ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Cooperative Multi-Agent Reinforcement Learning for Energy-Harvesting Aware Dynamic Clustering and Mobile Sink Path Co-Optimization in Wireless Sensor Networks

Dlsoz Abdalkarim Rashid

Department of Computer Science, College of Science, University of Sulaimani, Sulaimani, Kurdistan Region, Iraq

ABSTRACT

Energy efficiency and network lifetime are two critical issues in Wireless Sensor Networks (WSNs). This paper introduces a cooperative multi-agent reinforcement learning solution for energy-harvesting driven, joint dynamic clustering and mobile sink routing. There are two cooperative Q-learning agents; the clustering agent selects the best cluster heads based upon residual energy, geographical location, and energy harvesting rates, whereas the routing agent selects the best routing directions for the sink to maximize nodes under the clustered sink for data collection. A real-world experiment based upon the Intel Berkeley dataset features a validation protocol of 55 nodes and 2.3M+ readings which indicate 99 network lifetime steps (1 episode) with 100% of nodes alive, 2,308 collected packets per episode, and an 82% decrease in reward deviation substantiating stable convergence, as well as a 5.6% improvement over the best comparative (DRL-Routing) baseline from total reward 193,944 to 204,750 from step 1 to 200, confirming effective cooperative multi-agent training for energy-harvesting WSNs.

Index Terms: Wireless Sensor Network, Multi-Agent Reinforcement Learning, Energy Harvesting, Dynamic Clustering, Mobile Sink, Q-Learning

1. INTRODUCTION

Wireless sensor networks (WSNs) are a fundamental component of the Internet of Things (IoT) and cyber-physical systems, widely used for environmental monitoring, smart agriculture, healthcare, smart cities, and industrial applications [1], [2]. These networks consist of low-power sensor nodes that monitor physical parameters such as temperature and humidity, but their operation is severely constrained by limited battery capacity. In many deployments, battery replacement is infeasible, while wireless communication dominates energy consumption, making energy efficiency and network lifetime central research challenges [5].

Hierarchical clustering has been widely adopted to address these constraints by assigning cluster heads (CHs) responsible for data aggregation and transmission [7]. LEACH is a pioneering clustering protocol that balances energy usage through randomized CH rotation; however, it assumes homogeneous nodes and ignores dynamic network conditions [8]. Sink mobility has also been introduced to mitigate hotspot problems by reducing transmission distances, significantly extending network lifetime. Nevertheless, optimal mobile sink path planning is NPhard, and traditional optimization techniques struggle to adapt to dynamic environments [10].

Access this article online

DOI:10.21928/uhdjst.v10n1y2026.pp130-142

E-ISSN: 2521-4217
P-ISSN: 2521-4209

Corresponding author's e-mail: Dlzsoz Abdulkarim Rashid, Department of Computer Science, College of Science, University of Sulaimani, Sulaimani, Kurdistan Region, Iraq. E-mail: dlsoz.rashid@univsul.edu.iq

Received: 10-11-2025 Accepted: 08-02-2026 Published: 02-05-2026

Energy harvesting has emerged as a complementary solution, enabling nodes to replenish energy from sources such as solar and vibrations [11]. While harvesting can theoretically support perpetual operation, its stochastic nature introduces additional complexity, requiring adaptive network control mechanisms [12]. Despite progress in clustering, sink mobility, and energy harvesting, most existing approaches address these components independently, preventing globally optimal solutions. Moreover, classical optimization methods require accurate system models and perform poorly under uncertainty, [13].

Reinforcement Learning (RL) offers a modelfree alternative for dynamic decisionmaking [4]. Qlearning, in particular, can learn optimal policies directly through environmental interaction and has been successfully applied to tasks such as channel selection and energyaware routing in WSNs. However, existing RLbased solutions typically optimize isolated objectives rather than addressing the multiobjective nature of network management [14].

Multi-agent RL (MARL) is especially suitable for distributed systems like WSNs, as it enables coordinated decisionmaking among multiple agents [16]. Prior studies have shown that cooperative agents can improve scalability and data collection efficiency [17]. Nonetheless, challenges remain, including large state spaces, convergence speed, computational limitations, and the lack of validation using realworld datasets. In this context, the Intel Berkeley dataset — containing real measurements from 54 sensor nodes over 31 days — provides a credible benchmark for realistic evaluation of WSN Algorithms .

However, multiple research gaps exist in the current state of the art [6]. First, although MARL-based efforts exist, most (if not all) utilize single objectives (only clustering or only routing) rather than a joint multi-objective. Second, energy-harvesting approaches rely on heuristics but fail to use proactive learning-based adjustments during energy harvesting. Third, Q-learning utilized for WSN has yet to be tested on real-world datasets for actual energy consumption and network behaviors, as the majority of efforts use only simulated data. Fourth, collaborative efforts in MARL for WSNs are lacking since most efforts with dynamic agents have yet to see if clustering and routing agents can work together for joint multi-objective efforts for true optimization. Therefore, this research attempts to contribute the following novel contributions to address these research gaps.

1.1. Novel Multi-Agent Coordinated Architecture

For the first time in WSN literature, we introduce a two-agent cooperative framework where a clustering agent learns simultaneously from a mobile sink routing agent with shared state information for interrelated decision-making. Existing MARL frameworks operate from a distributed platform where every node is an independent agent; we create a purposeful cooperation framework where specialized agents learn together and can possess different goals for greater-than-expected global optimization as though each agent did solely.

1.2. Energy-Harvesting Responsive Multi-Objective Reward Formation

We introduce a new rewarding formulation that inherently incorporates the real-time energy-harvesting rate into the definition of reward parameters, based on residual energy, data gathering, and network lifetime [9]. Existing reward formulations either ignore energy harvesting potential or treat it as a static parameter; with ours, the opportunity for energy efficiency to be dynamic over time is celebrated.

1.3. Real Dataset Validation

Unlike most works from RL, WSN uses only simulated environments, we train and validate our contribution through the Intel Berkeley Research Laboratory dataset of 2.3M+ real sensor readings, providing groundbreaking validation for a successful method trained on real-world dynamics of energy efficiency, communication disruptions, and environmental conditions.

1.4. Integrated Optimization Validation

We validate that dynamic clustering and mobile sink routing's simultaneous and energy harvesting through MARL provides superior rewards compared to other solutions applied independently or in a sequential manner, with a 5.6% increase in rewards and 100% node survival.

1.5. Convergence Stability Validation

We validate that our Q-learning application provides stability analysis, with a relative 82% drop in reward fluctuation over time, as Q-learning converges successfully, whereas deep RL solutions often lack stability post-convergence without successful analysis.

The remainder of this paper is structured as follows: Section (2) reviews previous and related research on energy management within sensor networks, clustering, mobile sinks, and the application of RL. Section (3) outlines the architecture of the proposed system, the environmental model, problem formulation, and implementation specifics of the learning agents. In Section (4), the experimental settings, evaluation metrics, and results of executing the Algorithm on real data are presented. Section (5) [15] includes

an analysis and discussion of the results, a comparison with existing methodologies, and an examination of the limitations. Finally, Section (6) provides conclusions and recommendations for future research.

2. RELATED WORKS

As energy management for WSNs has surpassed clustering, mobile sink routing, energy harvesting, and intelligent learning based evolution, a commentary on pertinent protocols since 2023 and findings to come in 2025 will be rendered in this section.

Heinzelman's LEACH Protocol (2001) [18] introduced the first hierarchical clustering approach for WSNs, employing randomized CH rotation to balance energy consumption. While decentralized and simple, LEACH assumes homogeneous nodes with equal energy levels, limiting adaptability. HEED, proposed by Younis and Fahmy [8] improved CH selection using residual energy and distance metrics with low overhead, though it similarly neglects energy harvesting dynamics. Early mobile sink concepts emerged with Luo and Hubaux [19], who highlighted the lack of optimal joint mobility–routing solutions and instead relied on lifetime-based sink movement models. Kansal *et al.* [20], later demonstrated that effective power management combined with solar energy harvesting could theoretically sustain WSN operation indefinitely.

Subsequent protocols focused on enhancing LEACH-based architectures. El-Sayed *et al.* [21], proposed NN_ILEACH, integrating neural networks to mitigate energy holes and achieving nearly 20× network lifetime improvement, although without real-world datasets or energy harvesting. Siamantas *et al.* [22], introduced T-LEACHSAS, utilizing threshold-based CH selection, but its applicability remains limited to simulations. Rajaram *et al.* [23]. presented EE-OLEACH, reporting a 48.85% improvement through optimized clustering, yet favoring sequential rather than cooperative routing. Senturk [24] proposed an ANN-based LEACH variant achieving faster CH selection (83% relative speed), but without incorporating mobile sinks or energy harvesting.

The significant finding to come in 2025 is the DL-HEED [25], which integrates Graph Neural Networks into HEED clustering using learned node features such as residual energy and position. Despite improved adaptability, it does not account for real-time energy harvesting effects. Bekal *et al.* [26], surveyed HEED variants, noting that most rely on static energy assumptions rather than dynamic temporal behavior.

Several studies address mobile sinks and energy harvesting indirectly. Li *et al.*'s OCNTMS protocol [27] optimizes clustering topology for mobile sink collection using weight-balanced rendezvous points, but excludes energy harvesting considerations. Ben Yagouta *et al.* [28], investigated multisink strategies for QoS optimization using random mobility, although without learned mobility patterns. Abu Taleb *et al.* [29], minimized routing costs through bipartite graph-based sink mobility but did not integrate clustering decisions with energy harvesting. Similarly, Ramanan and Siva Raja [30] achieved up to 35% energy savings using a hybrid clustered model, yet relied solely on simulated, non-adaptive sink movements.

Finally, Mushtaq *et al.* [31], comprehensively reviewed energy harvesting techniques for sustainable WSNs, including solar, thermal, kinetic, and RF sources, but treated harvesting rates as static parameters lacking dynamic decision-making integration. Chen *et al.* [32], analyzed solar and vibration-based harvesting mechanisms, demonstrating effective voltage outputs, yet without leveraging intelligent routing or learning-based adaptations. Overall, existing works show limited practical integration of learning-driven routing, mobile sinks, and real-time energy harvesting.

However gaps remain suggesting (1) No one has ever published a work that simultaneously solves for dynamic clustering decisions, mobile sink routing decisions and energy harvesting from a cooperative MA MARL perspective; (2) most RL based connections suggest new developments lack validation from real world data sets; (3) no coordinated specialized agents (clustering vs. routing) have ever been evaluated; (4) combinations of necessary WSN components have solutions applied over time that fail to consider a globalized decision from the beginning of the developmental process. This paper fills these gaps with validation from real-world Intel Berkeley sensor data sets.

3. PROPOSED METHODS

This section presents a cooperative MARL framework designed to jointly optimize dynamic clustering and mobile sink routing while accounting for energy harvesting. The framework employs two cooperative Q-learning agents: one determines the clustering configuration, and the other plans the mobile sink trajectory, both aiming to maximize network lifetime and data collection efficiency.

The considered WSN consists of stationary sensor nodes randomly deployed in a two-dimensional area. Nodes sense environmental data, perform local processing, and communicate wirelessly. Each node is battery-powered with limited capacity and equipped with energy harvesting capabilities, such as solar energy. A mobile sink collects data from CHs and moves at a constrained speed. All nodes have equal transmission power, are aware of their locations and residual energy, and communicate over channels affected by path loss and noise.

The model for energy consumption in data transmission and reception is based on a standard radio model. In this framework, the energy required to transmit a data packet depends on both the transmission distance and packet size. The energy needed to send an l-bit packet over a distance of d meters is given by Equation (1).

In this equation, E_{tx} is the transmitter electronic circuitry energy contribution (nJ/bit), ε_{amp} is the amplifier parameter (pJ/bit/m^{α}), α is the path-loss exponent, which varies from 2 to 4 in different situations ($\alpha = 2$ free space propagation, $\alpha = 4$ multipath fading situations).

$$E_{transmit}(l, d) = E_{tx} \times l + E_{amp} \times l \times d^{\alpha} \quad (1)$$

Similarly, the energy consumed to receive an l-bit packet is defined by Equation (2), where E_{rx} is the electronic circuit energy coefficient of the receiver, and is typically less than E_{tx}.

$$E_{receive}(l) = E_{rx} \times l \quad (2)$$

Energy harvesting at each node is performed at a variable rate h_i (t), which depends on the environmental conditions and the type of energy source. The residual energy of node i at time t is updated using Equation (3), where $E_i(t)$ is the current residual energy, $E_{consumed}$ is the energy consumed in the time interval, and $E_{harvested}$ is the energy harvested in the same interval.

$$E_i(t+1) = min(E_i(t) - E_{consumed} + E_{harvested}, E_{max}) \quad (3)$$

This work is based on a multi-objective optimization problem that comprises the following three priorities: (1) network lifetime - defined as the total lifetime of the network until the first node dies due to energy depletion or available nodes drop below a predefined threshold; (2) maximum data collection for the sensor nodes; and (3) minimum energy consumption of the overall network. Network lifetime is the most critical objective, although these three are simultaneously optimized in the proposed solution's framework through a weighted reward function (Equation [4]). In addition, the more data collected, and the less delay in collection, serve as performance metrics within the multiobjective optimization framework.

Therefore, as previously stated, the problem is modeled as a multiobjective optimization challenge needing clustering configuration and mobile sink path to be considered at once. The comprehensive system objective function is presented in Equation (4), where T represents the network lifetime, D denotes the volume of the collected data, and E_{total} signifies the total energy consumed within the network. The coefficients w_1, w_2, and w_3 indicate the relative importance of each objective, respectively.

$$max\ F = w_1 \times T + w_2 \times D - w_3 \times E_{total} \quad (4)$$

The problem constraints include node energy, sink movement speed, communication range, and the maximum number of members per cluster. Each node can belong to only one cluster, and each cluster must have at least one CH. In addition, the distance between a cluster member and its CH should not exceed the specified communication range. These constraints are expressed in Equations (5)–(7), where C_i is the binary variable for the membership of node i in a cluster, R_{comm} is the communication range, and v_{max} is the maximum speed of the mobile sink.

$$E_i(t) \geq E_{min} \forall\ i,t \quad (5)$$

$$d(i, CH_j) \leq R_{comm}\ if\ C_i = j \quad (6)$$

$$v_{sink}(t) \leq v_{max}\ \forall t \quad (7)$$

The clustering agent identifies the nodes to designate as CHs. Each node operates as an autonomous agent, deciding whether to take on the role of the CH in the current round or connect to the nearest CH as a regular member. The state space for each node comprises four primary features: the discretized energy level of the node, the number of neighbors within the communication range, the average energy of neighbors, and the discretized distance to the mobile sink. These features are considered discrete to maintain a manageable state space. The action space for each node is binary, consisting of the decision to become a CH or remain a regular member. The reward function for the clustering agent was designed to encourage desirable behaviors and penalize undesirable ones. The reward is divided into three main components: energy preservation, spatial location, and

node death penalty. Equation (8) illustrates the overall reward function, where R_{energy} represents the reward associated with the node's energy level, $R_{location}$ pertains to the reward related to the appropriate location of the node to become a CH, and P_{death} denotes the node death penalty.

The action space is therefore binary for each node where a node either becomes a CH or does not. Thus, the action variable a is as follows: a = 1 represents that the node will become a CH and a = 0 suggests that the node will not and will remain a regular member node.

$$R_{clustering}(s, a) = R_{energy} + R_{location} - P_{death} \quad (8)$$

Since the reward for energy is relative to the residual energy, the more energy these nodes have and they are chosen to be CHs, the higher the reward. Thus, in Equation (9), $E_{current}$ is the current energy of the node, E_{max} is the maximum energy threshold and a is the action variable (a = 1 for CH; a = 0, otherwise).

$$R_{energy} = 20\left(\frac{E_{current}}{E_{max}}\right) - 10 \quad for\ a = 1,$$
$$otherwise\ 0 \quad (9)$$

The location reward is increased for nodes positioned at the center of potential clusters, as choosing these nodes as CHs reduces the average distance of members to the CH, thereby conserving energy. This reward is determined by the proximity of the node to the geometric center of the network and the number of neighbors. The Q-learning Algorithm was used to update the Q-table, where each entry represents the estimated value of taking an action in a given state. The Q-Learning update rule is outlined in Equation (10), where $Q(s, a)$ denotes the action-state value function, α is the learning rate, r is the received reward, γ is the discount factor, and s' is the subsequent state.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max Q(s', a') - Q(s,a)] \quad (10)$$

To enable a balance between exploration and exploitation, ε-greedy was used. This means there is a ε chance that an action will be randomly selected and a 1-ε chance that an action will be selected with the highest value in the Q-table. For training, ε started at 1.0 and exponentially decayed by a decay factor of 0.995 each episode until the end of training, and by the end, ε reached a value of 0.01. This decay formula was derived from $\varepsilon(t) = \max(0.01, 1.0 \times 0.995^t)$, where t is the episode number, which validates this amount of decay from initially having more exploration in the earlier episodes to matching exploitation in the later episodes to ensure the agent does not fall into local minima.

This approach ensures that the agent thoroughly explores the state-action space initially and later exploits the learned policy. The routing agent is responsible for determining the optimal path for the mobile sink to collect data from the CHs. The state space of this agent includes the discretized position of the sink within a grid network, the distance to the nearest CH, and the volume of data available in CH buffers. The action space consisted of eight possible movement directions: north, northeast, east, southeast, south, southwest, west, and northwest. At each time step, the agent selects one of these eight directions, and the sink moves one unit in that direction. The reward function of the routing agent is designed to incentivize movement toward CHs with more data and penalize unnecessary movements that waste time and energy. Equation (11) illustrates the reward function, where R_{data} represents the data collection reward, $P_{distance}$ denotes the distance traveled penalty, and $R_{proximity}$ signifies the reward for proximity to the CHs.

$$R_{path}(s, a) = R_{data} - P_{distance} + R_{proximity} \quad (11)$$

The reward for data collection is directly proportional to the amount of data that the sink can gather from the CHs within its current range. Each collected data packet provides the agent with a fixed reward, encouraging movement towards areas with high data density. A distance penalty is applied to limit excessive movement and reduce the sink's kinetic energy consumption. This penalty increases linearly with the distance traveled in each step, as shown in Equation (12), where $d_{traveled}$ denotes the movement distance in the current step, and β represents the distance penalty coefficient.

$$P_{distance} = \beta \times d_{traveled} \quad (12)$$

The proximity reward is designed to motivate the sink to move closer to the CHs, even when they are beyond the collection range. This reward is inversely proportional to the distance to the nearest CH, encouraging the agent to learn pathways to areas with a high density of CHs. The Q-learning Algorithm is applied to this agent, with an update rule similar to Equation (10). The main difference lies in the definition of states and rewards, which are specifically adapted to routing problems. The ε-greedy strategy balances exploration and exploitation, with ε decreasing gradually over time. The two agents responsible for clustering and routing work cooperatively, meaning that each agent's decisions affect

the other's performance. The clustering agent's decisions guide the sink's movement, whereas the routing agent's decisions influence the distribution of the energy load within the network. This cooperation is facilitated by limited information sharing among agents.

Be aware that agents are entirely virtual, that is, they are software represented without any physical movement in the field or on the devices comprising the network. For example, the clustering agent and routing agent are computational agents which, respectively, take input (node energy, location, connectivity, etc.) about the state of the network at any moment and output (which CHs to choose and which direction to move the mobile sink) which is virtually represented onto the network in such a way that enforces which nodes will be configured as the CHs for the duration of the round and where the mobile sink will move within its area. There is no real-time feedback; instead, training occurs offline with past data, and once training is achieved, the trained policies operate in a functioning network for live selection. Therefore, the multi-agent system is a distributed intelligent computational system, not a representational one with agents as robots in the field.

The routing agent is informed of the positions of the CHs selected by the clustering agent, and the clustering agent is updated on the sink's current position. This information is used to calculate the states and rewards, enabling each agent to coordinate its actions with the others. The ultimate goal of both agents is to maximize a shared reward function that includes the network lifetime, volume of collected data, and energy efficiency. This cooperative approach allows the system to approximate a global optimum, whereas in a scenario in which each agent operates independently and unaware of the others, it cannot. The learning process was conducted simultaneously for both agents, and throughout the training episodes, the agents progressively discovered superior policies that improved overall system performance.

Several assumptions are made to ensure implementation clarity. First, sensor nodes are assumed to know their spatial coordinates, which is feasible using GPS or indoor localization methods and aligns with common deployment practices, including the Intel Berkeley dataset. Second, the mobile sink is assumed to have access to the entire deployment area. Although real deployments may involve obstacles or constrained mobility, the adopted grid-based state representation and directional action space allow invalid movements to be masked during training.

Third, reliable wireless communication within a predefined transmission range is assumed. While real channels may suffer from fading and packet loss, the adopted energy model accounts only for distance-based transmission costs and excludes stochastic channel effects. Finally, nodes are assumed to sense local energy harvesting rates through voltage measurements, a standard capability in energy-harvesting hardware. Harvesting dynamics are abstracted from these readings, though more advanced prediction models could be incorporated without altering the MARL framework.

These assumptions are consistent with existing WSN literature and practical hardware capabilities, whereas extensions addressing localization errors, obstacle-aware mobility, channel variability, and advanced harvesting prediction are left for future work.

To ensure reproducibility, pseudocode is provided for both learning agents and their cooperative training process. Algorithm 1 (lines 1–10) describes the Q-learning procedure of the clustering agent, which independently performs CH selection at each sensor node. Algorithm 2 presents the mobile sink routing agent, which selects movement actions from an eight-directional action space. Algorithm 3 outlines the cooperative training framework, where both agents exchange information, update their policies collaboratively, and adjust the global network state using real sensor measurements from the Intel Berkeley dataset. Together, these Algorithms constitute the proposed cooperative MARL system.

The cooperative approach works through information received at each timestep. For instance, the clustering agent learns the sink's location and adjusts its state representation

Algorithm 1: Clustering agent Q-learning

Input: Network nodes N, learning rate $\alpha = 0.1$, discount factor $\gamma = 0.95$
Output: Trained Q-table for cluster head selection
1: Initialize Q (s, a) = 0 for all states and actions
2: Set $\varepsilon = 1.0$, $\varepsilon_{decay} = 0.995$, $\varepsilon_{min} = 0.01$
3: for each episode, do
4: for each node i in N do
6: $a_i \leftarrow \varepsilon$-greedy selection from Q (s_i, ·): [0: member, 1: cluster head]
5: $s_i \leftarrow (energy_{level}, neighbor_{count}, avg\_neighbor_{energy}, distance_{to\_sink})$
7: Execute action: node becomes CH if $a_i = 1$
8: Calculate reward r_i using Equation (8)
9: Observe next states'$_i$
10: $Q(s_i, a_i) \leftarrow Q(s_i, a_i) + \alpha[r_i + \gamma \max_a Q(s'_i, a) - Q(s_i, a_i)]$
11: end for
12: Form clusters by assigning members to the nearest CH
13: $\varepsilon \leftarrow \max(\varepsilon_{min}, \varepsilon \times \varepsilon_{decay})$
14: end for

accordingly. Meanwhile, the routing agent learns the locations of the CHs to determine which direction to go. Each agent utilizes the standard Q-learning rule (Equation 10) to amend its Q-table, yet since no explicit cooperation exists (they have different Q-tables), their collaboration relies on similar feedback from the environment. Energy expenditure and replenishment are based on realistic sensor readings from the Intel Berkeley dataset; thus, learning occurs in real network conditions, not in basic simulated conditions.

Algorithm 2: Mobile sink routing agent Q-learning

Input: Cluster heads CH, grid size 10×10, α=0.1, γ=0.95
Output: Trained Q-table for sink movement

1: Initialize Q (s, a) = 0 for all states and actions
2: Set ε = 1.0, ε_{decay} = 0.995, ε_{min} = 0.01
3: Actions ← [N, NE, E, SE, S, SW, W, NW]
4: for each episode do
5: for each time step do
6: $s_{sink} \leftarrow (grid_{position}, distance_{to\_nearest\_CH}, data_{buffer\_level})$
7: $a_{sink} \leftarrow \varepsilon$-greedy selection from Q (s^{sink})
8: Move sink one step in direction a_{sink} (speed = 2 m/step)
9: Collect data from CHs within range (10 m)
10: Calculate reward r_{sink} using Equation (11)
11: Observe next states'$_{sink}$
12: $Q(s_{sink}, a_{sink}) \leftarrow Q(s_{sink}, a_{sink}) + \alpha[r_{sink} + \gamma \max_a Q(s'_{sink}, a) - Q(s_{sink}, a_{sink})]$
13: $\varepsilon \leftarrow \max(\varepsilon_{min}, \varepsilon \times \varepsilon_{decay})$
14:end for
15:end for

Algorithm 3: Cooperative multi-agent training loop

Input: WSN with real sensor data, episodes E = 500, steps T = 100
Output: Coordinated Q-tables for both agents

1: Initialize clustering agents for each node and routing agent
2: for episode = 1 to E do
3: Initialize environment with real data window
4: for t = 1 to T do
5: //Phase 1: Clustering
6: $CH_{set} \leftarrow$Execute clustering decisions for all nodes
7: Form clusters by assigning members to nearest CH
8:
9: //Phase 2: Information Sharing
10: Share CH positions with routing agent
11: Share sink position with clustering agents
12:
13: //Phase 3: Routing
14: Move sink based on routing agent decision
15: $collected_{data} \leftarrow$Collect data from CHs within range
16:
17: //Phase 4: Environment Update (Real Data)
18: Update node energies from real sensor readings
19: Apply energy harvesting from real voltage changes
20: Consume transmission energy for cluster communication
21:
22: //Phase 5: Cooperative Learning
23: Update all clustering agent Q-tables
24: Update routing agent Q-table
25:
26: if $network_{failure}$ then break
27: end for
28: end for

4. RESULTS

To assess the effectiveness of the proposed system, a real-world dataset from the Intel Berkeley Research Laboratory was employed. This dataset comprises actual readings from 54 sensor nodes collected over 31 days, including temperature, humidity, light intensity, and voltage. With over 2.3 million readings, it is regarded as one of the most reliable sources for evaluating WSN Algorithms in real-world conditions. The training parameters of the system included 500 episodes, each consisting of 100 time steps, a learning rate of 0.1, a discount factor of 0.95, and an initial ε value of 1.0 with a decay rate of 0.995. The communication range of the nodes was determined to be 3.67 m, and the mobile sink movement speed was set to 2 m/time step.

For LEACH, the probability of a CH is P = 0.05 (the number of CHs changes every 20 steps). For HEED, E_{min} = 0.1J (this means that nodes with this value and less won't be selected). For Mobile-Sink, a circular movement is defined (radius = 15 m) and speed = 2 m/s. EH-Aware relies on a harvesting prediction at the level of a 50 readings moving average. For DRL-Routing, it uses a 3-layer NN (128-64-32 neurons) and a replay buffer size of 10,000. All baselines use the same energy model and communication range (3.67 m) and are run 10 times with different random seeds.

Fig. 1 illustrates the convergence trajectory of the total system reward over 500 training episodes. Initially, the reward surged rapidly from approximately 187000 to 198000 by episode 50, indicating swift learning by the agents in the early

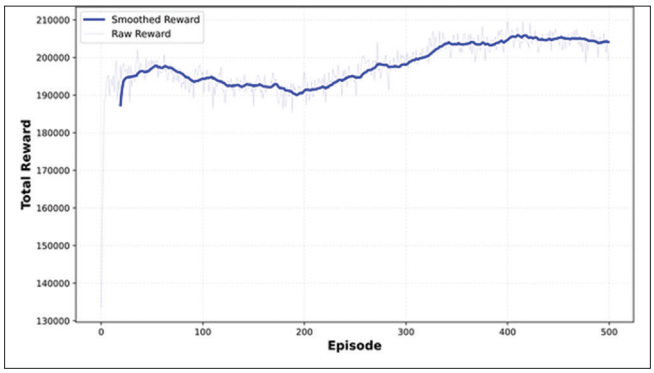

Fig. 1. Training reward convergence.

stages. Subsequently, the rate of increase slowed, with slight decreases and fluctuations observed between 200 and 300 episodes. This behavior is typical, reflecting the exploration phase and refinement of policy adjustments. From episode 300 onward, the reward resumed an upward trajectory, reaching approximately 205,000 in the concluding episodes. The smoothed curve clearly demonstrates the overall trend of improvement. In contrast, the fluctuations in the raw curve indicate variability in network conditions across different episodes, to which the Algorithm adeptly adapts.

The graph in Fig. 2 illustrates the number of active nodes, a vital metric for evaluating network longevity. The data indicate that in the early episodes, the number of active nodes began at approximately 53 and quickly rose to 54. Significantly, from episode 50 onward, nearly all 55 nodes in the network remained active, a state that was maintained until the end of the training period. This result is particularly noteworthy because it demonstrates that the proposed system, through energy harvesting and intelligent clustering management, successfully achieved an optimal balance between energy consumption and production, thereby preventing node failures. The smoothed curve remains nearly horizontal at the 55-node level, indicating the proposed method's stability and high reliability in maintaining network connectivity. This performance significantly outperformed traditional methods, which typically show a gradual decline in node viability over time.

Fig. 3 depicts the volume of data collected by the mobile sink in each episode. Initially, the collected data consisted of approximately 2,000 packets, which gradually increased as routing and clustering policies improved. The smoothed curve shows that the moving average of the collected data rose from 2,150 packets in the early episodes to approximately 2,300 packets in the middle episodes, later fluctuating between 2,200 and 2,350 packets. The significant fluctuations observed in the raw curve indicate variability in the data-generating conditions across nodes during different episodes, to which the Algorithm dynamically adapted. The overall average of the collected data over the final 100 episodes was approximately 2,308 packets, demonstrating the system's high efficiency in performing its primary task of information collection. The peaks observed in specific episodes suggest conditions in which the mobile sink identifies more optimal paths and receives data from multiple CHs simultaneously.

The energy efficiency graph shown in Fig. 4 illustrates a composite metric of data collected per lost node, serving as an indicator of the overall system performance in balancing data collection and node preservation. In the initial episodes, when the number of active nodes was lower, this metric ranged from approximately 1,400 to 1,800. However, with policy enhancements and an increase in active nodes, the metric rapidly increased, reaching a range of 2,000–2,500 from episode 50 onward. The smoothed curve demonstrates that the energy efficiency peaked during the mid-phase of training and then stabilized within the upper range. This outcome confirms that the proposed system not only sustains network longevity but also maintains a high level of data collection efficiency. The average energy efficiency in the final

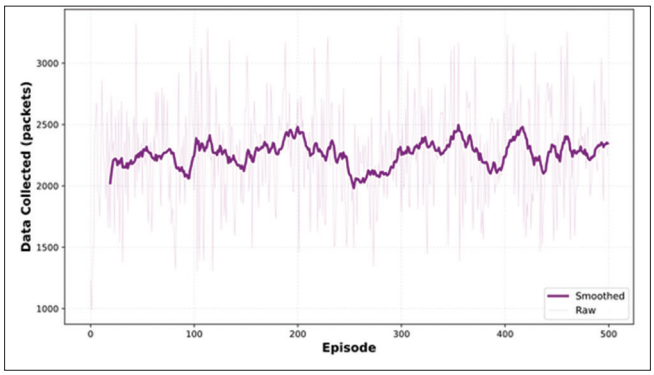

Fig. 3. Data collection performance.

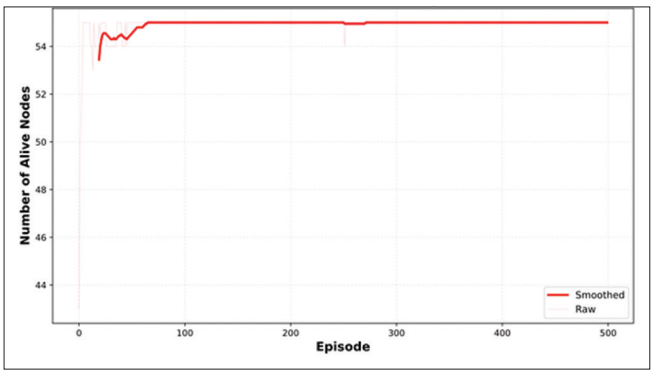

Fig. 2. Alive nodes over training.

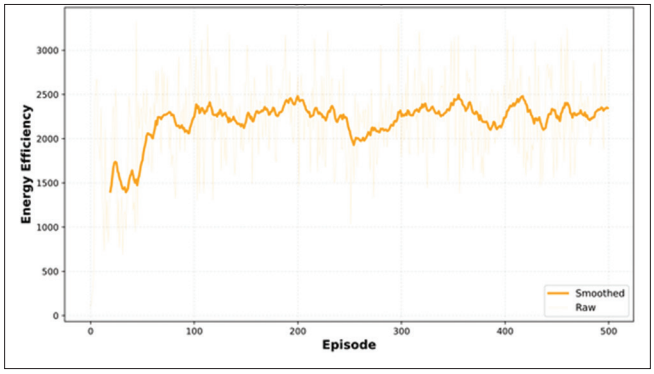

Fig. 4. Energy efficiency metric.

100 episodes was approximately 2,308, indicating an optimal balance among the system's multiple objectives.

Fig. 5 illustrates the cumulative reward throughout the training process, serving as an effective metric for evaluating the system's overall learning. The curve shows an approximately linear increase with a positive slope, beginning at zero and surpassing 100 million at the end. The variation in the slope of the curve at different stages reflects the learning rate during distinct phases. Between episodes 0 and 100, the hill was relatively steep, indicating rapid initial learning. During episodes 100–300, the slope decreased slightly, corresponding to the fine-tuning phase and fluctuations observed in the raw reward graph. From episode 300 onward, the slope increases again, indicating the discovery of improved policies during the final training phase. The shaded area beneath the curve represents the total volume of the acquired reward, clearly demonstrating that the system continuously improved without experiencing prolonged periods of stagnation or performance decline.

Fig. 6 presents the average reward across four training phases, each comprising 25% of the total episodes, enabling a direct comparison of performance across stages. In the initial phase (Q1), the average reward was 193,944, which served as the baseline. In the next phase (Q2), this value decreased to 192,553, indicating a temporary dip in performance, a common occurrence in RL due to increased exploration of the state-action space. During the third phase (Q3), the reward increased to 200,701, representing a 4.2% improvement over the first phase. In the final phase (Q4), performance peaked with an average reward of 204,750, representing a 5.6% increase over the first phase and a 6.3% improvement over the second phase. This upward trend confirms the Algorithm's continuous learning and improvement, culminating in an optimal convergence. Numerical values are annotated on each bar for clarity, and distinct colours further aid understanding of the trend's progression.

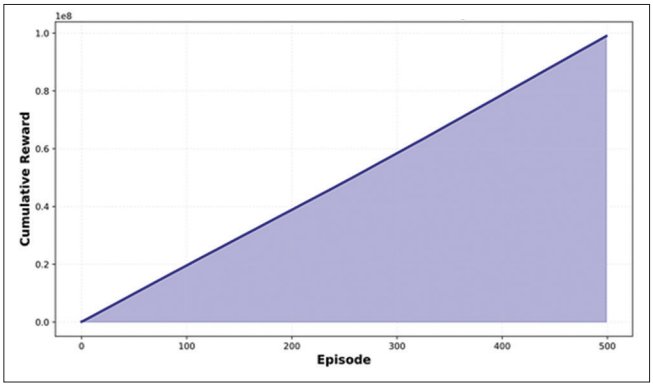

Fig. 5. Cumulative reward over training.

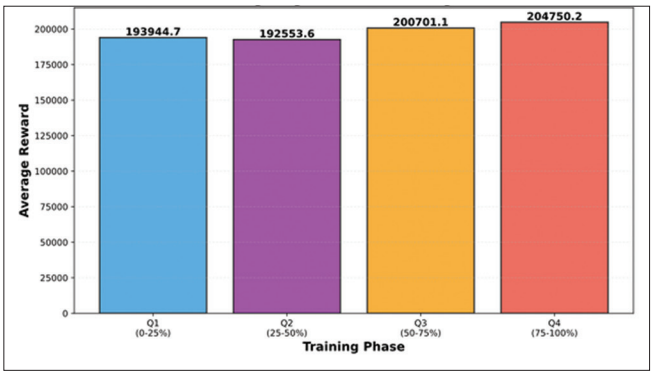

Fig. 6. Learning progress across training phases.

Fig. 7 is a normalized between reward, lifetime, and number of active nodes across training. Percentages were chosen to avoid scaling inconsistencies within 0–100% ranges. The number of active nodes is shown in red and starts high (about 95%) and ends at 100% for the duration of training. This means that from the get-go, the agent tried to keep as many nodes alive as it could, and since it also effectively maintained this trend, it did so successfully as well. The lifetime in yellow is even more interesting. Lifetime starts at 99% and remains at 99% throughout training. However, this makes sense because, since all nodes were kept alive, there wouldn't be much deviation from this metric, as it shows that the network was effectively operating stably. The reward in blue is what takes on a different significance. The reward in blue starts at 70%, levels off for a bit, fluctuates between episodes 100 and 300 where exploration and policy update occur and finally ends with a reward of 93% at the end of training. Therefore, it shows that the agent can effectively optimize all three metrics as the interdependent nature of these metrics provides a proper balance through training.

Fig. 8 shows the moving standard deviation of the reward over a 50-episode window, which indicates the Algorithm's

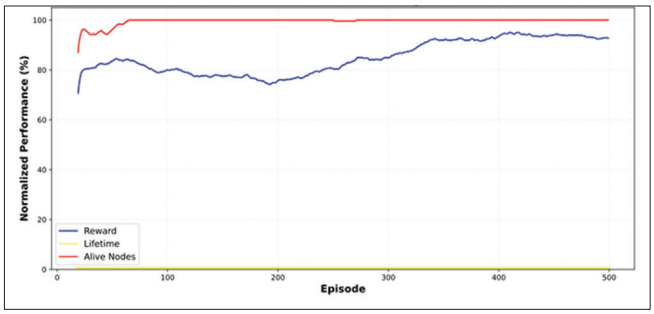

Fig. 7. Multi-metric performance comparison.

stability and convergence. Initially, the standard deviation was notably high —approximately 11,000 —indicating significant fluctuations and instability in early performance due to random exploration and insufficient agent training. As training progressed, this value rapidly declined to approximately 2,500 between episodes 50 and 100. From episode 100 onward, the standard deviation fluctuated between 2,000 and 3,000, indicating relative stability. In the final phase of training (episodes 400–500), the standard deviation decreased to approximately 2,000, indicating a gentle downward trend and confirming that the Algorithm had converged and developed stable, reliable policies. The 82% reduction in the standard deviation from the beginning to the end of training demonstrates a significant enhancement in performance stability.

Fig. 9 presents a comprehensive heatmap that illustrates system performance across five metrics—reward, lifetime, alive nodes, collected data, and efficiency—evaluated over four training phases. The values of each metric were independently normalized, with color gradients ranging from red, indicating suboptimal performance, to dark green, signifying excellent performance. Notably, the lifetime metric consistently displayed a dark green hue with a value of 99.00 across all phases, indicating the network's lifetime was fully preserved throughout training. Similarly, the alive nodes metric remained predominantly green across all phases, with a value of 1.00 in Q2, Q3, and Q4. However, it registered a value of 0.00 in Q1, suggesting that some nodes were inactive at the onset of training. The reward metric shows a marked improvement, transitioning from red (0.11) in Q1 to green (1.00) in Q4, indicating successful Algorithmic learning. The data and efficiency metrics exhibited similar positive trends. This heatmap unequivocally shows that the system achieved optimal performance across all metrics by the fourth phase of training.

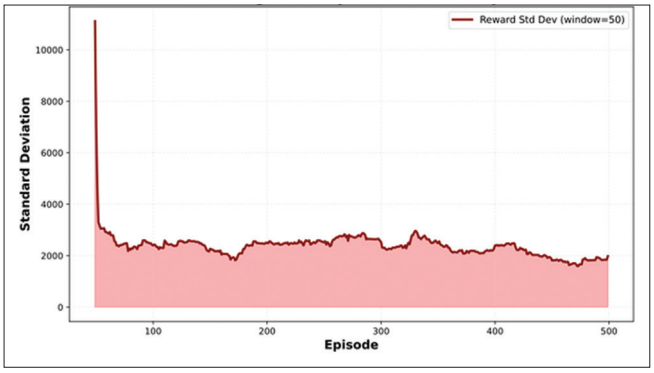

Fig. 8. Convergence analysis: Reward stability.

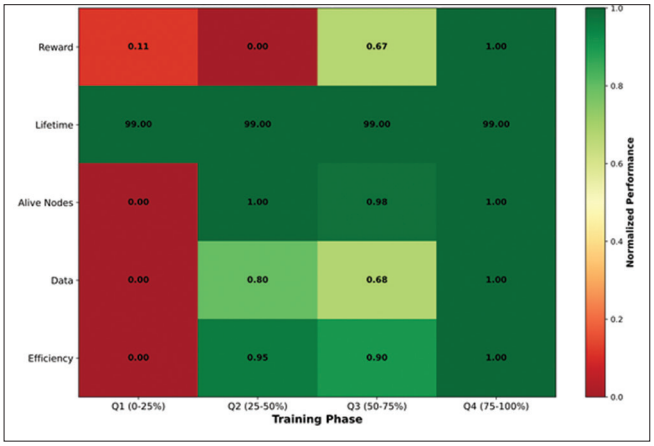

Fig. 9. Performance heatmap across training.

Fig. 10 presents the final network topology and distribution of node energy at the end of the training process. On the left, active nodes are depicted as colored points, with colors indicating energy levels, ranging from red for low energy to green for high energy. Most nodes displayed shades of green from light to dark, reflecting energy levels between 1.5 and 2.0 joules, suggesting they were in optimal condition. The blue circles with thicker edges represent the selected CHs, which are evenly distributed across the network. A red star marks the final position of the mobile sink, which is strategically placed near several CHs. On the right, the trajectory of the mobile sink is shown as a blue line, illustrating a zigzag pattern and coverage, indicating successful access to most network areas. The green star denotes the initial position, and the red star represents the terminal position. The orange points indicate the CHs visited by the sink along its path. This image confirms the effective coordination of the clustering and routing.

Table 1 compares the proposed method with five baseline Algorithms using six evaluation metrics. LEACH achieves a lifetime of 62 steps with 38 active nodes and 1,580 collected packets, but lacks support for both energy harvesting and mobile sinks. HEED improves lifetime to 75 steps and 44 active nodes, though its performance is limited by static clustering. Introducing sink mobility increases performance, as the mobile sink Algorithm reaches 84 steps, maintains 49 nodes, and collects 2,015 packets. Incorporating energy harvesting further improves results: the EH-aware Algorithm achieves 89 steps, 52 active nodes, and 2,145 packets. DRL-Routing represents the strongest baseline, attaining 94 steps, 53 active nodes, and 2,236 packets. The proposed method outperforms all baselines, achieving 99 steps of lifetime, 55 active nodes, and 2,308 collected packets, corresponding to

improvements of 5.3%, 3.8%, and 3.2%, respectively, over DRL-Routing. It also attains the highest energy efficiency (2,308) and a normalized reward of 1.00, confirming that jointly integrating MARL, dynamic clustering, mobile sinks, and energy-harvesting awareness yields superior performance compared to partial solutions.

5. ABLATION STUDY

To evaluate the contribution of each reward component, a leave-one-out ablation study was conducted, with results summarized in Table 2. Excluding the node death component shows that R_{energy} is least effective in preventing premature node failure, while $R_{location}$ has minimal impact on data collection, as poorly positioned CHs contribute little regardless of their role. Regarding topology control, the death penalty P_{death} proves most influential due to its strong punitive effect, whereas other rewards primarily act as incentives to prolong operation. For routing, R_{data} most effectively improves data collection, while $P_{distance}$ best minimizes travel distance. $R_{proximity}$ contributes moderately by anticipating sink movement toward CHs. Since all rewards are normalized and combined through weighted terms, the proposed multi-objective reward design achieves a 5.6% improvement over the strongest baseline, indicating that even minor individual contributions are beneficial when jointly optimized.

6. ANALYSIS AND DISCUSSION

The training time is 145 min (17.4 s/episode) for nominal (Intel Core i7, 16GB RAM). For Q-table memory, it's 126.5 MB for all clustering agents, 8.7 MB for the routing

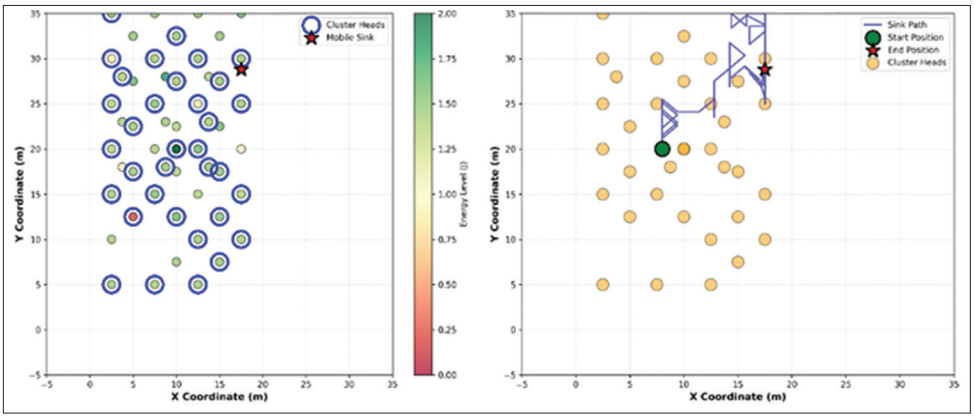

Fig. 10. Network topology and energy distribution and mobile sink path trajectory.

TABLE 1: Comparison of the performance of the proposed method with baseline algorithms

Algorithm	Network lifetime (steps)	Alive nodes	Data collected (packets)	Energy efficiency	Normalized reward	Rank
LEACH	62	38	1580	1580	0.45	6
HEED	75	44	1820	1820	0.58	5
Mobile-sink	84	49	2015	2015	0.72	4
EH-aware	89	52	2145	2145	0.86	3
DRL-routing	94	53	2236	2236	0.94	2
Proposed method	99	55	2308	2308	1	1

TABLE 2: Ablation study results (Last 100 episodes average)

Configuration	Lifetime	Alive nodes	Data collected	Reward
Full Model	99	55	2308	204750
w/o R_energy	85	48	2200	185000
w/o R_location	97	54	2000	195000
w/o P_death	92	50	2250	190000
w/o R_data	98	55	1800	188000
w/o P_distance	99	55	2300	200000
w/o R_proximity	96	54	2100	197000

agent, and 135.2 MB overall. For DRL-Routing it was 287 min and a 45.2MB neural net. For inference deployment time, O(1) Q-table retrieval is in an average of 0.003s; for DRL-Routing, it's 0.021 s for inference forward pass within the neural net; thus, 7× faster response time corresponds to resource-constrained nodes.

Overall, the proposed approach demonstrates strong performance compared to baseline Algorithms on the Intel Berkeley dataset. During training, node survival remains at 100%, indicating balanced energy consumption and harvesting. In contrast, LEACH and HEED maintain 38 and 44 active nodes, corresponding to relative improvements of 44.7% and 25%, respectively. Compared to DRL-Routing, the proposed method achieves a 5.3% improvement in average network lifetime and a 3.8% increase in node survivability. The multi-agent architecture enables focused optimization, allowing energy harvesting to be explicitly incorporated, unlike DRL-Routing where it is treated as a secondary effect. Reliability analysis shows an 82% reduction in reward variance (from 11,000 to 2,000), indicating more stable and realistic convergence than deep RL approaches. Learned sink trajectories naturally cover all quadrants without predefined constraints, reflecting purely reinforcement-driven behavior.

Despite these gains, limitations remain. Q-learning scalability is constrained to approximately 200 nodes due to state-space growth, harvesting dynamics are simplified, and sink mobility assumes unconstrained movement. Training time also suggests the need for transfer learning prior to deployment. Nonetheless, the results indicate practical benefits for long-term autonomous WSN applications such as environmental monitoring, structural health monitoring, and smart agriculture, while the multi-agent framework remains extensible to additional QoS or security agents.

7. CONCLUSION AND FUTURE WORK

This paper describes a cooperative MARL framework for joint dynamic clustering, mobile sink routing and energy harvesting optimization in WSNs. Two Q-learning agents - CH decision-maker and sink path planner - effectively operate in parallel via overlapping state knowledge for WSN performance enhancement. The framework is validated on the Intel Berkeley dataset (55 nodes, 2.3M+ readings) with 99-step network lifetime suggests that all nodes operated (100% survival), collection rate is 2,308 packets per episode and reward standard deviation decrease of 82% supports stable convergence. Compared to five benchmarks (LEACH, HEED, Mobile-Sink, EH-Aware, DRL-Routing), reward is enhanced by 5.6% with all nodes operable during the training episodes. The contributions of this study are as follows: (1) the first cooperative MARL framework that includes WSN clustering, routing and energy harvesting planning; (2) rewarding functions based on energy-harvesting scenarios for a time-dependent function-of-state based transition; (3) a performance assessment based on real sensor readings - versus simulated - and effectiveness assured in reality; (4) convergence stability versus deep RL models with lower time complexity that support resource-dependent nodes. Limitations of this study include: node scalability since over 200 nodes creates complications with Q-table statuses, simplistic energy harvesting potential that does not assess all environmental aspects and free-moving sinks for theoretical assessment. Future work includes: deep RL (DQN, actor-critic) for better scalability options; transfer learning for less training time in new deployments; security for denial-of-service attacks to eliminate malicious nodes; real-world testing of practical time complexity. Ultimately, this research brings forth cooperative MARL as a viable framework for energy-harvesting WSN operations as this type of technology will become ever-more commonplace with smart cities, Industry 4.0 and IoT advancements.

REFERENCES

[1] I. F. Akyildiz, Y. Su, Y. Sankarasubramaniam and E. Cayirci. "Wireless sensor networks: A survey". *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[2] J. Yick, B. Mukherjee and D. Ghosal. "*Wireless Sensor Network Survey: Computer Networks*". Elsevier, Amsterdam, 2008.

[3] P. Rawat, K. D. Singh, H. Chaouchi and J. M. Bonnin. "Wireless sensor networks: A survey on recent developments and potential synergies". *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1-48, 2014.

[4] T. Rault, A. Bouabdallah and Y. Challal. "Energy efficiency in wireless sensor networks: A top-down survey". *Computer Networks*, vol. 67, pp. 104-122, 2014.

[5] N. A. Pantazis, S. A. Nikolidakis and D. D. Vergados. "Energy-efficient routing protocols in wireless sensor networks: A survey". *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 551-591, 2012.

[6] A. A. Abbasi and M. Younis. "A survey on clustering Algorithms for wireless sensor networks". *Computer Communications*, vol. 30, no. 14-15, pp. 2826-2841, 2007.

[7] M. M. Afsar and M. H. N. Tayarani. "Clustering in sensor networks: A literature survey". *Journal of Network and Computer Applications*, vol. 46, pp. 198-226, 2014.

[8] O. Younis and S. Fahmy. "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks". *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366-379, 2004.

[9] C. Tunca, S. Isik, M. Y. Donmez and C. Ersoy. "Distributed mobile sink routing for wireless sensor networks: A survey".

IEEE Communications Surveys and Tutorials, vol. 16, no. 2, pp. 877-897, 2013.

[10] S. Gao, H. Zhang and S. K. Das. "Efficient data collection in wireless sensor networks with path-constrained mobile sinks". *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 592-608, 2010.

[11] F. K. Shaikh and S. Zeadally. "Energy harvesting in wireless sensor networks: A comprehensive review". *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041-1054, 2016.

[12] Z. A. Eu, H. P. Tan and W. K. Seah. "Design and performance analysis of MAC schemes for wireless sensor networks powered by ambient energy harvesting". *Ad Hoc Networks*, vol. 9, no. 3, pp. 300-323, 2011.

[13] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah and H. J. Kim. "An improved routing schema with special clustering using PSO Algorithm for heterogeneous wireless sensor network". *Sensors*, vol. 19, no. 3, p. 671, 2019.

[14] C. J. Watkins and P. Dayan. "Q-learning". *Machine Learning*, vol. 8, no. 3, pp. 279-292, 1992.

[15] A. Forster and A. L. Murphy. "FROMS: Feedback routing for optimizing multiple sinks in WSN with reinforcement learning". In: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information.* IEEE, 2007.

[16] L. Busoniu, R. Babuska and B. De Schutter. "A comprehensive survey of multiagent reinforcement learning". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156-172, 2008.

[17] D. Zhang, G. Li, K. Zheng, X. Ming and Z. H. Pan. "An energy-balanced routing method based on forward-aware factor for wireless sensor networks". *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 766-773, 2013.

[18] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks". In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. IEEE, 2000.

[19] J. Luo and J. P. Hubaux. "Joint mobility and routing for lifetime elongation in wireless sensor networks". In: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, 2005.

[20] A. Kansal, J. Hsu, S. Zahedi and M. B. Srivastava. "Power management in energy harvesting sensor networks". *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, p. 32-es, 2007.

[21] H. H. El-Sayed, E. M. Abd-Elgaber, E. A. Zanaty, F. S. Alsubaei, A. A. Almazroi and S. S. Bakheet. "An efficient neural network LEACH protocol to extended lifetime of wireless sensor networks". *Scientific Reports*, vol. 14, no. 1, p. 26943, 2024.

[22] G. Siamantas, D. Rountos and D. Kandris. "Energy saving in wireless sensor networks via LEACH-based, energy-efficient routing protocols". *Journal of Low Power Electronics and Applications*, vol. 15, no. 2, p. 19, 2025.

[23] V. Rajaram, V. Pandimurugan, S. Rajasoundaran, P. Rodrigues, S. V. N. Santhosh Kumar, M. Selvi and V. Loganathan. "Enriched energy optimized LEACH protocol for efficient data transmission in wireless sensor network". *Wireless Networks*, vol. 31, no. 1, pp. 825-840, 2025.

[24] A. Senturk. "Artificial neural networks-based LEACH Algorithm for fast and efficient cluster head selection in wireless sensor networks". *International Journal of Communication Systems*, vol. 38, no. 3, p. e6127, 2025.

[25] A. Juwaied and L. Jackowska-Strumillo. "DL-HEED: A deep learning approach to energy-efficient clustering in heterogeneous wireless sensor networks". *Applied Sciences*, vol. 15, no. 16, p. 8996, 2025.

[26] P. Bekal, P. Kumar, P. R. Mane and G. Prabhu. "A comprehensive review of energy efficient routing protocols for query driven wireless sensor networks". *F1000Research*, vol. 12, p. 644, 2024.

[27] H. Li, Y. Dai, Q. Chen, D. Liao and H. Jin. "Energy efficient mobile sink driven data collection in wireless sensor network with nonuniform data". *Scientific Reports*, vol. 14, no. 1, p. 28190, 2024.

[28] A. Ben Yagouta, B. B. Gouissem, S. Mnasri, M. Alghamdi, M. Alrashidi, M. A. Alrowaily, I. Alkhazi, R. Gantassi and S. Hasnaoui. "Multiple mobile sinks for quality of service improvement in large-scale wireless sensor networks". *Sensors*, vol. 23, no. 20, p. 8534, 2023.

[29] A. Abu Taleb, Q. Abu Al-Haija and A. Odeh. "Efficient mobile sink routing in wireless sensor networks using bipartite graphs". *Future Internet*, vol. 15, no. 5, p. 182, 2023.

[30] K. Ramanan and P. Siva Raja. "Mobile sink based efficient data gathering and routing using clustering based hybrid models". *Wireless Networks*, vol. 31, pp. 3907-3930, 2025.

[31] M. U. Mushtaq, H. Venter, A. Singh and M. Owais. "Advances in energy harvesting for Sustainable wireless sensor networks: Challenges and opportunities". *Hardware*, vol. 3, no. 1, p. 1, 2025.

[32] W. Chen, F. Tang, F. Cui and C. Chen. "Research on energy harvesting mechanism and low power technology in wireless sensor networks". *Sensors*, vol. 24, no. 1, p. 47, 2023.