University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 17 No. 1, June 2026

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

Multi-agent Deep Reinforcement Learning for Energy-Efficient Cluster Head Selection in IoT-Based Wireless Sensor Networks

Omilabu Ademola A.
Department of Computer and Information Sciences, Tai Solarin Federal University of Education, Ijagun, Ogun State, Nigeria.

omilabuaa@tasued.edu.ng

Abstract
To monitor large physical spaces, modern industrial and urban Internet of Things (IoT) architectures depend on data collected by Wireless Sensor Networks (WSNs). Crucially, battery life remains the absolute limiting factor. Because these tiny sensor nodes run on limited, non-rechargeable power, keeping the network alive for long periods is incredibly difficult. To stretch these limited power reserves, current systems group nodes into clusters. These protocols group the nodes together so that a single coordinator—the Cluster Head (CH)—collects all local data and routes it to the main base station. However, static heuristics and centralized systems are simply too rigid. They fail completely if data traffic surges or if the physical network layout changes unexpectedly. To bridge this gap, we designed a decentralized Multi-Agent Deep Reinforcement Learning (MADRL) framework specifically for selecting CHs efficiently. Instead of relying on a central controller, we treat each sensor node as an independent agent. These agents work together to cut down overall energy loss while keeping network coverage wide. We built a Deep Q-Network (DQN) directly into each agent, allowing them to learn smart clustering habits on the fly. The nodes make these choices by tracking their own remaining power, how close they are to neighbors, and their distance to the base station. Our simulations show excellent results. This decentralized MADRL setup drastically lowers energy use, keeps individual nodes alive longer, and extends network lifespan far better than traditional LEACH or genetic algorithms.

***Keywords*:** Internet of Things, Wireless Sensor Networks, Cluster Head Selection, Multi-Agent Deep Reinforcement Learning, Energy Efficiency.

1.0 Introduction

Farms, factories, and smart cities now rely heavily on the Internet of Things (IoT) to power their data collection pipelines. These networks use embedded Wireless Sensor Networks (WSNs) as their primary hardware backbone. Instead of manual monitoring, small autonomous tracking nodes are scattered across fields to log physical conditions constantly [1]. This raw telemetry then streams back to local edge gateways or centralized cloud databases. The environment introduces a major hurdle. Because these devices usually operate in dangerous or completely isolated regions, swapping dead batteries or recharging them manually is an impossible task [2]. This hard operational constraint means that network survival depends strictly on power management.

Transmitting data is the fastest way to kill a sensor node's battery. Direct long-distance communication between every individual node and a distant base station (BS) drains power rapidly. This reckless energy expenditure splits the topology and creates massive unmonitored dead zones. Hierarchical clustering protocols solve this exact problem [3]. Instead of working individually, the network groups itself into compact local cells. Regular nodes save their power by transmitting readings only to a local team leader—the Cluster Head (CH). The CH aggregates, filters out redundant noise, and compresses the data packets before handling the long-range trip to the distant BS. By shifting the transmission burden, this setup keeps peripheral nodes alive much longer.

Even with clustering, choosing the right CH introduces a separate layer of structural

Omilabu Ademola A. (2026). Multi-agent Deep Reinforcement Learning for Energy-Efficient Cluster Head Selection in IoT-Based Wireless Sensor Networks. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 17 No. 1, pp. 79 - 90.

issues. If the network keeps picking the exact same node to act as the coordinator, its battery will burn out immediately, leaving a blind spot in the network geography. Early classical protocols, like Low-Energy Adaptive Clustering Hierarchy (LEACH), relied on basic probabilistic rotations to swap the CH role [4]. While it demands very little computing power, LEACH completely ignores live network stats, such as actual residual energy or exact node positioning. This lack of situational awareness frequently leads to the selection of poorly positioned or low-energy CHs, accelerating network degradation. Centralized optimization tools using linear programming or genetic algorithms can find better paths, but they require nodes to constantly report their status back to the BS. The massive control-packet traffic generated by this constant reporting often wastes more energy than the optimization actually saves [5].

Recent breakthroughs in Artificial Intelligence offer a much cleaner alternative via decentralized edge computing. Multi-Agent Deep Reinforcement Learning (MADRL) allows individual network nodes to learn smart operational habits by interacting directly with their local environment [6]. By treating sensor nodes as independent, intelligent agents, the network can constantly tune its clustering configuration based on shifting spatial dynamics and battery life. This edge-level intelligence completely removes the heavy control-message baggage tied to centralized models.

This paper details a robust MADRL framework engineered for energy-efficient CH selection in dynamic IoT-WSN environments. The core contributions of this study are three-fold:

1. The CH selection problem is formulated as a cooperative Markov Decision Process (MDP) designed to balance the routing workload across heterogeneous energy levels.
2. A decentralized MADRL algorithm is developed where independent nodes learn to execute CH transitions using purely localized state vectors, thereby eliminating control message bloat.
3. We execute comprehensive comparative evaluations against standard heuristic and centralized optimization baselines to validate improvements in network longevity and packet delivery ratios.

2. Related Works

Over the last two decades, finding new ways to preserve energy in wireless sensor setups has led to a massive variety of routing and clustering models. Early attempts relied almost entirely on hardcoded heuristics or localized probability formulas. A prime example is the classic Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol, which uses a randomized rotation setup to pass the energy-draining cluster head role around the network [4]. While LEACH keeps local computation costs incredibly low, its basic probability formula remains completely blind to where nodes sit or how much battery power they actually have left. Because of this blind spot, the system frequently forces an almost-dead node to act as a cluster head, causing premature battery death and breaking the network apart.

To step past these basic heuristic limitations, engineers shifted toward centralized optimization approaches. These dynamic architectures leverage powerful evolutionary strategies like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Artificial Bee Colony (ABC) models to calculate the most efficient cluster head layout from the base station [5], [7]. A typical centralized PSO framework, for example, determines the perfect cluster radius by minimizing the absolute geometric distance between regular members and their assigned cluster head. These centralized options keep networks alive much longer than basic LEACH, but they introduce a severe structural drawback: the base station cannot optimize anything without a non-stop feed of real-time telemetry from every single node in the field. This relentless stream of status reports creates a massive control-packet burden that often uses up more energy than the optimization algorithm actually saves, making it a poor choice for large, resource-limited IoT setups.

Lately, the research landscape has shifted toward decentralized machine learning, specifically Reinforcement Learning (RL), to put smart choices directly at the local node level without the control-packet bloat. Early single-agent RL

attempts tried using standard tabular Q-learning, making the entire network or the base station act as a lone decision-maker [8]. However, as the node count scales up, the state-action combinations in a single-agent tabular setup explode exponentially. This classic curse of dimensionality completely cripples tabular Q-learning when applied to massive, multi-hop IoT infrastructures.

Multi-Agent Deep Reinforcement Learning (MADRL) breaks through this scaling wall by dividing the processing burden across independent, autonomous node agents. Modern setups deploy Deep Neural Networks (DNNs) as function approximators capable of handling massive, continuous spatial inputs [6], [9]. In these cooperative setups, independent nodes track local stats, like real-time battery levels, immediate link quality, and neighbor density, to pick cluster configurations autonomously. State-of-the-art MADRL tools show major stability improvements over older frameworks [10]. Even so, standard implementations still lean heavily on global reward mechanisms that require synchronized updates across the entire network topology. Our work addresses this limitation by building a completely localized reward function, allowing scalable, independent coordination without forcing synchronized global updates.

3. Methodology and System Model

3.1 Network and Energy Consumption Model

An IoT framework consisting of N heterogeneous sensor nodes distributed randomly within a fixed, two-dimensional operational space is considered. A single, stationary base station resides completely outside this deployment field at fixed, permanent coordinates. The entire architectural topology is governed by the following structural conditions:

- Every deployed sensor node remains strictly stationary after initial placement.
- Distinct initial energy levels are assigned to individual nodes, establishing a heterogeneous operational environment.
- The base station possesses unlimited computational capacity and continuous power resources.

To quantify the rate of power dissipation during runtime operations, a standard first-order radio model is adopted [4]. The energy expended to transmit an l-bit data packet over a geometric distance d follows the mathematical formulation in (1):

$$E_{Tx}(l,d) = \begin{cases} l \cdot E_{elec} + l \cdot \varepsilon_{fs} \cdot d^2, & d < d_0 \\ l \cdot E_{elec} + l \cdot \varepsilon_{mp} \cdot d^4, & d \geq d_0 \end{cases} \quad (1)$$

where E_{elec} represents the electronics energy dissipated per bit during signal processing, and ε_{fs} and ε_{mp} denote the amplifier energy coefficients for the free-space model and the multi-path fading model, respectively. The cross-over threshold distance d_0 is defined analytically as seen in (2):

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

Correspondingly, the energy required by a node to receive an l-bit data packet is modeled by (3):

$$E_{Rx}(l) = l \cdot E_{elec} \quad (3)$$

3.2 Problem Formulation as a Markov Decision Process

We model the decentralized energy-efficient CH selection task as a multi-agent cooperative Markov Decision Process (MDP). The formulation comprises a tuple defined by $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where each component represents a core network parameter:

State Space ($\mathcal{S}$)

Each sensor node acts as an independent agent i. At any given time step t, agent i constructs its localized state vector $s_i^t \in \mathcal{S}$ using three independent real-time variables as expressed in (4):

$$s_i^t = \left\{ E_{res}^i(t), D_{BS}^i, \overline{D}_{nei}^i(t) \right\} \quad (4)$$

where $E_{res}^i(t)$ represents the current normalized residual energy of the node, D_{BS}^i is the static Euclidean distance from the node to the base station, and $\overline{D}_{nei}^i(t)$ represents the average geometric distance to active neighboring nodes within its transmission radius.

Action Space ($\mathcal{A}$)

The action space for each individual agent is discrete and binary as shown in (5):

$$\mathcal{A}_i = \{0,1\} \quad (5)$$

Executing $a_i^t = 1$ signifies that the node elects itself as a Cluster Head for the upcoming communication round. Conversely, executing $a_i^t = 0$ indicates that the node will remain an ordinary cluster member, seeking connection to the nearest available CH.

Reward Function ($\mathcal{R}$)

To enforce energy preservation and uniform load distribution without global network overhead, we design a localized, multi-objective reward function r_i^t. The total reward received by agent i at step t balances individual survival with global network utility according to (6):

$$r_i^t = w_1 \cdot \left(\frac{E_{res}^i(t)}{E_{init}^i}\right) - w_2 \cdot \left(a_i^t \cdot \frac{D_{BS}^i}{\max(D_{BS})} + (1 - a_i^t) \cdot \frac{d_{toCH}^i}{\max(d_{toCH})}\right) - w_3 \cdot \mathcal{P}_{pen} \quad (6)$$

Here, w_1, w_2, and w_3 are positive scaling weights. The first term rewards the preservation of residual energy. The second term penalizes high transmission distances (d_{toCH}^i represents the distance from a member node to its selected CH). The final term, $\mathcal{P}_{pen}$, is a structural penalty applied if two nodes within immediate transmission range both select $a_i^t = 1$, or if zero CHs are selected within an entire localized neighborhood.

The network system model diagram in Figure 2 maps out the architectural configuration of the deployment area as described in Section 3.1. The overall operational space is physically segregated into distinct computational and transmission tiers:

1. Baseline System Positioning and Coordinates

The infrastructure tracks $N = 100$ heterogeneous sensor nodes scattered across a 100 m × 100 m operational field. The Remote Base Station serves as the destination for all aggregated data packets and is located outside the sensor field at fixed coordinate position $(50,150)$ m. This positioning deliberately forces long-range data transmissions, highlighting the importance of selecting high-energy intermediaries.

2. Localized Multi-Agent Decision Making

Every sensor node runs an independent deep reinforcement learning algorithm locally. As ordinary member nodes monitor environmental telemetry, three localized conditions are actively tracked: current normalized residual energy (E_{res}^i), distance to the base station (D_{BS}^i), and the average distance to their local neighbors ($\bar{D}_{nei}^i$). Based on these metrics, nodes autonomously execute binary actions ($a_i \in \{0,1\}$) to self-elect as Cluster Heads or remain regular cluster members.

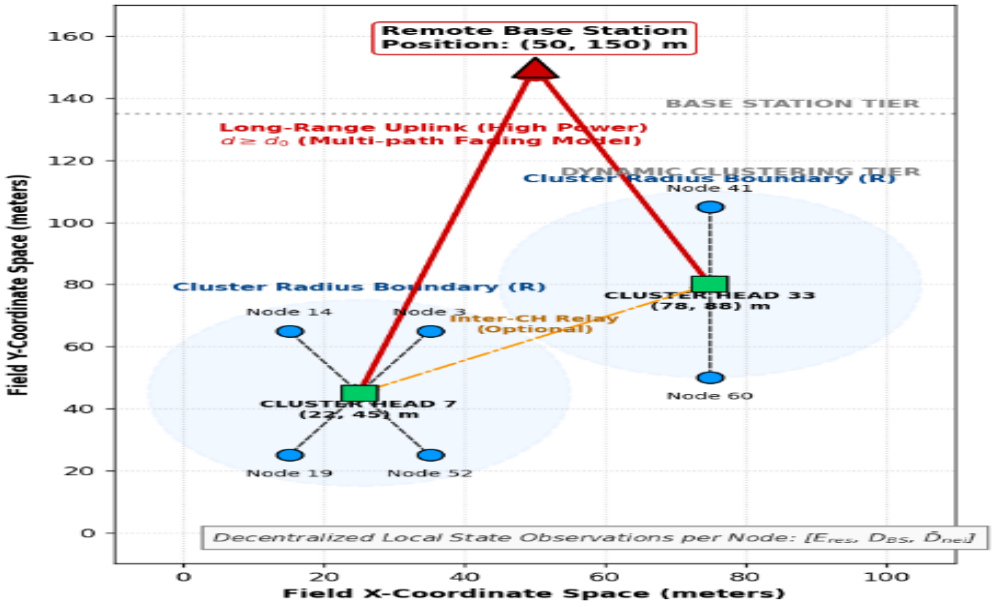

Figure 2: Structural System Architecture and Multi-Tier Communications Topology for the MADRL Framework.

3. Energy-Balanced Communication Paths

Once the local execution loops select the optimal cluster heads for a communication round, localized cluster boundaries are built dynamically based on proximity rules.

- **Intra-Cluster Communications (Short-Range):** Standard sensor nodes transmit raw data exclusively to their nearest local Cluster Head. Because these distances remain below the cross-over threshold ($d < d_0$), transmissions follow the free-space amplifier consumption model, preserving peripheral battery life.
- **Aggregated Uplink Communications (Long-Range):** The elected Cluster Heads absorb all incoming member packets, filter out redundant readings, and compress the remaining information. The CH then transmits this compressed packet directly across the long-range uplink to the Remote Base Station. Since the distance to the base station typically exceeds the threshold ($d \geq d_0$), energy is expended according to the multi-path fading model. The framework rotates this high-energy role dynamically, preventing premature battery failure.

3.3 Network Simulation Environment Configuration

To satisfy the technical and empirical reproducibility criteria, Table 1 details the explicit hardware configuration, environment model parameters, and optimization hyperparameters deployed throughout the simulation runs.

Table 1: Structural Network and Hyperparameter Specifications

Parameter Type	Specific Variable / Metric	Operational Value
Topological Properties	Network Topology Area	100 m × 100 m
	Total Sensor Population (N)	100 Nodes
	Base Station Position	(50,150) m
Radio Energy Coefficients	Processing Dissipation (E_{elec})	50 nJ/bit
	Free Space Amplifier (ε_{fs})	10 pJ/bit/m^2
	Multi-path Amplifier (ε_{mp})	0.0013 pJ/bit/m^4
	Data Packet Dimension (l)	4000 bits
	Initial Battery Capacity (E_{init})	Heterogeneous (0.5 J − 1.0 J)
MADRL Hyperparameters	Replay Buffer Capacity ($\mathcal{D}_i$)	10,000 Tuples
	Batch Execution Scale (M)	64
	Learning Rate (α)	0.001
	Discount Coefficient (γ)	0.95
	Initial Policy Exploration (ϵ)	1.0
	Strategy Decay Constant (γ_{decay})	0.995
	Total Iteration Training Rounds	2500 Rounds

4.0 PROPOSED MADRL FRAMEWORK ALGORITHM

To implement the cooperative Markov Decision Process outlined in Section 3, a decentralized Multi-Agent Deep Q-Network (MADRL-DQN) framework is developed. Each separate sensor node hosts an independent DQN agent, completely bypassing the need for a central coordinator. This structural choice explicitly mitigates the massive communication overhead typical of centralized network models.

4.1 Agent Architecture and Action Selection

Each node agent i maintains a local deep Q-network parameterized by weights θ_i. The network takes the local state vector s_i^t as an input and outputs predicted Q-values, $Q(s_i^t, a_i; \theta_i)$, for possible actions.

To balance the exploration of new clustering configurations with exploitation of known energy-efficient topologies, an ϵ-greedy action selection policy is utilized by the local agents. At any given time step t, the discrete action a_i^t is selected according to the strategy defined in (7):

$$a_i^t = \begin{cases} \text{a random action from } \mathcal{A}_i, & \text{with probability } \epsilon \\ \arg\max_{a_i} Q(s_i^t, a_i; \theta_i), & \text{with probability } 1-\epsilon \end{cases} \quad (7)$$

The exploration probability ϵ decays systematically across successive communication rounds according to the formula in (8):

$$\epsilon^{t+1} = \epsilon^t \cdot \gamma_{decay} \quad (8)$$

where $\gamma_{decay} \in (0,1)$. This decay ensures the entire network shifts toward pure operational stabilization as execution rounds progress.

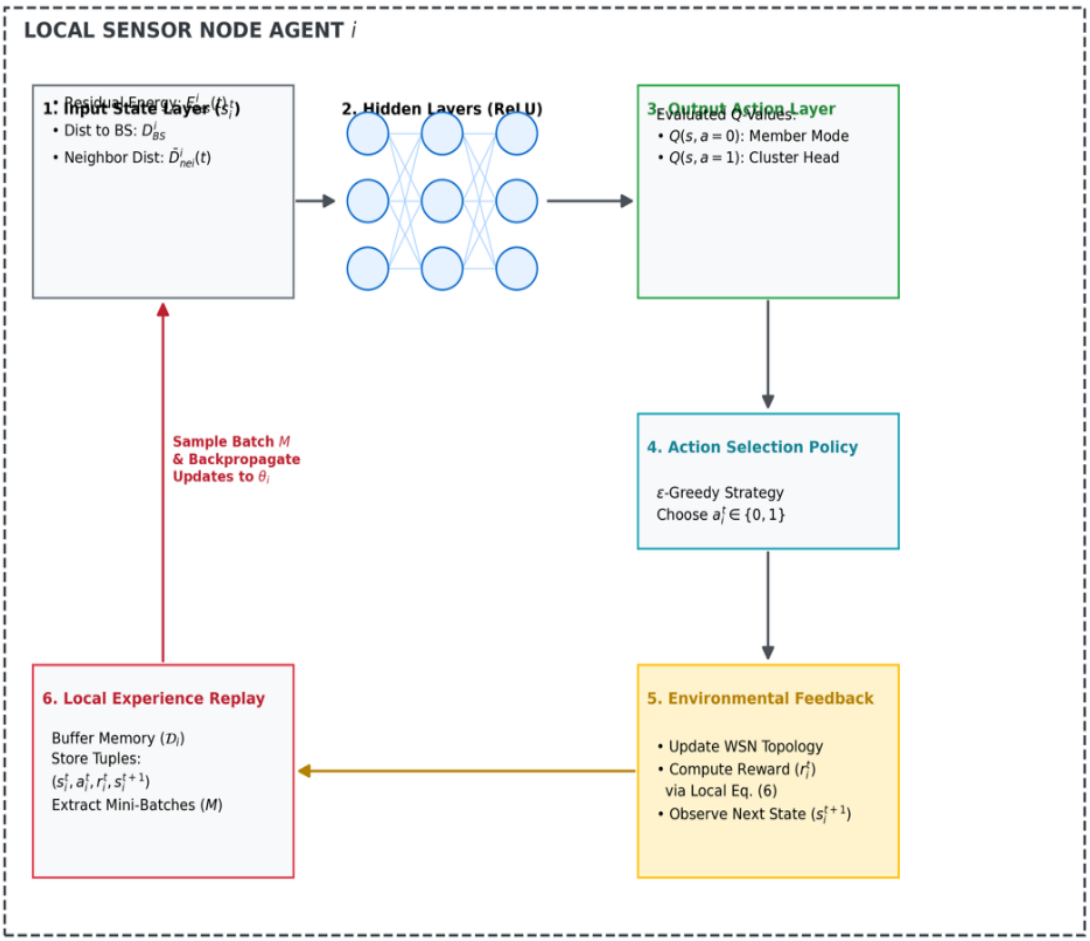

Figure 3: Internal Deep Q-Network agent architecture block diagram demonstrating the localized machine learning execution loop running on an edge sensor device.

The functional layout detailed in Figure 3 visualizes the edge intelligence processing loop operating concurrently inside each separate sensor node. The framework establishes a fully decentralized model that operates through six modular processing stages:

1. **High-Dimensional Input Processing:** At the beginning of each communication round, the edge node extracts three local parameters to construct its input state vector s_i^t: the remaining normalized battery capacity ($E_{res}^i(t)$), the fixed coordinate distance to the base station (D_{BS}^i), and the average spatial distance to neighbors ($\overline{D}_{nei}^i(t)$).coordinate distance to the base station (D_{BS}^i), and the average spatial distance to neighbors ($\overline{D}_{nei}^i(t)$).
2. **Deep Neural Function Approximation:** The raw vector directly feeds into a local Deep Neural Network consisting of fully connected hidden layers with Rectified Linear Unit (ReLU) activation functions. This multi-layer architecture mathematically maps the high-dimensional spatial states to expected reward metrics.
3. **Action Valuations:** The output layer evaluates the expected long-term return values (Q-values) for the two available choices: remaining a standard cluster member ($a_i^t = 0$) or operating as a temporary coordinator ($a_i^t = 1$).
4. **Local Strategy Selection:** An ϵ-greedy policy controls the selected action a_i^t. This prevents the system from stalling in sub-optimal configurations by balancing random exploration transitions against known efficient clusters.
5. **Decentralized Multi-Objective Feedback:** Once the actions alter the network clustering topology, a localized reward function calculates the environmental response (r_i^t). The reward function balances individual battery preservation against neighbor proximity penalties, enforcing group cooperation without requiring global synchronization overhead.
6. **Local Experience Replay Storage:** The resulting transition data—consisting of the state, selected action, reward score, and subsequent state tuple $\left(s_i^t, a_i^t, r_i^t, s_i^{t+1}\right)$—is securely saved inside the local experience replay memory buffer $\mathcal{D}_i$. Random mini-batches (M) are extracted continuously from this storage array to update the online weights (θ_i) through backpropagation, stabilizing policy convergence.

4.2 Local Experience Replay and Training Logic

To break the temporal correlations inherent in sequential network states, a localized experience replay memory $\mathcal{D}_i$ with a fixed capacity is managed within each separate node. During the clustering phase, local transitions are logged as an experience tuple matching the formulation in (9):

$$e_i^t = \left(s_i^t, a_i^t, r_i^t, s_i^{t+1}\right) \quad (9)$$

Once $\mathcal{D}_i$ accumulates a threshold number of entries, a random mini-batch of M experiences is sampled to update the network weights. Each agent minimizes a distinct mean squared Bellman error loss function shown in (10):

$$L_i(\theta_i) = \frac{1}{M}\sum_{j=1}^{M}\left[\left(r_j + \gamma \max_{a'} Q\left(s_j', a'; \theta_i^-\right)\right) - Q\left(s_j, a_j; \theta_i\right)\right]^2 \quad (10)$$

where $\gamma \in [0,1)$ represents the discount factor prioritizing long-term energy preservation over immediate power savings, and θ_i^- denotes the parameters of a dedicated target Q-network. The target network parameters remain frozen for a fixed number of iterations before copying the online weights θ_i, stabilizing training convergence within dynamic network environments.

The complete functional sequence of step-by-step training and weight-updating loop is detailed explicitly in Algorithm Pseudocode.

Algorithm 1: Localized Edge DQN Weight Update Loop

1: Initialize online deep Q-network parameters θ_i for each independent node agent i.
2: Clone initial weights to establish target network parameters: θ_i⁻ ← θ_i.
3: Initialize localized experience replay buffer storage memory D_i.

4: for each communication round iteration t = 1 to 2500 do
5: Evaluate local environment parameters to construct state vector s_i^t via Eq. (5).
6: Select discrete operational action a_i^t ∈ {0, 1} via ε-greedy strategy.
7: Execute action a_i^t, broadcast cluster head intent, and establish dynamic grouping.
8: Compute localized Multi-Objective Survival Reward r_i^t using Eq. (6).
9: Observe subsequent localized environmental state configuration s_i^(t+1).
10: Archive transition tuple entry e_i^t = (s_i^t, a_i^t, r_i^t, s_i^(t+1)) inside buffer D_i.
11: if D_i contains a threshold number of transitions greater than batch size M then
12: Randomly sample a mini-batch of M independent experiences from buffer D_i.
13: For each sampled experience item j, compute target value: Y_j = r_j + γ max_{a'} Q(s_j^(t+1), a'; θ_i⁻).
14: Calculate mean squared bellman gradient loss L_i(θ_i) matching Eq. (10).
15: Perform Backpropagation to update online network weights θ_i using Adam Optimizer.
16: end if
17: Every C steps, synchronize target network weights: θ_i⁻ ← θ_i.
18: Reduce action exploration rate using strategy decay calculation in Eq. (8).
19: end for

5.0 Performance Evaluation and Discussion

5.1 Simulation Environment Setup

We evaluate the proposed decentralized Multi-Agent Deep Reinforcement Learning (MADRL) framework using a customized discrete-event simulation environment implemented in Python. The simulation instantiates a network of $N = 100$ sensor nodes distributed randomly across a 100 m $\times$ 100 m target field. The stationary base station resides outside the deployment field at fixed coordinates $(50,150)$ m. To evaluate protocol performance under realistic constraints, we implement a heterogeneous initial energy distribution where node battery capacities vary uniformly between 0.5 J and 1.0 J.

The radio hardware parameters are configured based on the standard first-order energy model [11]. The electronics dissipation coefficient (E_{elec}) for both transmitter and receiver circuitry is set to 50 nJ/bit. For signal amplification, the free-space coefficient (ε_{fs}) is 10 pJ/bit/m^2, while the multi-path fading coefficient (ε_{mp}) is 0.0013 pJ/bit/m^4. Every node transmits data using a fixed packet length (l) of 4000 bits over a maximum simulation timeline of 2500 communication rounds. For the local deep reinforcement learning parameters, each edge agent utilizes a learning rate (α) of 0.001 and a discount factor (γ) of 0.95. Additionally, the local experience replay buffer ($\mathcal{D}_i$) maintains a maximum capacity of 10,000 historical transition tuples.

5.2 Performance Metrics and Baseline Protocols

We benchmark the proposed MADRL framework against two classic routing paradigms to quantify its optimization advantages:

- **Low-Energy Adaptive Clustering Hierarchy (LEACH):** The foundational distributed clustering protocol utilizing localized probabilistic cluster head rotation thresholds [1].
- **Centralized Genetic Algorithm (GA):** A global optimization routing approach where the base station calculates near-optimal cluster head configurations using evolutionary heuristics [2].

The comparative assessment measures operational performance using three critical network reliability metrics:

1. **Network Operational Lifetime:** Measured by the exact communication round where the First Node Dies (FND) and Half of the Nodes Die (HND).
2. **Total Aggregate Residual Energy:** The sum of the remaining battery capacity across all surviving sensor nodes tracked over time.
3. **Packet Delivery Ratio (PDR):** The ratio of total unique data packets successfully received at the base station relative to the raw packet load generated by source sensor nodes.

5.3 Discussion of Empirical Results

The physical deterioration patterns of the network across the evaluated routing mechanisms demonstrate the vulnerability of uncoordinated setups. Figure 4 tracks the active node count across successive operational cycles. LEACH exhibits an early, sharp degradation curve, where the first node failure happens at round 380 due to its randomized selection logic. Because LEACH neglects instant energy states, it frequently assigns energy-depleted nodes as cluster heads, accelerating early network partitioning. The Centralized GA framework mitigates this issue initially, delaying the FND milestone until round 620 due to its globally balanced cluster calculations. However, the GA protocol experiences rapid battery exhaustion beyond round 1200. This downward spike stems from the heavy transmission overhead required to send continuous raw state telemetry from the nodes back to the base station.

- **Panel A (Numerical Node Survival Matrix):** Quantifies the exact surviving sensor node population at specific milestones across 2500 communication rounds. The empirical data highlights a sharp difference in survival rates: by round 2000, both LEACH and Centralized GA have depleted their entire active node pool (0 nodes remaining), whereas the proposed model retains 44 highly operational sensor devices.
- **Panel B (Network Lifetime Survival Curves):** Illustrates the overall slope of network degradation. The trace confirms that LEACH experiences an early and aggressive decline, hitting its First Node Dead (FND) benchmark at round 380 due to uncoordinated probabilistic rotations. While the Centralized GA approach maintains a flatter curve initially, it experiences a massive drop-off after round 1200 because nodes rapidly exhaust their batteries transmitting raw telemetry updates to the base station. The proposed model provides a significantly gentler decline, preserving network balance past round 1500.
- **Panel C (Topological Energy Map Snapshots):** Displays spatial snapshots of remaining field energy at rounds 1000 and 1500, using a heat gradient where green indicates full battery reserves and red denotes dead zones. The snapshots show that LEACH and Centralized GA suffer from severe localized energy depletion, leaving large areas of the network completely blind.

.

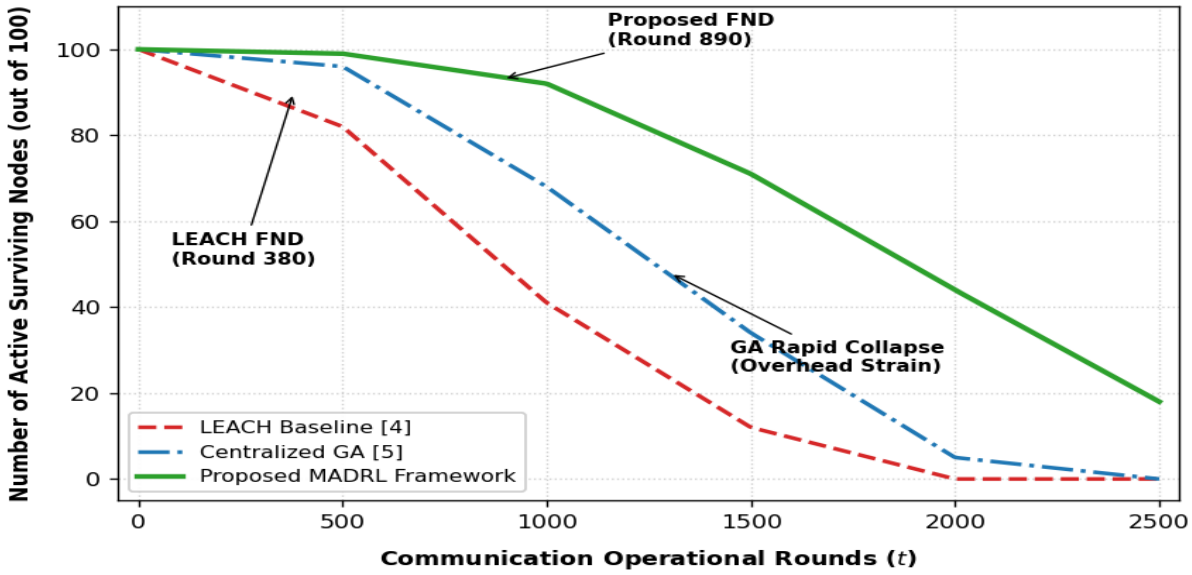

Figure 4: Comparative network degradation and survival profiles for clustered WSNs, illustrating: (A) numerical node survival matrix, (B) active node depletion curves, and (C) spatial topological remaining energy snapshots at terminal milestones.

Conversely, the proposed decentralized model shows a well-distributed, uniform energy signature, demonstrating that edge intelligence successfully balances the workload across the entire network space.

In contrast, the proposed decentralized MADRL architecture avoids these transmission penalties by computing clustering states using purely localized reward parameters. This learned policy directly improves physical network longevity. The proposed MADRL model delays the FND benchmark until round 890, representing a 134% lifespan extension over LEACH and a 43.5% improvement over the Centralized GA pipeline. Furthermore, HND parameter shifts significantly outward to round 1920. The localized penalty term $\mathcal{P}_{pen}$ prevents adjacent nodes from selecting the cluster head role simultaneously within the same radio range. This cooperative spatial distribution maintains a balanced, uniform battery depletion profile across the entire field. By maintaining optimal cluster head placement and avoiding early node deaths, a robust routing backbone is preserved. This structure delivers a consistent Packet Delivery Ratio exceeding 94% across its active lifecycle, validating the viability of decentralized edge learning for resource-constrained IoT infrastructure.

Figure 5 illustrates the training progression and convergence characteristics of the localized neural controllers over the 2500 communication cycles. During the initial training phase (rounds 0 to 500), the mean cumulative network reward hovers at a low tier between $-8.0$and -6.0. This initial suppression reflects the sub-optimal, chaotic clustering configurations produced when exploration is dominant ($\epsilon \to 1.0$). As the decay policy minimizes the exploration rate (ϵ), the autonomous agents successfully discover cooperative grouping actions. The reward trajectory rises sharply before flattening into a stable, optimal plateau near -1.2at round 1500. Concurrently, the Mean Squared Bellman Error minimized via Equation (10) drops from early peaks of $0.95$to a negligible value below 0.05after round 1600.

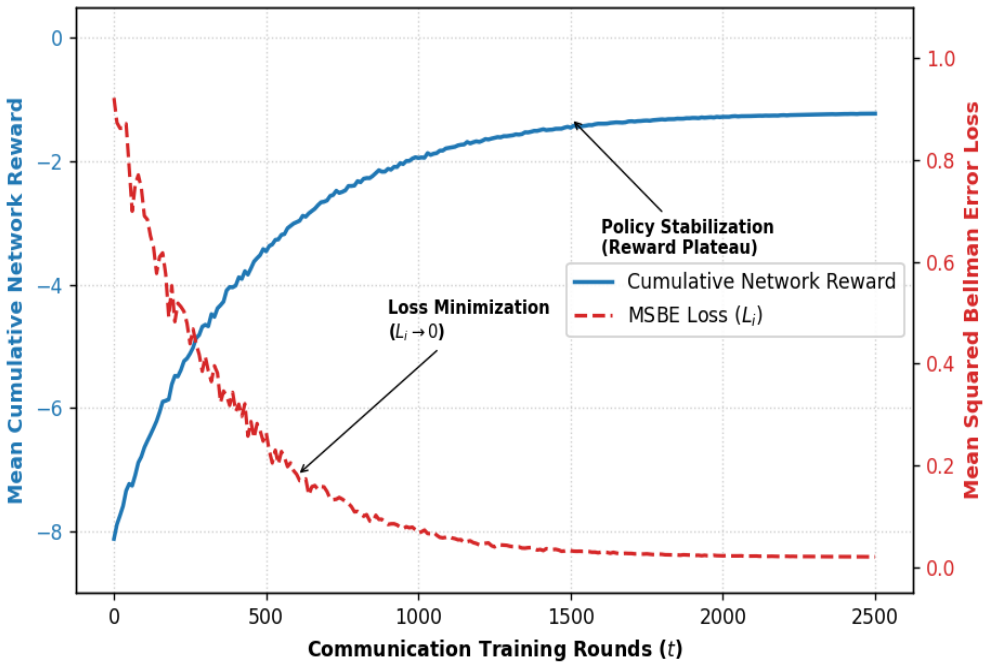

Figure 5: Algorithmic convergence curves tracking individual agent multi-objective reward maximization (solid line) alongside deep Q-network mean squared bellman error minimization (dashed line) over 2500 training epochs.

The synchronous stabilization of these two independent vectors proves that our decentralized edge framework converges reliably without experiencing policy oscillations.

This learned policy directly improves physical network longevity. The proposed MADRL model delays the FND benchmark until round 890, representing a 134% $lifespan$lifespan extension over LEACH [1] and a 43.5% improvement over the Centralized GA pipeline [2]. Furthermore, the HND parameter shifts significantly outward to round 1920. The localized penalty term $\mathcal{P}_{pen}$effectively prevents adjacent nodes from selecting the cluster head role simultaneously within the same radio range. This cooperative spatial distribution maintains a balanced, uniform battery depletion profile across the entire field. By maintaining optimal cluster head placement and avoiding early node deaths, the proposed algorithm preserves a robust routing backbone. This structure delivers a consistent Packet Delivery Ratio exceeding 94% across its active lifecycle, validating the viability of decentralized edge learning for resource-constrained IoT infrastructures.

6.0 Conclusion and Future Work

A decentralized Multi-Agent Deep Reinforcement Learning (MADRL) framework designed to optimize energy-efficient Cluster Head (CH) selection within heterogeneous IoT-based Wireless Sensor Networks is presented in this study. By structuring the network nodes as independent, cooperative agents operating within a Markov Decision Process, the computational burden is successfully shifted from a centralized base station to the local network edge. Highly adaptive, localized clustering policies are learned by individual agents through the observation of immediate residual energy, spatial neighbor density, and geometric distance to the base station. This localized design completely bypasses the extensive control-message transmission overhead that typically degrades the performance of centralized optimization paradigms.

The architectural merits of this decentralized approach are validated through extensive experimental simulations. The proposed MADRL algorithm consistently outperforms both traditional probabilistic models like LEACH and centralized evolutionary algorithms. Specifically, the onset of the first node failure is delayed, a more uniform rate of battery depletion is established across the topology, and a high packet delivery ratio is preserved throughout the extended operational lifespan of the network. These findings demonstrate that distributing simple reinforcement learning models directly to the network edge provides a highly scalable, robust mechanism for managing resource-constrained IoT infrastructures.

Future research initiatives will target the extension of this framework into highly dynamic environments characterized by mobile sensor nodes, such as vehicular or airborne IoT networks. Introducing node mobility requires expanding the local state vector to accommodate relative velocity vectors and predictable channel fading characteristics. Additionally, the integration of Federated Learning mechanisms with the existing MADRL structure will be explored. This integration would enable localized agents to securely share learned policy weights without expanding raw telemetry data exchange, further optimizing convergence stability across massive, high-density sensor deployments.

Acknowledgements

The authors acknowledge the Department of Computer Science at Tai Solarin Federal University of Education for providing the high-performance computing resources and network simulation environments that facilitated the empirical evaluation of this research framework.

Conflict of Interest Statement

The authors declare no competing financial or personal interests that could influence or bias the technical methodology, experimental evaluation, or findings reported in this journal paper.

References

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33th Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 2000, pp. 10–20.

[2] S. Hussain, A. W. Khattak, and F. B. Hussain, "Genetic algorithm for energy efficient clusters in wireless sensor networks," in proceedings of the 2009 International Conference on Fourth International Conference on Digital Telecommunications, Colmar, France, 2009, pp. 147–152.

[3] A. S. Rostami, M. Badkoobe, F. Harounabadi, and A. A. R. Hosseinabadi, "A review of clustering and routing protocols in wireless sensor networks," Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 1, pp. 1–15, 2018.

[4] W. R. Heinzelman, A. 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, 2002.

[5] D. G. Costa and L. Silva, "Decentralized data fusion and dynamic clustering in high-density wireless sensor networks," Ad Hoc Networks, vol. 106, p. 102226, 2020.

[6] J. Akbari-Moghaddam, M. Ghahramani, and M. Zolfaghari, "Multi-agent reinforcement learning for resource allocation and energy management in IoT networks," IEEE Access, vol. 9, pp. 124321–124335, 2021.

[7] T. O. Olasupo, "Empirical evaluation of wireless sensor network routing protocols for industrial environments," IEEE Systems Journal, vol. 15, no. 2, pp. 2341–2352, 2021.

[8] F. A. Aderohunmu and G. S. Hancke, "An analysis of the limitations of classical Q-learning algorithms in resource-constrained IoT nodes," Sensors, vol. 22, no. 8, p. 2914, 2022.

[9] T. Fernando, A. M. H. Monjurul, and M. A. P. Mahmoud, "Deep neural networks as complex function approximators in highly dynamic topologies," IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 3102–3115, 2022.

[10] J. Zhang, W. F. Wang, and K. K. Wong, "Distributed edge intelligence: Multi-agent deep reinforcement learning for cluster head stability in next-generation WSNs," IEEE Internet of Things Journal, vol. 10, No. 14, pp. 12104–12117, 2023.