

A Centralized Energy-Efficient Clustering and Routing Approach for Wireless Sensor Networks via Graph Neural Networks

Sayyed Majid Mazinani¹, Morteza Saadati² and Ali Akbar Khazaei²

Abstract-- In Wireless Sensor Networks (WSNs), energy efficiency is one of the main challenges. Clustering is an efficient method for prolonging the lifetime of WSNs. The canonical problem in this context is the choice of Cluster Heads (CHs). CHs consume more energy than member nodes; therefore, the rotation of the CH role among all nodes is essential to prevent the early death of CHs. As this problem is NP-hard, traditional approaches relied on random selection and rotation of the CH role, while others considered heuristics such as position, member count, and residual energy to select CHs in either a centralized or distributed fashion. In this paper, we present a CH selection method based on Graph Neural Network (GNN) to replace ad hoc heuristics with a neural network that effectively considers both the state of nodes and the intricate relationships between them to select CH with the goal of prolonging the lifetime of the network. To this end, we trained a GNN on WSNs with a few nodes, in which solving with exact methods is feasible. Thanks to the scalability feature of GNNs, we can apply this trained GNN to larger networks for CH selection. Also, to avoid long-range communications that consume a considerable amount of energy, a multi-hop routing scheme is proposed to balance energy consumption. Simulation results confirm the effectiveness of our GNN in comparison to state-of-the-art protocols. Furthermore, our results highlight the promising usefulness of GNNs in addressing other challenges in WSNs.

Index Terms-- Energy Efficiency, Clustering, Graph Neural Network, Routing, WSN

I. INTRODUCTION

Internet of Things (IoT) is a technology with the potential to revolutionize many aspects of our lives. WSN is one of the critical contributors to the IoT and is regarded as an integral part

of it [1] and as an enabler of low-cost implementation of new technologies [14]. Recent advances in IoT, wireless networks, and Micro-Electro-Mechanical Systems (MEMS) have rendered WSNs as one of the promising tools to achieve industrial automation, smart agriculture, military and civil surveillance, wildlife monitoring, smart cities, health, and many more interesting applications [25, 14, 31, 4, 54]. WSNs have the advantages of low cost, convenient deployment, and the ability to work independently and in harsh environments [14, 1]. They are composed of one or a few powerful base stations and many miniaturized, independent, and low-powered sensor devices capable of sensing various environmental factors, processing data, and wireless communication. Sensors monitor the environment and send the sensed data back to the base station(s) and ultimately to end-users via the internet. Data acquisition can occur periodically (time-driven), only when an event happens (event-driven), or upon a query from the end user (query-driven) [1]. The sensor node's battery cannot be recharged or replaced after deployment. This causes one of the main challenges that WSNs face. As the energy available to devices is limited, the networks' lifetime will be bounded, making energy the most critical resource in a WSN [1, 8]. Therefore, designing energy-efficient communication protocols is a major focus of research in the field of WSN [10, 8, 16, 23].

Energy is spent in a node for different purposes, including sensing, processing, and wireless communication. Most of the energy is dissipated due to wireless communication [17, 1, 4, 38]. Therefore, the main focus of studies on energy efficiency is on optimizing the communication. According to both free-

Received; 2026-04-25

Revised; 2026-05-21

Accepted; 2026-08-23

1. Department of Electrical and Computer Engineering, Imam Reza International University, Mashhad, Iran.

2. Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

*Corresponding author Email: smajidmazinani@imamreza.ac.ir

Cite this article as:

Mazinani, S. M., Saadati, M., and Khazaei, A. A. (2026). A Centralized Energy-Efficient Clustering and Routing Approach for Wireless Sensor Networks via Graph Neural Networks. *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)*. Semnan University Press. 6 (4), 1-12. DOI: <https://doi.org/10.22075/mseee.2026.40942.1266>

space and multi-path propagation models, the energy required to transmit a message grows exponentially as the distance to the recipient increases. This behavior suggests that it is more energy-efficient to send a message over multiple close by hops than to transmit it directly [8]. On the other hand, many-hop communication exhibits more delay and leads to the emergence of the well-known hotspot problem [19]. As all the routes converge on the BS, the hotspot problem arises from excessive load on nodes near the BS, leading to their early death and resulting in coverage holes and the early death of other nodes. [11, 38, 2]

To mitigate these challenges, the hierarchical clustering protocols are the most successful [9, 14, 38]. These protocols exploit local correlations in sensed data to aggregate information and send more accurate and compact data to the BS [8]. To achieve this, the time is divided into multiple rounds. In each round, the network is partitioned into multiple local clusters, and one CH node is elected for each cluster. Nodes send the sensed parameters to their respective CHs, and the CH aggregates the received data into a single message and sends it to the BS via intermediary relay nodes [23, 2]. As the clustering structure has a significant effect on the lifetime of WSNs, the conical question is how to choose CHs and relays to get the maximum lifetime for the network. The lifetime of a WSN is defined as the number of rounds until the first, half, and all of the nodes are dead. These metrics are referred to as FND, HND, and LND, respectively. Other lifetime definitions based on coverage and throughput are also proposed; for a discussion on WSN lifetime metrics, refer to [26].

The energy-efficient clustering in WSNs is a well-known NP-hard problem [9, 4]. Also, [8] further proved that no Fully Polynomial Time Approximation Scheme (FPTAS) exists for this problem. The literature in this area is focused on different types of clustering; one taxonomy is based on static vs. dynamic clusters. In a static clustering method, the clusters are decided at the onset of network operation and remain fixed throughout the network's lifetime. In contrast, in a dynamic clustering protocol, the clusters are re-selected at different rounds of network operations. Another taxonomy is based on the centralized vs. distributed nature of the protocols. In the centralized approach, all required information for clustering is gathered, and the clustering decisions are made at the BS. In contrast, in a distributed protocol, nodes form clusters in a collaborative manner.

A. Graph Neural Networks

GNNs are a class of deep learning models designed to operate on graph-structured data. Unlike traditional neural networks that handle regularly-shaped data, such as images or word sequences [36], GNNs can effectively learn representations from data where relationships between entities are inherently irregular and complex, e.g., computer or social networks, molecular structures, and transportation systems [45, 39]. By leveraging the connectivity patterns and node attributes within a graph, GNNs enable the extraction of rich, contextual information that captures both local and global structure.

The core mechanism of GNNs involves the iterative aggregation and transformation of information from a node's neighbors to update its representation. This process, often referred to as message passing, allows each node to integrate features from its immediate and extended neighborhood, facilitating tasks such as node classification, link prediction, and graph classification [45].

GNNs are particularly useful for problems in computer networks, as the underlying structure of the network in terms of topology is best represented by a graph [35]. Also, they offer generalization and scalability. Unlike CNNs, in which the size of inputs and outputs is fixed after training, a GNN can be applied to graphs of varying sizes without losing performance. This ability proves very useful for applications in computer networks, as training for different network sizes is not feasible [28, 39]. Additionally, GNNs are implemented by employing efficient message passing protocols, where information is concurrently aggregated and updated across nodes. This parallelism significantly reduces computation time [39]. GNNs have been successfully utilized to solve various problems in wireless networks. For example, in [39, 22, 35, 42, 53]. For a detailed review of GNN applications in wireless networks, refer to [15, 50].

B. Motivation and Contributions

Extensive literature exists on energy efficiency in WSNs, with many methods and algorithms proposed over the past few years. However, due to the NP-hardness of the problem, research is still actively pursuing ways to further optimize energy consumption in these battery-limited wireless networks. Designing an energy-efficient clustering and routing protocol for WSN entails two objectives: to both minimize and balance energy consumption among nodes. As GNNs have served as a powerful tool to tackle various problems in computer networks, we aim to incorporate GNNs to solve energy-efficient CH selection in WSNs.

This paper proposes a novel application of GNNs to enhance energy efficiency in WSNs by selecting CHs, with the overarching goal of prolonging network lifespan. We introduce a methodological framework that employs a GNN, trained on data from small-scale WSNs where exact solutions are attainable, to predict the suitability rank of nodes for CH designation. Additionally, we develop a multi-hop routing scheme to achieve a more equitable distribution of communication tasks among all nodes, thus achieving more balanced energy consumption among the nodes. Through comprehensive simulations and comparisons with recent and classical WSN clustering techniques, we validate the efficacy of our approach. The results highlight the promising application of GNNs in addressing energy efficiency challenges in WSNs and suggest broader utility for tackling similar issues across wireless networks and other graph-based combinatorial optimization problems.

C. Organization

The rest of this paper is organized as follows. Related works are reviewed in Section II. The system model and problem definition are presented in Section III. In Section IV, the

proposed solution is explained in detail. Section V contains the performance evaluation simulation results. Furthermore, the conclusion relies on the last section.

II. RELATED WORK

In LEACH [17], an early and widely adopted clustering protocol for WSNs, CHs are selected probabilistically based on a random number in each node. If the number is below a threshold, the node is designated as the role of CH and informs its neighboring nodes of its role. The member nodes join the nearest CH based on the strength of the received signal. LEACH has several shortcomings. Firstly, the use of random numbers for CH selection can lead to selecting CHs with low energy or unsuitable locations while leaving some nodes in sparse parts of the field without a CH [25].

Several protocols aim to improve LEACH. LEACH-C [16] is a centralized counterpart to LEACH. It aims to minimize the energy consumption of member nodes by minimizing the total distance between a member and its CH using a simulated annealing optimization algorithm. To prevent low-energy nodes from becoming CHs, LEACH-C considers only nodes with energy above average as CH candidates. However, due to the use of single-hop communication between CHs and the BS, the overall energy consumption of LEACH-C is not optimal. MEACH [7] is another improvement over LEACH. It is a distributed protocol that forms clusters based on the radio range and the number of cluster members. To improve energy efficiency, a multi-hop routing is proposed where some CHs act as relays for more distant CHs. It solves some limitations of previous protocols, but due to the high number of control messages, energy efficiency and delay are increased. PEGASIS [24] uses a greedy algorithm at the BS to form chains of nodes to minimize communication distance and ultimately energy consumption. Each node in the chain sends its sensed data to the next node, and the next node combines the received data with that of its own and forwards it. Finally, a chain leader sends the data to the BS. To prevent the early death of the chain leader, a new leader is selected at the beginning of the next round. PEGASIS shortcomings include failing to account for energy when choosing the leader node, and the delays as a result of many-hop routing and inapplicability for large WSNs due to a single chain leader.

Beyond LEACH and its variations, numerous protocols have been proposed to enhance clustering efficiency through the use of meta-heuristic and swarm intelligence algorithms. Namely, Genetic Algorithm [29, 32, 30], Particle Swarm Optimization (PSO) [33, 52, 18, 34], Gray Wolf Optimizer [9, 51], Salp Swarm Algorithm [8, 41], and Spider Monkey Optimization [27].

Neural networks have also been widely used to tackle energy-efficient clustering in WSN. In [3], a multi-hop routing protocol based on reinforcement learning (RL) is proposed to minimize delay in energy harvesting WSNs. Both centralized and distributed RL algorithms are examined. However, using binary features to approximate the Q-value and delayed convergence of RL algorithms in general makes this protocol sub-optimal. RL is also used in [12] to select the next hop for a node in a

distributed fashion. Also, in [13], a distributed coverage hole recovery method based on RL by re-adjusting the sensing range of nodes is proposed. Authors of [5] developed a routing algorithm based on a novel deep belief network framework. An RL algorithm is adopted for CH selection, while routing is handled using a deep learning approach. In [6], a GNN is proposed for deep structural clustering. The GNN is enhanced to solve the problem of structural information vanishing.

Many other works utilized RL for solving problems in WSNs, including [21, 37, 40, 44, 46, 47, 48, 49]. For an extensive study of clustering protocols in WSNs, refer to [11, 19]. These reviews highlight various advancements and approaches in clustering protocols, each aiming to improve energy efficiency, stability, and adaptability in WSNs.

III. SYSTEM MODEL AND PROBLEM DEFINITION

A. Network Model

In this work, we considered a network consisting of a set of homogeneous sensor nodes randomly deployed in a 2-dimensional sensing field denoted by N . Furthermore, we assumed that the nodes know their location and do not move after deployment. Additionally, the wireless channel is symmetric, and the BS is within the communication range of all nodes. We assumed a time-driven paradigm for data collection, meaning that all nodes sense and report data to the BS periodically at each round. Finally, the CHs can aggregate and compress data received from cluster members.

B. Energy Model

We are using equations from the system model presented in the LEACH [17]. Here, the dissipated energy to send a message to a distance d is given by (1).

$$E_{TX}(l, d) = \begin{cases} l * E_{elec} + l \times E_{fs} \times d^2, & d \leq d_0 \\ l * E_{elec} + l \times E_{mp} \times d^4, & d > d_0 \end{cases} \quad (1)$$

where l is the length of the message, E_{elec} is the amount of energy dissipated by the electronic circuit per bit, E_{fs} and E_{mp} depend on the transmitter amplifier model, and the threshold value d_0 is given by (2).

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (2)$$

On the other hand, the dissipated energy for receiving a message of length l is given by (3).

$$E_{RX}(l) = l \times E_{elec} \quad (3)$$

The CHs need to use the processing unit to aggregate and compress the received data from the cluster members, resulting in energy dissipation as given by (4), where E_{da} is the per-bit energy required for aggregation.

$$E_{DA}(l) = l \times E_{da} \quad (4)$$

In this system model, nodes can take three roles. Namely, cluster member, CH, and Relay. According to (1)-(4), the

energy consumption of nodes based on their role in the network can be formulated. Here, we do not account for the energy required for sensing [9] as it is relatively negligible [8]. The energy consumed by a member node in a single round is given by (5).

$$E_c^{member}(i) = E_{TX}(l, \text{dist}(i, CH_i)) \quad (5)$$

where CH_i is the CH of node i and $\text{dist}(\cdot)$ is the distance between any two nodes. On the other hand, if a node is CH, the amount of energy it consumes is approximated by (6).

$$E_c^{CH}(j) = E_{RX}(l \times CM_j) + E_{DA}(l \times (CM_j + 1)) + E_{TX}(l, \text{dist}(j, \text{next}(j))) \quad (6)$$

where CM_j is the number of members in cluster j , and $\text{next}(j)$ denotes the next hop for node j according to the current routing. The next hop might be a relay node or the BS. This formulation corresponds to the energy required for receiving l -bit messages from all cluster members, aggregating the data, and sending one l -bit message to the next hop.

Similarly, the energy consumed by a relay is given by (7), which corresponds to receiving the l -bit message and sending it as-is to the next hop.

$$E_c^{relay}(r) = E_{RX}(l) + E_{TX}(l, \text{dist}(r, \text{next}(r))) \quad (7)$$

C. Problem Definition

Here, we propose to maximize the network lifetime (FND) by selecting optimum CHs and relays in different rounds. The member nodes will join the closest CH to form clusters. Therefore, the problem is to choose CHs and relays such that FND is maximized. We define the function $\mathcal{F}(\cdot)$, given by (8a), which represents a formulation for FND.

$$\text{Maximize}_{\mathcal{C}, \mathcal{R}} \mathcal{F}(\mathcal{C}, \mathcal{R}, \mathcal{Z}) = \min_{n \in \mathcal{N}} (t_n^d) \quad (8a)$$

$$\text{s. t.} \quad \mathcal{C} \in \mathcal{X}^c \quad (8b)$$

$$\mathcal{R} \in \mathcal{X}^r \quad (8c)$$

In this formulation, $\mathcal{C}$ is the CH selection matrix such that $C_{(t,n)}$ is 1 if node n is CH at round t and zero otherwise, and $\mathcal{R}$ is the relay selection matrix such that $R_{(t,n)}$ is the relay of CH n at round t . Also, the problem parameters, such as location and residual energy of nodes, are contained in $\mathcal{Z}$. $\mathcal{X}^c$ and $\mathcal{X}^r$ are feasible sets for clustering and routing, respectively. Finally, t_n^d , formulated by (9), represents the round at which node n dies.

$$t_n^d = \min\{t | t = 0, \dots, T \wedge e_n^t \leq 0\} \quad (9)$$

where e_n^t is the residual energy of node n at round t , and T is the last round.

IV. PROPOSED SOLUTION

Achieving an exact solution for problem (8) is not feasible for large networks [9]. However, it is possible to find CH configurations that optimize the objective function (8a) within smaller networks. For instance, to find three optimum CHs for a network comprising 60 nodes, $\binom{60}{3} = 34,220$ solutions must be examined, a task feasible within a practical time. Our proposed solution is to train a GNN by generating a training dataset consisting of small networks solved by exhaustive search, then use the trained GNN on larger network topologies.

A. Graph Representation

To utilize GNN for this task, we first need to construct a graph representation of the problem. In this graph, each node corresponds to a sensor node in the WSN, and an edge exists between two nodes if their distance is below a defined threshold (d_t). This construction results in a neighborhood graph that captures the connectivity of the WSN. Furthermore, each node is associated with a 2-dimensional feature vector containing its residual energy and the energy the node would consume if selected as the CH. Also, the label of a node indicates its suitability for becoming a CH. We call this value CH rank and denote it by r_n^{CH} for node n , the lower the rank, the more suitable a node is to become CH. Similarly, each edge (u, v) is associated with a feature vector containing the amount of energy that node v consumes if node u is CH, denoted as $e_v^{ch=u}$, which is equal to the energy required to send one message from v to u as formulated by (5). Fig. 1 shows a sample of this graph. With this graph representation established, the objective of the GNN is to predict node labels (CH rank) by leveraging the graph structure along with the feature vectors of both nodes and edges. Node representation learning is a suitable GNN framework to tackle this task.

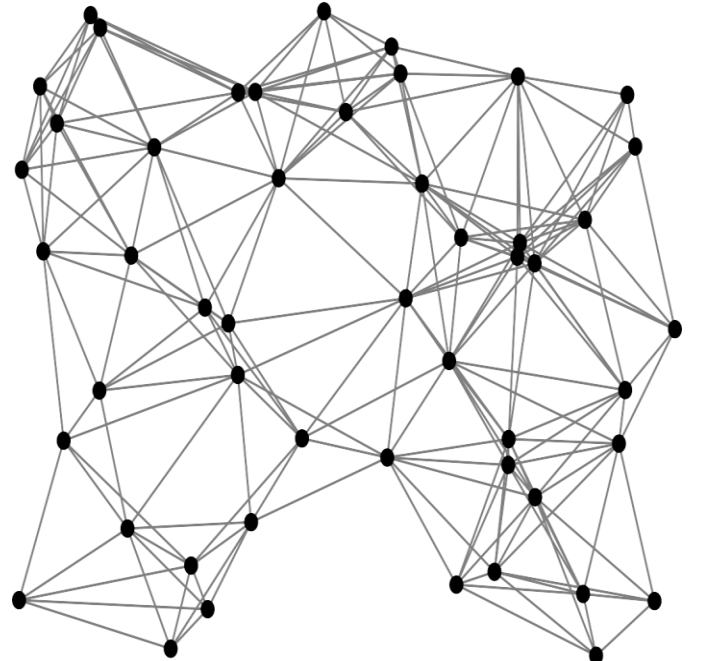


Fig. 1. The neighborhood graph on a sample WSN with 50 nodes.

Node Representation Learning

Node representation learning aims to extract effective and compact node embeddings by training a neural network that maps the input graph (features of nodes and edges, along with the graph topology) to the node labels. A common framework for GNN is the Message Passing Neural Network (MPNN). In the MPNN framework, multiple layers of message passing are stacked on top of each other [45]. The basic idea of MPNN is that relevant information about the nodes, edges, and the entire graph can be encoded in fixed-size vectors [35]. Denote $\mathbf{h}_k^{(l)}$ as the hidden state of the k -th node in the l -th layer. Starting from the initial node representation, each layer consists of the following three stages as formulated in (10) and depicted in Fig. 2:

- **Message:** For each edge, the hidden state of nodes at the two ends of the edge is fed to an MSG function, and the output is called the message from node v to u , denoted as $m_{vu}^{(l)}$. This step is formulated by (10a).
- **Aggregate:** Given by (10b), each node aggregates the messages from all incoming edges by using an AGG function. The aggregation is denoted by $a_v^{(l)}$.
- **Combine:** Finally, applying a UPT function, each node combines the aggregated messages with its own hidden state, given by (10c).

$$\text{MSG: } m_{vu}^{(l)} \leftarrow \text{MSG}(\mathbf{h}_v^{(l-1)}, \mathbf{h}_u^{(l-1)}), \forall (v, u) \in \mathcal{E} \quad (10a)$$

$$\text{AGG: } a_v^{(l)} \leftarrow \text{AGG}(\{m_{vu}^{(l)} | u \in \mathcal{N}_v\}, \forall v \in \mathcal{N}) \quad (10b)$$

$$\text{UPT: } \mathbf{h}_v^{(l)} \leftarrow \text{UPT}(\mathbf{h}_v^{(l-1)}, a_v^{(l)}), \forall v \in \mathcal{V} \quad (10c)$$

where $\mathcal{N}_v$ is the set of neighbors of node v .

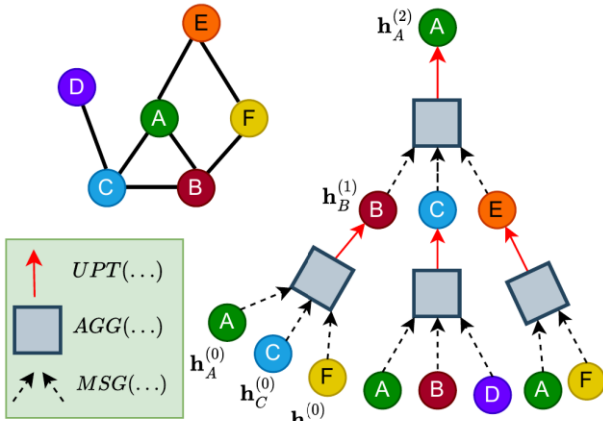


Fig. 2. The process of obtaining a latent node representation in the MPNN framework [36].

B. Proposed GNN Structure

In the proposed GNN, three layers of the Edge Convolutional Graph Neural Network (ECGNN) [39] are stacked on top of each other. ECGNN is a GNN architecture designed to be used on weighted graphs [39] and uses the following formula to update the hidden state of graph nodes.

$$\mathbf{m}_{uv}^{(l)} = \text{MLP}(\mathbf{h}_v^{(l-1)} \cdot \mathbf{f}_{uv} \cdot \mathbf{h}_u^{(l-1)}, \mathcal{W}_{MLP}^{(l)}), \forall (v, u) \in \mathcal{E} \quad (11a)$$

$$\mathbf{h}_v^{(l)} = \text{mean}(\{\mathbf{m}_{vu}^{(l)} | u \in \mathcal{N}_v\}, \forall v \in \mathcal{V}) \quad (11b)$$

Where $\mathbf{f}_{uv}$ is the edge feature vector of the edge between u and v , $\cdot$ is the concatenation operator, MLP is a two-layer fully connected neural network with ReLU activation function, $\mathcal{W}_{MLP}^{(l)}$ is the learnable weight matrix of the MLP at later l , and mean is the element-wise mean function. Eq. (11a) and (11b) correspond to the message and aggregate stages of the MPNN framework, respectively, while the update stage is the identity function. Additionally, a readout network consisting of a two-

Algorithm 1 Dataset Generation Algorithm

Input: n

Output: set of P training samples D

1: **for** $i = 0$ to P **do**

2: $\mathcal{N} =$ set of N_t nodes with random position and energy.

3: $\mathcal{C}_{ch} = \mathcal{N}$

4: $r = 0$

5: **while** $\mathcal{C}_{ch} \neq \emptyset$ **do**

6: Select relays as described in section 4.F

7: $\mathcal{C} =$ set of cluster heads from $\mathcal{C}_{ch}$ that optimize (8)

8: $r_n^{ch} = r, \forall n \in \mathcal{C}$

9: remove selected CHs from: $\mathcal{C}_{ch}$

10: $r = r + 1$

11: **end while**

12: **end for**

12: **return** D

layer MLP with ReLU activation function is used to map the hidden state of nodes in the final layer to the CH ranks. This encoder can also be viewed as the update stage of the last layer. Table I lists the numeric details about the structure of the GNN.

TABLE I
Numeric Details of the GNN

Parameter	Value
Number of node features	2
Number of edge features	1
Number of node hidden features at layer 1	16
Number of node hidden features at layer 2	32
Number of node hidden features at layer 3	64
Number of Encoder MLP layers	3
Encoder MLP structure	64 to 32 to 1

C. Training the GNN

generated a synthetic dataset to train the GNN. As mentioned before, an exact solution for problem (8a) can be found for small WSNs in a feasible time. The training dataset contains 60-node WSNs where the position and residual energy of the nodes are drawn randomly from uniform distributions. For each of these samples, all the permutations of selecting 3 CHs from the set of candidates were examined to find the one that maximizes the objective function. Initially, all nodes are considered CH candidates. The best CH are given the rank value of 0 and

removed from the candidate set. Then the process repeats for the remaining candidates, while in each iteration the rank value is increased by one. This results in assigning a rank value to every node. Algorithm 1 shows this process in a formal way.

The generated training samples are converted to the graph representation and are used to train the GNN based on the mean absolute error (MAE) loss function.

D. Application

To employ the trained GNN to select CHs, the network in its current state should be converted to a graph representation. To calculate the energy a node consumes when selected as CH, the next hop should be known. Therefore, in our proposed protocol, the routing precedes the CH selection. The trained GNN can predict optimal CH ranks when a suitable graph is fed into it. First, the graph is constructed based on the description in Section 4. A, then it is fed into the GNN. The GNN output is the CH ranks. We select the top k nodes with the lowest ranks as CH. This procedure is detailed in Algorithm 2. To assign CH to relays, each CH will be connected to the relay corresponding to the cluster center closest to it.

Theorem 1. The time complexity of the GNN-based CH

Algorithm 2 GNN-based CH selection

```

1:  $\mathbf{h}_{v=n}^{(0)} = \langle e_n^t, e_n^{ch=n} \rangle \forall v \in \mathcal{V}$ 
2:  $\mathbf{f}_{uv} = (e_u^{ch=v}) \forall \langle u, v \rangle \in \mathcal{E}$ 
3: for  $l \in \{1, 2\}$  do
4:   for all  $\langle u, v \rangle \in \mathcal{E}$  do
5:      $m_{uv} = \text{MLP}(\mathbf{h}_v^{l-1} \cdot \mathbf{f}_{uv} \cdot \mathbf{h}_v^{(l-1)}, \mathcal{W}_{MLP}^{(l)})$ 
6:   end for
7:   for all  $v \in \mathcal{V}$  do
8:      $\mathbf{h}_v^{(l)} = \text{mean}(\{\mathbf{m}_{vu}^{(l)} | u \in \mathcal{N}_v\})$ 
9:   end for
10: end for
11: for all  $v \in \mathcal{V}$  do
12:    $r_{n=v}^{CH} = \text{MLP}_{\text{READOUT}}(\mathbf{h}_v^{(l=2)})$ 
13: end for
14:  $\mathcal{C}_{(t,n)} = 1$  for  $k$  nodes with least rank  $r_n^{CH}$ , and zero for other nodes.
15: return  $\mathcal{C}_{(t,n)} \forall n \in \mathcal{N}$ 

```

selection method, as shown in Algorithm 2, is $O(n^2)$, where n is the number of nodes in the network.

Proof. We consider the time complexity of the algorithm by going through the pseudo code. At line 1, $\mathbf{h}_v^{(0)}$ is constructed for each graph node, and as each graph node corresponds to a node in the network, this line is completed in $O(n)$. In line 2, edge feature vectors are initialized. The number of edges in the graph is in $O(n^2)$. At line 3, a *for* loop that repeats 2 times starts. In line 4, a nested loop that repeats for each edge of the graph starts. In every iteration of this loop, the message $\mathbf{m}_{uv}$ is calculated by running an MLP. As the size of MLP layers and their inner weights in each layer are fixed and do not grow with the size of the problem, it can be regarded as a constant time computation. Line 7 declares another loop that repeats for each node. Inside the loop, all messages from neighbors of a node

are aggregated, and as the number of neighbors of a node is in $O(n)$, the time complexity of line 8 is $O(n)$. This is multiplied by the repeat count of the enclosing loop. Therefore, the total time complexity of the loop at line 7 is $O(n^2)$. Line 11 is a similar loop that runs the readout MLP for each node, and as the MLP has constant runtime, the total time complexity of the loop at line 11 is $O(n)$. In line 14, the top k nodes among n nodes are found. As finding the least element of a list is in $O(n)$, repeating this k times is in $O(nk)$, and as n is an upper-bound for k , $O(nk)$ can be replaced by $O(n^2)$. Therefore, the total time complexity is (12).

$$O(n) + O(n^2) + 2 \times (O(n^2) \times O(1) + O(n) \times O(n)) + O(n) \times O(1) + O(n^2) = O(n) + O(n^2) + 2 \times O(2n^2) + O(n) + O(n^2) = O(6n^2 + 2n) = O(n^2) \quad (12)$$

E. Relay Selection

The proposed GNN-based CH selection requires estimating nodes' energy consumption; therefore, the next hop for CH candidates should be known before CH selection. To mitigate this, we choose a grid-based division of the sensing field and designate the centers of grid cells as placeholders for CHs. Afterward, we chose an exclusive relay chain from this point to the BS. Routing in this context resembles the approach described in [36,9,8]. This strategy prevents relays from being overloaded, resulting in more evenly distributed energy consumption and mitigating hotspot issues. Starting from the furthest cluster centers, a relay is selected for the cluster center, and repeatedly, the next hop is chosen for the relay until the BS is reached. This procedure is then repeated for each cluster.

To determine the next hop for a given node or cluster center, a set of relay candidates is formed by considering the nodes that are not already a relay for another cluster and have more residual energy than the average. To choose the next hop, the nodes in the relay candidate set are examined, and the closest node that satisfies the following conditions is selected as the relay. Firstly, the relay should be closer to the BS than the current node. Secondly, the relay should not be closer than a threshold D_{min}^r to the current node. This constraint is to avoid selecting very close relays, which results in increased hop count and delay. Finally, the detour via relay to BS should not be much longer than the direct route from the current node to the BS. The first, second, and final conditions are given by (13), (14), and (15), respectively.

$$\text{dist}(r, BS) < \text{dist}(n, BS) \quad (13)$$

$$\text{dist}(n, r) < D_{min}^r \quad (14)$$

$$\text{dist}(n, r) + \text{dist}(r, BS) < \text{dist}(n, BS) + D_{max}^r \quad (15)$$

Where D_{max}^r is a threshold controlling the maximum allowed difference between the direct node to BS distance and node to relay plus relay to BS distance. The complete routing procedure is given in Algorithm 3.

V. PERFORMANCE EVALUATION AND DISCUSSION

We conducted a series of simulations to evaluate the

Algorithm 3 The routing algorithm

Input: Location of nodes and cluster centers

Output: $R_{(t,:)}$

```

1:  $\mathcal{C}$  = clusters sorted by distance from center point to BS in descending order
2:  $\mathcal{R}$  = set of nodes with higher energy than average.
3: for all  $C \in \mathcal{C}$ 
4:    $n$  = center point of cluster  $C$ 
5:   while  $n \neq BS$  do
6:      $\mathcal{R}'$  = set of relay candidate that satisfy formula (13) to (15)
7:     if  $\mathcal{R}' = \emptyset$  then
8:        $r = BS$ 
9:     else
10:       $r$  = closest relay candidate in  $\mathcal{R}'$  to  $n$ 
11:    end if
12:     $R_{(t,n)} = r$ 
13:    Remove  $r$  from  $\mathcal{R}$ 
14:     $n = r$ 
15:  end while
16: end for
17: return  $R_{(t,:)}$ 

```

effectiveness of our proposed protocol and compare it against several recent protocols to establish benchmarks. These simulations were carried out in Python. The GNN was implemented using PyTorch and the Deep Graph Library (DGL) [43], a comprehensive library specifically designed for deep learning on graphs. During the training phase, we employed the Adam optimizer [20], a popular choice in the deep learning community for its adaptive learning rate optimization capabilities. This optimizer dynamically adjusts learning rates for each parameter, ensuring efficient convergence. The convergence of the loss function is depicted in Fig. 3.

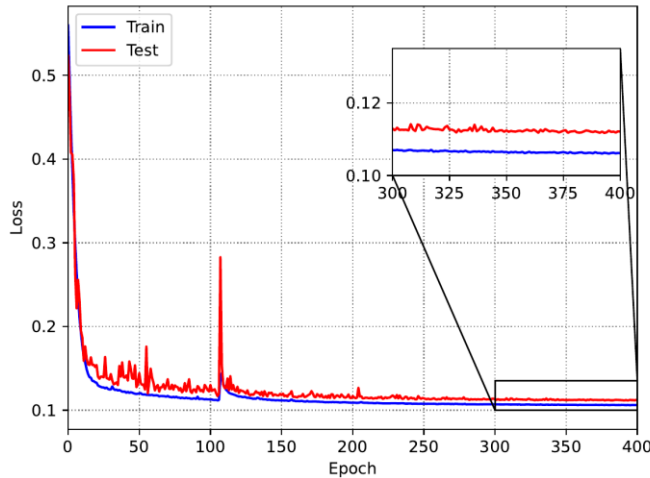


Fig. 3. Loss per epoch for the train and test datasets.

A. Metrics

We have used several common metrics to assess the performance of our proposed method and compare it with other protocols. These are listed below:

- **Alive nodes per round:** The number of alive nodes in each round until all nodes are dead. FND, HND, and LND are extracted from this metric and presented separately.
- **Throughput:** The total amount of data that is delivered to the BS; this is a function of alive nodes per round, as each sensing node sends one fixed-size packet for each round in which it is alive.
- **Radius of the largest coverage hole:** We previously proposed this metric in [36]. This is defined as the largest distance between a dead node and its closest alive node. The radius of the largest coverage hole is intended to quantify how well a protocol distributes energy consumption throughout the sensing field and its ability to avoid the hotspot problem.

B. Benchmarks

We have employed several cutting-edge protocols to benchmark our work, including:

LEACH (2000) [17]: The most common and earliest WSN energy-efficient clustering protocol. In LEACH, CH are selected in a distributed fashion based on randomly chosen numbers in each node. It does not propose a solution for routing and relies on direct CH to BS transmission, which has a drastic negative effect on its energy efficiency [8]. However, it is widely used as a baseline to assess the performance of other protocols.

CSMC (2023) [8]: A recently proposed centralized clustering and multi-hop routing protocol utilizing the Salp Swarm Algorithm. Its routing scheme shares many similarities with the current work. An exclusive chain of relays is selected for each CH based on a heuristic algorithm to mitigate the hotspot problem.

GECP (2024) [36]: Our previous work was the GNN-based equal clustering protocol. This work aims to equalize the member count across static clusters to better balance energy consumption for CHs. It also used a routing similar to [8].

TCSAA (2022) [41]: Aims to address the main limitation of the LEACH protocol, i.e., the multi-hop routing. Clustering in this protocol is similar to LEACH, but instead of direct transmission from CHs to BS, it devises multi-hop routing for both inter-cluster and intra-cluster transmissions. The routing problem is solved via a Taylor-based Cat Salp Swarm Algorithm.

ESCVAD (2022) [25]: Forms static clusters of equal size based on Voronoi division and a heuristic post processing. The equalization of clusters is aimed at as a means of balancing energy consumption among CHs. The multi-hop routing proposed in this protocol is executed in a distributed manner using a heuristic weighting mechanism that accounts for distance and energy.

TABLE II
Parameters of Scenarios

	Area	BS location	Nodes
Scenario 1	200m*200m	(100,100)	100
Scenario 2	300m*300m	(150,500)	500

C. Simulation Setup

Two scenarios are devised to assess the performance of the proposed protocol and to compare it with benchmarks. The scenarios differ in terms of the dimensions of the sensing field and the number of sensor nodes. These parameters are listed in Table II, while Table III lists the details of other parameters used in the simulation that remain unchanged across the two scenarios.

TABLE III
Common Parameters Used in Simulation

Parameter	Value
Initial Energy	0.5J
Percentage of CHs	5%
Packet length	200bits
Message Length	4000bits
Max hop	5
m_{min}	0.8
m_{max}	1.5
E_{elec}	50nJ/bit
E_{fs}	10pJ/bit/m ²
E_{mp}	0.0013pJ/bit/m ²
E_{DA}	5nJ/bit/signal
d_0	87m
d_t for scenario 1	94m
d_t for scenario 2	141m
GNN Training epochs	400
GNN initial learning rate	10^{-3}
$ N $ for train and test	60
Number of samples in dataset, P	2000

D. Computation Overhead

To provide insight into the computational efficiency of the proposed method, we analyze the overhead associated with data generation and model training. Dataset construction involves an exhaustive search to determine the optimal clustering configuration (e.g., selecting 3 optimal Cluster Heads among 60 nodes, corresponding to approximately 35,000 combinations). This process was executed using a straightforward Python script. By leveraging parallel processing across 4 CPU cores, generating 500 training samples required approximately 40 minutes. Furthermore, the GNN training phase, conducted over 400 epochs with an average duration of 30 seconds per epoch, was completed in less than 3 hours and 20 minutes. Consequently, the total time required for both data generation and model training was under 4 hours on standard CPU-based hardware. This demonstrates that the proposed approach is computationally efficient, and the training overhead could be further reduced significantly by utilizing GPU-accelerated environments.

To evaluate the feasibility of the proposed GNN model for real-time deployment, we analyze the computational requirements of the inference phase at the BS. The proposed architecture is a lightweight, shallow GNN with the following configuration: 2 node features and 1 edge feature with 3 message-passing layers with hidden dimensions of 16, 32, and 64, respectively, and finally, a 3-layer MLP with a structure of 64 to 32 to 1. Given this architecture, the model contains approximately 9,500 trainable parameters. Executing this model involves a small number of floating-point operations (FLOPs), which can be processed in a few milliseconds on standard hardware. Since the BS is typically assumed to be a resource-rich node with sufficient processing power and an uninterrupted power supply, the computational overhead of this lightweight GNN is negligible. Consequently, the model is highly suitable for practical WSN scenarios, as the real-time inference latency is orders of magnitude smaller than the communication intervals of the network.

E. Results

The number of alive nodes per round for the first scenario is depicted in Fig. 4. Our protocol led to an almost equal FND compared to GECP, but managed to overcome this limitation by showing a 32% higher HND compared to the best protocol in terms of HND and 33% in comparison with GECP. As for FDN, we also see a 19% increase compared to GECP, which is the best-performing protocol. The total throughput of our method for scenario 1 is 54.44MB, which is 25% higher than that of the runner-up protocol GECP. Fig. 6 plots the total throughput of all protocols for the first scenario.

Our protocol has several features that aid it in achieving better results in comparison to benchmarks, including the adaptation of FND as the objective function and using GNN to find a solution for maximizing FND. Also, using multi-hop routing and selecting an exclusive route for each cluster to the BS ensures balanced energy consumption among members, CHs, and relays. Finally, omitting the setup phase in some rounds in which the previously selected CHs and relays are still good enough is another factor for the energy efficiency of our protocol.

Balancing energy consumption between nodes also limits the growth of coverage holes. The largest coverage hole metric is designed to assess the ability of a protocol to spatially distribute energy consumption across the sensing field to prevent loss of coverage over large areas. Fig. 5 plots the radius of the largest coverage hole over time for the proposed and benchmark protocols. The largest coverage hole for the competitive protocol GECP grows with a higher slope until around round 850 at 200m and grows more slowly afterward. In comparison, our protocol limits the growth of coverage holes throughout the network's lifetime. Preventing the use of near-death nodes for higher energy-demanding tasks via designating the FND and the objective function will result in prolonging the lifetime of a node as much as possible, and is reflected in these results.

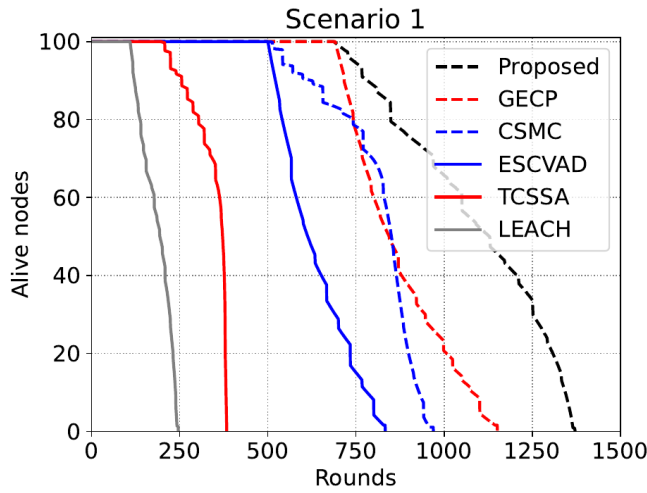


Fig. 4. Alive nodes per round for scenario 1.

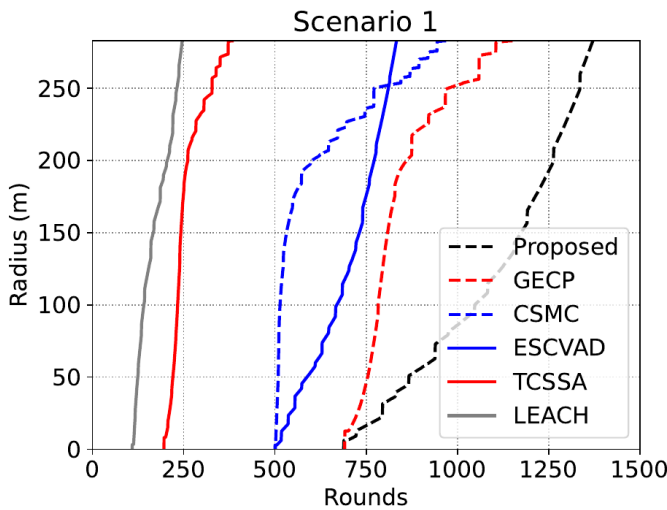


Fig. 5. Radius of largest coverage hole per round for scenario 1.

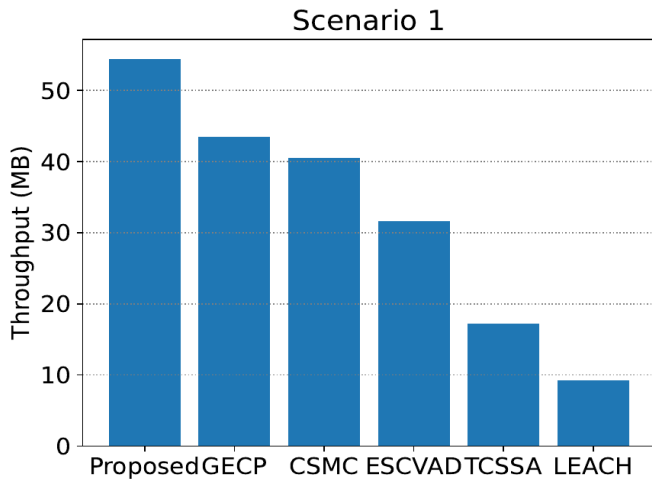


Fig. 6. Total throughput scenario 1.

The second scenario tests the ability of the protocols for larger WSNs. It consists of 500 sensor nodes in a larger field of 300m*300m, while the BS is located outside the field at a far distance. In this scenario, the routing plays a more vital role, and the emergence of the hotspot problem is more prevalent as the nodes close to the BS are the most spatially suitable relays.

Fig. 7 shows the alive nodes per round for the second scenario. In this scenario, the proposed protocol yields better results in FND, HDN, and LND, signifying the suitability of this design for larger networks in comparison with the benchmarks.

When analyzing the results more carefully, a complementary trend can be observed in the FND metric. In Scenario 1, FND improvement is negligible (0%), whereas in Scenario 2, FND is improved by approximately 52%. In other words, the smaller LND gain in Scenario 2 is compensated by a significantly stronger improvement in FND. The reason lies in the characteristics of the two scenarios. Scenario 1 corresponds to a smaller and denser network, which can be considered an “easier” setting. During the stable period (from network start until FND), all nodes are alive, and energy consumption is relatively balanced across methods, hence the similar FND performance. The main differentiation appears after the first node dies, where node depletion accelerates, and LND differences become more visible.

In contrast, Scenario 2 involves a larger deployment area, leading to longer communication distances and increased transmission energy. In such conditions, maintaining a strong FND becomes more challenging. The proposed method significantly improves FND (52%), thereby delaying the onset of network degradation more effectively. Although the LND improvement is modest, the throughput results ($\approx 15\%$ higher than CSMC) indicate that more nodes remain functional for a longer period. Therefore, the method prioritizes maintaining a longer stable operational phase, which is often more critical in practical WSN applications than merely delaying the final node death.

The growth of the coverage hole in the second scenario is plotted in Fig. 8. We observe a sharper slope for the growth of the largest coverage hole in comparison with the first scenario, as the network field is much larger this time. However, the proposed protocol also outperforms the benchmarks in this setup. Overall, keeping more nodes alive for a longer time will enable the transmission of more data messages containing sensed parameters and finally a higher throughput, as plotted in Fig. 9. Finally, the summary of the results for both scenarios is presented in Table IV.

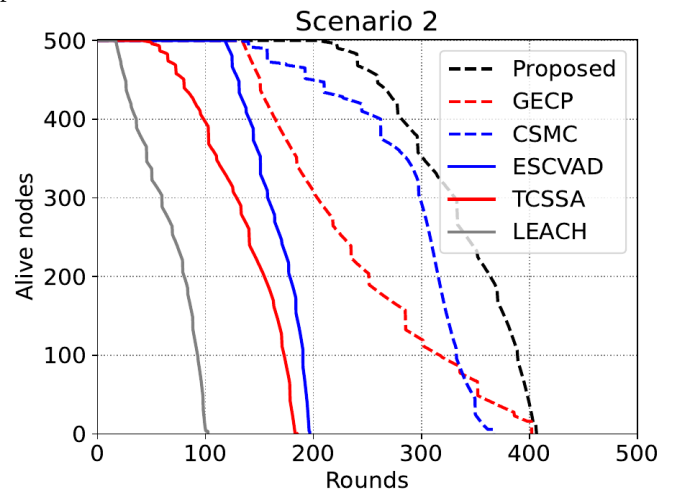


Fig. 7. Alive nodes per round for scenario 2.

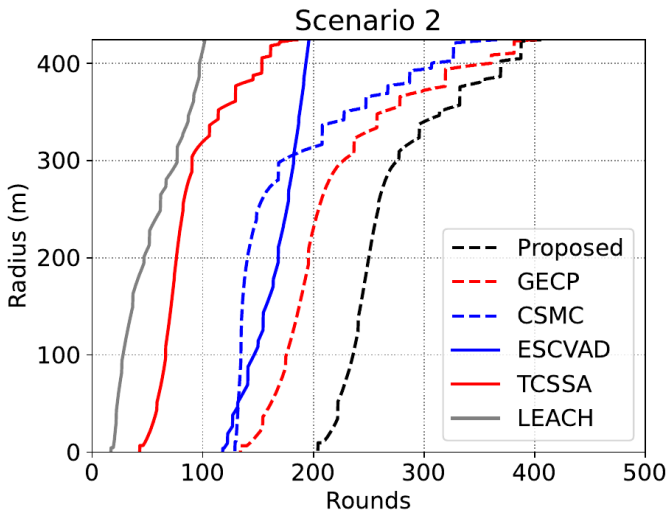


Fig. 8. Radius of largest coverage hole per round for scenario 2.

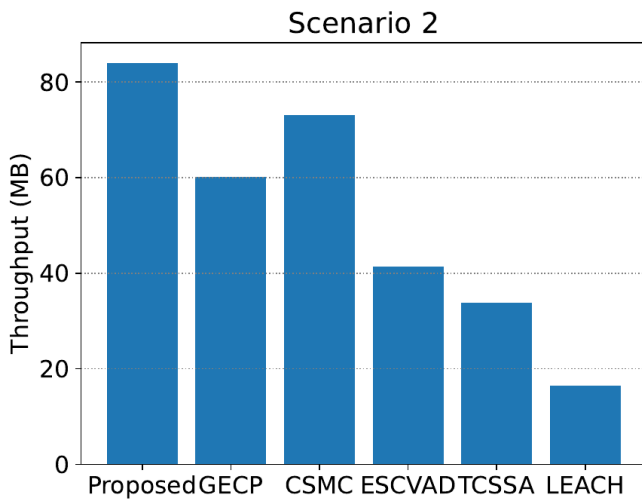


Fig. 9. Total throughput scenario 2.

TABLE IV
Common Comparison of Lifetime Metrics

	Scenario 1			Scenario 2		
	FND	HND	LND	FND	HND	LND
Proposed	691	844	1151	134	227	402
GECP	499	850	969	129	308	366
CSMC	500	613	833	118	169	196
ESCVAD	196	371	383	43	140	185
TCSSA	109	194	246	17	70	102
Proposed Improved	688	1129	1371	204	342	406
	0%	32%	19%	52%	11%	1%

VI. LIMITATIONS AND FUTURE WORK

This study has several limitations that should be acknowledged. First, although the proposed method is compared with several representative clustering protocols, it is not evaluated against some of the most recent deep learning and GNN-based WSN protocols. Faithfully reproducing those approaches requires implementing their full training pipelines and protocol assumptions, which were beyond the scope of the present study.

Future work will include a more comprehensive benchmarking against recent learning-based protocols.

Second, to construct the input features for the GNN, the routing procedure must be executed before the CH selection stage. To break this circular dependency between routing and CH selection, the routing algorithm assumes that candidate CHs are located at the centers of grid cells when estimating relay paths. While this assumption provides a practical way to generate consistent energy estimates, it may introduce a degree of sub-optimality in the estimated routing cost due to the difference between proxy and actual CH locations.

Third, the system model follows a commonly adopted WSN assumption of static sensor nodes and relatively stable environmental conditions. Factors such as node mobility, sudden node failures, and severe environmental effects (e.g., strong signal attenuation due to obstacles or weather) were not explicitly modeled in this work. Although this assumption is consistent with many studies in the literature, evaluating the proposed method under more dynamic and harsh conditions remains an important direction for further research.

Several promising directions exist for future work. One potential extension is to integrate reinforcement learning within the GNN framework so that the protocol can learn adaptive clustering strategies directly from interaction with the environment, eliminating the need for labels generated via exhaustive search. Another direction is the development of a neural network-based routing mechanism, which could provide more accurate relay selection and offer a more principled way of addressing the routing-clustering dependency while further improving the overall efficiency of the network.

VII. CONCLUSION

We introduced a novel method for selecting energy-efficient CHs in WSNs to extend network lifespan. To address this challenge, we employed a GNN trained using data from small-scale WSNs for which obtaining exact solutions is feasible. Subsequently, the trained GNN was utilized to predict the suitability rank of nodes for CH designation. Furthermore, to achieve more balanced energy consumption, we devised a multi-hop routing scheme to distribute communication tasks among all nodes as evenly as possible. By conducting simulations and comparing them with both recently proposed and classic WSN clustering approaches, we demonstrated the effectiveness of our specific design choices. The success of GNNs in addressing this particular challenge in WSNs highlights their promising potential to tackle similar issues within wireless networks and other graph-based combinatorial optimization problems in general.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICTS OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] A. M. K. Abdulzahra, A. K. M. Al-Qurabat, and S. A. Abdulzahra, "Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods," *Internet of Things*, vol. 22, Art. no. 100765, Jul. 2023.
- [2] M. Adnan, L. Yang, T. Ahmad, and Y. Tao, "An unequally clustered multi-hop routing protocol based on fuzzy logic for wireless sensor networks," *IEEE Access*, vol. 9, pp. 38531–38545, 2021.
- [3] H. Al-Tous and I. Barhumi, "Reinforcement learning framework for delay-sensitive energy harvesting wireless sensor networks," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 7103–7113, Mar. 2021.
- [4] H. Ali, U. U. Tariq, M. Hussain, L. Lu, J. Panneerselvam, and X. Zhai, "ARSH-FATI: A novel metaheuristic for cluster head selection in wireless sensor networks," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2386–2397, 2021.
- [5] G. Arya, A. Bagwari, and D. S. Chauhan, "Performance analysis of deep learning-based routing protocol for efficient data transmission in 5G WSN communication," *IEEE Access*, vol. 10, pp. 9340–9356, 2022.
- [6] R. Bai, R. Huang, L. Zheng, Y. Chen, and Y. Qin, "Structure-enhanced deep clustering network via a weighted neighborhood auto-encoder," *Neural Networks*, vol. 155, pp. 144–154, Nov. 2022.
- [7] R.-S. Chang and C.-J. Kuo, "An energy-efficient routing mechanism for wireless sensor networks," in *Proc. 20th Int. Conf. Advanced Information Networking and Applications (AINA)*, vol. 2, Apr. 2006, pp. 5 pp.–.
- [8] S. M. M. H. Daneshvar and S. M. Mazinani, "On the best fitness function for WSN lifetime maximization: A solution based on a modified salp swarm algorithm for centralized clustering and routing," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.
- [9] S. M. M. H. Daneshvar, P. A. A. Mohajer, and S. M. Mazinani, "Energy-efficient routing in WSN: A centralized cluster-based approach via grey wolf optimizer," *IEEE Access*, vol. 7, pp. 170019–170031, 2019.
- [10] K. Eledlebi, D. Ruta, H. Hildmann, F. Saffre, Y. Al Hammadi, and A. F. Isakov, "Coverage and energy analysis of mobile sensor nodes in obstructed noisy indoor environment: A Voronoi approach," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2745–2760, 2022.
- [11] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550–586, 2017.
- [12] W. Guo, Y. Wang, Y. Gan, and T. Lu, "Energy-efficient and reliable routing in wireless body area networks based on reinforcement learning and fuzzy logic," *Wireless Networks*, vol. 28, no. 6, pp. 2669–2693, Aug. 2022.
- [13] F. Hajjej, M. Hamdi, R. Ejbali, and M. Zaied, "A distributed coverage hole recovery approach based on reinforcement learning for wireless sensor networks," *Ad Hoc Networks*, vol. 101, Art. no. 102082, Apr. 2020.
- [14] A. A.-H. Hassan, W. M. Shah, A.-H. H. Habbab, M. F. I. Othman, and M. N. Al-Mhiqani, "An improved energy-efficient clustering protocol to prolong the lifetime of WSN-based IoT," *IEEE Access*, vol. 8, pp. 200500–200517, 2020.
- [15] S. He, S. Xiong, Y. Ou, J. Zhang, J. Wang, Y. Huang, and Y. Zhang, "An overview on the application of graph neural networks in wireless networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2547–2565, 2021.
- [16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
- [17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annual Hawaii Int. Conf. System Sciences (HICSS)*. IEEE Computer Society, 2000.
- [18] T. Kaur and D. Kumar, "Particle swarm optimization-based unequal and fault-tolerant clustering protocol for wireless sensor networks," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4614–4622, Jun. 2018.
- [19] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, "Communication protocols for wireless sensor networks: A survey and comparison," *Heliyon*, vol. 5, no. 5, Art. no. e01591, May 2019.
- [20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2014.
- [21] M. Krishnan and Y. Lim, "Reinforcement learning-based dynamic routing using mobile sink for data collection in WSNs and IoT applications," *Journal of Network and Computer Applications*, vol. 194, Art. no. 103223, Nov. 2021.
- [22] V. Lima, M. Eisen, K. Gatsis, and A. Ribeiro, "Resource allocation in large-scale wireless control systems with graph neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2634–2641, 2020.
- [23] D. Lin and Q. Wang, "An energy-efficient clustering algorithm combining game theory and a dual-cluster-head mechanism for WSNs," *IEEE Access*, vol. 7, pp. 49894–49905, 2019.
- [24] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerospace Conf.*, vol. 3, 2002, pp. 3–3.
- [25] N. Ma, H. Zhang, H. Hu, and Y. Qin, "ESCVAD: An energy-saving routing protocol based on Voronoi adaptive clustering for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 9071–9085, 2022.
- [26] N. H. Mak and W. K. G. Seah, "How long is the lifetime of a wireless sensor network?" in *Proc. 2009 Int. Conf. Advanced Information Networking and Applications*, 2009, pp. 763–770.
- [27] N. Mittal, U. Singh, R. Salgotra, and B. S. Soh, "A Boolean spider monkey optimization-based energy-efficient clustering approach for WSNs," *Wireless Networks*, vol. 24, no. 6, pp. 2093–2109, Feb. 2017.
- [28] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Adaptive wireless power allocation with graph neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [29] A. S. Nandan, S. Singh, and L. K. Awasthi, "An efficient cluster head election based on optimized genetic algorithm for movable sinks in IoT-enabled HWSNs," *Applied Soft Computing*, vol. 107, Art. no. 107318, Aug. 2021.
- [30] A. S. Nandan, S. Singh, R. Kumar, and N. Kumar, "An optimized genetic algorithm for cluster head election based on movable sinks and adjustable sensing ranges in IoT-based HWSNs," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5027–5039, 2022.
- [31] T. S. Panag and J. S. Dhillon, "Dual head static clustering algorithm for wireless sensor networks," *AEU—International Journal of Electronics and Communications*, vol. 88, pp. 148–156, 2018.
- [32] Y. B. Patil and S. Kohle, "A high-scalability and low-latency cluster-based routing protocol in time-sensitive WSNs using genetic algorithm," *Measurement: Sensors*, Art. no. 100941, Nov. 2023.
- [33] P. C. S. Rao, P. K. Jana, and H. Banka, "A particle swarm optimization-based energy-efficient cluster head selection algorithm for wireless sensor networks," *Wireless Networks*, vol. 23, no. 7, pp. 2005–2020, Apr. 2016.
- [34] P. Rawat and S. Chauhan, "Particle swarm optimization-based sleep scheduling and clustering protocol in wireless sensor network," *Peer-to-Peer Networking and Applications*, vol. 15, no. 3, pp. 1417–1436, May 2022.
- [35] K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260–2270, Oct. 2020.
- [36] M. Saadati, S. M. Mazinani, A. A. Khazaei, and S. J. S. M. Chabok, "Energy-efficient clustering for dense wireless sensor network by applying graph neural networks with coverage metrics," *Ad Hoc Networks*, vol. 156, Art. no. 103432, 2024.
- [37] C. Savaglio, P. Pace, G. Aloï, A. Liotta, and G. Fortino, "Lightweight reinforcement learning for energy-efficient communications in wireless sensor networks," *IEEE Access*, vol. 7, pp. 29355–29364, 2019.
- [38] I. K. Shah, T. Maity, Y. S. Dohare, D. Tyagi, D. Rathore, and D. S. Yadav, "ICIC: A dual-mode intra-cluster and inter-cluster energy minimization approach for multihop WSN," *IEEE Access*, vol. 10, pp. 70581–70594, 2022.
- [39] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.
- [40] B.-N. Trinh, L. Murphy, and G.-M. Muntean, "A reinforcement learning-based duty cycle adjustment technique in wireless multimedia sensor networks," *IEEE Access*, vol. 8, pp. 58774–58787, 2020.
- [41] A. Vinitha, M. S. S. Rukmini, and Dhirajsunehra, "Secure and energy-aware multi-hop routing protocol in WSN using Taylor-based hybrid optimization algorithm," *Journal of King Saud University—Computer and Information Sciences*, vol. 34, no. 5, pp. 1857–1868, May 2022.
- [42] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1367–1376, Feb. 2022.
- [43] M. Wang et al., "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315*, 2019.

- [44] Z. Wei, Y. Zhang, X. Xu, L. Shi, and L. Feng, "A task scheduling algorithm based on Q-learning and shared value function for WSNs," *Computer Networks*, vol. 126, pp. 141–149, Oct. 2017.
- [45] L. Wu, P. Cui, J. Pei, and L. Zhao, Eds., *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Nature, 2022.
- [46] M. U. Younus, M. K. Khan, M. R. Anjum, S. Afridi, Z. A. Arain, and A. A. Jamali, "Optimizing the lifetime of software-defined wireless sensor network via reinforcement learning," *IEEE Access*, vol. 9, pp. 259–272, 2021.
- [47] M. U. Younus, M. K. Khan, and A. R. Bhatti, "Improving the software-defined wireless sensor networks routing performance using reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3495–3508, Mar. 2022.
- [48] W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10737–10750, 2021.
- [49] D. Zhou, B. Yan, C. Li, A. Wang, and H. Wei, "Relay selection scheme based on deep reinforcement learning in wireless sensor networks," *Physical Communication*, vol. 54, Art. no. 101799, Oct. 2022.
- [50] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [51] Y. Zhou, L. Zhang, and W. Li, "Partitioned uneven cluster routing algorithm based on gray wolf optimization in WSNs," *Ad Hoc Networks*, vol. 163, Art. no. 103564, Oct. 2024.
- [52] Y. Zhou, N. Wang, and W. Xiang, "Clustering hierarchy protocol in wireless sensor networks using an improved PSO algorithm," *IEEE Access*, vol. 5, pp. 2241–2253, 2017.
- [53] T. Zhu, X. Chen, L. Chen, W. Wang, and G. Wei, "GCLR: GNN-based cross-layer optimization for multipath TCP by routing," *IEEE Access*, vol. 8, pp. 17060–17070, 2020.
- [54] M. Navaei and P. Rezaei, "Investigating the effect of blood dielectric changes in human tissue model by a microwave sensor," *Journal of Modeling and Simulation in Electrical and Electronics Engineering*, vol. 2, no. 4, pp. 1–5, 2023.

BIOGRAPHIES



Morteza Saadati received the B.S. and M.S. degrees in Telecommunications Engineering from Islamic Azad University, Iran, in 2014 and 2017, respectively. He is currently a PhD candidate in Telecommunication Systems from Islamic Azad University, Iran. His research interests include, wireless networks, machine learning, and neural networks.



Sayyed Majid Mazinani was born in Mashhad, Iran on 28 January 1971. He received his Bachelor degree in Electronics from Ferdowsi University, Mashhad, Iran in 1994 and his Master degree in Remote Sensing and Image Processing from Tarbiat Modarres University, Tehran, Iran in 1997. He worked in IRIB from 1999 to 2004. He also received his Ph.D. in Wireless Sensor Networks from Ferdowsi University, Mashhad, Iran in 2009. He is currently associate professor at the faculty of Engineering in Imam Reza International University, Mashhad, Iran. He was the head of Department of Electrical and Computer Engineering from 2009 to 2012. His research interests include Computer Networks, Wireless Sensor Networks and Smart Grids.



Ali Akbar Khazaei was born in Mashhad, Iran on May 09, 1969. He received his B.S. degree in Electronics engineering from Ferdowsi University of Mashhad in 1992, his M.Sc. and Ph.D degrees in Communication engineering from Tarbiat Modares University, Tehtan, Iran, in 1996 and Islamic Azad University, Science and Research Branch, Tehran, Iran in 2012 respectively. He has been with the Electrical Engineering Department of Islamic Azad University, Mashhad Branch, Mashhad, Iran, from 2008, where he is an assistant professor now. From 2000 to 2008, he was with the Satellite Communication group, Iran Telecommunication Research Center (ITRC), Tehran, Iran.

UNCORRECTED