Journal of Rare Cardiovascular Diseases
ISSN: 2299-3711 (Print) | e-ISSN: 2300-5505 (Online)
www.jrcd.eu
RESEARCH ARTICLE

Congestion Clustering Aware Routing (CCAR) Protocol and Blockchain-Based Deep-Learning for Quality Routing in Wireless Sensor Network (WSN)

Mrs. C. Saranya[1] and Dr. L. Sudha[2]
[1]Research Scholar, Department of Computer Science, VET Institute of Arts and Science (Co education) collegeErode, TamilNadu, India.
[2]Associate Professor, Department of Computer Science, VET Institute of Arts and Science (Co-education) college, Erode, Tamil Nadu, India.

*Corresponding Author
Mrs. C. Saranya

Article History
Received: 21/09/2025
Revised: 30/09/2025
Accepted: 27/10/2025
Published: 24/11/2025

Abstract: Wireless Sensor Network (WSN) have become an important and promising technology owing to their wide range of applications in disaster response, battle field surveillance, wildfire monitoring, radioactivity monitoring, etc. Congestion control is important issue to the routing algorithms of WSN. It's crucial for maintaining network performance, extending network lifetime, and ensuring reliable data delivery. In this paper, Congestion Clustering Aware Routing (CCAR) protocol is introduced to effectively detect attacker nodes and reduce congestion via optimal routing through clustering. Dynamic Weight Deep Q-Network (DWDQN) model is introduced with a node queue to detect the level of load on the node. DWDQN model carries a threshold value to keep the arriving data packets on the basis of priority queues. The algorithm takes alleviating congestion as major issue which considers the traffic of the node itself and local network traffic. Chaotic Wild Horse Optimization (CWHO) is introduced for Cluster Head Selection (CHS) based on the residual energy, distance among sensor nodes, CH and Base Station (BS) distance, Node degree, Node centrality, traffic load, and weight time. Furthermore, to overcome the single point of failure issue, a decentralized blockchain is deployed on CHs and BS. Additionally, Malicious Nodes (MNs) are removed from the network using real-time message content validation (RMCV) and Deep Learning (DL) techniques like Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU) are trained and tested on the conventional protocol. Simulation results show that proposed protocol can prolong the network lifetime, quality of service (QoS) metrics.

Keywords: Wireless Sensor Networks (WSN), Congestion Control, Deep Q-Network (DQN), Cluster Head Selection (CHS), Blockchain-based Security

INTRODUCTION

Wireless Sensor Network (WSN) have become one of the developing research field, as they are envisioned to have wide applications with different phenomenon related to environmental tracking, emergency response, security monitoring in manned or unmanned missions [1-2]. A WSN is composed of huge, low power intelligent sensors with high power sink, which are responsible for establishing paths among themselves with certain transmission regulations [3]. Wireless sensors are more advantageous due to their simple installation, self-identification, self-diagnosis and time awareness for coordination with other sensors to form dynamic self-organized networks. Due to this diversity of sensor nodes, the applications of WSNs are huge in a range that starts with healthcare, military, defense, agriculture to our day to day life. Despite huge applications, WSN faces many typical challenges like limited energy sources, computational speed, memory, and limited communication bandwidth, making the sensor network degrade in performance and decreasing the network lifetime [3].

WSNs are of two types like homogenous and heterogeneous. In a homogeneous network, nodes are identical. Computational heterogeneity describes those networks where nodes differ in microprocessor's power and storage capacity. Hence, the powerful nodes can perform complex data processing and long-term storage. When nodes differ with bandwidth, the networks come under the link heterogeneity category and are suitable for reliable data transmission. Both link and computational heterogeneity consume a considerable amount of energy, reducing the network lifetime. Developing different algorithms for different applications is quite a challenging task. In particular, the designer of WSNs must emphasize on various issues like data aggregation, clustering, routing, localization, fault detection, task scheduling, event tracking, etc.

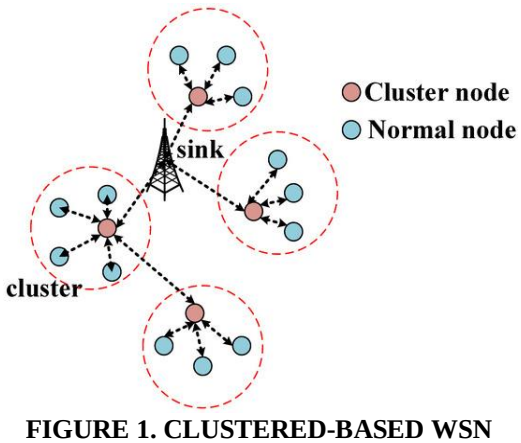

FIGURE 1. CLUSTERED-BASED WSN

Clustering is an energy-efficient method for the hierarchical organization of sensor nodes in a network [4,5]. However, each node in an ad hoc network communicating directly with the sink node leads to problems such as data collision, network congestion, and unnecessary drainage of power [6]. Low Energy Adaptive Clustering Hierarchy (LEACH) is a classical cluster-based protocol proposed to minimize energy consumption by efficiently selecting cluster heads [7]. Forming small clusters within the network helps overcome these crucial issues through efficient resource utilization. For each cluster, a cluster head (CH) is elected to act as a hop between the sensing nodes and the sink (as shown in Figure 1), thereby reducing the transmission distance. The CHs are elected dynamically after a certain interval to reduce the overhead.

The clustered network helps the system to maintain a longer life term by scheduling a duty cycle between nodes in a cluster without affecting the normal functionalities of the network [7]. The CH sets a time division multiple access (TDMA) schedule for data transmission to prevent any collision of messages. The non-CH node sends its data to the respective CHs with the Direct Sequence Spread Spectrum (DSSS) communication, in which each cluster has its unique propagation code to reduce interference. The CHs add received data and send it to BS through a fixed propagation code with Carrier-Sense Multiple Access (CSMA). The sensor nodes send the data towards destination without any human involvement in the WSNs, after sensing it from the surrounding environment. Traditional clustering methods in terms of scalability, reliability, fault tolerance, amount of data delivered, energy consumption, better coverage of the experimental field, and the increase of the network lifetime becomes very difficult task. The clustering on WSN remains a Non-deterministic Polynomial (NP) hard optimization problem which cannot be solved efficiently by traditional approaches [7-8]. Thus, a more accurate resolution of the NP optimization of the clustering is made possible by using approaches based on recent research on Computational Intelligence (CI) and Machine Learning (ML) [9-10].

However, the data traffic of WSNs involves significant upstream traffic. The data traffic is closely related to the WSNs application and has the characteristics of periodicity or continuity [11]. A large amount of upstream data traffic can overload nodes and exceed their processing power, resulting in increased latency, packet loss, and retransmission. The deterioration of underlying network performance adversely affects the reliability of monitored applications. In addition to these factors, resources are constrained in WSNs, such as processing power, available memory, and battery power. Resource constraints, node overload, error-prone communication links, and irregular upstream traffic can ultimately lead to network congestion. Congestion significantly reduces network performance and results in increased packet loss. Heterogeneous nature of WSN makes them prone to various attacks [12]. Blockchain is a promising technology which addresses the data security and third party involvement issues. The blockchain is not only used as a cryptocurrency, but it is also used in various other fields like healthcare [13], manufacturing, smart cities, etc., [13-14]. It provides a distributed ledger in which records are added through consensus mechanism after performing the validation process.

In this paper, Congestion Clustering Aware Routing (CCAR) protocol is introduced to effectively detect attacker nodes and reduce congestion via optimal routing through clustering. Dynamic Weight Deep Q-Network (DWDQN) model is introduced which considers the traffic of the node itself and local network traffic. Chaotic Wild Horse Optimization (CWHO) is introduced for Cluster Head Selection (CHS) based on the residual energy, distance among sensor nodes, CH and BS Distance, Node degree, Node centrality, traffic load, and weight time. Additionally, Malicious Nodes (MNs) are removed from the network using real-time message content validation (RMCV) and DL techniques like CNN, Bidirectional Long Short-Term Memory (BiLSTM), and Bidirectional Gated recurrent unit (BiGRU) are trained and tested on the conventional protocol.

LITERATURE REVIEW

Pravin et al., [15] proposed a Congestion Aware Clustering with Improved Ant Colony Algorithm (CAC-IACA). This mechanism involves two steps (i) identifying the best route by following the Ant Colony Optimization (ACO) algorithm and (ii) data segmentation using rendezvous mobile nodes. The Rendezvous nodes are present in each cluster to reduce the congestion rate on receiver side during data transmission. This proposed methodology mainly concentrates on reducing coverage cost for 3 Dimensional (3D) environmental monitoring. Simulation results are analyzed and the efficiency of the proposed scheme proves better than the conventional method in terms of network lifetime, energy consumption, control overhead and transmission delay.

Farsi et al., [16] introduced a Congestion-aware Clustering and Routing (CCR) protocol to alleviate the congestion issue over the network. The CCR protocol is proposed to decrease end-to-end delay time and prolong the network lifetime through choosing the suitable Primary Cluster Head (PCH) and the Secondary Cluster Head (SCH). The experimental results demonstrate that the effectiveness of the CCR protocol to satisfy the Quality of Service (QoS) requirements in increasing the network lifetime and raising the number of packets sent alike. Moreover, the CCR outperforms other state-of-the-art techniques in decreasing the overflow of data, and thus the network bandwidth usage is reduced than other methods.

Patil et al., [17] presented an Optimum Cluster-Based Congestion aware Multipath routing Protocol (OCC-MP). Improved Atom Search Optimization (IASO) approach is introduced for efficient clustering. Then the HSIPO method is used for decision making, which computes each node's trust degree. The Hybrid Swarm Intelligent Pyramid Optimization (HSIPO) algorithm selects the CH from a group of nodes. Then a Deep Recurrent Neural Network (DRNN) is used to monitor congestion and provide congestion aware routing. Finally, multiple simulation situations like various node densities and at various simulation rounds supported in the proposed OCC-MP technique outperformed current methodologies with respect to energy consumption, throughput, traffic load overflow, delivery ratio, and number of nodes alive.

Patil et al., [18] proposed a novel congestion control system to diminish the congestion on network and to enhance the throughput of the network. Initially, Cluster Head (CH) selection is achieved by exhausting K-means clustering algorithm. After the selection of cluster head, an efficient approach for congestion management is designed to select adaptive path by using Adaptive Packet Rate Reduction (APTR) algorithm. Finally, ACO is utilized for enhancement of wireless sensor network throughput. The objective function increases the wireless sensor network throughput by decreasing the congestion on network. The proposed system is simulated with Network Simulator (NS-2). The proposed K-means C-ACO-ICC method attains higher throughput lower delay minimum congestion level. Finally, the simulation consequences demonstrate that proposed system may be capable of minimizing that congestion level and improving the throughput of the network.

Maheshwari et al., [19] employed Butterfly Optimization Algorithm (BOA) to choose an optimal CH from a group of nodes. The CH selection is optimized by the residual energy of the nodes, distance to the neighbors, distance to the base station, node degree and node centrality. The route between the CH and the base station is identified by using ACO, it selects the optimal route based on the distance, residual energy and node degree. The performance measures of this proposed methodology are analyzed in terms of alive nodes, dead nodes, energy consumption and data packets received by the BS. The outputs of the proposed methodology are compared with traditional approaches and compared with some existing methods.

Moussa and Belrhiti Alaoui[20] proposed an algorithm named Energy-efficient Cluster-based Routing Protocol using Unequal Clustering Algorithm (ECRP-UCA) and improved ACO techniques. ECRP-UCA divides the network into unequal clusters based on residual energy, distance to the sink, number of neighbor nodes, and a new parameter named number of backward relay nodes in previous round to properly balance the load among CH. Batch-based clustering method is introduced that allows the network to function several rounds without requiring control overhead for its configuration. Additionally, devised an improved ACO based routing technique for efficient and reliable inter-cluster routing from CHs to the sink. The proposed routing protocol is intensively experimented and compared with recent and relevant existing protocols. The simulation results show that the proposed ECRP-UCA outperforms these protocols in terms of various interesting metrics.

Reddy et al., [21] designed an energy-efficient routing protocol to find the optimal CH in WSN. As the main contribution deals with the Cluster Head Selection (CHS), ACO Integrated Glowworm Swarm Optimization (ACI-GSO) approach which is the hybridization of GSO and ACO algorithms. The objective of the CHS is to reduce the distance among the selected CH node. It makes the fitness function using multiple objectives like distance, delay, and energy. Finally, the performance of the proposed work is evaluated and the efficiency of the proposed work is proved over other conventional works in terms of alive nodes, dead nodes, energy consumption and data packets received by the BS.

Deepa and Suguna[22] proposed an Optimized Quality of Service-based Clustering with Multipath Routing Protocol (OQoS-CMRP) for WSNs. It reduces the energy consumption in sink coverage area by applying the Modified Particle Swarm Optimization(MPSO)-based clustering algorithm to form clusters to select CH in the sink coverage area which solves energy hole problem. The SingleSink-AllDestination algorithm is used to find near optimal multi-hop communication path from sink to sensors for selecting the next hop neighbor nodes. The Round-robinPathsSelection algorithm is used for transferring data to sink. According to QoS metrics, the performance of the proposed communication protocol is evaluated and compared with other existing protocols. OQoS-CMRP achieves prominent data communication with reasonable energy conservation. It also reduces transmission delay and communication overhead on the basis of ensuring the outcome of the entire network.

Amjad et al., [23] proposed a blockchain-based node authentication model for the Internet of sensor things (IoST). Distance, Degree, and Residual energy-based low-energy adaptive clustering hierarchy (DDR-LEACH) protocol is used to replace CHs with the ordinary nodes based on maximum residual energy, degree, and minimum distance from BS. Furthermore, storing a huge amount of data in the blockchain is very costly. External data storage, named as Interplanetary File System (IPFS) is used for ensuring data security, which performs better than the existing encryption schemes. Moreover, a huge computational cost is required using a proof of work consensus mechanism to validate transactions. To solve this issue, proof of authority (PoA) consensus mechanism is used in the

proposed model. DDR-LEACH outperforms LEACH in terms of energy consumption, throughput, and improvement in network lifetime with CH selection mechanism.

Khan et al., [24] proposed a routing protocol based on Energy Temperature Degree-Low Energy-Adaptive Clustering Hierarchy (ETD-LEACH). In the protocol, nodes consume less energy when transmitting data, which improves the network lifetime. The proposed protocol selects the CHs on the bases of degree, temperature, and energy to perform routing. Moreover, for solving the issue of a single point of failure, the blockchain is utilized. The data transactions are also housed in the blockchain, which is deployed on the CHs and Base Station (BS) in blockchain, multiple nodes take part. Therefore, to perform a consensus between them, a PoA consensus mechanism is used in the underlying work. In the blockchain, the secure hashing algorithm-256 (SHA-256) is used for secure hashing of data transactions. Furthermore, malicious nodes are detected during the routing using the Real-Time Message Content Validation (RMCV) scheme in the ETD-LEACH protocol. The proposed model is evaluated under the Denial-of-Service (DoS) attack, the Man-In-The-Middle (MITM) attack, and the smart contract analysis performed by the Oyente tool. The performance of the proposed model is evaluated through simulations. ETH-LEACH protocols are compared using different parameters like number of alive nodes, energy consumption, throughput, and delay.

Jibreel et al., [25] proposed a Heterogeneous Gateway-based Energy-Aware multi-hop routing protocol (HMGEAR). The proposed routing scheme is based on the introduction of heterogeneous nodes in the existing scheme, selection of the head based on the residual energy, introduction of multi-hop communication strategy in all the regions of the network, and implementation of energy hole elimination technique. All these strategies are aiming at reducing energy consumption and extend the life of the network. Results show that the proposed routing scheme outperforms two existing ones in terms of stability period, throughputs, residual energy, and the lifetime of the network. Mehta and Saxena[26] presented a Multi-Objective Based Clustering and SailFish Optimizer (SFO) guided routing method to sustain energy efficiency in WSNs. CH is selected, based on effective fitness function which is formulated from multiple objectives. It helps to minimize energy consumption and reduces number of dead sensor nodes. After CH selection, SFO is used to select an optimal path to sink node for data transmission. The proposed approach is analytically analyzed and results are compared with the similar existing approaches in terms of energy consumption, throughput, packet delivery ratio, and network lifetime. The simulation results show that proposed method has performed better in terms of energy consumption and number of alive sensor nodes respectively when compared to existing methods. Further, it shows significantly better results than other optimization-based approaches.

PROPOSED METHODOLOGY

In this paper, Congestion Clustering Aware Routing (CCAR) protocol is introduced to effectively detect attacker nodes and reduce congestion via optimal routing. DWDQN model is introduced to detect the level of load on the node, and data packets congestion is detected based on the priority queues. Chaotic Wild Horse Optimization (CWHO) is introduced for Cluster Head Selection (CHS) based on the fitness function. Malicious Nodes (MNs) are removed and detected using DL techniques. Routing protocols results are measured with QoS metrics. Figure 2 shows the overall process of flow diagram.

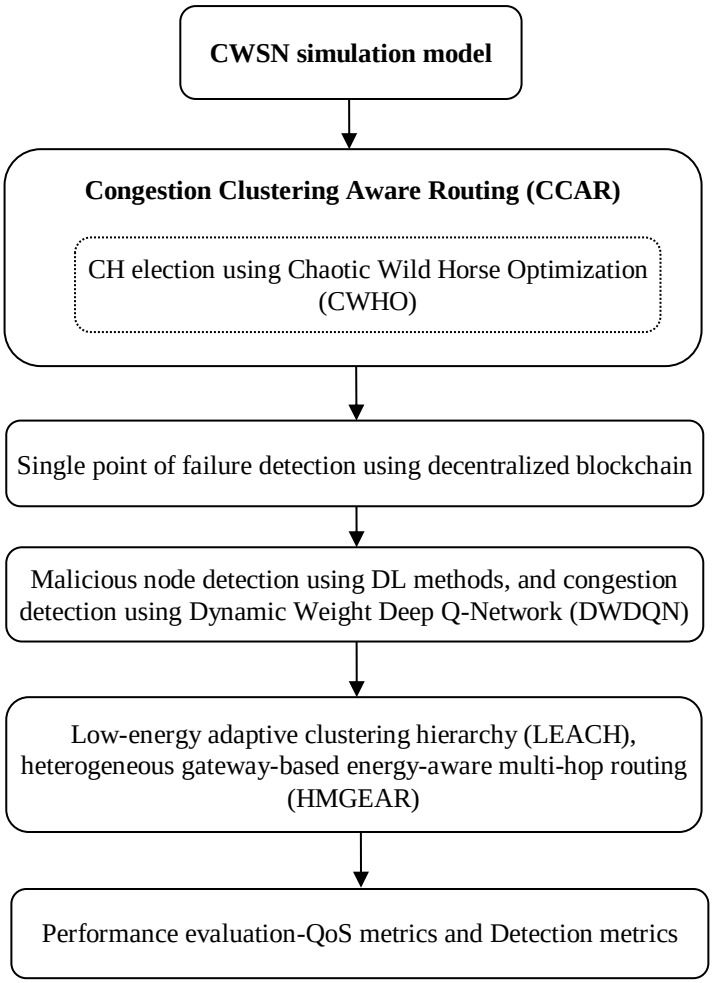

FIGURE 2. OVERALL FLOW DIAGRAM OF PROPOSED SYSTEM

1.1. Network Model

Network model, CWSN model is used in which N sensor nodes are evenly arranged in a circular area of diameter M. The BS in the center of the network area has strong computing power. Because the BS energy can be self-replenished, the energy loss of the BS is not considered in this work. It comprising sensor nodes, CHs and BS.

1.2. Energy Consumption Model

According to the actual transmission distance from the CHs to the BS, the free space model and the multipath fading channel model both need to be analyzed by considering only the multipath fading channel model [27]. Therefore, the expression of the total energy consumption of the model will undergo some changes. $E_T(e, d)$ indicates the energy consumed by the wireless transmitter to transmit a set of e bits of information. The expression is as follows:

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

$E_R(e)$ indicates the energy required to receive the information of the e bit. The expression is as follows:

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

In Equations (1) and (2), E_{elec} is the energy consumed per bit by the transmitter or receiving circuit and d is the distance between the transmitter and the receiver. In Equation (2), when $d < d_0$, use the free space model and ε_{fs} acts as the energy factor per bit. Otherwise, the multipath fading channel model is used, and ε_{mp} acts as the energy factor per bit. In addition, d_0 is used as the distance threshold. As long as it is input as an independent variable into the free space model and the multipath fading channel model to establish an equation (3) the following expression can be obtained:

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

Therefore, the calculation formula for the energy consumed by transmitting e bit information is defined as follows:

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

In Equation (4), $d^2_{\text{to CH}}$ represents the distance from the cluster member node to the CH. The CH receives information from the cluster member nodes in the cluster, and then fuses the information that which it senses from the environment, eventually transmits the merged information to the BS, information size obtained by the CH is e bit. The energy consumed in the process is calculated as follows:

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

In Equation (5), n is the number of nodes surviving in the monitored area, k is the number of clusters to be divided, E_{DA} is the energy consumed by the CH to process each bit of data, and $d_{\text{to BS}}$ is the distance between the CH and the BS.

1.3. Clustering model

In general, the inter-cluster communication traffic in WSN increases as the number of clusters increases, and the intra-cluster communication traffic increases as the number of clusters decreases. The determination of the optimal cluster number of the network is of great significance to the network's communication. Determine the optimal number of clusters k in combination with the network structure model and energy consumption model. The main steps of the clustering protocol are described as follows [27]:

(1) Calculate the number k of clusters required according to the calculation formula of the network optimal cluster number.
(2) Through the AGglomerative NESting (AGNES) algorithm with balanced energy consumption optimization, can build the required k clusters.
(3) Implement the selection mechanism of the CH in each cluster, and then can implement the BCOA division of the energy balance strategy and node selection mechanism for the large cluster area before and after the death of the first node, respectively.
(4) Data transmission and energy update.

To minimize the total energy consumption and balance the energy consumption of the nodes in the network, perform the node death decision after each round of data transmission in the network (once the node dies, return to Step 1; otherwise, return to Step 3). In addition, Steps 1, 2, and 3 are collectively called the preparation phase of the protocol. Step 4 is called the stabilization phase of the protocol.

The distances between some CHs and BS in the model may be larger than d_0, so it is necessary to simultaneously refer to the free space model and the multipath fading channel model when considering the energy consumption between clusters. The energy consumed by all of the clusters in the region in one round is follows [27]:

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

AGNES algorithm is a hierarchical clustering algorithm [27]. First, several objects are input, each one constitutes an initial cluster by itself. Then the two clusters with the shortest distance are continuously merged into one cluster until the number of clusters obtained reaches the number of clusters k satisfying the termination condition. Finally, the resulting k clusters are the target clusters of algorithm. In this algorithm, each cluster equals a sample set, and the merger between clusters equals the merger between sets. The merging standard is the distance between the two clusters, which usually assumes three forms: (1) the longest distance, (2) the shortest distance, and (3) the average distance. By adding the two clusters average distance, and the variance of the distance set of two clusters in the cluster setup process, can construct a cluster setup factor. The two clusters corresponding to the largest cluster-merging factor can be merged until the number of clusters reaches the preset number of clusters k [27].

1.3.1. Fitness function

From the sensors group in the network, the optimal CH is selected by the CWHO fitness function. The residual energy is considered in the fitness function to evade a dead node as a CH. Higher centrality is used to reduce the distance of transmission among the members. The mathematical forms of fitness functions and their definitions are discussed below [27]:

CH Residual energy: CH collects data from ordinary sensor nodes and transmits it to BS in a network. Because the CH takes more energy to perform the preceding activities, the node with the most residual energy is the strongest choice to be a CH. The following Equation (7) describes residual energy (f_1) [27]:

$$f_1 = \sum_{i=1}^{m} \frac{1}{E_{CH_i}} \quad (7)$$

where the i^{th} CH residual energy is E_{CH_i}.

Distance among sensor nodes: It specifies the range among usual sensor nodes as well as its CH. The dissipation of energy for the node mostly depends on the transmission path distance. If the chosen node has a minimum transmitting distance near BS, then the node's energy consumption is small. Sensors to CH (f_2) the distance is displayed in Equation (8) [27]:

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

where $D\left(s_i, \frac{CH_j}{I_j}\right)$ denoted the sensor i and CH_j distance and I_j is the sensor node's quantity belongs to CH [27].

CH and BS Distance: The energy consumption of the node is evaluated based on the distance via the transmitting track. When BS is situated away from CH, for example, data transmission will require a lot of energy. As a result, the abrupt drop in CH could be related to increased energy use. Therefore, throughout information transfer, the node that is closest to BS is selected. The optimal solution of distance among the BS (f_3) and the CH is represented by the following Equation (9) [27]:

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

where $D(CH_j, BS)$is the distance between BS and CH_j.

Node degree: It specifies how many sensor nodes each CH. Because CHs with more cluster members lose their energy for a shorter period, the CHs with fewer sensors are chosen. Equation (10) expresses the degree of node (f_4) [27]:

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

where $I_i$is the number of CHi sensor nodes.

Node centrality: Node centrality (f_5) is an expression that expresses how far a node is from its neighbors, represented in Equation (11) [27]:

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

where n(i) is the number of CH_i neighboring nodes and L is the network dimension. Every objective value is assigned a weight value. Several objectives are combined into a single function in this situation.

Traffic Load: To reduce energy overheads, the traffic load for nodes with a trust value is higher than the value of the trust threshold is identified. Further, to detect network congestion, 'CT_{Min}' and 'CT_{Max}' thresholds that fall within the range of buffer length in the queue are set. Depending on the fixed threshold values, network congestion can be classified into three types, as follows,

Less: If the queue length falls below the threshold CT_{Min}, the network is considered as having congestion.
Medium: If the queue length is between $CT_{Min}$and CT_{Max}, the network is considered as having medium congestion.
High: If the queue length is higher than the threshold CT_{Max}, then the congestion level is considered as being high.

Let T_s^k represent the buffer length of the 'k[th]' node with source node s represented as follows,

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

Distance: The distances between the sender, the potential next node, and the sink are computed. Let d_1 represent the distance between the recent source and the probable next nodes and let d_2 denote the distance between the next node and the sink. The matching distance from the sender d_1^M and the matching distance from sink d_2^M are considered.

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

Hence, the distance of the potential next node associated with matching distances $d_1^M$and d_2^Mcan be determined as follows,

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

where α and β are the adjustment parameters for $d_1^M$and d_2^M.

Waiting Time of the Nodes: The waiting time at each node in the cluster is computed to find out whether a node in the cluster can be selected as CH or not.

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

RE is denoted as the RE of a node, IE is denoted as the Initial energy (IE) of a node, $WT_{max}$is denoted as the maximum pre-defined time to wait, $Avg\left(S_i^k\right)$ is denoted as the average speed of adjacent nodes within the range of transmission, and $S_n$signifies the speed of the node. S1, S2, S3, S4, S5, S6, and S7 are the weighted values. Equation (16) depicts the single objective function [27]:

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

where, $\sum_{r=1}^{7} Sr = 1\ Sr \in (0, 1)$, the values of Sr are 0.25, 0.20, 0.15, 0.15, 0.15, and 0.15 correspondingly [27]. The overall fitness of the energy level recommends that nodes with increased distances include a higher value of overall fitness. In this scenario, those nodes with a higher value of overall fitness have the lowest possibility of becoming CHs. Conversely, the nodes that have a large number of neighbors and the nodes with increased energy will have a lower value of overall fitness. The nodes with a lower value of overall fitness will have a greater possibility of becoming CHs. Further, they aggregate the data to reduce redundancy. The aggregation of data conserves energy within the network since unnecessary data are not transmitted to the BS (sink). Once data are collected and a CH is chosen, every node with a specific energy level sets its overall fitness to a particular value. A node with a lower overall fitness value involves less energy consumption and has the possibility of becoming a CH. The chosen CH forwards data packets to other nodes within its radius.

1.3.2. CH Selection Using Chaotic Wild Horse Optimization (CWHO)

Chaotic Wild horse optimizer (CWHO) is a metaheuristic algorithm that simulates the social behavior of wild horses in nature for optimal selection of CH in the WSN model. WHO simulates horse behavior in which they leave their group and join another group before becoming adults to prevent mating between siblings or daughters for optimal selection of CH [30]. Both stallions and mares live together and interact with each other in grazing for optimal CH selection. Foals leave their groups after they increase and join other groups to establish their own families for optimal CH selection. WHO shows competitive performance compared to some algorithms, it suffers from low exploitation capability and stagnation in local optima. It has been solved by using chaotic operator. CWHO algorithm is a metaheuristic swarm-based algorithm inspired by the social behavior of horses, such as grazing, domination, leadership hierarchy, and mating [28].

Sensor nodes parameter and random initialization: Initially, the node's global population, including Np packets, each with Ns nodes are initialized. If N individuals and G groups exist, then the number of non-leaders (mares and foals) is N–G, and the number of leaders is G (optimal number of CH in the Ns nodes). The proportion of stallions is defined as PS, which is G/N [28].

Grazing Behavior: In order to simulate the grazing phase for optimal CH selection, assume that the stallion position existed in the grazing area center. The equation (17) is used to enable other individuals to move [28],

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

where $X_{G,j}^{i}$ and $St_{G,j}$ are the positions of the ith group member and stallion in the jth group for optimal CH selection, R is a random number between −2 and 2, and Z is an adaptive parameter computed by Equation (18) [28],

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

where P is a vector containing 0 and 1, and its dimension equals the dimension of the CH election problem, $\overrightarrow{R_1}$ and $\overrightarrow{R_3}$ are random vectors between 0 and 1, and $R_2$is a random number between 0 and 1. TR is a linearly decreasing parameter computed by Equation (19) [28],

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

where t and T are the current and maximum iterations, respectively.

Horse Mating Behavior: As stated previously, one of the unique behaviors of horses compared to other animals is separating foals from their original groups prior to their reaching puberty and mating for optimal CH selection. To be able to simulate the behavior of mating between horses with the equation (20) [28],

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

$$Crossover = Mean$$

where $X_{G,k}^{p}$ is the position of horse p in group k, which is formed by positions of horse q in group i and horse z in group j. In the basic WHO, the probability of crossover is set to a constant named PC.

Group Leadership: Group leaders (stallions(St) will lead other group members to a suitable area (waterhole). Group leaders (stallions) will also compete for the waterhole, leading the dominant group to employ the waterhole first for optimal CH selection. Equation (21) is used to simulate this behavior [28],

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

where $\overline{St}_{G,j}$ and $St_{G,j}$ are the candidate position and the current leader position in the j[th] group, respectively, and WH is the position of the waterhole.

Exchange and Selection of Leaders: At first, leaders are selected randomly as CH. After that, leaders (best CH) are selected based on their fitness values. To simulate the exchange between leader positions and other individuals, the equation (22) is used as follows [28],

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

where $f\left(X^i_{G,j}\right)$ and $f\left(St_{G,j}\right)$are the fitness values of foal (Fo) and stallion, respectively. In the equation (18), random vectors are replaced by Logistic chaotic map (ρ) to avoid returning to same grazing,

$$P = \overrightarrow{\rho_1} < TR, IDX = (P == 0), Z = \rho_2 \Theta IDX + \overrightarrow{\rho_3}\Theta(\sim IDX) \quad (23)$$

$$P_{q+1} = lP_q\left(1 - P_q\right), l = 4 \quad (24)$$

The algorithm 1 shows the CWHO with optimal CH selection in WSN model.

Algorithm 1. Pseudocode of CWHO

Start

1. Input CWHO parameters: $PC = 0.13, PS = 0.2$
2. Set population size (N) and the maximum number of iterations (T).
3. Initialize the population of horses at random.
4. Create foal groups and select stallions.
5. **While ($t \leq T$)**
6. Calculate TR using Equation (19).
7. Calculate Z using Equation (20).
8. **For the number of stallions**
9. **For the number of foals**
10. **If** rand > PC
11. Update the position of the foal using Equation (17).
12. **Else if**
13. Update the position of the foal using Equation (20).
14. **End if**
15. **End for**
16. **If** rand > 0.5
17. Generate the candidate position of stallion using Equation (21).
18. **Else if**
19. Generate the candidate position of stallion using Equation (22).
20. **End if**
21. **If** the candidate position of the stallion is better
22. Replace the position of the stallion using the candidate position.
23. **End if**
24. **End for**
25. Exchange foals and stallions position using Equation (22).
26. t = t + 1
27. **End While**
28. Output the best solution obtained by CWHO.

End

1.4. Priority-Based Congestion control using Dynamic Weight Deep Q-Network (DWDQN)

Priority-Based Congestion control includes application, routing, queuing, and neighboring node modules. The application module classifies the incoming data packets into control and data packets, respectively. Data traffic is determined based on QoS demands. The packets can be classified as (i) reliable, (ii) normal, (iii) delayed, and (iv) critical. Control packets are transmitted to neighboring nodes such that every node locally updates the neighbors with the routing decision. If a packet of data is broadcast, this application module enables the following routing module to establish a route based on

packet type. The queuing module is responsible for allocating priority to every packet before sending the packet for channel access. The neighboring node module observes the forwarded packet to share details of node quality and node state among its neighbors. The routing module is responsible for the selection of optimal routes based on the attributes of neighbors that represent node quality. The selection of nodes and CH will be based on the clustering technique and fitness function. The queuing module is responsible for categorizing packet priority into two types: (i) Request Traffic (RET), and (ii) Non Request Traffic (NRET) queues. Delay-sensitive packets are assigned high priority and are put into the RET queue. The remaining classes of data packets are put into the NRET buffer. The attempt to categorize incoming packets into real and NRET buffers produces reduced latency for data packets within a pre-defined time. Further, the Dynamic Weight Deep Q-Network (DWDQN) is used by queuing model for assigning priority to data packets. The application module utilizes a traffic classification as presented in Figure 3.

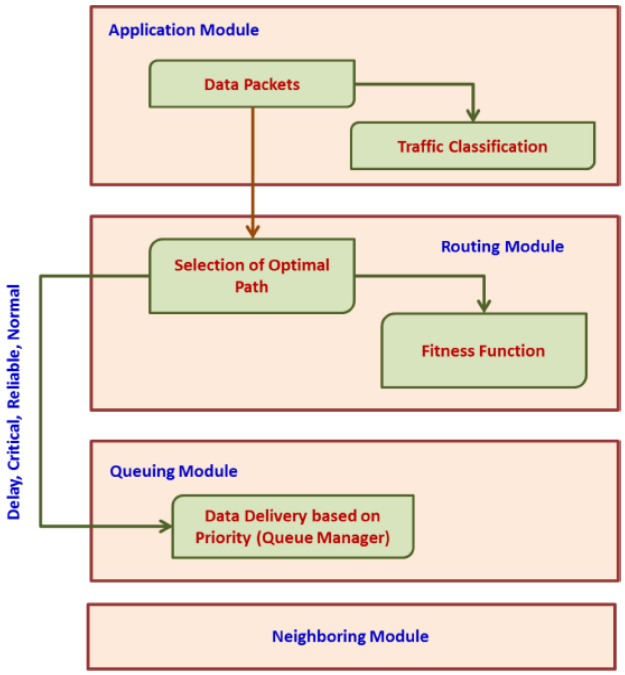

FIGURE 3. TRAFFIC CLASSIFICATION

This reduces delays in the data traffic as in Figure 4.

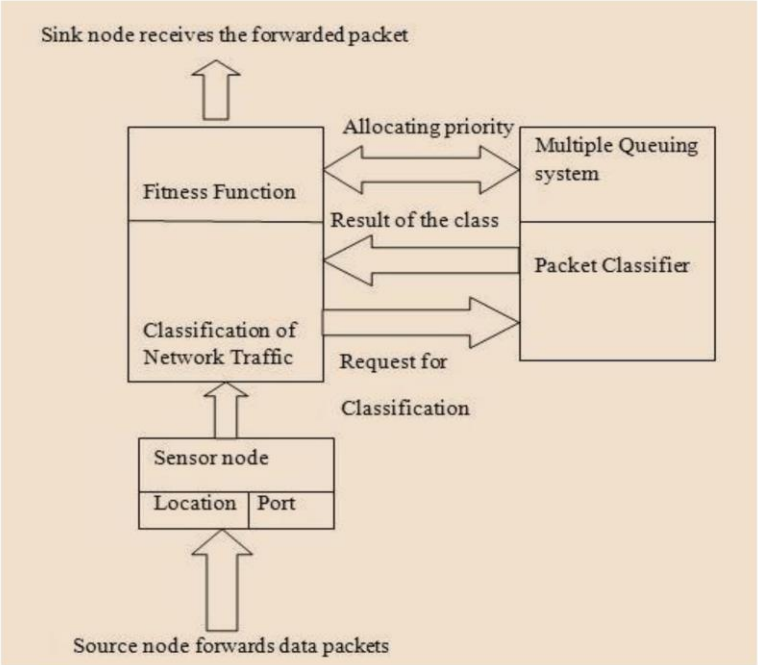

FIGURE 4. NETWORK TRAFFIC CLASSIFICATION MODEL

Priority-based data delivery allocates priority to every packet that is forwarded in the communicating medium depending on the traffic type. A multi-queuing priority policy is established to enable the nodes to construct several queues of priority for various classes of traffic by assigning a dissimilar level of importance to each class in a service policy. Every queuing model contains a particular value of threshold for the incoming packets based on the priority queues that handle network congestion effectively and also enhance the overall network performance. The data types involve several levels of importance and quality and therefore include diverse priorities in routing data packets.

DWDQN is a reinforcement learning method that integrates deep learning techniques with Q-learning for QoS demands, and data delivery. In DQN, the state space S encompasses all possible states of the environment, and the action space A includes all possible actions an agent can execute [29-30]. The policy π, typically parameterized by a neural network $Q(s, a; \theta)$, maps states $s \in S$ to actions $a \in A$, with θ representing the neural network parameters for data delivery. The reward function R(s, a) defines the immediate reward received by the agent after transitioning from state s to a new state s' through action a. The discount factor γ, generally set between 0 and 1, calculates the present value of future rewards, prioritizing data over more distant rewards. DQN is to learn a policy π^* which maximizes the total expected return, i.e., the cumulative discounted rewards obtained by policy π from any initial state s as data delivery. Mathematically is represented as follows,

$$Q^*(s,a) = \mathbb{E}\left[R(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \tag{25}$$

where $Q^*(s,a)$ is the maximum expected return obtainable from taking action a in state s. In practical training, DQN iteratively updates the neural network parameters to approximate the optimal Q-function Q∗(s, a), using the update equation (26),

$$\theta_{t+1} = \theta_t + \alpha[y_t - Q(s_t, a_t : \theta_t)]\nabla_{\theta_t} Q(s_t, a_t; \theta_t) \tag{26}$$

where $y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta_t)$ is the target Qvalue, r_t is the immediate reward received, and α is the learning rate. Target is to find the optimal policy π^* that maximizes the expected cumulative reward, which can be mathematically expressed as follows,

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))\right] \tag{27}$$

where $\pi(s_t)$ is denoted as the action taken in state st according to policy π, $R(s_t, a_t)$ is the reward function, γ is the discount factor, representing the importance of future rewards. The experience replay mechanism in DQN which stores transitions (s_t, a_t, r_t, s_{t+1}) from agent-environment interactions. Dynamic weighting mechanism is applied to the experience replay component of DQN. Each transition in the replay buffer is denoted by (s_t, a_t, r_t, s_{t+1}) with a dynamic weight w_t that influences is sampled for training the network. In this way, it ensures that more informative experiences exert a greater influence on model updates.

Temporal Difference Error Calculation: Specifically, the TD error for a transition is fundamental in updating the weights and is calculated using the following equation (28),

$$\delta_t = r_t + \gamma \max_{a} Q(s_{t+1}, a'; \theta) - Q(s_t, a_t; \theta) \tag{28}$$

where r_t is the reward received after taking action at in state s_t, γ is the discount factor representing the importance of future rewards, $Q(s_{t+1}, a'; \theta)$ estimates the maximum future reward from the next state s_{t+1}, and θ are the parameters of the Q-network. Equation (28) reflects the difference between the predicted reward for the action taken in the current state and the maximum predicted reward for the best possible data delivery action in the next state, adjusted by the immediate reward received and discounted by γ. The magnitude of δ_t indicates the degree to which the current Q-value predictions are off-target, which in turn informs how much the model needs to adjust its predictions. A high absolute value of δ_t suggests a significant error in prediction, implying that the learning from this particular transition could lead to substantial improvements in policy performance.

Weight Adjustment: The weight w_t of each transition in the experience replay is dynamically adjusted based on the absolute temporal difference error,

$$w_t = \exp(\lambda|\delta_t|) \tag{29}$$

where λ is a positive scaling factor which determines how sensitive the weight adjustments are to the TD error. Equation (29), exponential function is used to scale the weights of transitions in the experience replay, with the scaling factor λ modulating the extent of the adjustment. A larger λ increases the responsiveness of the weight to changes in the TD error, enhancing the priority of transitions with larger errors during training.

Sampling Mechanism Based on Updated Weights: The probability of sampling a particular transition from the experience replay buffer is adjusted in proportion to its weight,

$$P(t) = \frac{w_t}{\sum_i w_t} \quad (30)$$

where P(t) is the probability of sampling transition t, w_t is the weight of transition t, and $\sum_i w_t$ is the sum of weights for all transitions stored in the replay buffer. Equation (30) increases the frequency of sampling transitions with higher weights during training. Calculate the sampling probability P(t) by weighting it with w_t, the transitions weight.

THREE-LAYERED ARCHITECTURE

Three-layered architecture as shown in Figure 5, middle layer is the blockchain enabled RMCV and DL network layer. Whereas, the first and the third are the CWSN layers, comprising sensor nodes, CHs and BS. Moreover, ANN, CNN, Bidirectional Long Short-Term Memory (BiLSTM), and Bidirectional Gated recurrent unit (BiGRU) are trained and tested on the LEACH protocol generated dataset. The nodes in the CWSN perform routing using both LEACH and HMGEAR protocols. During routing, some nodes in the network may perform malicious activities. To minimize such activities in the network, the MNs are detected using the trained model and RMCV scheme. The nodes involved in the routing mechanism are classified using DL and RMCV techniques. Once classification is performed, the registration of LNs is done using PoA. While, routing is performed using LEACH and HMGEAR protocols after MND.

BLOCKCHAIN NETWORK: The blockchain enabled RMCV and CNN detect the presence of MNs in the WSN. It is deployed on CHs and BS. After MND, the LNs are registered in the blockchain using the smart contract. The PoA consensus mechanism is used when LNs are registered in blockchain network.

CWSN: The CWSN is composed of N number of nodes that perform routing using LEACH and HMGEAR protocols. After data processing, the data is collected by the sensor nodes from the surrounding environment and sent towards the CHs for processing. Further data processing is performed by the CHs and the integrated data is sent towards BS. In the WSNs, some nodes may perform malicious activities and may send false messages to disturb the data traffic or drop the data packets. Therefore, MND is performed using RMCV and DL-driven architecture. RMCV calculates the nodes' trustworthiness; other side nodes' classification is performed by ANN, CNN, BiLSTM, and BiGRU. The RMCV scheme detects the MNs in the WSNs. In the WSNs, the victim nodes may find the adversary messages sent by adversary nodes. The messages are evaluated based on their trustworthiness and integrity parameters, which are calculated by satisfying the following three conditions: message path, message conflict and message similarity. The MNs are removed from the network after being detected. While, routing is performed only between the LNs using LEACH and HMGEAR protocols.

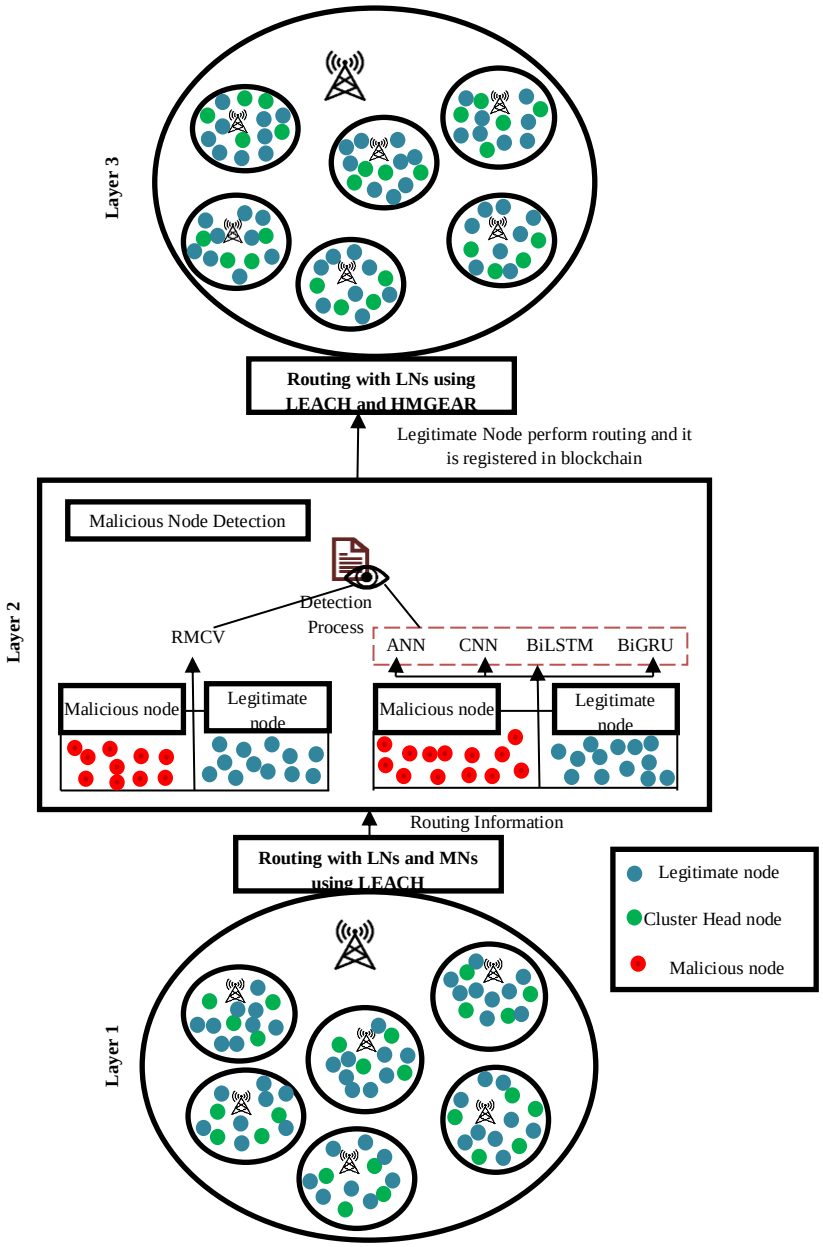

FIGURE 5. NEWLY PROPOSED 3-LAYERED SYSTEM MODEL

While the ANN contains only three layers: input, hidden, and output. The hidden layer processes the given input and forwards it to the fully connected neurons where the output is generated [31]. Therefore, CNN gives more accuracy than ANN. CNN is a regularized type of Feed-Forward Neural Network (FFNN) that learns features by itself via filter optimization. FFNN are usually fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. It is a multi layered architecture where the layers work well for feature extraction and classification purposes [32].

Bidirectional Long Short-Term Memory (BiLSTM) is an advanced version of the LSTM network and it is designed to effectively detect malicious nodes while preserving memory. BiLSTM can process the dataset forward in time, and the other processes it backward, enabling the consideration of both past and future information.

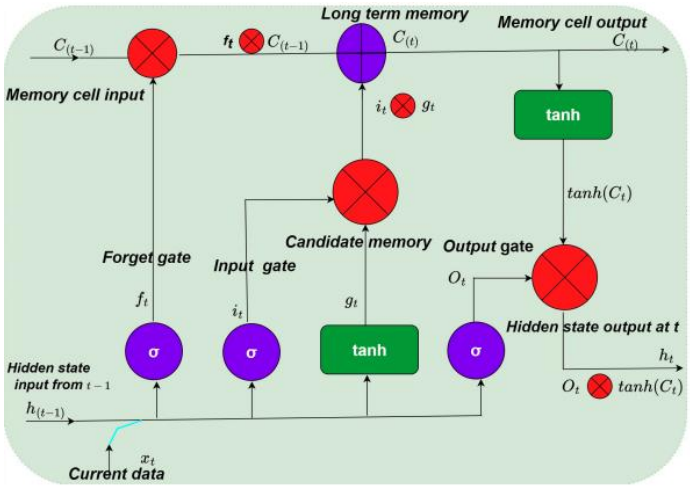

FIGURE 6. LSTM ARCHITECTURE

Figure 6 illustrates the detailed architecture of the LSTM model, emphasizing its unique ability to comprehensively analyze and detect malicious nodes [33]. The key components of LSTM include cell states and several gating mechanisms: forget, input, and output gates. Both the forget gate and the input gate determine which information is erased from and added to the cell state. When these two points are known, the cell state can be updated. Finally, the network's final output is determined by the output gate. These mechanisms can regulate the flow of malicious nodes between different gates using a sigmoid function [33]. An LSTM unit can be precisely described as follows,

$$i_t = \sigma(Y_i.x_t + W_i.h_{t-1} + b_i) \quad (31)$$

where Y and W is denoted as the weight matrix, b signifies the bias term, it represents the input gate at time t, · denotes the matrix multiplication process, σ denotes the sigmoid function, x_t signifies the nodes at a time (t), and h_{t-1} signifies the output of the previous LSTM unit. The input gate is crucial in identifying the specific node information from the previous unit that necessitates modification [33],

$$f_t = \sigma\left(Y_f.x_t + We_f.h_{t-1} + b_c\right) \quad (32)$$

where $f_t$is denoted as the forget gate and it is responsible for computing the significance of the nodes information and forgetting old malicious nodes,

$$\tilde{c}_t = tanh(Y_c.x_t + We_c.h_{t-1} + b_c) \quad (33)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (34)$$

where the $c_t$state of the candidate is determined through the utilization of the tangent activation function, as illustrated in Equation (32). Subsequently, the current cell state is evaluated and provided in Equation (35), where ⊙ represents point-to-point multiplication,

$$g_t = \sigma\left(Y_g.x_t + We_g.h_{t-1} + b_g\right) \quad (35)$$

$$h_t = g_t \odot \tanh(c_t) \quad (36)$$

The output gate $g_t$is calculated in Equation (35), where h_t represents the output of the LSTM unit, as given in Equation (36). BiLSTM consists of two LSTM layers that operate in both the forward and backward directions, as depicted in Figure 7. v_t is the output layer in the BiLSTM and it is formulated as follows [33],

$$v_t = [\vec{h}_t, \overleftarrow{h}_t] \quad (37)$$

The forward and backward outcomes of the LSTM units are represented by the symbols $\overleftarrow{h}_t$ and $\vec{h}_t$. The combination of these two LSTM units forms the output v_t. The fundamental concept underlying the RNN resides in the notion that, rather than transmitting the complete set of nodes information to the neural network in a WSN, progressively introduce the data one by one sequentially, effectively incorporating the temporal variable as well. Subsequently for the following output, feed the subsequent input alongside the previous output.

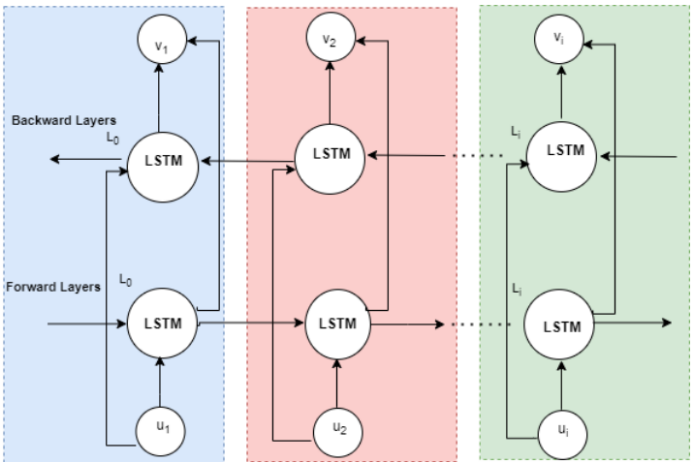

FIGURE 7. FRAMEWORK DIAGRAM OF BiLSTM

Figure 7, BiLSTM computes the bidirectional hidden states by running LSTM layers in both the forward and backward directions. The forward LSTM processed the sequence from the first to the last element, whereas the backward LSTM processed it in reverse order. The resulting hidden states from both directions were concatenated to form the final bidirectional hidden state. In deep learning, weights and biases are generated through training by minimizing a loss function. Weights and biases are modified by the training to minimize the difference between the output model and observed values. Adadelta optimizer, its learning rate is varied with values between 0 and 2. A very large value of learning rate may cause the model to exceed the optimal point, and a very small value may generate a slow training model [34].

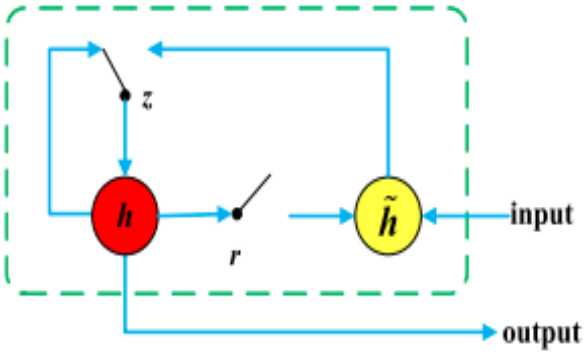

FIGURE 8. ILLUSTRATION OF THE GATED RECURRENT UNIT

GRU solves gradient vanishing problem using update and reset gate. If the gradient shrinks, then it back propagates over time and affects the learning, which makes the model untrainable. GRU gives the advantage over LSTM as it uses less memory and works better than LSTM. GRU also works faster than LSTM. On the other side, the LSTM works better on the datasets having long sequences. DL models find the maximum MNs in the network and remove them. Moreover, the Legitimate Nodes (LNs) identified by DL models are registered in the blockchain network using a smart contract. There are two gates in GRU: an update gate z and a reset gate r. They modulate whether information is updated or forgotten, as shown in Figure 8. To be specific, the update gate determines how many memories in the previous cell can survive, and the reset gate determines how to combine the new cell with the previous memory. Figure 8 shows the illustration of the Gated Recurrent Unit, which is the basis of Bidirectional Gated Recurrent Unit. There are four very important parts in GRU: reset gates r, update gates z. The activation of h_t^j of the GRU at time t is a linear interpolation between the previous activation h_t^{j-1} and the candidate activation $\tilde{h}_t^j$, which can be computed by equation (38),

$$h_t^j = \left(1 - z_t^j\right) h_t^{j-1} + z_i^j \tilde{h}_t^j \tag{38}$$

how much the cell updates its activation is determined by the update gate z_t^j , which is computed by equation (39),

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1}) \tag{39}$$

where σ is a nonlinear function such as logistic sigmoid function, x_t denotes a vector of the sequences at time step t, W_z and U_z are weights that can be trained to update z_t^j. Similar to the traditional recurrent cell, the candidate activation $\tilde{h}_t^j$ can be computed by equation (40),

$$\tilde{h}_t^j = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{40}$$

where r_t denotes a set of reset gates and is an elementwise multiplication. The reset gate is portrayed as the decision maker, which means that it determines how many of the previous states h_{t-1} can survive. When the reset gate is zero, $\tilde{h}_t^j$ forgets all previous states. Following the update gate, the reset gate r_t can be computed by equation (41),

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{41}$$

BiGRU includes two sequences: one forward and one backward. To fully consider the expression of information in different directions, element-wise summation is used to combine forward and backward sequences,

$$h_i = h_i^F \oplus h_i^B \tag{42}$$

where h_i^F represents the forward sequence, h_i^B denotes the backward sequence, μ denotes the mean value of the sentence of the semantic distribution, and $H = [h_1, \ldots, h_l]$ denotes the hidden representation. In the proposed model, the time for transmission, in the t[th] communication round, to the BS from the n[th] client is given using Equation (43) [35],

$$\tau_n^{up}(t) = \frac{\gamma_n i_n(t)}{Blog_2\left(1 + \frac{P_n(t)h_n(t)}{BN_0}\right)} \tag{43}$$

where the system bandwidth, transmission power of the nth node, noise power spectral density and the number of bits are given by n[th] node are given by $B, P_n(t), N_0$ and γ_n. In addition, Equation (44) gives the energy consumed by the n[th] node in the t[th] communication round [34],

$$E_n^{up}(t) = \frac{P_n(t)\gamma_n i_n(t)}{Blog_2\left(1 + \frac{P_n(t)h_n(t)}{BN_0}\right)} \tag{44}$$

Moreover, legitimate nodes (LNs) are registered in the blockchain network using proof-of-authority consensus protocol. The protocol outperforms proof-of-work in terms of computational cost.

SIMULATION RESULTS AND DISCUSSION

Simulation WSN-DS, a specialized dataset for WSN to detect DoS attacks. LEACH and protocol was used to collect the dataset because it is one of the most common and widely used routing protocols in WSNs. WSN-DS contains 374661 records that represent four types of DoS attacks: Blackhole, Grayhole, Flooding, and Scheduling attack to the normal behavior (no-attack) records (Table 1). Table 1 dataset has been collected from https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds. The dataset is separated to 60.00% of training and 40.00% of testing. Network Simulator 2 (NS2) was used to gather the required data and simulate the model. Simulation parameters are summarized in Table 2. Experiments are conducted using hardware platform of Intel (R) Xeon (R), i5-3570 CPU with the clock frequency of 3.50 GHz and 16.00 GB memory, 10 MB cache. Simulation is performed for different number of sensor nodes various percentage of CHs at different locations of BS in the sensing field of area $200 \times 200\ m^2$. The sensor nodes change within a range of 100 to 400 and the number of selected CHs vary from 15% to 30% while BS is positioned at (50,50), (100, 100), (150, 150).

TABLE 1. WSN-DS DATASET DETAILS

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.VARIOUS PARAMETERS USED IN SIMULATION

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

For better CH selection, distance to BS, node centrality, residual energy, node degree, and distance to neighbors are the inputs given to BCOA. Distance, node degree, and residual energy are also inputs to the EEBCDL. Because these methods are commonly employed to enhance the WSNs energy efficiency, this proposed method (CCAR) is contrasted to several established approaches such as the DDR-LEACH [23], ETD-LEACH [24], HMGEAR [25], OQoS-CMRP [22], and BCOA-EEBCDL (Beetle Chimp Optimization Algorithm- Energy Efficiency Blockchain Deep Learning).

QoS Metrics: Simulation results show that proposed protocol can prolong the network lifetime, improve the network throughput, increased Packet Delivery Ratio (PDR), reduced Packet Loss Ratio (PLR) and reduced average energy consumption.

Throughput: Throughput is the amount of data that passes through the network in a given time period. It's usually measured in bits per second (bps) or megabits per second (Mbps).

Packet Delivery Ratio (PDR): Packet Delivery Ratio (PDR) can be expressed as follows,

$$R_{Success} = \frac{P_{succ}}{P_{all}} \times 100\% \tag{45}$$

where $R_{Success}$represents the ratio of packet successful, P_{succ}denotes the number of packets successfully sent by the router, P_{all} indicates the total number of packets transmitted through the router.

Packet Loss Ratio (PLR): Packet Loss Ratio (PLR) can be expressed as follows,

$$R_{drop} = \frac{P_{drop}}{P_{all}} \times 100\% \tag{46}$$

where R_{drop}represents the ratio of packet loss, P_{drop}denotes the number of packets dropped by the router, P_{all} indicates the total number of packets transmitted through the router.

Average energy consumption: During each iteration, it specifies how much energy each node uses on average. It has been measured in terms of mJ.

Detection metrics: Because different performance metrics are appropriate in different settings, performance metrics like True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Overall Accuracy (A) has been used to assess the performance of intrusion detection methods. TPR represents the rate of attack cases identified correctly, TNR represents the rate of normal (no-attack) cases identified correctly, FPR represents the rate of no-attack cases identified as attacks by the system, and FNR represents the rate of attack cases identified as normal ones. A is the total rate of correct decisions whether identifying an attack correctly or deciding there is no attack when really there is no attack.

$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \times 100\%$	(47)
$\text{True Negative Rate (TNR)} = \frac{TN}{TN + FP} \times 100\%$	(48)
$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \times 100\%$	(49)
$\text{False Negative Rate (FNR)} = \frac{FN}{FN + TP} \times 100\%$	(50)
$\text{A} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$	(51)

where TP is the number of attack cases classified correctly as attacks. TN is the number of normal (no-attack) cases classified correctly as normal (no-attack). FP is the number of normal (no-attack) cases classified incorrectly as attacks.

FN is the number of attack cases classified incorrectly as normal (no-attack). Attack detection results of DL methods with respect to evaluation metrics are illustrated in table 3.

TABLE 3. PERFORMANCE METRICS OF INTRUSION DETECTION SYSTEMS

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

TABLE 4. PERFORMANCE ANALYSIS COMPARISON OF ROUTING METHODS

Number of nodes	**Network lifetime (rounds)**					
	DDR-LEACH	**ETD-LEACH**	**HMGEAR**	**OQoS-CMRP**	**BCOA-EEBCDL**	**CWHO-CCAR**
100	892	1058	1234	1462	1645	1859
200	1241	1395	1586	1742	1921	2158
300	1625	1753	1975	2086	2258	2475
400	1858	2026	2327	2455	2587	2856
500	2215	2377	2581	2726	2814	3067
Number of nodes	**Throughput(Mbps*10^4)**					
	DDR-LEACH	**ETD-LEACH**	**HMGEAR**	**OQoS-CMRP**	**BCOA-EEBCDL**	**CWHO-CCAR**
100	8.22	10.18	12.51	13.85	15.73	17.25
200	10.85	12.97	15.19	16.93	18.47	19.84
300	13.71	15.46	17.96	19.25	20.32	21.95
400	15.68	17.05	19.67	20.78	21.83	23.89
500	17.45	18.63	20.84	22.56	23.75	25.48
Number of nodes	**Packet Delivery Ratio (PDR) (%)**					
	DDR-LEACH	**ETD-LEACH**	**HMGEAR**	**OQoS-CMRP**	**BCOA-EEBCDL**	**CWHO-CCAR**
100	82.18	84.59	87.44	89.45	90.52	92.67
200	80.66	83.21	86.15	88.57	89.71	91.84
300	79.72	81.47	84.81	87.45	88.54	90.69
400	78.26	80.43	83.95	86.21	87.46	89.78
500	77.35	79.24	82.63	85.39	86.47	88.62
Number of nodes	**Packet Loss Ratio (PLR) (%)**					
	DDR-LEACH	**ETD-LEACH**	**HMGEAR**	**OQoS-CMRP**	**BCOA-EEBCDL**	**CWHO-CCAR**
100	17.82	15.41	12.56	10.55	9.48	7.33
200	19.34	16.79	13.85	11.43	10.29	8.16
300	20.28	18.53	15.19	12.55	11.46	9.31
400	21.74	19.57	16.05	13.79	12.54	10.22
500	22.65	20.76	17.37	14.61	13.53	11.38
Number of Rounds	**Average Energy Consumption (mJ)**					
	DDR-LEACH	**ETD-LEACH**	**HMGEAR**	**OQoS-CMRP**	**BCOA-EEBCDL**	**CWHO-CCAR**
500	0.362	0.335	0.294	0.265	0.234	0.205
1000	0.394	0.359	0.315	0.286	0.262	0.248
1500	0.425	0.397	0.356	0.319	0.284	0.267
2000	0.458	0.423	0.378	0.343	0.298	0.271
2500	0.487	0.456	0.404	0.366	0.321	0.305
3000	0.529	0.504	0.446	0.415	0.358	0.316

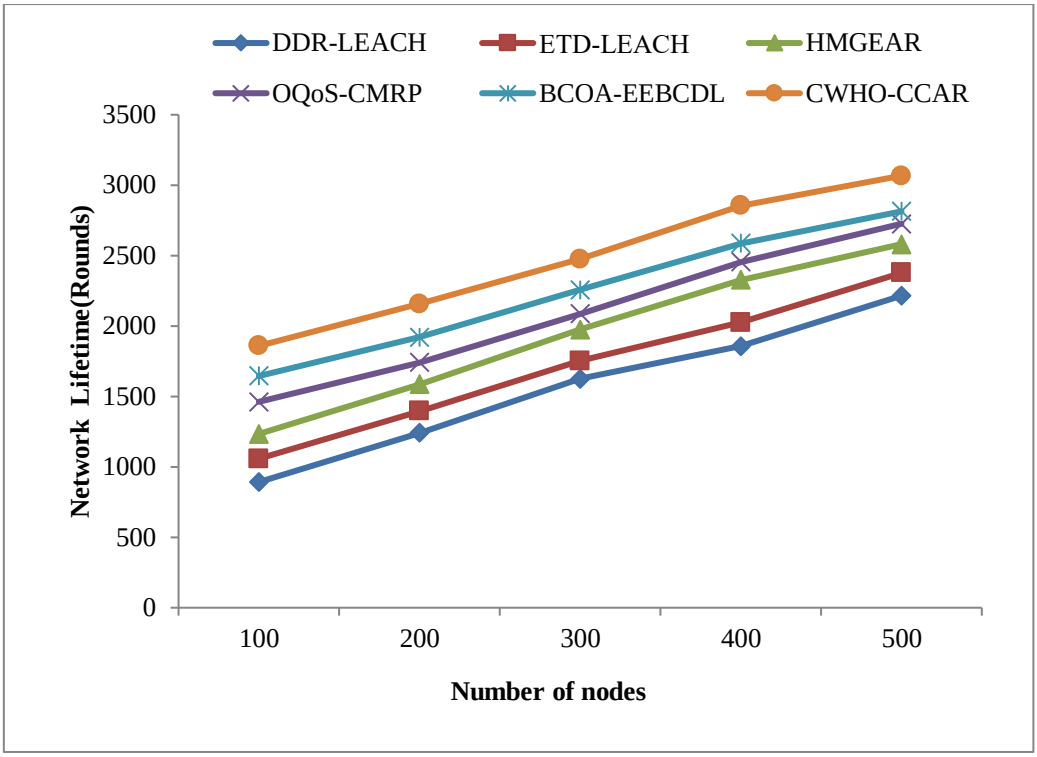

FIGURE 9. NETWORK LIFETIME COMPARISON OF ROUTING METHODS

Network lifetime in terms of rounds with conventional methods and existing methods is illustrated in figure 9. The proposed system has the highest network lifetime when compared with existing methods of 100 to 500 nodes. CWHO-CCR has the highest network lifetime of 3067 rounds; other methods such as DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL have the lowest network lifetime of 2215 rounds, 2377 rounds, 2581 rounds, 2726 rounds, and 2814 rounds for 500 nodes (see Table 4). Proposed system has highest network lifetime due to optimal CH election with CWHO. CWHO is a metaheuristic algorithm that simulates the social behavior of wild horses in nature for optimal selection of CH in the WSN model.

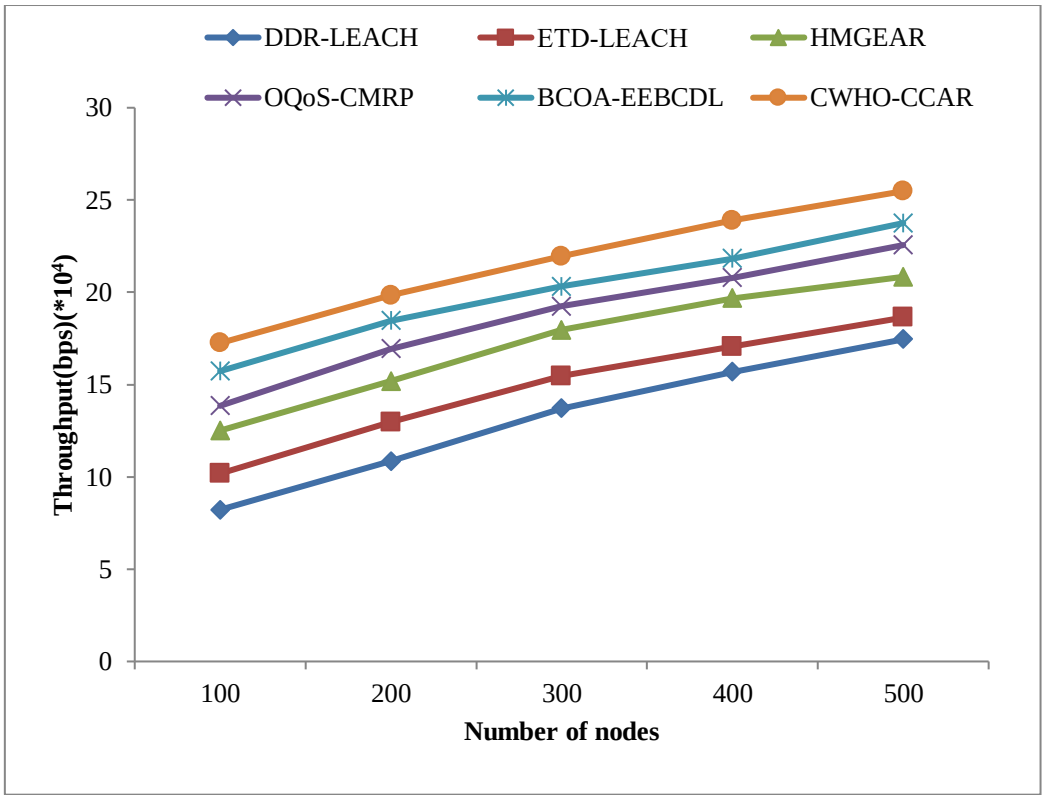

FIGURE 10. THROUGHPUT COMPARISON OF ROUTING METHODS

The proposed method and conventional methods like DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL with respect to throughput are illustrated in figure 10. The proposed system has the highest throughput of $25.48*10^4$ bps; other methods such as DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL have the lowest throughput of $17.45*10^4$ bps, $18.63*10^4$ bps, $20.84*10^4$ bps, $22.56*10^4$ bps, and $23.75*10^4$ bps for 500 nodes (see Table 4). Proposed system has highest throughput due to optimal CH election with CWHO, and congestion control using Dynamic Weight Deep Q-Network (DWDQN).

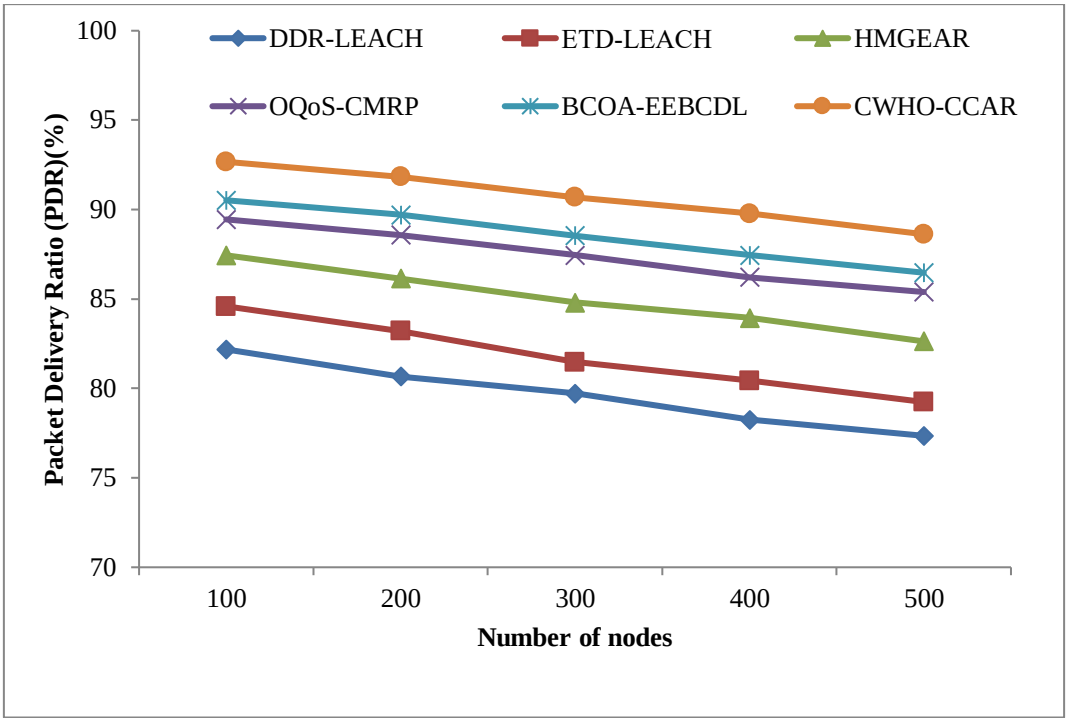

FIGURE 11. PDR COMPARISON OF ROUTING METHODS

PDR results of DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, BCOA-EEBCDL, and CWHO-CCAR with 100 and 500 nodes are shown in Figure 11. The proposed system has the highest PDR of 92.67%; other methods such as DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL have the lowest PDR of 82.18%, 84.59%, 87.44%, 89.45%, and 90.52% for 100 nodes (see Table 4). Dynamic Weight Deep Q-Network (DWDQN) is used by queuing model for assigning priority to data packets which shows proposed system has highest PDR than other methods.

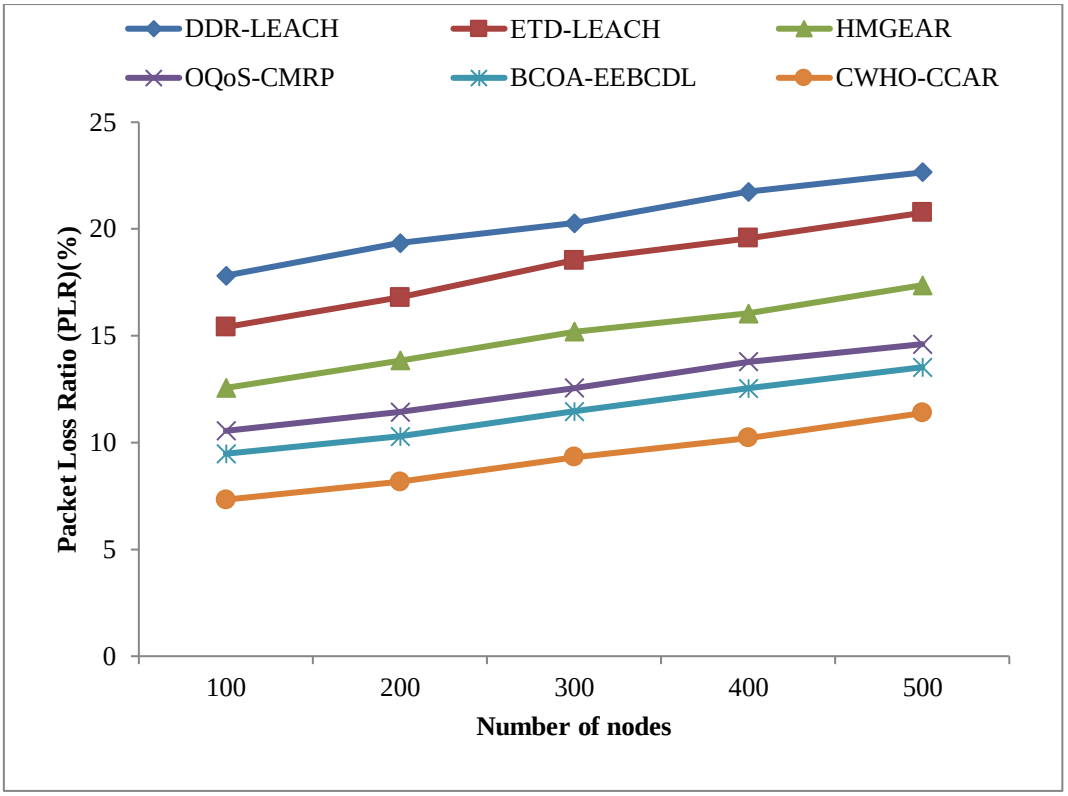

FIGURE 12. PLR COMPARISON OF ROUTING METHODS

The proposed system with existing techniques like DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL in terms of PLR with 100 to 500 nodes is illustrated in figure 12. PLR of the proposed method is compared with existing methods of network size 100 to 500 nodes. The proposed system has the lowest PLR of 7.33%; other methods such as DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL have increased PLR of 17.82%, 15.41%, 12.56%, 10.55%, and 9.48% for 100 nodes (see Table 4). DWDQN is introduced for assigning priority to data packets which overcomes congestion, proposed system has lowest PLR than other methods.

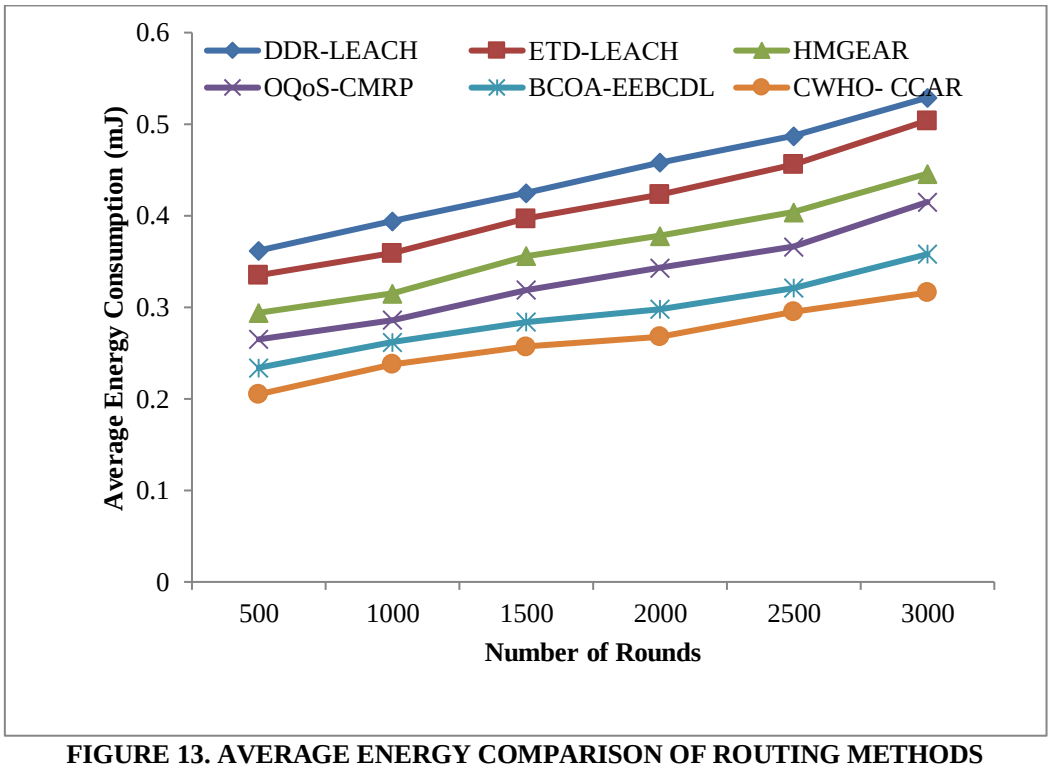

FIGURE 13. AVERAGE ENERGY COMPARISON OF ROUTING METHODS

Figure 13 shows the average energy comparison of DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, BCOA-EEBCDL, and CWHO-CCAR with a network size of 100 to 500 nodes. The proposed system has the lowest average energy consumption of 0.316 mJ; other methods such as DDR-LEACH, ETD-LEACH, HMGEAR, OQoS-CMRP, and BCOA-EEBCDL have increased average energy consumption of 0.529 mJ, 0.504 mJ, 0.446 mJ, 0.415 mJ, and 0.358 mJ for 500 nodes (see Table 4).

CONCLUSION AND FUTURE WORK

In this paper, Congestion Clustering Aware Routing (CCAR) protocol is introduced to alleviate the congestion issue over the WSN model. The CCAR protocol is proposed to decrease end-to-end delay time and prolong the network lifetime through choosing the suitable CH and the cluster member. Chaotic Wild Horse Optimization (CWHO) model is introduced for CH election, and it simulates the social behavior of wild horses of horses such as grazing, domination, leadership hierarchy, and mating. WHO suffers from low exploitation capability and stagnation in local optima has been solved by using logistic chaotic. CWHO is based on the residual energy, distance among sensor nodes, CH and BS distance, Node degree, and Node centrality. Dynamic Weight Deep Q-Network (DWDQN) is introduced which control packets are transmitted to neighboring nodes such that every node locally updates the neighbors with the routing decision. A multi-queuing priority policy is established to enable the nodes to construct several queues of priority for various classes of traffic by assigning a dissimilar level of importance to each class in a service policy. Malicious nodes (MNs) are detected using the Deep Learning (DL) techniques, and Real-Time Message Content Validation (RMCV) scheme. The experimental results demonstrate that the effectiveness of the CCAR protocol to satisfy the quality of service (QoS) requirements in increasing the network lifetime, Throughput, Packet Delivery Ratio (PDR), Packet Loss Ratio (PLR), and reduced average energy consumption. Moreover, security analysis is performed, which shows that blockchain network is resilient against different vulnerabilities. Furthermore, other performance metrics such as end-to-end delay, computation time are to be studied and a real network rather than simulation should be established to further evaluate routing protocol.

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. Nayak, P., Swetha, G.K., Gupta, S. and Madhavi, K., 2021. Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities. Measurement, 178, pp.1-15.
4. Riaz, M.N., 2018. Clustering algorithms of wireless sensor networks: a survey. International Journal of Wireless and Microwave Technologies (IJWMT), 8(4), pp.40-53.
5. Al-Sulaifanie, A.I., Al-Sulaifanie, B.K. and Biswas, S., 2022. Recent trends in clustering algorithms for wireless sensor networks: A comprehensive review. Computer Communications, 191, pp.395-424.
6. Wohwe Sambo, D., Yenke, B.O., Förster, A. and Dayang, P., 2019. Optimized clustering algorithms for large wireless sensor networks: A review. Sensors, 19(2), pp.1-27.
7. 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.
8. Gherbi, C., Aliouat, Z. and Benmohammed, M., 2017. A survey on clustering routing protocols in wireless sensor networks. Sensor Review, 37(1), pp.12-25.
9. Fanian, F. and Rafsanjani, M.K., 2019. Cluster-based routing protocols in wireless sensor networks: A survey based on methodology. Journal of Network and Computer Applications, 142, pp.111-142.
10. Del-Valle-Soto, C., Rodríguez, A. and Ascencio-Piña, C.R., 2023. A survey of energy-efficient clustering routing protocols for wireless sensor networks based on metaheuristic approaches. Artificial Intelligence Review, 56(9), pp.9699-9770.
11. Pandey, D.; Kushwaha, V. An exploratory study of congestion control techniques in Wireless Sensor Networks. Comput. Commun. **2020**, 157, 257–283.
12. Jiang Q., S. Zeadally, J. Ma, and D. He, ‘‘Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,’’ IEEE Access, vol. 5, pp. 3376–3392, 2017.
13. Mishra L. and S. Varma, ‘‘Middleware technologies for smart wireless sensor networks towards Internet of Things: A comparative review,’’ Wireless Pers. Commun., vol. 116, no. 3, pp. 1539–1574, 2021.
14. Padmavathi U. and N. Rajagopalan, ‘‘Concept of blockchain technology and its emergence,’’ in Blockchain Applications in IoT Security. Hershey, PA, USA: IGI Global, 2021, pp. 1–20.
15. Pravin, R.A., Shiny, X.A., Vennila, V.B., Selvaraju, P. and Mageswari, R.U., 2024. Congestion aware clustered WSN based on an improved ant colony algorithm. Measurement: Sensors, 34, pp.1-6.
16. Farsi, M., Badawy, M., Moustafa, M., Ali, H.A. and Abdulazeem, Y., 2019. A congestion-aware

clustering and routing (CCR) protocol for mitigating congestion in WSN. IEEE Access, 7, pp.105402-105419.
17. Patil, K.K., Kumaran, T.S. and Mathapat, M., 2024. OCC-MP: An optimal cluster based congestion aware technique multipath routing protocol in WSN using hybrid evolutionary techniques. Measurement: Sensors, 31, pp.1-9.
18. Patil, K.K., Senthil Kumaran, T. and Prasad, A.Y., 2023. Improved congestion control in wireless sensor networks using clustering with metaheuristic approach. Journal of Interconnection Networks, 23(02), p.2250005.
19. Maheshwari, P., Sharma, A.K. and Verma, K., 2021. Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. Ad Hoc Networks, 110, pp.1-52.
20. Moussa, N. and El Belrhiti El Alaoui, A., 2021. An energy-efficient cluster-based routing protocol using unequal clustering and improved ACO techniques for WSNs. Peer-to-Peer Networking and Applications, 14(3), pp.1334-1347.
21. Reddy, D.L., Puttamadappa, C. and Suresh, H.N., 2021. Merged glowworm swarm with ant colony optimization for energy efficient clustering and routing in wireless sensor network. Pervasive and Mobile Computing, 71, pp.1-18.
22. Deepa, O. and Suguna, J., 2020. An optimized QoS-based clustering with multipath routing protocol for wireless sensor networks. Journal of King Saud University-Computer and Information Sciences, 32(7), pp.763-774.
23. Amjad, S., Abbas, S., Abubaker, Z., Alsharif, M.H., Jahid, A. and Javaid, N., 2022. Blockchain based authentication and cluster head selection using DDR-LEACH in internet of sensor things. Sensors, 22(5), pp.1-20.
24. Khan, A.U., Sajid, M.B.E., Rauf, A., Saqib, M.N., Zaman, F. and Javaid, N., 2022. Exploiting Blockchain and RMCV-Based Malicious Node Detection in ETD-LEACH for Wireless Sensor Networks. Wireless Communications and Mobile Computing, 2022(1), pp1-15.
25. Jibreel, F., Tuyishimire, E. and Daabo, M.I., 2022. An enhanced heterogeneous gateway-based energy-aware multi-hop routing protocol for wireless sensor networks. Information, 13(4), pp1-15.
26. Mehta, D. and Saxena, S., 2020. MCH-EOR: Multi-objective cluster head based energy-aware optimized routing algorithm in wireless sensor networks. Sustainable Computing: Informatics and Systems, 28, pp.1-34.
27. 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.
28. 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.
29. Zhang, X., Zhang, J., Si, W. and Liu, K., 2024. Dynamic Weight Adjusting Deep Q-Networks for Real-Time Environmental Adaptation. *arXiv preprint arXiv:2411.02559*, pp.1-8.
30. Hafiz A., "A survey of deep q-networks used for reinforcement learning: state of the art," Intelligent Communication Technologies and Virtual Mobile Networks: Proceedings of ICICV 2022, pp. 393–402, 2022.
31. Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J., 2021. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), pp.6999-7019.
32. Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H.I.K., Frisky, A.Z.K. and ul Haq, Q.M., 2022. Understanding of Convolutional Neural Network (CNN): A review. International Journal of Robotics and Control Systems, 2(4), pp.739-748.
33. 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.
34. 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.
35. 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.