theta)}\exp\big(-E(v,h|\theta)\big) \tag{37}$$

Where $Z(\theta) = \sum\sum \exp(-E(v,h|\theta))$ is the normalization factor. By determining the sample's maximum logarithmic likelihood function using equation (38), for the number N training sample, it is possible to determine parameter θ.

$$\theta^* = \underset{\theta}{\text{argmax}} L(\theta) = \underset{\theta}{\text{argmax}} \sum_{n=1}^{N} logp(v^n|\theta) \tag{38}$$

Where $p(v|\theta) = \frac{1}{Z(\theta)} \sum_h \exp\big(-E(v,h|\theta)\big)$ is v's likelihood function.

Contrastive Divergence (CD) improves network parameter training efficiency and RBM learning. In this method, visible layer vector values are used to calculate hidden layer neuron states. To update the hidden representation, the hidden layer states are determined from the visible neurons once the visible layer is reconstructed. Since the neurons within the same RBM layer operate independently, the (j)th hidden neuron is activated when determined from the visible layer states, and its activation probability is expressed in Equation (39).:

$$p\big(h_j = 1|v,\theta\big) = \sigma\left(b_j + \sum_i v_i w_{ij}\right) \tag{39}$$

Equation (40) represents the activation probability of the hidden layer's reconstruction,

$$p(v_i = 1|h,\theta) = \sigma\left(a_j + \sum_j w_{ij}\, h_j\right) \tag{40}$$

Additionally, equation (41) provides the revised equations for the RBM weights and bias parameters.

$$\begin{cases} w_{ij}^{k+1} = w_{ij}^{k} + \varepsilon\left(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{recon}\right) \\ a_i^{k+1} = a_i^{k} + \varepsilon(\langle v_i\rangle_{data} - \langle v_i\rangle_{recon}) \\ b_{ij}^{k+1} = b_j^{k} + \varepsilon\left(\langle h_j\rangle_{data} - \langle h_j\rangle_{recon}\right) \end{cases} \quad (41)$$

Among them, $\langle . \rangle_{data}$ represents the original MN data model, $\langle . \rangle_{recon}$ represents the reconstructed model, ε represents the learning rate, k represents the CD algorithm iterations, w_{ij}^{k+1} represents the updated weight matrix, and a_i^{k+1} and b_{ij}^{k+1} show the bias vectors once the hidden and visible layers have been updated.

Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is an enhanced form of the LSTM network developed to improve malicious node detection while maintaining long-term memory. It processes the data sequence in two directions: one layer analyzes the data in the forward direction, whereas the opposite is accurate for the other. Three gating mechanisms are the forget, input, and output gates and the cell state that make up an LSTM [32]. Equation (42) shows how an LSTM unit operates conceptually.

$$i_t = \sigma(Y_i . x_t + W_i . h_{t-1} + b_i) \quad (42)$$

The weight matrix is represented by *Y* and *W*, the nodes at a time (t) are represented by x_t, the previous LSTM unit's output is represented by h_{t-1}, the sigmoid function is represented by σ, the bias term by σ, and the input gate at time t by b, σ. Equation (43) illustrates the significance that the input gate identifies the preceding unit's node data that requires update [32]:

$$f_t = \sigma\left(Y_f . x_t + We_f . h_{t-1} + b_c\right) \quad (43)$$

where f_t is the forget gate that computes node importance and forgets past detrimental nodes per equations (44-45).:

$$\tilde{c}_t = tanh(Y_c . x_t + We_c . h_{t-1} + b_c) \quad (44)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (45)$$

where Equation (42) illustrates how the tangent activation function is used to determine the candidate's c_t state. Equation (47), where ⊙ stands for point-to-point multiplication, then evaluates and provides the current cell state:

$$g_t = \sigma\left(Y_g . x_t + We_g . h_{t-1} + b_g\right) \quad (46)$$

$$h_t = g_t \odot \tanh(c_t) \quad (47)$$

Equation (46) computes the output gate g_t, where h_t is the LSTM unit's output, as shown by Equation (47). As seen in Figure 7, two LSTM layers comprise a BiLSTM. The BiLSTM's output layer, v_t, is represented by equation (48): [32],

$$v_t = [\vec{h}_t, \overleftarrow{h}_t] \quad (48)$$

The representations $\overleftarrow{h}_t$ and $\vec{h}_t$ represents the LSTM units' forward and backward outputs, respectively. The final result, v_t, is obtained by integrating these two outputs. The fundamental concept of a Recurrent Neural Network (RNN) is that the data are presented progressively, adding temporal information, rather than giving the neural network in a WSN the whole set of node information at once. For the next step, the new input is processed together with the output obtained from the previous step.

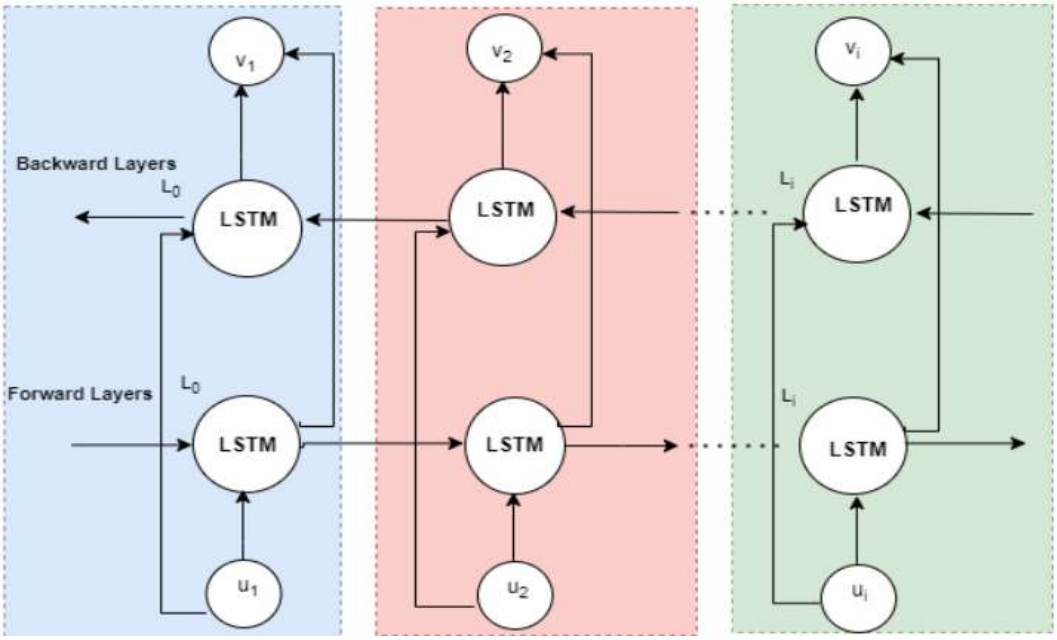

FIGURE 7. FRAMEWORK DIAGRAM OF BiLSTM

BiLSTM executes LSTM layers both forward and backward to obtain bidirectional hidden states, as shown in Figure 7. While the backward LSTM evaluates the sequence in the opposite way, the forward LSTM examines it from the first element to the last. The final bidirectional hidden state is produced by combining the hidden states acquired from the forward and backward directions. When the learning rate ranges between 0 and 2, the Adadelta optimizer is used [33].

Bidirectional Gated Recurrent Unit (BiGRU)

The GRU addresses the gradient vanishing issue by using reset and update gates. When gradients diminish during backpropagation through time, the learning process is affected, making the model difficult to train. Compared with LSTM, GRU requires less memory and often provides better computational efficiency. It also operates faster than LSTM. However, LSTM generally performs better when handling datasets that contain long sequences. The malicious nodes (MNs) in the network are found and removed using deep learning (DL) models. While these models identify the legitimate nodes (LNs), a smart contract is used to register them on the blockchain network. The update gate (z) and the reset gate (r) are the two primary gates employed in a GRU unit. By selecting what to keep and what to discard, these gates regulate the flow of information. In particular, the update gate sets the proportion of previous cell memory kept, whereas the reset gate controls how new cell information is combined with past memory.

BiGRU's Gated Recurrent Unit is shown in Figure 8. The linear interpolation between the preceding activation and the GRU activation at time t, h_t^{j-1} and the candidate activation $\tilde{h}_t^j$ may be calculated using equation (49):

$$h_t^j = \left(1 - z_t^j\right)h_t^{j-1} + z_i^j \tilde{h}_t^j \tag{49}$$

Equation (50) computes the update gate z_t^j, this controls the cell's activation update rate:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1}) \tag{50}$$

where W_z and U_z are weights for updating z_t^j, x_t time step t vector of sequences, and σ is a logistic sigmoid. The candidate activation $\tilde{h}_t^j$ may be calculated using equation (51) as with regular recurrent cells.

$$\tilde{h}_t^j = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{51}$$

where is element wise multiplication and r_t is reset gates. At the reset gate, the number of previous states h_{t-1} may survive is decided. Zero reset gate causes $\tilde{h}_t^j$ to forget previous states. In equation (52) after the update gate, the reset gate r_t is determined.,

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{52}$$

Two sequences one forward and one backward are included in BiGRU. Forward and backward sequences are combined using element-wise addition to entirely account for the expression of information in various directions.,

$$h_i = h_i^F \oplus h_i^B \tag{52}$$

The forward sequence is represented by h_i^F, the backward sequence by h_i^B, the hidden representation is represented by $H = [h_1,...,h_1]$, and the sentence's semantic distribution mean value is indicated by μ. Equation (53) is used in the suggested model to determine the transmission time from the nth client to the BS in the tth communication round [34],

$$\tau_n^{up}(t) = \frac{\gamma_n i_n(t)}{Blog_2\left(1 + \frac{P_n(t)h_n(t)}{BN_0}\right)} \tag{53}$$

where B, $P_n(t)$, N_0, and γ_n provide the system bandwidth, the nth node's transmission power, the amount of bits produced by the nth node, in addition to the noise power spectral density. Additionally, Equation (54) provides the tth communication process's nth node's energy consumption,

$$E_n^{up}(t) = \frac{P_n(t)\gamma_n i_n(t)}{Blog_2\left(1 + \frac{P_n(t)h_n(t)}{BN_0}\right)} \tag{54}$$

Moreover, proof-of-authority consensus is used on the blockchain network to register LNs. In terms of computational cost, the technique is superior than proof-of-work.

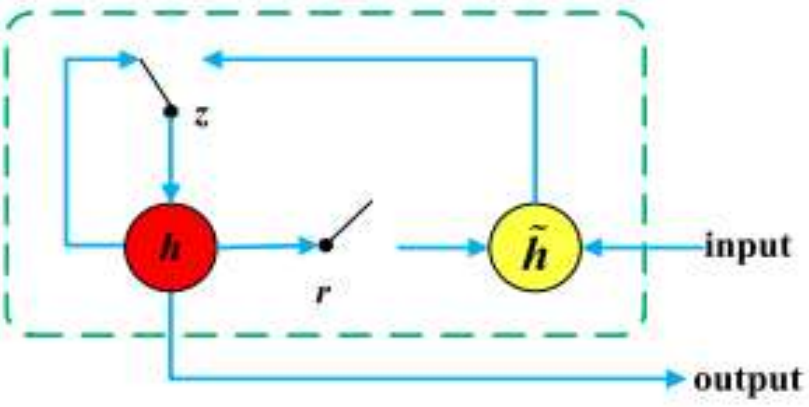

FIGURE 8. GATED RECURRENT UNIT ILLUSTRATION

4. SIMULATION RESULTS AND DISCUSSION

DoS assaults are identified using the WSN-DS dataset, which was developed especially for WSNs. The LEACH procedure was used to get the data. Table 1 describes the dataset's 374,661 records, DoS attacks include Blackhole, Grayhole, Flooding, Scheduling, and normal (no-attack) attacks. https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds for the dataset. Training uses 60% of the dataset, while testing uses 40%. The required data was gathered and the suggested model was simulated using Network Simulator 2 (NS2). Table 2 lists the important simulation parameters. An Intel® Xeon® i5-3570 CPU with a 3.50 GHz clock speed, 16 GB of RAM, and a 10 MB cache was used for the experiments. Within a 200 × 200 m^2 sensing area, simulations were operated with varied numbers of sensor nodes and percentages of CHs at many base station (BS) positions. The BS positioned itself in coordinates (50, 50), (100, 100), and (150, 150), between 100 and 400 sensor nodes were present and the percentage of CHs varied from 15% to 30%.

TABLE 1. DETAILS OF WSN-DS DATASET

ATTACK	TRAINING SET	TESTING SET
Blackhole	6029	4020
Grayhole	8758	5838
Flooding	1988	1324
Scheduling	3982	2656
Normal	204039	136027
Total	224796	149865

TABLE 2. SIMULATION-USED PARAMETERS

Parameter	Value
Network area	$200 \times 200\ m^2$
Base Station location	(50,50), (100,100), (150,150)
Number of sensor nodes	500 nodes
Number of clusters	20
Initial Sensor Node Energy	2.0 J
$\mathbf{E_{elec}}$	50 nJ/bit
$\boldsymbol{\varepsilon_{fs}}$	10 pJ/bit/m^2
$\boldsymbol{\varepsilon_{mp}}$	0.0013 pJ/bit/m^4
$\boldsymbol{d_0}$	30 m
$\boldsymbol{d_{max}}$	200 m
Packet Size	4000 bits
% of CHs	10% to 30%
Routing Protocol	LEACH, HMGEAR
MAC Protocol	CSMA/TDMA
Attacker intensity	10%, 30%, 50%
Network Topology	Random distribution
Number of rounds	3000

The BS distance, node degree, residual energy, centrality, and nearby nodes are supplied to EWHO to optimize CH selection. CCAR uses residual energy, node degree, and distance. These parameters are

extensively used to improve WSN energy efficiency, the proposed method, CCEARA, is compared against several established approaches, including ESMCH [18], CHSA-MNDA [19], CHLNNet [21], CNN-FL [23], BCOA-EEBCDL (Beetle Chimp Optimization Algorithm–Energy Efficient Blockchain Deep Learning), and CWHO-CCAR (Chaotic Wild Horse Optimization–Congestion Clustering Aware Routing).

QoS Metrics: According to simulation findings, the suggested protocol could improve the PDR, decrease PLR, increase network throughput, and decrease average energy usage.

Throughput: The quantity of data that transfers across a network in a certain length of time is referred to as throughput.

The expression for PDR is,

$$R_{Success} = \frac{P_{succ}}{P_{all}} \times 100\% \tag{55}$$

where P_{succ} shows the amount of packets the router is able to deliver successfully, P_{all} shows the number packets were sent via the router overall, and $R_{Success}$ is the percentage of packet success.

PLR can be expressed as,

$$R_{drop} = \frac{P_{drop}}{P_{all}} \times 100\% \tag{56}$$

While P_{drop} indicates how many packets were dropped by the router, P_{all} shows the entire number of packets delivered by the router, and R_{drop} represents the ratio of packet loss.

Average energy consumption: It indicates the average amount of energy used by each node throughout each time. The unit of measurement has been mJ.

Detection metrics: Since different scenarios require different evaluation criteria, various performance indicators, such as overall accuracy (A), are used to evaluate the efficacy of intrusion detection techniques, including True Positive Rate (TPR), True Negative Rate (TNR), FPR, and FNR. TPR measures attack detection, whereas TNR measures normal (no-attack) detection. FPR shows how many usual incidents are mislabeled assaults, while the percentage of assaults that are mistakenly labelled as normal is shown by FNR. Overall Accuracy (A) reflects the total proportion of correct decisions, including both correctly detected attacks and correctly identified normal cases.

The performance metrics is described following equations (57-61):

$$\text{TPR} = \frac{TP}{TP + FN} \times 100\% \tag{57}$$

$$\text{TNR} = \frac{TN}{TN + FP} \times 100\% \tag{58}$$

$$\text{FPR} = \frac{FP}{FP + TN} \times 100\% \tag{59}$$

$$\text{FNR} = \frac{FN}{FN + TP} \times 100\% \tag{60}$$

$$\text{A} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{61}$$

where TP is the number of assaults properly recognized. TN is the number of normal (no-attack) items. FP represents normal (no-attack) events classified as attacks. The number of assault instances that are mistakenly categorized as normal (no-attack) is known as FN. Table 3 shows the attack detection results of DL techniques in relation to assessment metrics.

TABLE 3. INTRUSION DETECTION SYSTEM PERFORMANCE METRICS

METHODS	TPR (%)	TNR (%)	FPR (%)	FNR (%)	A (%)
ANN	84.12	85.61	14.39	15.88	85.37
CNN	86.69	87.45	12.55	13.31	87.62
LSTM	89.54	90.89	9.11	10.46	90.41
GRU	91.49	92.08	7.92	8.51	92.17
BiLSTM	90.51	91.44	8.56	9.49	91.35
BiGRU	92.58	93.67	6.33	7.42	93.46
DBN	95.82	97.18	2.82	4.18	96.51

TABLE 4. ROUTING METHODS PERFORMANCE ANALYSIS COMPARISON

Number of nodes	Network lifetime (rounds)						
	ESMCH	CHSA-MNDA	CHLNNet	CNN-FL	BCOA-EEBCDL	CWHO-CCAR	EWHO-CCEARA
100	892	1058	1234	1462	1645	1859	2030
200	1241	1395	1586	1742	1921	2158	2325

300	1625	1753	1975	2086	2258	2475	2620
400	1858	2026	2327	2455	2587	2856	3020
500	2215	2377	2581	2726	2814	3067	3450
Number of nodes	**Throughput(Mbps*10⁴)**						
	ESMCH	**CHSA-MNDA**	**CHLNNet**	**CNN-FL**	**BCOA-EEBCDL**	**CWHO-CCAR**	**EWHO-CCEARA**
100	8.22	10.18	12.51	13.85	15.73	17.25	18.72
200	10.85	12.97	15.19	16.93	18.47	19.84	20.85
300	13.71	15.46	17.96	19.25	20.32	21.95	23.54
400	15.68	17.05	19.67	20.78	21.83	23.89	25.68
500	17.45	18.63	20.84	22.56	23.75	25.48	27.92
Number of nodes	**Packet Delivery Ratio (PDR) (%)**						
	ESMCH	**CHSA-MNDA**	**CHLNNet**	**CNN-FL**	**BCOA-EEBCDL**	**CWHO-CCAR**	**EWHO-CCEARA**
100	82.18	84.59	87.44	89.45	90.52	92.67	93.88
200	80.66	83.21	86.15	88.57	89.71	91.84	92.12
300	79.72	81.47	84.81	87.45	88.54	90.69	91.25
400	78.26	80.43	83.95	86.21	87.46	89.78	90.55
500	77.35	79.24	82.63	85.39	86.47	88.62	89.68
Number of nodes	**Packet Loss Ratio (PLR) (%)**						
	ESMCH	**CHSA-MNDA**	**CHLNNet**	**CNN-FL**	**BCOA-EEBCDL**	**CWHO-CCAR**	**EWHO-CCEARA**
100	17.82	15.41	12.56	10.55	9.48	7.33	6.12
200	19.34	16.79	13.85	11.43	10.29	8.16	7.88
300	20.28	18.53	15.19	12.55	11.46	9.31	8.75
400	21.74	19.57	16.05	13.79	12.54	10.22	9.45
500	22.65	20.76	17.37	14.61	13.53	11.38	10.32
Number of Rounds	**Average Energy Consumption (mJ)**						
	ESMCH	**CHSA-MNDA**	**CHLNNet**	**CNN-FL**	**BCOA-EEBCDL**	**CWHO-CCAR**	**EWHO-CCEARA**
500	0.362	0.335	0.294	0.265	0.234	0.205	0.185
1000	0.394	0.359	0.315	0.286	0.262	0.248	0.218
1500	0.425	0.397	0.356	0.319	0.284	0.267	0.232
2000	0.458	0.423	0.378	0.343	0.298	0.271	0.248
2500	0.487	0.456	0.404	0.366	0.321	0.305	0.276
3000	0.529	0.504	0.446	0.415	0.358	0.316	0.298

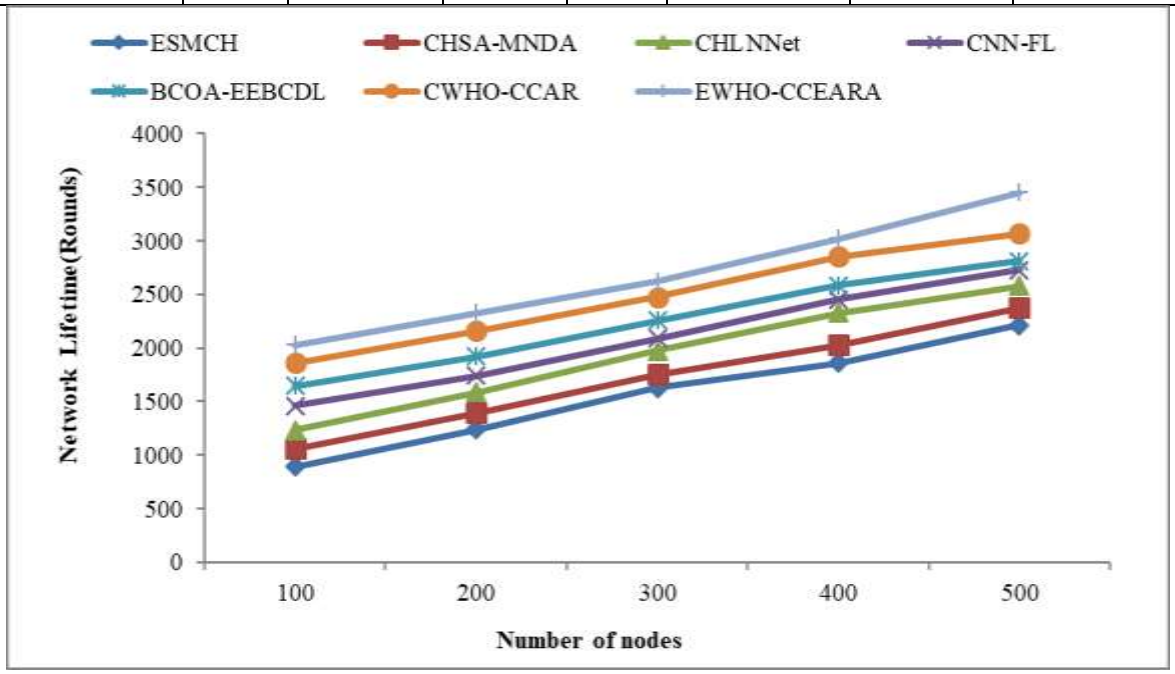

FIGURE 9. COMPARING ROUTING METHODS FOR NETWORK LIFETIME

Figure 9 shows the intended and current network lifetimes in rounds. The proposed system achieves the longest network lifetime across networks with 100 to 500 nodes. Specifically, EWHO-CCEARA reaches a maximum network lifetime of 3,450 rounds, while other approaches ESMCH, CHSA-MNDA, CHLNNet, CNN-FL, BCOA-EEBCDL, and CWHO-CCAR achieve lower lifetimes of 2,215, 2,377, 2,581, 2,726, 2,814, and 3,067 rounds, respectively, for 500 nodes (see Table 4). The optimum CH selection utilizing EWHO is responsible for the suggested system's increased network lifetime,

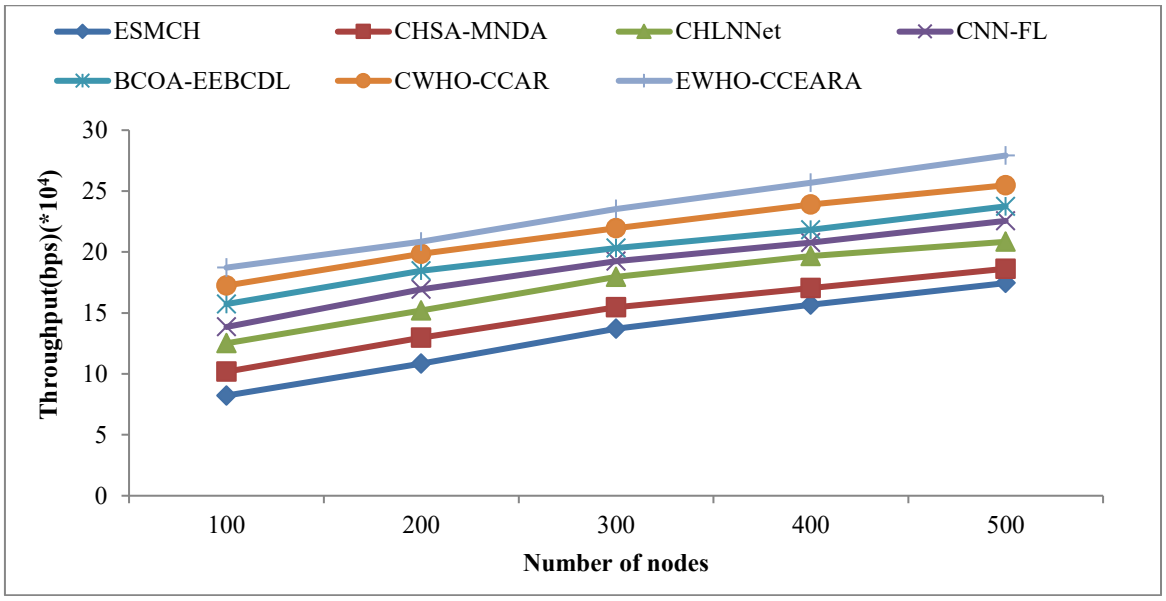

FIGURE 10. THROUGHPUT COMPARISON OF ROUTING METHODS

Figure 10 presents the throughput comparison between the proposed method and conventional approaches, including ESMCH, CHSA-MNDA, CHLNNet, CNN-FL, BCOA-EEBCDL, and CWHO-CCAR. While the other approaches record lesser results, the suggested system obtains the maximum throughput of 27.92 × 10^4bps. (see Table 4). The superior throughput of the appropriate CH selection is responsible for the suggested system using EWHO and effective congestion control through the DDQN.

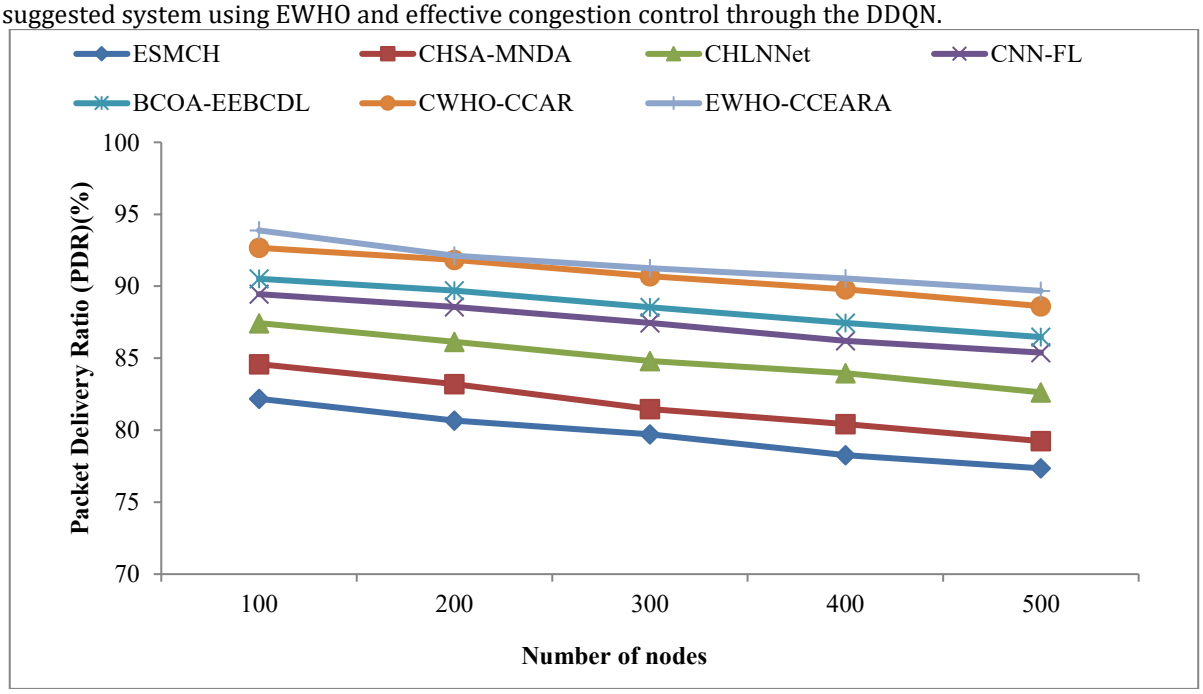

FIGURE 11. PDR ROUTING METHODS COMPARISON

Figure 11 presents the PDR results for ESMCH, CHSA-MNDA, CHLNNet, CNN-FL, BCOA-EEBCDL, and CWHO-CCAR with 100 and 500 nodes. The proposed system achieves the highest PDR of 93.88%, whereas the other methods record lower values of 82.18%, 84.59%, 87.44%, 89.45%, 90.52%, and 92.67% for 100

nodes (see Table 4). The superior PDR of the proposed system is attributed to the use of the DDQN within the queuing model, which effectively prioritizes data packets.

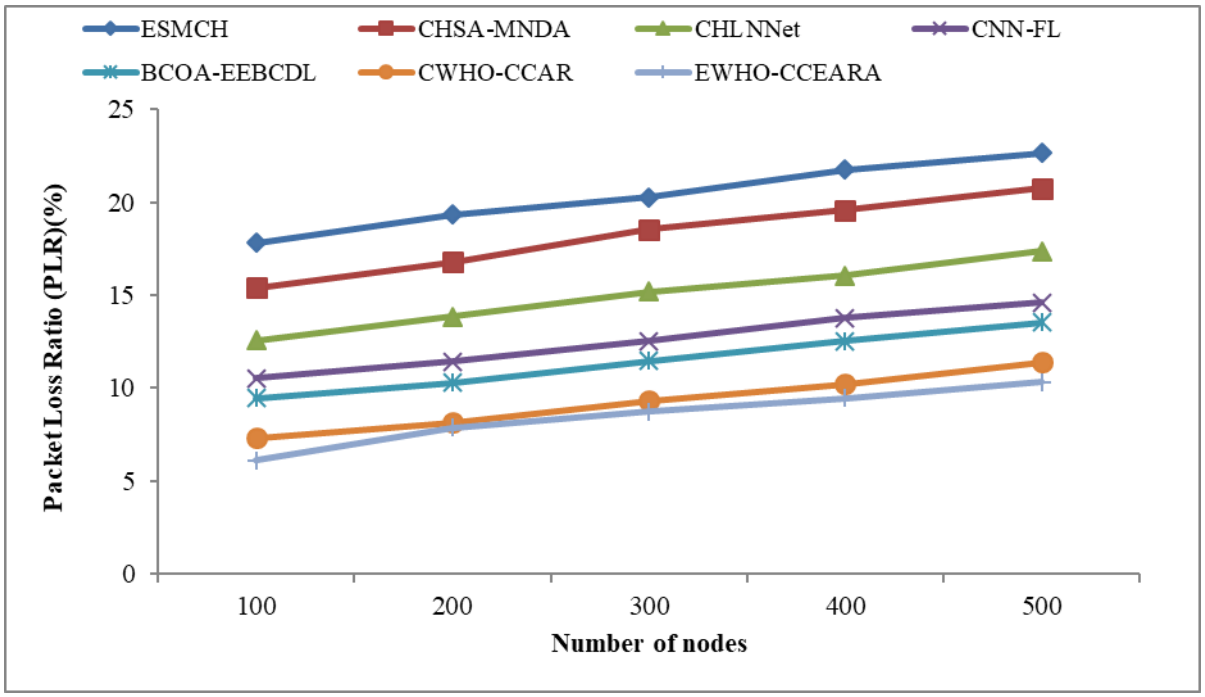

FIGURE 12. PLR COMPARISON OF ROUTING METHODS

In Figure 12, the PLR of the suggested system is contrasted with current techniques for networks with 100–500 nodes, including ESMCH, CHSA-MNDA, CHLNNet, CNN-FL, BCOA-EEBCDL, and CWHO-CCAR. The proposed system achieves the lowest PLR of 6.12%, while the other methods record higher PLR values of 17.82%, 15.41%, 12.56%, 10.55%, 9.48%, and 7.33% for 100 nodes (see Table 4). The reduced PLR in the proposed system is attributed to the use of the DDQN for prioritizing data packets and alleviating network congestion.

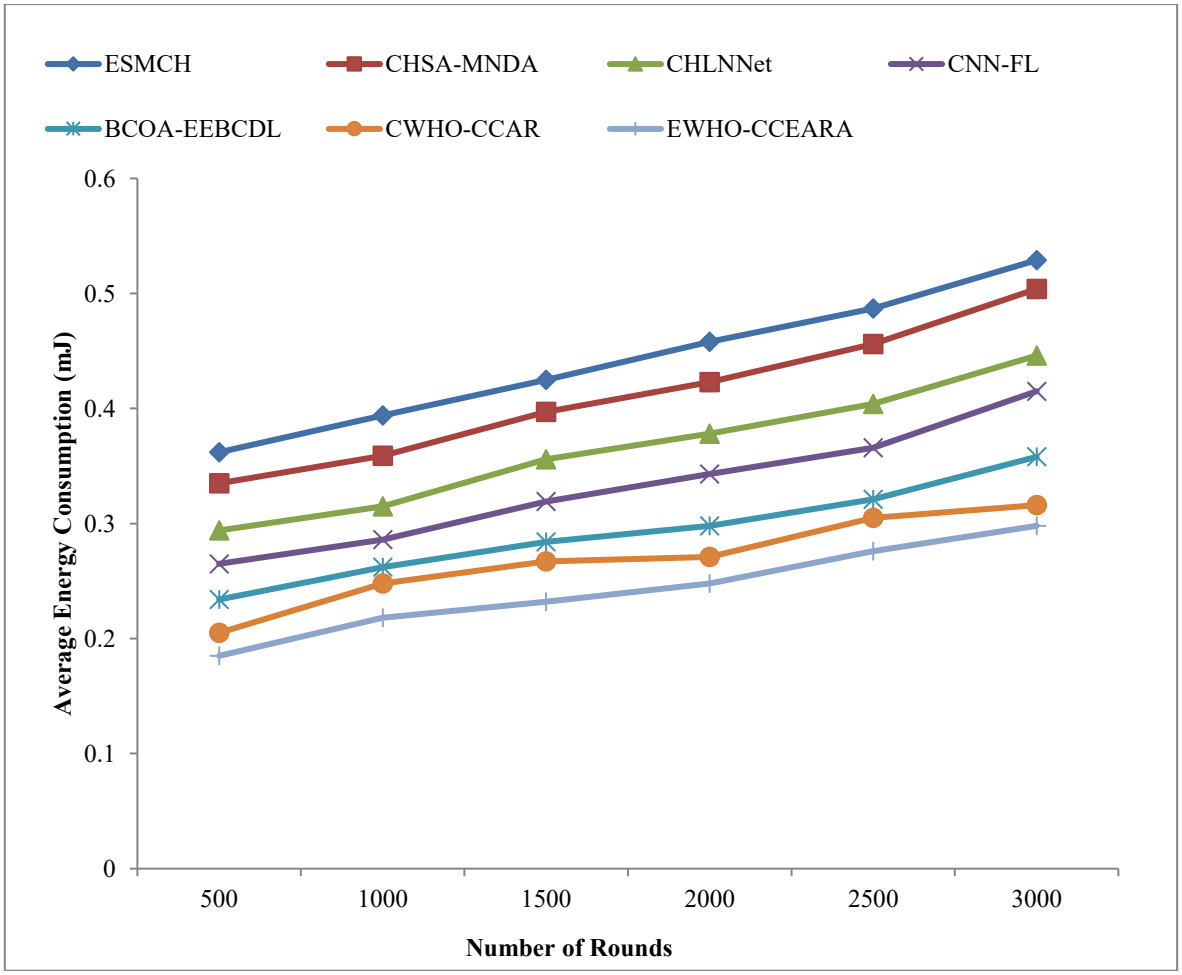

FIGURE 13. ROUTING METHODS' AVERAGE ENERGY COMPARISON

Figure 13 presents the comparison of average energy consumption for ESMCH, CHSA-MNDA, CHLNNet, CNN-FL, BCOA-EEBCDL, and CWHO-CCAR across network sizes ranging from 100 to 500 nodes. At an average energy usage of 0.298 mJ, the suggested system has the lowest value compared to the other approaches (see Table 4).

5. Conclusion And Future Work

This paper presents the CCEARA to address congestion issues in WSNs. The CCEARA protocol aims to decrease end-to-end delay and extend network lifetime by selecting optimal CHs and cluster members. For CH selection, the Enhanced Wild Horse Optimization (EWHO) model is employed, which mimics the social behaviors of wild horses, including grazing, dominance, leadership hierarchy, and mating. The limitations of WHO, including low exploitation capability and tendency to get stuck in local optima, are addressed using a logistic function. The Enhanced Wild Horse Optimization (EWHO) considers factors such as residual energy, distances between sensor nodes, the distance between CHs and the BS, node degree, and node centrality. Additionally, a DDQN is employed to manage control packet transmission, ensuring that each node updates its neighbors locally with the routing decisions. Nodes may build several queues for distinct traffic classes using a multi-queue priority policy, assigning each class a distinct level of importance within the service policy. MNs are identified using a DLEM, which integrates a (DBN, BiLSTM, and BiGRU, along with RMCV scheme. Experimental results demonstrate that the CCEARA protocol effectively meets QoS requirements by improving network lifetime, throughput, and PDR, reducing PLR, and lowering mean energy consumption. Additionally, a security analysis is conducted, demonstrating that the blockchain network is robust against various vulnerabilities. Future work includes: (1) assessing other performance indicators like calculation time and end-to-end delay, and implementing a real network environment instead of simulations to further assess the routing protocol, and (2) applying hybrid swarm-intelligence-based optimization algorithms for cluster head selection (CHS) to reduce end-to-end delay and extend network lifetime.

References

1. Huanan, Z., Suping, X. and Jiannan, W., 2021. Security and application of wireless sensor network. *Procedia Computer Science*, *183*, pp.486-492.
2. Behera, T.M., Samal, U.C., Mohapatra, S.K., Khan, M.S., Appasani, B., Bizon, N. and Thounthong, P., 2022. Energy-efficient routing protocols for wireless sensor networks: Architectures, strategies, and performance. *Electronics*, *11*(15), pp.1-26.
3. Ma, J., Wang, S., Meng, C., Ge, Y. and Du, J., 2018. Hybrid energy-efficient APTEEN protocol based on ant colony algorithm in wireless sensor network. *EURASIP Journal on Wireless Communications and Networking*, *2018*(1), pp.1-13.
4. Zheng, H., Guo, W. and Xiong, N., 2017. A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(12), pp.2315-2327.
5. Manchanda, R. and Sharma, K., 2020. Energy efficient compression sensing-based clustering framework for IoT-based heterogeneous WSN. *Telecommunication Systems*, *74*(3), pp.311-330.
6. Zhang, J., Lin, Z., Tsai, P.W. and Xu, L., 2020. Entropy-driven data aggregation method for energy-efficient wireless sensor networks. *Information Fusion*, *56*, pp.103-113.
7. Huibin, X. and Mengjia, Z., 2022. An Energy-Efficient Clustering Routing for Wireless Sensor Networks Based on Energy Consumption Optimization. *International Journal of Digital Multimedia Broadcasting*, *pp*.1-11.
8. Singh, D., Kumar, B., Singh, S. and Chand, S., 2019. SMAC-AS: MAC based secure authentication scheme for wireless sensor network. *Wireless Personal Communications*, *107*, pp.1289-1308.
9. El Khediri, S., Khan, R.U., Nasri, N. and Kachouri, A., 2020. MW-LEACH: Low energy adaptive clustering hierarchy approach for WSN. *IET Wireless Sensor Systems*, *10*(3), pp.126-129.
10. Nigam, G.K. and Dabas, C., 2021. ESO-LEACH: PSO based energy efficient clustering in LEACH. *Journal of King Saud University-Computer and Information Sciences*, *33*(8), pp.947-954.
11. Al-Ghushami, A., Karie, N. and Kebande, V., 2019. Detecting Centralized Architecture-Based Botnets using Travelling Salesperson Non-Deterministic Polynomial-Hard problem-TSP-NP Technique. In *2019 IEEE Conference on Application, Information and Network Security (AINS)*, pp.77-81.
12. Jilani, S.A., Koner, C. and Nandi, S., 2020. Security in wireless sensor networks: attacks and evasion. In *2020 national conference on emerging trends on sustainable technology and engineering applications (NCETSTEA)*, pp.1-5.
13. Priya, R.M., Suganthi, B., Rajeswari, S. and Dharani, R., 2026. The Enhanced Detection of Malicious Nodes in Wireless Sensor Networks Using Multimodal Contrastive Domain-Sharing GANs and Blockchain Distributed Storage. *International Journal of Communication Systems*, pp.1-15.
14. Yang, H., Zhang, X. and Cheng, F., 2021. A novel algorithm for improving malicious node detection effect in wireless sensor networks. *Mobile Networks and Applications*, *26*(4), pp.1564-1573.
15. Gunjal, M. and Kulkarni, P., 2025. Adaptive Deep Learning-Driven Framework for Malicious Node Detection in Wireless Sensor Networks. In *2025 Global Conference in Emerging Technology (GINOTECH)*, pp.1-7.
16. Nouman, M., Qasim, U., Nasir, H., Almasoud, A., Imran, M. and Javaid, N., 2023. Malicious node detection using machine learning and distributed data storage using blockchain in WSNs. *IEEE Access*, *11*, pp.6106-6121.
17. Zawaideh, F. and Salamah, M., 2019. An efficient weighted trust-based malicious node detection scheme for wireless sensor networks. *International Journal of Communication Systems*, *32*(3), pp.e3878-e3900.
18. Morsi, A.M., Barakat, T.M. and Nashaat, A.A., 2020. An efficient and secure malicious node detection model for wireless sensor networks. *International Journal of Computer Networks & Communications (IJCNC)*, *12*, pp.97-108.
19. Guo, Y., Tang, X. and Sun, H., 2022. A malicious node detection model for wireless sensor networks security based on CHSA-MNDA algorithm. In *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp.860-865.
20. Khashan, O.A., 2024. Blockchain-machine learning fusion for enhanced malicious node detection in wireless sensor networks. *Knowledge-Based Systems*, *304*, pp.1-16.
21. Obaid, M.A., Awad, A.S. and Khaleel, I.S., 2022. An optimal cluster head selection with trusted path routing and classification of intrusion in WSN employing CHLNNet. *Ingenierie des Systemes d'Information*, *27*(5), pp.685-693.
22. Das, R. and Dwivedi, M., 2024. Cluster head selection and malicious node detection using large-scale energy-aware trust optimization algorithm for HWSN. *Journal of Reliable Intelligent Environments*, *10*(1), pp.55-71.

23. Shanmathi, M., Sonker, A., Hussain, Z., Ashraf, M., Singh, M. and Syamala, M., 2024. Enhancing wireless sensor network security and efficiency with CNN-FL and NGO optimization. *Measurement: Sensors*, *32*, pp.1-10.
24. Kumar, M., Mukherjee, P., Verma, K., Verma, S. and Rawat, D.B., 2021. Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks. *IEEE Transactions on Network Science and Engineering*, *9*(5), pp.3272-3281.
25. Zhao, X., Zhong, W. and Navaei, Y.D., 2022. A Novel Energy-Aware Routing in Wireless Sensor Network Using Clustering Based on Combination of Multiobjective Genetic and Cuckoo Search Algorithm. *Wireless Communications and Mobile Computing*, *2022*(1), pp.1-14.
26. Zhao, X., Zhong, W. and Navaei, Y.D., 2022. A Novel Energy-Aware Routing in Wireless Sensor Network Using Clustering Based on Combination of Multiobjective Genetic and Cuckoo Search Algorithm. *Wireless Communications and Mobile Computing*, 2022(1), pp.1-14.
27. Zheng, R., Hussien, A.G., Jia, H.M., Abualigah, L., Wang, S. and Wu, D., 2022. An improved wild horse optimizer for solving optimization problems. *Mathematics*, *10*(8), pp.1-30.
28. Ahmed, U., Lin, J.C.W., Srivastava, G. and Djenouri, Y., 2021. A deep q-learning sanitization approach for privacy preserving data mining. In *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, pp.43-48.
29. Ding, Y., Ma, L., Ma, J., Suo, M., Tao, L., Cheng, Y. and Lu, C., 2019. Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. *Advanced Engineering Informatics*, *42*, pp.1-29.
30. Wang, X.Z., Zhang, T. and Wang, R., 2017. Noniterative deep learning: Incorporating restricted Boltzmann machine into multilayer random weight neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(7), pp.1299-1308.
31. Xue-juan, C., Xiang, W., Zhong-qiang, Y., Xiang, C., Yu-wu, Z. and Chun-xiang, C., 2017. Spectral characteristics and species identification of rhododendrons using a discriminative restricted Boltzmann machine. *Spectroscopy Letters*, *50*(2), pp.65-72.
32. Abduljabbar, R.L., Dia, H. and Tsai, P.W., 2021. Unidirectional and bidirectional LSTM models for short-term traffic prediction. *Journal of Advanced Transportation*, *2021*(1), pp.1-16.
33. Kılıçarslan, S., Aydın, H.A., Adem, K. and Yılmaz, E.K., 2024. Impact of optimizers functions on detection of Melanoma using transfer learning architectures. *Multimedia Tools and Applications*, pp.1-21.
34. Deng, X., Li, J., Ma, C., Wei, K., Shi, L., Ding, M., Chen, W. and Poor, H.V., 2022. Blockchain assisted federated learning over wireless channels: Dynamic resource allocation and client scheduling. *IEEE Transactions on Wireless Communications*, *22*(5), pp.3537-3553.