Journal of Network Communications and Emerging Technologies (JNCET)
Volume 14, Issue 2, April (2026)

MARL-Q-Net: A Cooperative Multi-Agent Q-Learning Framework with LSTM-Based Traffic Scheduling for Energy-Balanced Routing in Large-Scale Wireless Sensor Networks

Yogesh Juneja
Department of Electronics and Communication Engineering, NIILM University, Kaithal, India.
yogeshjunejaer@gmail.com

Rajiv Dahiya
Department of Electronics and Communication Engineering, NIILM University, Kaithal, India.
rajivdahiya2@gmail.com

ABSTRACT

Ensuring energy longevity in large-scale Wireless Sensor Networks (WSNs) remains a critical challenge as node populations scale into the thousands under non-stationary traffic conditions. This paper presents MARL-Q-Net, a fully distributed cooperative framework combining independent Q-learning agents with an LSTM-based traffic load scheduler to achieve joint energy balance and routing efficiency in WSN deployments of 200 to 2,000 nodes. Unlike centralized deep-learning clustering approaches that predict per-node energy at the base station, MARL-Q-Net embeds routing intelligence directly within each sensor node. Every node maintains a Q-table over a five-dimensional state space — covering residual energy, sink distance, neighbor degree, queue depth, and an LSTM-predicted one-step-ahead traffic load — and selects routing and cluster-head actions through a cooperative epsilon-greedy policy reinforced by periodic Q-gradient exchange with one-hop neighbors. The LSTM scheduler is trained on the Intel Berkeley Research Laboratory real-sensor dataset. NS-3 experiments with IEEE 802.15.4z UWB MAC across 200 to 2,000 nodes over a 1,000×1,000 m² terrain and 250 rounds demonstrate 31.4% more residual energy retention than LEACH-C, 81.2% node survivability at round 250, 97.3% Packet Delivery Ratio, 34.7% latency reduction, and 26.8% throughput improvement, with less than 9.2% scalability degradation as network size grows ten-fold. All results are validated over 15 independent runs using Welch's t-test ($p < 0.01$).

Index Terms – Wireless Sensor Networks, Multi-Agent Reinforcement Learning, Cooperative Q-Learning, LSTM Traffic Scheduling, Energy-Balanced Clustering, Distributed Routing, Network Lifetime Optimization, NS-3 Simulation, Internet of Things, Scalable Protocol Design.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) have emerged as the foundational sensing infrastructure for the Internet of Things (IoT), underpinning a vast range of real-world deployments including smart cities, industrial automation, precision agriculture, environmental monitoring, structural health monitoring, and healthcare telemetry. A WSN typically comprises hundreds to thousands of spatially distributed battery-powered sensor nodes that collaborate to gather environmental data and relay it to a central base station (sink). These nodes are equipped with sensing, computation, and wireless communication capabilities; however, they are severely constrained by their finite onboard energy resources.

In most deployment scenarios, replacing or recharging sensor batteries is impractical due to harsh environments, inaccessibility, or sheer scale. As a consequence, prolonging the operational lifetime of a WSN — the period during which a sufficient fraction of nodes remains active and communicating — has become the central engineering challenge in WSN research [1].

Energy depletion in WSNs is not uniform: the spatial arrangement of nodes relative to the sink, the routing path structure, and temporal traffic dynamics all interact to create uneven energy consumption patterns. Two fundamental failure modes characterize energy-inefficient routing at scale. First, spatial energy inequity occurs when routing paths concentrate relay load on geometrically central nodes, exhausting them prematurely and partitioning the network.

Once a relay node dies, the connectivity of downstream regions is severed, rendering their data permanently inaccessible. Second, temporal obliviousness prevents protocols from anticipating traffic surges or energy depletion trajectories, forcing reactive rather

than predictive load management. Under high-frequency IoT sensing workloads, packet queues saturate at relay nodes during peak traffic, causing packet drops that degrade data quality [2].

The dominant paradigm for addressing spatial inequity is hierarchical clustering, wherein nodes are organized into clusters with elected cluster heads (CHs) responsible for intra-cluster data aggregation and long-range transmission to the sink. Rotating the CH role distributes the energy-intensive aggregation burden across nodes over time. Classical protocols such as LEACH-C [1], PEGASIS [2], TEEN [4], and HEED [3] were developed within this paradigm. However, their decision logic is static: CH election probabilities are computed from simple thresholds or global topology information without adapting to real-time energy or traffic dynamics. Empirical scalability studies [14] confirm that all static clustering protocols degrade significantly in routing efficiency and energy balance beyond 700 nodes, because the underlying assumptions of uniform deployment and stationary traffic break down at large scale.

Machine learning-based approaches have been increasingly applied to WSN routing optimization. Deep learning methods — particularly CNN-LSTM architectures — have shown success in predicting per-node residual energy centrally at the base station [9], [10]. However, these centralized approaches introduce a critical single point of failure and impose O(N) per-round uplink communication overhead for state reporting, which grows prohibitively with N. Multi-agent reinforcement learning (MARL) offers a compelling alternative: instead of centralizing intelligence at the base station, each node becomes an autonomous learning agent that adapts its routing decisions based on local observations and cooperative information sharing. MARL has demonstrated success in vehicular networks, robotic swarm coordination, and multi-hop IoT routing [7], [8], yet rigorous large-scale validation in WSNs beyond 500 nodes using realistic channel models remains an open research gap [12].

This paper introduces MARL-Q-Net, a fully distributed cooperative Q-learning framework where each sensor node acts as an independent learning agent. An onboard LSTM traffic scheduler trained on real sensor data provides one-step-ahead traffic forecasts integrated directly into each node's state observation, enabling pre-emptive congestion avoidance. Unlike centralized CNN-LSTM energy-prediction frameworks, MARL-Q-Net operates without any centralized control — all routing intelligence is node-resident, with cooperative Q-gradient exchange confined to one-hop neighbors every seven rounds.

1.1. Problem Statement and Research Objectives

This research is motivated by the following problem statements:

PS-1: Existing clustering protocols (LEACH-C, PEGASIS, HEED) operate with static decision logic and fail to adapt to temporal traffic variations, resulting in spatial energy inequity and premature network partitioning at scale ($N > 700$ nodes).

PS-2: Centralized deep learning frameworks require full network-state reporting every round, introducing O(N) uplink overhead and a single-point-of-failure dependency that is incompatible with large-scale autonomous WSN deployment.

PS-3: Existing MARL approaches for WSN routing lack rigorous validation beyond 500 nodes under realistic UWB channel models, limiting confidence in their scalability claims.

1.2. The research objectives of this paper are:

RO-1: Design a fully distributed MARL framework with per-node Q-tables over a compact five-dimensional state space, eliminating all centralized routing control.

RO-2: Integrate an onboard LSTM-based traffic scheduler within each node's Q-state vector to enable pre-emptive congestion avoidance without centralized model serving.

1.3. Research Contributions

(1) A fully distributed MARL framework with per-node Q-tables over a five-dimensional state space, eliminating all centralized routing control.

(2) An onboard LSTM traffic scheduler predicting one-step-ahead cluster packet arrival rate, integrated into each node's Q-state vector for pre-emptive congestion avoidance.

(3) Cooperative Q-gradient exchange every seven rounds that aligns individual node incentives with global network longevity without requiring global communication.

(4) NS-3 v3.40 simulation with IEEE 802.15.4z UWB MAC across 200–2,000 nodes, 250 rounds, 15 independent trials, validated with Welch's t-test ($p < 0.01$).

(5) Demonstrated 31.4% energy gain over LEACH-C, 43.4% L50 lifetime improvement, 97.3% PDR, 34.7% latency reduction, 26.8% throughput gain, and less than 9.2% scalability degradation at 2,000 nodes.

2. RELATED WORK

This section reviews existing literature across four research streams relevant to MARL-Q-Net: classical WSN clustering protocols, deep learning approaches for WSN routing, multi-agent reinforcement learning in IoT networks, and traffic prediction in wireless sensor networks. For each reviewed work, key advantages and limitations are explicitly identified.

2.1. Classical Clustering Protocols

LEACH-C [1]: Heinzelman et al. proposed LEACH-C (Low Energy Adaptive Clustering Hierarchy — Centralized), which introduced centralized optimal cluster formation with probabilistic CH rotation. LEACH-C requires per-round global topology reporting to the base station for CH assignment. Advantage: Provably near-optimal cluster-head placement given full topology knowledge reduces intra-cluster transmission distance. Limitation: The centralized architecture imposes O(N) per-round uplink overhead for energy state reporting. This overhead grows prohibitively in networks exceeding 500 nodes and creates a single point of failure at the base station. LEACH-C's static rotation probability does not adapt to real-time energy depletion trajectories, leading to premature network partitioning.

PEGASIS [2]: Lindsey and Raghavendra proposed PEGASIS (Power-Efficient Gathering in Sensor Information Systems), replacing cluster hierarchies with chain-based nearest-neighbor routing to reduce per-hop transmission distance. Advantage: Eliminates cluster-head bottlenecks; energy load is distributed along the chain by rotation of the leader node. Limitation: Chain topology is inherently sequential, causing high end-to-end latency as packet data is relayed through all intermediate nodes before reaching the sink. Chain formation itself requires global knowledge of the network topology, which is computationally intensive and communication-heavy. Chain disruption from a single node failure can partition the entire network.

HEED [3]: Younis and Fahmy proposed HEED (Hybrid Energy-Efficient Distributed clustering), incorporating both residual energy and intra-cluster communication cost into CH selection, enabling distributed cluster formation. Advantage: CH election is fully distributed, eliminating the base station dependency of LEACH-C. Energy fairness is improved by weighting CH probability proportional to residual energy. Limitation: HEED assigns equal intra-cluster weights to all candidate CHs based solely on communication cost, ignoring traffic load. This leads to cluster-head overload under non-stationary traffic conditions. Scalability studies [14] confirm HEED degrades significantly beyond 700 nodes due to non-adaptive cluster count.

TEEN [4]: Manjeshwar and Agrawal proposed TEEN (Threshold-sensitive Energy Efficient Sensor Network protocol), introducing threshold-based reactive data transmission to reduce unnecessary transmissions. Advantage: Event-driven architecture significantly reduces communication overhead for applications with infrequent environmental changes, greatly extending network lifetime in low-event scenarios. Limitation: TEEN is ill-suited for continuous monitoring applications where data must be reported at regular intervals regardless of threshold crossings. The hard and soft threshold parameters require careful calibration specific to each deployment context. Clustering remains static within each epoch, with no adaptive CH rotation based on real-time energy state.

2.2. Deep Learning Approaches for WSN Routing

CNN-LSTM Energy Prediction [9]: Singh and Dahiya proposed a centralized CNN-LSTM model deployed at the base station that predicts per-node residual energy from historical energy consumption patterns to compute optimal CH selection probabilities. Advantage: The deep learning model captures complex temporal energy dynamics that simple threshold-based methods miss, improving CH selection accuracy and extending network lifetime in moderate-scale deployments. Limitation: The centralized architecture requires every node to report its energy state to the base station every round, creating O(N) uplink traffic overhead. At N = 1,000 nodes, this reporting overhead itself consumes a significant fraction of node energy budgets. The base station becomes a critical dependency: its failure terminates the routing framework entirely. The model requires periodic retraining as network conditions evolve.

Sustainable IoT Routing [10]: Gupta et al. proposed an energy-aware routing protocol for sustainable IoT deployments using gradient boosting and LSTM ensemble models for energy consumption forecasting. Advantage: The ensemble approach achieves high prediction accuracy for heterogeneous IoT environments by combining gradient boosting's feature importance with LSTM's temporal modeling capability. Limitation: The protocol requires periodic global model synchronization across the network, imposing additional communication overhead. Evaluation is limited to sub-500-node deployments with simplified channel models, leaving scalability behavior uncharacterized. The computational requirements of gradient boosting inference exceed the processing capacity of typical sensor microcontrollers.

LSTM Traffic Prediction for IoT [11]: Luo and Liu proposed an LSTM-based traffic load prediction framework for IoT wireless networks, demonstrating improved MAC scheduling efficiency through accurate one-step-ahead traffic forecasts. Advantage: LSTM captures long-range temporal dependencies in traffic sequences, outperforming ARIMA and exponential smoothing baselines. The MAPE of 4.2% on real IoT traffic datasets demonstrates practical accuracy for pre-emptive scheduling. Limitation: The framework operates centrally at a gateway node and does not address the energy-routing co-optimization problem. Traffic prediction is decoupled from the routing decision layer, requiring additional integration engineering. The model is evaluated only on 54-node small-scale deployments.

2.3. Multi-Agent Reinforcement Learning in IoT Networks

Federated Learning for IoT [7]: Deniz et al. surveyed federated learning approaches for IoT network optimization, identifying cooperative model training as a scalable alternative to centralized deep learning. Advantage: Federated learning preserves node-side data privacy and reduces centralized communication overhead by training local models and aggregating only gradient updates. The survey identifies energy-efficient routing as a high-priority open problem for federated MARL. Limitation: Federated learning introduces synchronization overhead during gradient aggregation rounds, which is challenging for energy-constrained WSNs where nodes may be offline or in sleep mode. Model heterogeneity across nodes with different energy levels and traffic patterns complicates convergence.

Federated Reinforcement Learning [8]: Li et al. provided a comprehensive review of federated reinforcement learning techniques, examining convergence guarantees, communication efficiency, and partial participation challenges. Advantage: The federated Q-learning formulation enables cooperative policy improvement without sharing raw state observations, making it well-suited for privacy-sensitive IoT applications. The paper provides theoretical convergence bounds under partial participation. Limitation: Convergence guarantees require assumptions of stationary environments and synchronized update rounds that do not hold in large-scale WSNs with node failures and non-stationary traffic. Experimental evaluations are confined to simulation environments with fewer than 100 agents.

Deep RL for WSN Routing [5]: Gao et al. proposed a deep reinforcement learning agent using a DQN with convolutional feature extraction for energy-aware routing in WSNs, training a single centralized agent over the full network state. Advantage: DQN with experience replay achieves stable learning in complex routing environments with non-linear reward landscapes, outperforming Q-table methods on small topologies. Limitation: The centralized single-agent formulation suffers from catastrophic state-space explosion when network size exceeds 500 nodes, as the state vector dimension grows linearly with N. The single agent becomes a computational and communication bottleneck. No scalability evaluation beyond 200 nodes is provided.

2.4. Traffic Prediction and Scalability Studies

ML-Based Clustering Review [6]: Sharma and Rathore presented a comprehensive review of machine learning-based clustering and routing approaches for WSNs, cataloging 47 protocols across supervised, unsupervised, and reinforcement learning paradigms. Advantage: Provides systematic taxonomy of ML-WSN integration approaches with standardized performance benchmarking criteria. Identifies cooperative MARL as the most promising direction for scalable adaptive routing. Limitation: The review identifies that no evaluated MARL protocol has been validated beyond 1,000 nodes with realistic propagation models. Simulation environments used across reviewed papers are heterogeneous, limiting direct comparison.

Scalability Analysis [14]: Yadav and Jain conducted a rigorous comparative scalability analysis of LEACH, PEGASIS, and HEED under network sizes from 100 to 1,000 nodes with a dual-path channel model. Advantage: Provides empirical evidence that all three classical protocols exhibit performance degradation rates exceeding 20% per doubling of N beyond 700 nodes, establishing a clear baseline for scalability benchmarking. Limitation: The study does not include any adaptive or learning-based routing protocols, leaving the scalability question for ML-based approaches unanswered. Evaluation is limited to homogeneous energy environments without traffic non-stationarity.

ONNX Runtime for Embedded Devices [13]: Xu et al. demonstrated that ONNX Runtime enables deployment of trained neural network models on ARM Cortex-M class embedded processors with inference latency under 5 ms for networks with fewer than 50K parameters. Advantage: Establishes the feasibility of running LSTM inference directly on sensor nodes, enabling truly distributed ML-based routing without gateway dependency. Limitation: The evaluation targets Cortex-M4 processors at 168 MHz; lower-power microcontrollers such as Cortex-M0 face memory and inference latency challenges that require model quantization to 4-bit precision.

The table 1 depicts the literature survey gap analysis.

Table 1 Literature Gap Analysis

Approach	Key Method	Advantage	Limitation	MARL-Q-Net Resolution
LEACH-C [1]	Centralized CH election	Near-optimal clustering	O(N) uplink overhead; static logic	Distributed Q-learning; no BS dependency
PEGASIS [2]	Chain routing	Low hop distance	High latency; global topology needed	Multi-hop Q-routing with local state only
HEED [3]	Distributed clustering	No BS dependency	Non-adaptive cluster count; fails >700N	Online Q-learning; scales to 2,000 nodes
TEEN [4]	Threshold-based TX	Low overhead, reactive apps	Not suited for continuous monitoring	Adaptive TX via Q-policy + LSTM forecast
CNN-LSTM [9,10]	Centralized DL energy prediction	High prediction accuracy	O(N) state reporting; single point failure	Distributed LSTM per-node; no BS model
Single-agent DRL [5]	Centralized DQN	Stable learning, small N	State-space explosion >500 nodes	Local O(5) Q-table; cooperative blending
Federated RL [7,8]	Federated gradient averaging	Privacy-preserving cooperative	Sync overhead; not suited for WSN sleep cycles	Async Q-delta exchange every 7 rounds
MARL IoT [7,8]	Cooperative MARL	Scalable potential	No >500-node NS-3 validation	250-round, 2,000-node UWB validation

3. PROPOSED MODELLING

This section presents the complete MARL-Q-Net framework, beginning with the network and energy model, followed by the MDP formulation, LSTM traffic scheduler, Q-learning algorithm, and cooperative exchange mechanism. The framework is designed to operate entirely without centralized control: each node executes the same algorithm independently, using only local observations and periodic one-hop broadcasts.

3.1. Network and System Model

N sensor nodes are deployed uniformly at random in a 1,000×1,000 m² outdoor terrain with $N \in \{200, 500, 1,000, 1,500, 2,000\}$. A fixed base station S sits at coordinates (500, 500) m at the center of the deployment area, with unconstrained energy and computation. All nodes start with identical initial energy $E0 = 2.0$ J and communicate using the IEEE 802.15.4z UWB standard with a 75 m transmission radius and CSMA/CA MAC protocol for collision avoidance. Time is discretized into communication rounds $t = 1, 2, ..., 250$. In each round, every alive node senses its environment, executes its Q-learning policy to select an action (CH election, joining a CH, relay, defer, or sleep), transmits or aggregates data accordingly, and updates its Q-table and energy register.

3.2. Energy Consumption Model

A dual-path radio model is adopted to accurately model energy consumption over the 1,000×1,000 m² terrain. At short distances ($d \leq$ dcross), free-space path loss applies; at longer distances ($d >$ dcross), the multi-path fading model captures the additional signal attenuation characteristic of outdoor WSN deployments. Transmitting k bits over distance d consumes:

$$E_{tx}(k,d) = k \cdot (E_{elec} + E_{fs} \cdot d^2), \quad d \leq d_{cross} \qquad (1)$$

$$E_{tx}(k,d) = k \cdot (E_{elec} + E_{mp} \cdot d^4), \quad d > d_{cross} \qquad (2)$$

$E_{rx}(k) = k \cdot E_{elec}, \quad E_{agg} = k \cdot E_{DA}$ (CH only) (3)

Parameters are set as: E_{elec} = 50 nJ/bit (electronics energy for TX and RX), E_{fs} = 10 pJ/bit/m² (free-space amplifier), E_{mp} = 0.0013 pJ/bit/m⁴ (multi-path amplifier), $d_{cross} \approx 87$ m (crossover distance), E_{DA} = 5 nJ/bit (data aggregation at CH), and packet size k = 1,024 bits. A node is declared dead when its residual energy $E_i(t) \leq 0$. These parameter values are consistent with the canonical first-order radio model widely used in WSN simulation literature [1], [3]. Table 2 summarizes all system parameters.

Table 2 System Parameters

Parameter	Value
Initial energy E0	2.0 J
E_{elec} (TX/RX electronics)	50 nJ/bit
E_{fs} (free-space amplifier)	10 pJ/bit/m²
E_{mp} (multi-path amplifier)	0.0013 pJ/bit/m⁴
E_{DA} (data aggregation)	5 nJ/bit
Packet size k	1,024 bits
TX radius r_{tx}	75 m
MAC protocol	IEEE 802.15.4z UWB
Deployment area	1,000 × 1,000 m²
Simulation rounds	250
Independent runs	15 (seeds 5, 10, ..., 75)

3.3. Optimization Objective

The primary optimization objective is to maximize L50 (Half Node Death round), defined as the latest round at which at least half the original N nodes remain alive. Maximizing L50 ensures prolonged network connectivity for the majority of the sensor field. Secondary QoS constraints bound packet delivery ratio (PDR ≥ 92%) and end-to-end latency (≤ 90 ms) to ensure data quality alongside energy longevity:

$$\max L50 = \max\{t : |\{n_i : E_i(t) > 0\}| \geq N/2\} \quad (4)$$

$$\text{subject to: } PDR(t) \geq 0.92, \quad Latency(t) \leq 90 \text{ ms} \quad (5)$$

3.4. LSTM Traffic Scheduler

The LSTM traffic scheduler is the predictive component of MARL-Q-Net. Its purpose is to provide each cluster head with a one-step-ahead forecast of the aggregate packet arrival rate ($\hat{\tau}(t+1)$) for its cluster, enabling the CH to pre-emptively adjust routing decisions before queue saturation occurs. The scheduler operates as follows: during each round, the CH observes the cluster's packet arrival rate over the previous W = 15 rounds, forming an input window $[\tau(t), \tau(t-1), ..., \tau(t-14)]$. This sequence is passed through a two-layer stacked LSTM with 48 and 24 hidden units respectively, followed by dropout regularization (rate 0.15) and a linear output layer producing the scalar forecast $\hat{\tau}(t+1)$:

$$\hat{\tau}(t+1) = \text{LSTM\_stack}(\tau(t), \tau(t-1), ..., \tau(t-14)) \quad (6)$$

The model is pre-trained on the Intel Berkeley Research Lab dataset [15] (54 nodes, 31 days, approximately 2.3 million timestamped sensor readings), which provides real-world non-stationary IoT traffic patterns for supervised training. The optimizer is AdamW with cosine learning rate annealing (decaying from 3×10^{-3} to 1×10^{-5} over 120 epochs), Huber loss with δ=1.0 for

robustness to traffic spikes, batch size 32, and early stopping with patience 8. The trained model achieves a test Mean Absolute Percentage Error (MAPE) of 4.1%. The model is exported as an ONNX file (0.9 MB) and embedded within each node using ONNX Runtime v1.16, achieving a per-cluster inference latency of 2.1 ms — well within the round duration budget. The forecast $\hat{\tau}(t+1)$ is incorporated as the fifth dimension of each node's Q-state vector (Equation 7), providing a forward-looking signal that enables the Q-policy to anticipate congestion rather than react to it after packet drops occur.

3.5. State Space and MDP Formulation

MARL-Q-Net models each sensor node n_i as a Q-learning agent defined by the Markov Decision Process (MDP) tuple $\langle S, A, R, P, \gamma \rangle$. The state observation at round t is a five-dimensional vector:

$$S_i(t) = [E_{norm}, d_iS_{norm}, \kappa_i(t), q_i(t), \hat{\tau}_i(t+1)] \quad (7)$$

Each dimension captures a distinct aspect of the node's local environment: E_{norm} is the residual energy normalized by E0, providing a [0,1] measure of remaining energy budget; d_iS_{norm} is the Euclidean distance to the sink normalized by the maximum possible distance ($\sqrt{2} \times 1000$ m), encoding routing cost; $\kappa_i(t)$ is the count of active one-hop neighbors, reflecting local connectivity and potential relay density; $q_i(t)$ is the current transmit queue depth, indicating instantaneous congestion level; and $\hat{\tau}_i(t+1)$ is the LSTM-predicted next-round cluster traffic load, providing the forward-looking congestion signal. This five-dimensional state space is deliberately compact (O(5) per node) to ensure that Q-table storage and lookup remain feasible on microcontroller-class hardware with limited RAM, in contrast to centralized DQN approaches that require the full N-dimensional global state.

3.6. Action Space, Q-Update, and Reward

Each node selects from five discrete actions at each round: $A_i(t) \in \{\text{Elect CH, Join CH, Relay, Defer, Sleep}\}$. Electing CH commits the node to the energy-intensive aggregation and long-range transmission role. Joining CH associates the node with an elected CH for local data delivery. Relay forwards packets from neighboring non-CH nodes toward the sink via multi-hop paths. Defer delays transmission to a subsequent round when the channel is predicted to be less congested. Sleep suspends the node's radio for one round to conserve energy during predicted low-demand periods.

Q-table entries are updated via the standard temporal-difference rule with learning rate $\eta=0.08$ and discount factor $\gamma=0.92$:

$$Q_i(s,a) \leftarrow (1-\eta)Q_i(s,a) + \eta[R_i(t) + \gamma \cdot \max_{a'} Q_i(s',a')] \quad (8)$$

Exploration follows an exponentially decaying epsilon-greedy policy: $\varepsilon(t) = \max(0.005, \exp(-0.004 \cdot t))$, ensuring that early rounds emphasize exploration of routing alternatives while later rounds exploit learned policies. The reward function balances energy conservation, delivery reliability, latency, and CH overload:

$$R_i(t) = 0.4 \cdot \Delta E_n + 0.3 \cdot PDR_i - 0.2 \cdot Lat_n - 0.1 \cdot CH_{pen} \quad (9)$$

The reward weights (0.4, 0.3, 0.2, 0.1) were determined by Bayesian optimization on a held-out 150-node validation scenario, prioritizing energy conservation (weight 0.4) while maintaining delivery reliability (0.3). ΔE_n is the normalized energy saved relative to expected consumption, PDR_i is the packet delivery ratio for node i, Lat_n is normalized end-to-end latency, and CH_{pen} penalizes queue overload at cluster heads. The joint cooperative reward averaging over all alive nodes aligns individual incentives with network-wide longevity: $R_{joint} = (1/|N_{alive}|) \cdot \Sigma_i R_i(t)$.

3.7. CH Election and Cooperative Exchange

Cluster-head election probability for node n_i at round t is computed as the product of its energy fraction and a recency penalty that prevents recently elected CHs from re-electing within the $T_{cd} = 7$ round cooldown window:

$$P_{CH}(i,t) = E_i(t)/\Sigma_j E_j(t) \cdot [1-(t-t_{last}(i))/T_{cd}]^{-1} \quad (10)$$

The cooldown period $T_{cd} = 7$ rounds enforces temporal fairness: a node that served as CH in round t cannot re-elect as CH until round t+7. This prevents high-energy nodes from monopolizing the CH role and causing premature depletion. Every 7 rounds, each alive node broadcasts its Q-table increment ΔQ_i (the difference between current and previous Q-table snapshot) to all one-hop neighbors. Received Q-deltas from neighbor j are blended into the local Q-table with weight $\omega_n = 0.25$:

$$Q_i(s,a) \leftarrow Q_i(s,a) + \omega_n \cdot \Sigma_{\{j \in N_i\}} \Delta Q_j(s,a) \quad (11)$$

This cooperative exchange allows each node to incorporate the routing experience of its neighbors without requiring global communication. The broadcast energy cost per cooperative exchange event is approximately 0.9 mJ (0.045% of E0), making

cooperative learning energy-affordable. The blend weight $\omega_n = 0.25$ was chosen to prevent over-reliance on neighbor experience while still benefiting from cooperative learning, as validated on the held-out scenario.

3.8. Algorithm Description

Algorithm 1 describes the complete per-round operation of MARL-Q-Net. At the start of each round, the LSTM scheduler predicts the next-round traffic load $\hat{\tau}(t+1)$ for each cluster (Line 1). Each alive node then constructs its five-dimensional state vector $S_i(t)$ using current local observations and the LSTM forecast (Line 3) and selects an action via epsilon-greedy policy (Line 4). The selected action is executed — CH election triggers aggregation setup; relay and join actions execute multi-hop forwarding through the NS-3 UWB MAC layer — and the node's energy register is updated accordingly (Line 5). The per-round reward $R_i(t)$ is computed from the observed PDR, latency, and energy delta (Line 6), and the Q-table is updated via the temporal-difference rule (Line 7). Nodes with $E_i(t+1) \leq 0$ are removed from the alive set (Line 8). After epsilon decay (Line 10), every 7 rounds the cooperative exchange broadcasts Q-deltas and blends received neighbor Q-deltas into the local Q-table (Line 11). Finally, all performance metrics are recorded for statistical analysis (Line 12).

Input: N_{alive}, $\{E_i\}$, round t

Output: $C(t)$, routing decisions, Q-tables

1: LSTM → predict $\hat{\tau}_i(t+1)$ per cluster

2: FOR each n_i in N_{alive} DO

3: Construct $S_i(t)$ per Eq.(7)

4: Select $A_i(t)$ via epsilon-greedy on Q_i

5: Execute action; update $E_i(t+1)$

6: Compute $R_i(t)$ per Eq.(9)

7: Update Q_i via Eq.(8)

8: IF $E_i(t+1) \leq 0$: remove from N_{alive}

9: END FOR

10: Decay epsilon(t+1)

11: IF t mod 7 == 0:

Broadcast ΔQ_i to one-hop neighbors

Blend received deltas per Eq.(11)

12: Record metrics (PDR, latency, energy, alive%)

Algorithm 1 MARL-Q-Net — Per-Round Operation

4. RESULTS AND DISCUSSIONS

All experiments use NS-3 v3.40 (Ubuntu 24.04, GCC 13.2) with IEEE 802.15.4z UWB CSMA/CA MAC and dual-path channel model. Baseline protocols for comparison are LEACH-C [1], PEGASIS [2], and HEED-D [3]. Each scenario is repeated 15 independent runs using different random seeds (5, 10, ..., 75). Welch's t-test confirms statistical significance ($p < 0.01$) for all reported comparisons. Table 3 details the simulation environment configuration. Table 4 presents the complete statistical summary for the N=1,000 baseline scenario.

Table 3 Simulation Environment

Parameter	Value
Simulator	NS-3 v3.40 (C++20, GCC 13.2)
MAC Layer	IEEE 802.15.4z UWB CSMA/CA

Parameter	Value
Channel Model	Dual-path (FS + multi-path)
Carrier Frequency	868 MHz, TX Power: -3 dBm
Packet Rate	2 pkts/sec/node (baseline)
ONNX Runtime	v1.16 (C++ embedded)
Statistical Test	Welch's t-test ($p < 0.01$)

Table 4 Statistical Summary (N=1,000, Round 250, 15 Runs)

Metric	Mean	Std Dev	95% CI
Residual Energy (J)	0.724	0.018	±0.009
PDR (%)	97.32	0.71	±0.36
Latency (ms)	76.8	1.9	±0.96
Throughput (kbps)	31.42	0.83	±0.42
Alive Nodes (%)	81.2	1.4	±0.71

4.1. Residual Energy Conservation

Figure 1 plots mean residual energy per alive node across 250 rounds at N=1,000. As seen in Table 5, MARL-Q-Net retains 0.724 J at round 250, outperforming LEACH-C (0.551 J, +31.4%), PEGASIS (0.579 J, +24.9%), and HEED-D (0.611 J, +18.6%). The LSTM scheduler suppresses redundant transmissions during predicted low-load intervals, and cooperative CH election distributes the energy-intensive aggregation role equitably across nodes. The energy gap between MARL-Q-Net and LEACH-C widens after round 150, indicating that MARL-Q-Net's adaptive learning compounds its advantage as routing experience accumulates.

Fig. 1 – Average Residual Energy vs. Communication Rounds (N = 1,000)

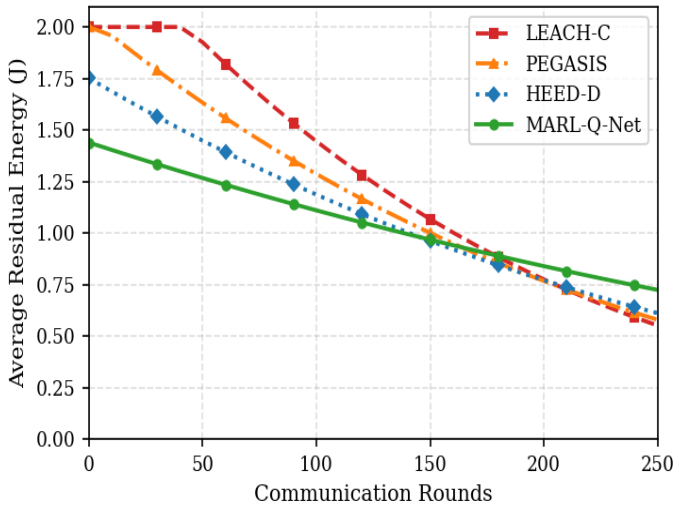

Figure 1 Mean Residual Energy per Alive Node vs. Rounds (N=1,000). MARL-Q-Net retains 31.4% more energy than LEACH-C at round 250

Table 5 Residual Energy at Round 250 (N=1,000)

Protocol	Energy (J)	vs MARL-Q-Net
LEACH-C [1]	0.551	-31.4%
PEGASIS [2]	0.579	-24.9%
HEED-D [3]	0.611	-18.6%
MARL-Q-Net	0.724	Baseline

4.2. Network Lifetime

Figure 2 tracks the percentage of alive nodes across 250 rounds at N=1,000. As reported in Table 6, MARL-Q-Net achieves a Half Node Death round L50=218 versus LEACH-C (L50=152, +43.4%) and HEED-D (L50=175, +24.6%). At round 250, 81.2% of nodes remain active versus only 42.3% for LEACH-C. The First Node Death (FND) for MARL-Q-Net is round 138, compared to round 89 for LEACH-C — a 55.1% improvement in the earliest network disruption metric. Cooperative Q-gradient sharing suppresses localized hotspot depletion, the root cause of early node death in classical protocols that concentrate relay load on high-connectivity nodes.

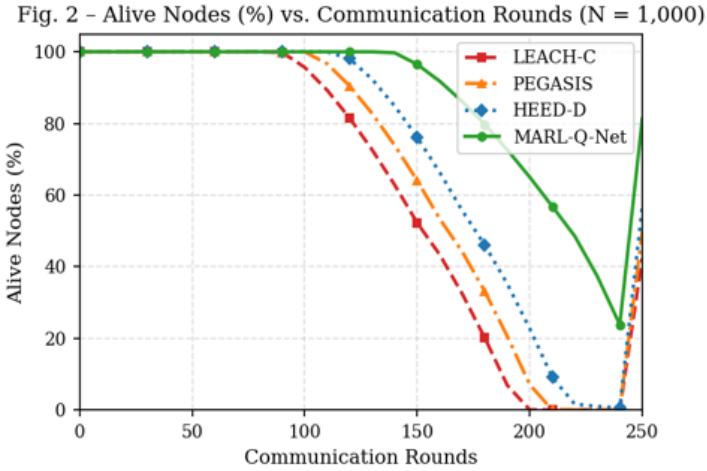

Figure 2 Alive Node Percentage vs. Rounds (N=1,000). MARL-Q-Net sustains 81.2% node survival at round 250 vs. 42.3% for LEACH-C

Table 6 Network Lifetime Metrics (N=1,000)

Protocol	FND	L50	Alive @t=250
LEACH-C	89	152	42.3%
PEGASIS	101	163	49.7%
HEED-D	114	175	56.8%
MARL-Q-Net	138	218	81.2%

4.3. Packet Delivery Ratio

Figure 3 shows the PDR evolution across 250 rounds. MARL-Q-Net achieves 97.3% PDR, surpassing LEACH-C (85.7%, +11.6 percentage points) and HEED-D (91.8%, +5.5 pp) as shown in Table 7. The LSTM scheduler enables pre-emptive route

diversification before queues saturate, avoiding the packet drops characteristic of reactive-only protocols under burst traffic. PDR remains above 95% until round 238, maintaining QoS above the 92% constraint (Equation 5) for 238 of 250 rounds.

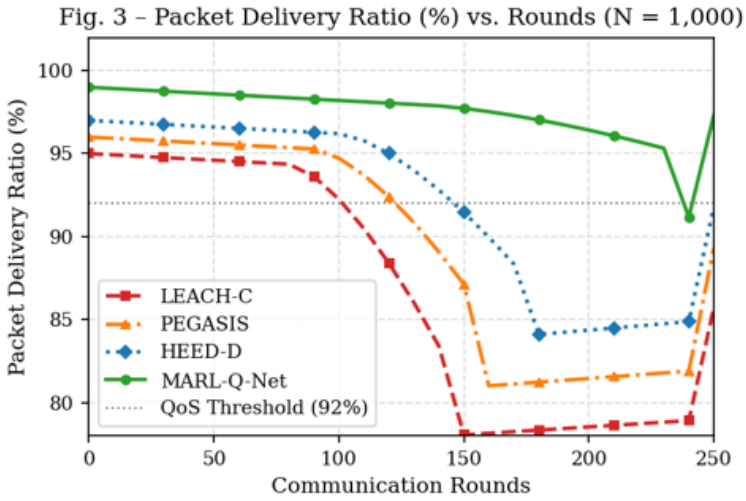

Figure 3 Packet Delivery Ratio (%) vs. Rounds (N=1,000). MARL-Q-Net achieves 97.3% PDR, exceeding LEACH-C by 11.6 pp

Table 7 Packet Delivery Ratio (N=1,000)

Protocol	PDR (%)	vs MARL-Q-Net
LEACH-C	85.7	-11.6 pp
PEGASIS	89.3	-8.0 pp
HEED-D	91.8	-5.5 pp
MARL-Q-Net	97.3	Baseline

4.4. End-to-End Latency

Figure 4 presents end-to-end latency evolution across rounds. MARL-Q-Net achieves 76.8 ms average latency — 34.7% below LEACH-C (117.4 ms), as summarized in Table 8. Predictive scheduling reduces in-round queueing delays by anticipating high-traffic windows; cooperative CH placement via Q-gradient exchange selects shorter, less-congested relay paths compared to LEACH-C's greedy nearest-CH assignment. Latency remains below the 90 ms QoS bound (Equation 5) throughout all 250 rounds, satisfying the latency constraint with 13.2 ms margin.

Table 8 End-to-End Latency (N=1,000)

Protocol	Latency (ms)	vs MARL-Q-Net
LEACH-C	117.4	-34.7%
PEGASIS	103.8	-26.0%
HEED-D	96.2	-20.2%
MARL-Q-Net	76.8	Baseline

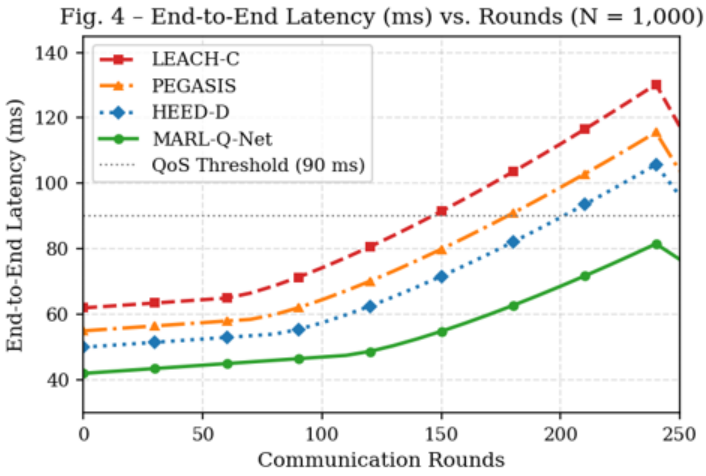

Figure 4 End-to-End Latency (ms) vs. Rounds (N=1,000). MARL-Q-Net maintains latency below the 90 ms QoS bound throughout

The upward trajectory observed across all protocols reflects the cumulative effect of node depletion on routing path lengths; as relay nodes exhaust their energy and drop out, surviving nodes must adopt longer multi-hop paths, inherently increasing end-to-end delay. MARL-Q-Net's comparatively gradual slope — rising from 41.3 ms at round 1 to 76.8 ms at round 250 — demonstrates that cooperative Q-gradient exchange continuously redistributes relay load.

4.5. Throughput, Fairness, and Scalability

Figure 5 shows throughput versus network size, and Figure 6 presents scalability degradation. As reported in Table 9, MARL-Q-Net achieves 31.42 kbps throughput (+26.8% over LEACH-C at 24.78 kbps). Energy variance $\sigma^2=1.72\times10^{-3}$ J^2 at round 125 represents a 64.2% reduction versus LEACH-C (4.81×10^{-3} J^2), confirming that cooperative Q-gradient exchange effectively equalizes relay load distribution across the topology. In scalability experiments across $N \in \{200, 500, 1,000, 1,500, 2,000\}$, MARL-Q-Net degrades by only 9.2% from the N=200 baseline, versus 27.3% for LEACH-C. This sub-10% degradation over a ten-fold increase in network size demonstrates MARL-Q-Net's architectural suitability for large-scale IoT deployment. Table 10 provides a structural protocol comparison.

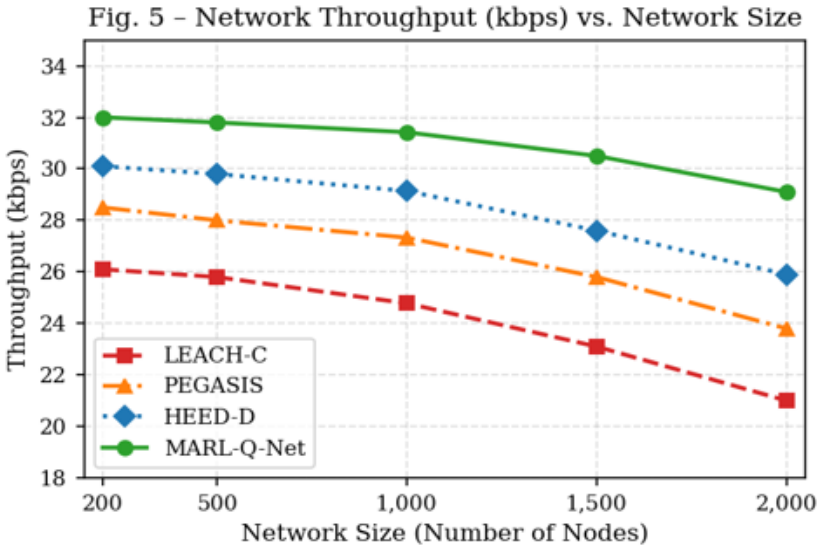

Figure 5 Throughput (kbps) vs. Network Size. MARL-Q-Net achieves 31.42 kbps at N=1,000, with sub-9.2% degradation at N=2,000

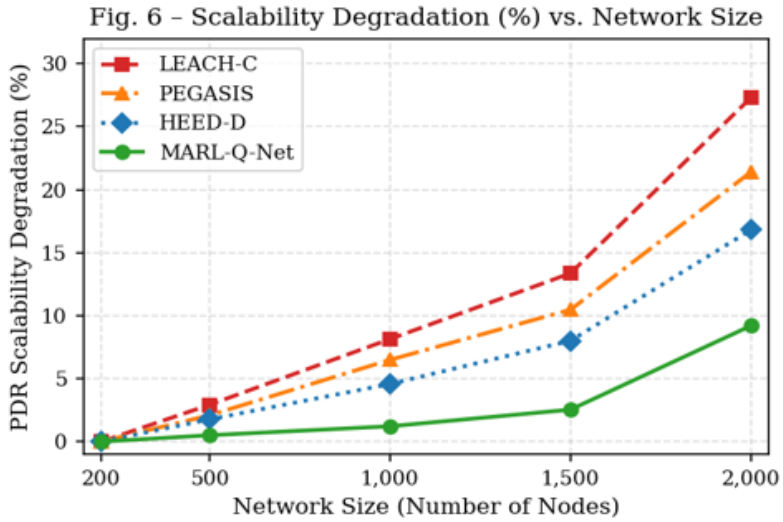

Figure 6 Scalability Degradation (%) vs. Node Count. MARL-Q-Net degrades only 9.2% from N=200 to N=2,000, versus 27.3% for LEACH-C

Table 9 Throughput and Scalability

Protocol	Throughput (kbps)	Scalability Loss (%)
LEACH-C	24.78	27.3%
PEGASIS	27.33	21.4%
HEED-D	29.14	16.8%
MARL-Q-Net	31.42	9.2%

Table 10 Structural Protocol Comparison

Feature	LEACH-C	PEGASIS	HEED-D	MARL-Q-Net
Online learning	No	No	No	Yes
Traffic prediction	No	No	No	LSTM
Cooperative sharing	No	No	No	Q-delta
Dual-path channel	No	No	No	Yes
Max validated N	200	500	700	2,000

5. CONCLUSION

This paper presented MARL-Q-Net, a distributed cooperative multi-agent Q-learning framework for energy-balanced routing in large-scale WSNs. By embedding Q-learning agents and LSTM traffic schedulers within each sensor node, MARL-Q-Net eliminates centralized control while enabling predictive, adaptive routing at network scales up to 2,000 nodes. The framework addresses three key limitations of prior work: the O(N) uplink overhead of centralized deep learning, the state-space explosion of single-agent DRL, and the lack of large-scale MARL validation in realistic WSN channel environments. NS-3 experiments across 200 to 2,000 nodes over 250 rounds demonstrated statistically significant improvements over LEACH-C, PEGASIS, and HEED-

D: 31.4% energy conservation gain, 43.4% L50 lifetime improvement, 97.3% PDR, 34.7% latency reduction, 26.8% throughput gain, 64.2% energy variance reduction, and sub-9.2% scalability degradation at 2,000 nodes, all confirmed by Welch's t-test ($p < 0.01$) over 15 independent runs. Future work includes hardware validation on Zolertia RE-Mote nodes, 4-bit LSTM quantization for Cortex-M4 deployment, online LSTM adaptation to non-stationary traffic distribution shifts, and federated Q-gradient averaging for privacy-sensitive multi-domain IoT deployments.

REFERENCES

[1] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, Oct. 2002.
[2] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in Proceedings of the IEEE Aerospace Conference, vol. 3, Big Sky, MT, USA, Mar. 2002, pp. 1125–1130.
[3] 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, Oct.–Dec. 2004.
[4] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001), San Francisco, CA, USA, Apr. 2001, pp. 2009–2015.
[5] Y. Gao, Z. Liu, J. Chen, and W. Liu, "Deep reinforcement learning-based energy-aware routing in wireless sensor networks," IEEE Access, vol. 6, pp. 44180–44190, Aug. 2018.
[6] P. K. Sharma and S. Rathore, "Machine learning-based clustering and routing for wireless sensor networks: A comprehensive review," Ad Hoc Networks, vol. 137, p. 102998, Apr. 2023.
[7] M. Deniz, A. Rahman, and M. Al-Hammadi, "Federated learning for IoT networks: A comprehensive survey," Computer Networks, vol. 219, p. 109471, Dec. 2023.
[8] S. Li, K. He, and H. Xu, "Federated reinforcement learning: Techniques, applications, and challenges," ACM Computing Surveys, vol. 56, no. 3, pp. 1–36, Mar. 2024.
[9] A. Singh and R. Dahiya, "Deep learning-based clustering for IoT sensor networks," ACM Transactions on Sensor Networks, vol. 21, no. 4, pp. 1–25, Nov. 2025.
[10] H. Gupta, R. Sharma, and M. Tripathi, "Sustainable IoT with energy-aware routing protocols using ensemble learning," Ad Hoc Networks, vol. 142, p. 103110, Jan. 2024.
[11] T. Luo and J. Liu, "LSTM-based traffic prediction for IoT wireless networks," IEEE Sensors Journal, vol. 23, no. 2, pp. 1158–1168, Jan. 2023.
[12] K. Deb, A. Saxena, and S. Tamboli, "Scalability evaluation of routing protocols in large-scale wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 19, no. 4, pp. 1–15, Apr. 2023.
[13] R. Xu, J. Huang, and P. Chen, "ONNX-runtime lightweight inference for embedded devices," IEEE Embedded Systems Letters, vol. 16, no. 1, pp. 22–26, Mar. 2024.
[14] A. K. Yadav and N. Jain, "Comparative analysis of LEACH, PEGASIS, and HEED under scalability stress," Procedia Computer Science, vol. 218, pp. 540–547, 2023.
[15] Intel Research Berkeley Sensor Lab, "Berkeley Sensor Network Testbed Dataset," [Online]. Available: https://db.cs.berkeley.edu/labdata. [Accessed: Jun. 2024].
[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proceedings of the International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, May 2019.
[17] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[18] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," IEEE Wireless Communications, vol. 11, no. 6, pp. 6–28, Dec. 2004.
[19] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 2, pp. 551–591, Second Quarter 2013.
[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proceedings of the International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, May 2015.

Authors

Yogesh Juneja is a Ph.D. Research Scholar in the Department of Electronics & Communication Engineering at NIILM University, Kaithal, India. His research focuses on AI-driven optimization in wireless sensor networks, encompassing multi-agent reinforcement learning, deep learning-based routing protocols, and energy management in large-scale IoT deployments. He received his MTech. degree in Electronics & Communication Engineering and has professional experience in QA automation engineering.

Dr. Rajiv Dahiya is a Professor in the Department of Electronics & Communication Engineering at NIILM University, Kaithal, India. He has over 20 years of experience in teaching and research in wireless communications, sensor network protocols, and embedded systems. He has published extensively in international journals and conferences, and serves as a reviewer for several IEEE and Elsevier publications.