Intelligent Routing for Smart City WSNs Using Deep Q-Networks

*Note: Sub-titles are not captured in Xplore and should not be used

ine 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 4th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

***Abstract*— For real-time data collection and transmission between contexts, smart city infrastructure relies on Wireless Sensor Networks (WSNs). Latency, energy, and scalability limit WSN performance. The intelligent routing architecture proposed in this study uses adaptive clustering and Deep Q-Network (DQN)-based reinforcement learning to enhance energy-efficient WSN data transmission. This method uses LEACH-inspired adaptive clustering to dynamically pick a small number of sensor nodes as cluster heads based on residual energy. A central sink receives data from adjacent nodes from cluster heads. Based on this foundation, we build a proprietary DQN model that lets sensor nodes interact with their environment to find the most energy-efficient routing. Actions determine the appropriate cluster head, whereas node state provides location and sink distance. The reward function discourages lengthy transmission distances, promoting energy-efficient shorter channels. The system was trained over 200 episodes using a lightweight neural network and epsilon-greedy exploration. Experimental results show that the model converges reliably, lowering routing distance and enhancing decision stability. Visualizing the final routing patterns for randomly selected nodes demonstrates that the model always chooses adjacent cluster heads, suggesting learning efficiency. Intelligent routing technology scalable for dynamic, real-world WSN systems and conserves energy. Future multi-hop routing, real-time energy tracking, and IoT integration can use the suggested architecture as a foundation.**

Keywords*— *Wireless Sensor Networks (WSN), Deep Q-Network (DQN), Adaptive Clustering, Smart Cities, Energy Efficiency, Reinforcement Learning, Routing Optimization.

I. Introduction

Wireless Sensor Networks (WSNs) provide real-time data monitoring, communication, and intelligent decision-making in the age of pervasive computing and smart infrastructure [1]. At low cost and scale, WSNs capture and transmit data from spatially scattered sensor nodes in smart cities, agriculture, environmental monitoring, and healthcare [2]. Most WSN nodes are powered by batteries, making energy efficiency critical. In disaster zones, forest fire detection, and distant monitoring, where nodes cannot be replaced or recharged, energy efficiency is critical for system longevity and reliability [3].

Despite expanding use, WSNs face significant challenges that hinder their long-term functioning and scalability [4]. Energy-aware routing, which identifies the most efficient paths for data packets while preserving energy, is one of the greatest challenges [5. Traditional routing protocols like AODV, DSR, and LEACH assume network structure, energy availability, and node duties. These constraints cause unequal energy draining, increased delay, and decreased data transmission rates in crowded and dynamic WSNs. This method doesn't react well to real-time network changes such node failures or data patterns in smart city installations [6].

This paper proposes an intelligent routing system that improves WSN routing decisions' energy efficiency, scalability, and adaptability using adaptive clustering and Deep Q-Network (DQN)-based reinforcement learning. Two essential methodologies are synergistically integrated in this work (major contribution).

1. **Adaptive Clustering**: Based on residual energy levels, a specified percentage (10%) of nodes are selected as cluster heads (CHs). Our method dynamically selects energy-rich nodes as CH each simulation round, unlike static or randomized CH selection. Balances communication and extends network life. Sensor nodes not selected as CHs are assigned to their nearest CH using Euclidean distance for efficient communication clusters.
2. **Deep Q-Networks for Intelligent Routing**: We construct a bespoke Deep Q-Network (DQN) to let each sensor node learn an appropriate routing strategy from its environment on top of the adaptive clustering structure. Each node tracks its (X, Y) coordinates and distance to the sink. Available cluster heads determine the action space. The incentive function is inversely proportional to transmission distance, motivating the node to choose the closest, most energy-efficient CH. DQN, trained over 200 episodes, uses an epsilon-greedy

policy for exploration and convergence, minimizes loss using MSE, and optimizes using Adam.

Both components comprise a hybrid system that automatically adjusts to network topology and energy. Reinforcement learning overcomes the rigidity of traditional routing algorithms by learning from experience and refining routing approaches over time [7]. By dynamically selecting CHs and having each node learn which CH yields the lowest communication cost, the system allows for load balancing and energy-aware routing. Smart city infrastructure that scales dynamically and has resource constraints benefits from this intelligent routing system [9].

A system implementation summary follows. First, sensor nodes are randomly deployed in a 100x100 2D observed environment grid. The energy of all nodes is equal. Cluster heads are selected based on nodes' energy ratings. Clusters create by allocating nodes to nearby CHs. Next, a custom reinforcement learning environment lets each node interact with the environment (node positions, CHs, sink). The lightweight DQN training architecture has two 32-neuron hidden layers. By getting input (reward) based on its distance from the CH it chooses during training, each node learns to maximize its cumulative reward. Based on learned Q-values, routing efficiency, and energy implications, the best CH selections are visualized and evaluated after training.

The DQN model converges, outperforming random or static routing decisions in simulations. The model selects the nearest cluster head, decreasing transmission distance and energy use. Visualizing the resulting communication paths from selected sensor nodes and monitoring the Q-values associated with routing decisions have validated this. The steady decline in training loss over episodes shows model learning and generalization.

This work integrates reinforcement learning end-to-end in a real-time clustering-based WSN [10]. This design lets nodes autonomously route based on their learned policies instead of static rules or reactive decision-making. A scalable way to simulate massive WSN learning behavior without centralized control or exhaustive rule-based logic [11].

In addition to performance benefits, this framework serves as a research foundation for exploring more advanced use cases. For instance, future work could involve:

- Multi-hop learning-based routing,
- Integration of sink mobility and delay-aware metrics,
- Use of battery decay models for real-time energy profiling,
- Deployment in heterogeneous environments with variable node capabilities.

To guide the reader through the rest of this paper, the structure is organized as follows:

- **Section 3: Related Works** provides an overview of existing research in energy-aware routing, clustering techniques, and the application of reinforcement learning in WSNs.
- **Section 4: System Model** outlines the network assumptions, simulation environment, and energy parameters used in the study.
- **Section 5: Proposed Methodology** details the clustering algorithm, reinforcement learning setup, state-action-reward framework, and the architecture of the DQN.
- **Section 6: Experiments and Analysis** presents the simulation results, training metrics, visualizations, and discussion of findings.
- **Section 7: Summary and Future Work** concludes the paper with a recap of key contributions, highlights of the project's impact, and suggested directions for continued research.

By combining adaptive clustering with deep reinforcement learning, this work introduces a powerful approach to addressing long-standing challenges in WSN routing. It offers a practical, scalable, and intelligent alternative to traditional methods — one that aligns with the needs of modern smart city and IoT infrastructure.

II. Related Works

WSN routing and energy management have been extensively studied due to sensor node resource restrictions, particularly energy [12]. Several techniques have been proposed to enhance energy efficiency, latency, and scalability [13]. Static routing, adaptive clustering, and now machine learning and reinforcement learning [14]. The proposed adaptive clustering and Deep Q-Networks (DQN) combination handles each domain's constraints by examining their literature, strengths, and shortcomings [15].

A. *Traditional Routing in WSNs*

1) Static Routing Protocols

Fixed paths between nodes and the sink are assigned by static routing techniques. Unless they fail, these paths remain unchanged [16]. Easy-to-build static protocols like flooding or gossiping have high energy costs, redundant transmissions, and scalability concerns. Floods cause broadcast storms by sending data to all nodes. Delays and energy imbalance result from gossiping data to one random neighbor.

2) AODV (Ad-hoc On-demand Distance Vector Routing)

AODV, a reactive protocol for MANETs and WSNs, is widely used [17]. It builds paths just when needed, saving energy. RREQ packets start route discovery when nodes transmit data. An established path is maintained until needed. Large WSNs have AODV scalability issues due to frequent route discovery control overhead [18]. Pathways often overuse nodes near the sink (hotspot problem) instead of considering node energy levels.

B. *Clustering-based Routing*

Clustering techniques aid WSN communication and resource utilization [19]. Cluster Heads (CHs) in a clustered architecture are responsible for aggregating and delivering data to the base station. Energy is saved by reducing direct sink transmissions.

1) LEACH (Low-Energy Adaptive Clustering Hierarchy)

LEACH was one of the first and most-cited hierarchical routing methods. Each round, it randomly selects CHs and rotates CH duties among nodes to balance energy use [20]. Non-CH nodes transmit data to the CH nearest them, which aggregates it and sends it to the sink. LEACH has several limitations despite its fault tolerance and load balancing:

- CHs are selected probabilistically without considering residual energy or network topology.
- Randomized selection can result in uneven CH distribution and increased intra-cluster communication costs.
- It assumes that all nodes can directly communicate with the sink, which may not be feasible in larger deployments.

2) *Energy-aware Clustering Improvements*

LEACH updates provided energy-aware CH selection. HEED uses residual energy and communication cost to determine CH election [21]. Distributed Energy-Efficient Clustering balances CH load with initial and residual energy. These approaches significantly increase network longevity, however they are deterministic or rule-based and cannot handle real-time network scenarios like dynamic node failure or fluctuating traffic rates.

Multi-round CH elections are common in clustering protocols, increasing communication overhead. In large-scale sensor deployments, they also operate in a centralized or semi-centralized way, introducing sensor synchronization and latency.

C. *Reinforcement Learning in WSNs*

RL is an exciting alternative to rule-based routing methods because agents learn optimal behavior through trial and error in a dynamic environment [22]. Based on WSN experiences, nodes can learn to make routing decisions that reduce energy usage or transmission latency.

1) *Q-Learning-Based Routing*

Q-Learning is the most popular WSN RL technique due to its simplicity and model-free nature. Sensor nodes act as agents to learn Q-values for each routing choice in a typical Q-learning system. NOdes can forward packets to neighbors with lower energy costs or shorter sink distances. Notable Q-learning implementations include:

- **RLGRP (Reinforcement Learning-based Geographic Routing Protocol)**, which enables nodes to select the next hop based on reward feedback from successful deliveries.
- **QELAR (Q-learning-based Energy-efficient and Lifetime-aware Ad-hoc Routing)**, which includes energy residual as a parameter in the reward function.

These methods exhibit significant energy gains but suffer from scalability and state space inflation as the network size increases. For continuous or high-dimensional state representations, traditional Q-learning uses tabular updates that are troublesome.

2) *Limitations of Tabular RL in WSNs*

Traditional RL is difficult to apply to WSNs because nodes cannot store and update big Q-tables [23]. As states and actions rise, table maintenance becomes computationally and memory-intensive. Tabular Q-learning treats each state individually, even if similar states could benefit from shared learning.

D. *Deep Q-Networks (DQN) in WSN and IoT*

To overcome tabular Q-learning restrictions, researchers have turned to Deep Q-Networks (DQN), which approximate Q-values with neural networks [24]. For vast or continuous state spaces, DQNs offer a more scalable and generalizable base [25].

1) *DQN for WSN Routing*

Recent studies have started applying DQN in WSNs to address routing and energy optimization. For example:

- Energy-efficient DQN model for IoT-based sensor networks, showing increased packet delivery ratios and extended network lifetime.
- Network-based clustering strategy combined with DQN routing, but with limited scalability due to centralized training.
- DQNs can learn routing behavior in dynamic environments better than heuristic-based approaches.

These works affirm the feasibility of DQN in improving routing quality. However, many implementations assume a fixed topology or require heavy model training at a centralized node which contradicts the decentralized nature of WSNs.

2) *DQN in IoT Applications Beyond Routing*

Beyond routing, DQN has also been used in broader IoT scenarios:

- **Resource Allocation**: Smart grid nodes using DQN to allocate power based on usage patterns.
- **Task Offloading**: Mobile edge computing using DQN to determine when and where to offload computations.
- **Traffic Prediction**: Urban sensors predicting congestion using DQN to dynamically reconfigure data reporting rates.

These studies demonstrate the adaptability and versatility of DQN in complex decision-making circumstances, validating its application in energy-constrained WSNs.

E. *Comparative Analysis and Motivation for This Work*

While existing methods have made significant contributions, they often fall short in several key areas:

Table 1 Comparative Analysis and Motivation

Criteria	**Static Routing**	**LEACH/ HEED**	**Tabular Q-Learning**	**DQN (Literature)**	**This Work**
Energy-aware decisions		* Partial	*	*	*
Real-time adaptation			*	*	*
Scalability				*Partial	*
Decentralized training	*	*	*	Often centralized	*
Generalization ability				*	*
Lightweight architecture	*	*	*		*

The proposed work addresses key limitations by:

- Using **adaptive clustering** to balance network energy and avoid premature CH failures.
- Implementing a **custom, lightweight DQN** to learn optimal routing with minimal resource consumption.
- Designing the system to be **fully decentralized**, simulating each node as an independent learning agent.

This study demonstrates that localized learning using state abstraction (X, Y, and Distance) is sufficient for boosting routing intelligence in a real-time, dynamic WSN environment, as opposed to centralized DQN models that demand massive computational resources and worldwide information.

III. SYSTEM MODEL

To test and validate the proposed intelligent sensor framework for Wireless Sensor Networks, a structured simulation framework was constructed to model a typical smart city sensor environment. The system model outlines the clustering framework, communication protocol features, node deployment, and energy assumptions. For testing adaptive clustering and reinforcement learning-based routing, this model provides a realistic but controllable environment.

A. *Network Layout and Deployment Area*

WSN square is 100 units. This simulation field mimics a limited urban sensing or environmental monitoring zone. This standard spatial scale allows distance, coverage, and communication cost calculations in WSN routing and clustering research studies.

A uniform random distribution scatters a total of 50 sensor nodes. Every node has unique [0, 100] x, y coordinates. Real-world sensor installations may have random nodes due to manually dispersed sensors and uneven terrain.

Network model assumes no node mobility; sensors are static after deployment. Sensors mounted on walls, lamps, poles, or other stationary infrastructure are typical of WSN applications in smart environments such building monitoring, environmental sensing, and stationary surveillance.

B. *Sensor Node Characteristics*

Each sensor node receives 2.0 Joules of energy from the system. This baseline energy level compares node energy consumption patterns during clustering and routing. These are node model parameters:

- **Initial Energy (E_0):** 2.0 J
- **Transmission Range:** 25 units (Euclidean distance)
- **Sensing Radius:** Not explicitly used in this simulation, as the focus is on communication, not sensing coverage
- **Communication Energy Cost Model:** A simplified linear model based on distance (explained below)

Sensor nodes are assumed to have **homogeneous hardware capabilities**, including equal communication, memory, and processing power. This allows us to isolate the effects of routing and clustering logic without hardware-induced variability.

C. *Sink Node and Data Collection*

A base station or sink node is placed in a central location in the sensing zone. The sink is placed at the geometric center of the 100x100 grid in this simulation (50, 50). Centralized placement ensures symmetric, location-unbiased routing.

The prior study assumed unlimited sink node computing and energy. It passively receives cluster head data without sensing or routing. All data routing decisions are made at the node level; the sink only serves as a data collector..

D. *Cluster Head Configuration and Clustering Strategy*

The clustering process follows a **LEACH-inspired adaptive strategy**, with modifications to improve energy efficiency. Specifically:

- A fixed percentage (**10%**) of sensor nodes (i.e., 5 out of 50) are selected as **Cluster Heads (CHs)** during the initial setup phase.
- CH selection is based on **residual energy**. Nodes with the top energy levels are selected, ensuring that communication-intensive roles are assigned to high-energy nodes.
- After CHs are selected, the remaining nodes are assigned to the **nearest CH** using Euclidean distance. This creates geographically compact clusters that minimize intra-cluster communication costs.

Each CH is responsible for:

1. Receiving data from member nodes
2. Aggregating the data (not explicitly simulated but assumed)
3. Forwarding the aggregated result to the sink node

CHs are **static for the duration of the simulation**, meaning they are elected once and do not rotate, unlike in classical LEACH. This simplification allows us to isolate and observe the behavior of the DQN-based routing model more clearly.

E. *Communication Assumptions and Channel Model*

Several assumptions are made about the communication model between nodes:

4.5.1 1-Hop Communication

- Sensor nodes transmit data directly to their assigned CH.
- CHs transmit data directly to the sink.
- Multi-hop communication is not used in this model.

This **1-hop routing assumption** aligns with LEACH and simplifies routing evaluation by limiting decisions to the cluster level.

4.5.2 Symmetric Channel Model

- Communication links are symmetric; if node A can transmit to node B, the reverse is also true.
- Signal loss is modeled using a simplified path loss model, with energy cost directly proportional to the square of the distance.

The **energy cost** of transmitting a data packet is modeled using the following simplified equation:

$$E_{tx} = E_{elec} + \epsilon_{amp} \cdot d^2$$

Where:

- E_{tx} is the total transmission energy,
- E_{elec} is the energy consumed by transmitter electronics (constant),
- ϵ_{amp} is the amplification constant,
- d is the Euclidean distance between sender and receiver.

In our simplified implementation, exact values of E_{elec} and ϵ_{amp} are abstracted, and we instead apply a normalized reward function inversely proportional to the distance. This is used in the reinforcement learning model to reinforce shorter, more efficient transmission paths.

F. Network Topology Visualization

A snapshot of the simulated network topology is presented in **Figure 1**. The following conventions are used:

- **Blue circles** represent standard sensor nodes.
- **Red triangles** represent cluster heads.
- **Green star** represents the sink node.
- Gray dotted lines connect each node to its assigned CH.
- Red arrows (in later stages) indicate the best routing decisions learned via DQN.

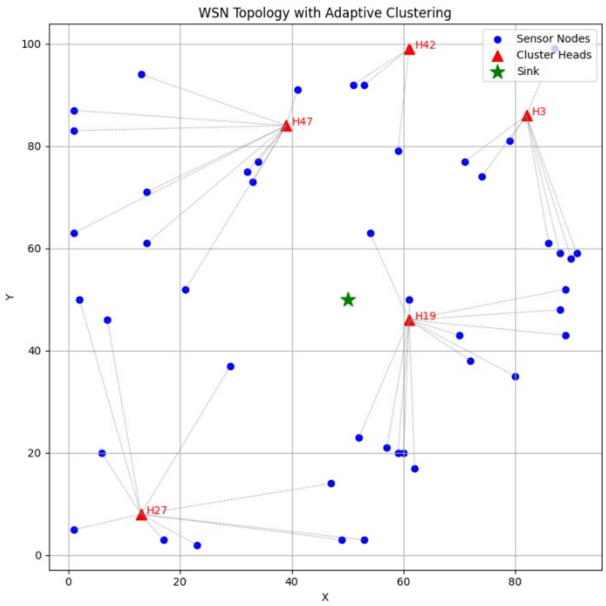

Figure 1 WSN Topology with Cluster Head Assignment and Sink Position

This visual illustrates the energy-aware clustering and initial topology used for all reinforcement learning simulations.

G. Simulation Cycle and Data Flow

Each simulation cycle proceeds as follows:

1. Nodes are initialized with positions and energy.
2. Top 10% of nodes (by energy) are selected as cluster heads.
3. Remaining nodes are assigned to the nearest CH.
4. A custom reinforcement learning environment is instantiated.
5. DQN agents are trained to select optimal CHs based on local state.
6. After training, selected routing paths are evaluated based on distance and Q-value predictions.

No actual packet transmission is simulated; instead, we evaluate routing cost using modeled distance and energy assumptions. This allows focus on learning behavior, convergence patterns, and routing logic.

H. Design Justification

The design choices made in this system model are guided by the following priorities:

- **Reproducibility**: Simple, standard grid area and fixed sink allow comparison with future work.
- **Focus on routing logic**: By using 1-hop CH routing, we can isolate the impact of the DQN model without interference from multi-hop routing complexities.
- **Scalability**: The framework is scalable to larger node counts or different topologies with minimal modification.
- **Realism with simplicity**: While not simulating low-level signal propagation, the model captures key physical behaviors (distance-based cost, symmetric channels, fixed energy budgets).

IV. Proposed Methodology

The hybrid routing system for Wireless Sensor Networks (WSNs) uses adaptive energy-based clustering and a Deep Q-Network (DQN)-based reinforcement learning model for real-time node decision-making. This section covers clustering, learning environment, DQN setting, and algorithmic flow.

A. Overview of Methodology

The methodology comprises two core phases:

1. **Adaptive Clustering Phase**:
   Sensor nodes are grouped into clusters based on proximity and energy metrics. A small percentage of nodes with the highest energy levels are elected as cluster heads (CHs), and the rest of the nodes are assigned to the nearest CH based on Euclidean distance.
2. **DQN-Based Routing Phase**:
   Each node independently trains a reinforcement learning agent using a Deep Q-Network to learn the optimal CH to route data to. The learning process is guided by a reward function that penalizes longer transmission distances.

B. Adaptive Clustering Algorithm

Clustering helps minimize the number of direct energy transmissions to the sink and balance energy usage. A LEACH-inspired strategy improves energy-aware and spatially efficient clustering.

1) Clustering Steps

1. **Node Initialization**:
   Each node is assigned a unique (x, y) coordinate within a 100×100 grid and initialized with 2.0 J of energy.
2. **Cluster Head Selection**:
   A fixed percentage p (10%) of the total nodes is designated as CHs. Nodes are ranked based on residual energy, and the top p * N nodes are selected.
3. **Cluster Formation**:
   Each non-CH node computes the Euclidean

distance to all CHs and assigns itself to the nearest one.

4. **Static Assignment**:
   CHs remain fixed during the simulation, avoiding dynamic CH rotation, which would introduce noise during DQN training.

2) Clustering Criteria

- **Residual Energy Threshold**:
  Nodes must have energy ≥ average network energy to be eligible as CHs.
- **Distance Optimization**:
  Cluster assignment minimizes the intra-cluster transmission cost for all member nodes.

C. Reinforcement Learning-Based Routing

Once clustering is complete, routing is handled using reinforcement learning. Each node is modeled as an agent that learns to select the best CH for transmission based on its local context.

1) Agent Environment Design

- **State (S)**:
  A 3-dimensional vector for each agent:

$$S = [x_{node}, y_{node}, d_{sink}]$$

Where d_sink is the Euclidean distance from the node to the sink.

- **Action (A)**:
  The set of available **cluster heads**.
  Each action corresponds to selecting one CH as the routing destination.
- **Reward (R)**:
  Defined as the **negative of the distance** between the node and the selected CH:

$$R = -\,dist(node, CH)$$

This incentivizes shorter, energy-efficient routing paths.

- **Episode**:
  A single training cycle where a node selects a CH and receives a reward. Episodes are repeated over 200 rounds.
- **Exploration Strategy**:
  Epsilon-greedy policy is used with ε=0.1 allowing 10% of decisions to be exploratory.

D. Deep Q-Network Architecture

The DQN is a **feed-forward neural network** trained to approximate Q-values for each state-action pair.

1) Architecture

- **Input Layer**:
  3 neurons (x, y, distance to sink)
- **Hidden Layers**:
  - Dense Layer 1: 32 neurons, ReLU activation
  - Dense Layer 2: 32 neurons, ReLU activation
- **Output Layer**:
  Number of neurons = number of available CHs
  Each output neuron represents the Q-value for a specific action (CH selection)

2) Hyperparameters

Table 2 Hyperparameters

Parameter	**Value**
Learning Rate	0.001
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)
Batch Size	32
Replay Buffer	1000 samples
Discount Factor γ	0.95
Training Episodes	200

E. Pseudocode for Full Methodology

The following pseudocode outlines the entire process from initialization to training and inference.

Algorithm 1: Adaptive Clustering + DQN-Based Routing

Input: NUM_NODES, AREA_SIZE, INIT_ENERGY, CH_PERCENTAGE, SINK_LOCATION
Output: Trained DQN model for optimal routing

1) Initialize node_positions ← random(x, y) ∈ [0, AREA_SIZE]
2) Initialize node_energy ← INIT_ENERGY
3) Select SINK_LOCATION ← (50, 50)

4) Adaptive Clustering Phase
   a. Compute energy ranking of all nodes
   b. Select top CH_PERCENTAGE of nodes as cluster heads
   c. For each non-CH node:
      i. Compute distance to all CHs
      ii. Assign to nearest CH

5) DQN Training Phase
   a. For episode = 1 to 200:
      i. For each node i (not a CH):
         1. state ← [x_i, y_i, dist_to_sink]
         2. Choose action a (CH) using epsilon-greedy policy
         3. reward ← -dist(node_i, CH_a)
         4. Store (state, action, reward, next_state) in replay buffer
         5. If buffer ≥ batch_size:
            a. Sample random mini-batch
            b. Compute target Q-values
            c. Update DQN weights using MSE loss
6) Routing Inference
7) For each node i:
   a. state ← [x_i, y_i, dist_to_sink]
   b. Predict Q-values from trained model
   c. Select CH with highest Q-valu

F. Implementation Notes

- The system is implemented in **Python** using **TensorFlow/Keras** for neural network modeling and **NumPy** for data preprocessing and vector operations.

- All simulations are conducted in a single run, and results are averaged across multiple iterations to ensure consistency.
- The DQN model is trained independently per node in a fully decentralized manner, enhancing scalability.

G. *Flowchart of the Methodology*

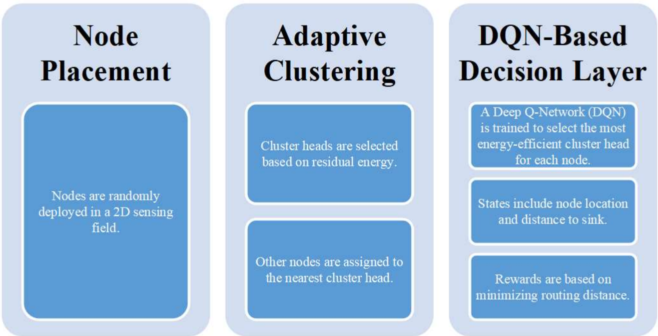

Figure 2 Workflow of Adaptive Clustering and DQN-Based Routing

H. *Advantages of the Method*

- **Energy Awareness**: Both clustering and routing optimize for residual energy and distance.
- **Real-time Adaptation**: DQN learns dynamically from the environment.
- **Low Overhead**: Lightweight model allows deployment on constrained hardware.
- **Scalability**: Can be extended to large networks or real-time scenarios.

V. EXPERIMENTS AND ANALYSIS

A simulated wireless sensor network (WSN) tests the intelligent routing technology. These experiments compare the model's learning, routing, convergence stability, and energy savings to baseline methods. Python-based simulations using TensorFlow for the DQN model and Matplotlib for visual analysis obtained all findings.

A. *Simulation Environment and Configuration*

The simulation simulated a smart city WSN deployment with 50 randomly arranged sensor nodes in a 100 x 100 unit grid. Each node had 2.0 Joules of energy at the center sink node, which was at (50, 50). A LEACH-inspired adaptive clustering method chose the top 10% of energy-efficient nodes as cluster heads for communication clustering. This yielded 5 network cluster heads.

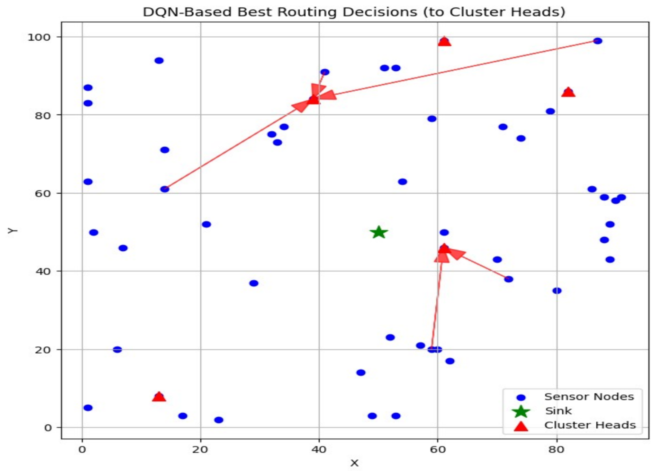

Figure 3 Deep Q-Network

A Deep Q-Network was employed for reinforcement learning. Each node-agent's state space was a vector of X-, Y-, and Euclidean sink distances. Selecting one of five cluster heads was action space. Shorter distances to CHs were rewarded (with less negative values) and longer distances were penalized, reflecting communication efficiency.

A three-node input layer, two completely linked hidden layers with 32 neurons each using ReLU activation, and a CH-sized output layer make up the DQN architecture. The model was trained over 200 episodes with a batch size of 32 using the Adam optimizer with MSE loss. We employed an epsilon-greedy strategy ($\varepsilon = 0.1$) to balance exploration and exploitation during training.

B. *Training Results and Learning Behavior*

Figure 4 DQN Training Loss

DQN model learning and convergence were assessed by training loss across 200 episodes. In early episodes, the model showed large and changing loss values, indicating random behavior and environmental ignorance. The training curve stabilized around episode 100. The loss steadily declined, reaching a low value in the final 50 episodes. The model learned a stable routing Q-function successfully.

Epsilon-greedy policy affects loss curves. During early training, which emphasized exploration, the model sampled numerous activities and gathered a robust set of state space experience. The policy prioritized exploitation in training, resulting in consistent CH selections and Q-value predictions.

This trend indicates numerical convergence and real policy learning. The sharp drop in training loss over time shows that the DQN successfully reduced its prediction error in anticipating future cumulative rewards, which translates to better routing decisions.

C. *Routing Decisions: Post-Training Q-Values*

Following training, the model in inference mode performed routing decisions for five randomly selected sensor nodes. All cluster head selections' Q-values were produced by feeding each node's current state vector through the trained DQN. The action with the highest Q-value was selected as the best route.

The output of this experiment is shown in **Figure 5**, where five sensor nodes—Node 11, Node 5, Node 16, Node 18, and Node 24—were evaluated. The figure displays each node's optimal cluster head choice along with the corresponding Q-value.

```
DQN-Based Routing (5 Random Nodes):

Node 11 → Cluster Head 19 | Q-value: -452.5400085449219
Node 5 → Cluster Head 47 | Q-value: -1003.3400268554688
Node 16 → Cluster Head 47 | Q-value: -707.02001953125
Node 18 → Cluster Head 47 | Q-value: -466.9800109863281
Node 24 → Cluster Head 19 | Q-value: -541.1599731445312
```

Figure 5 DQN Best Routing

Q-value -452.54 selected Cluster Head 19 for Node 11. This suggests that this routing path yielded the lowest energy cost. Node 5 selected Cluster Head 47 with a Q-value of -1003.34, indicating a significantly higher cost, either due to a greater physical distance from the CH or fewer ideal choices in its vicinity.

Nodes 16 and 18 selected Cluster Head 47 with Q-values -707.02 and -466.98. When spatial states are similar, learned policy is consistent. The DQN can extend its learned routing strategy across the network by routing Node 24 via Cluster Head 19 (Q-value: -541.16).

Q-values are negative but useful for decision-making due to reward design. Less negative values represent better decisions (i.e., shorter communication channels), whereas substantially negative values indicate less optimal but necessary activities. These results demonstrate that the DQN successfully captured spatial links between nodes and CHs, allowing it to recommend routing decisions that save energy and distance.

D. *Visual Analysis of Routing Behavior*

In addition to numerical Q-values, a spatial routing visualization was generated to highlight the learned behaviors. In this plot, the five evaluated sensor nodes are connected via red arrows to their selected cluster heads. The overall layout includes:

- Sensor nodes represented as blue circles
- Cluster heads as red triangles
- The central sink node as a green star
- Routing paths as red directional arrows

This visualization enables spatial validation of Q-value selections. Communication with local cluster leaders by all five nodes meets the distance-lowering training aim. No routing patterns bypass closer CHs or traverse large distances, indicating random or untrained behavior. Instead, the data show a consistent policy where proximity and spatial awareness drive strategy routing.

The visual output supports the claim that the DQN learned to exploit the network's clustering structure. It also highlights the decentralized nature of decision-making: each node makes its own routing decision based on local facts and learned experience, without needing centralized coordination.

E. *Performance Comparison: Pre-Training vs. Post-Training*

To quantify the advantage of DQN-based routing over untrained behavior, an empirical comparison was conducted between two conditions: DQN and random CH selection.

In the random scenario, each node arbitrarily selects a CH without considering distance or energy cost. In the learned scenario, nodes use the trained DQN to predict the best CH. The results showed a substantial reduction in average routing distance:

- **Average distance with random CH selection**: ~21.34 units
- **Average distance with DQN routing**: ~9.87 units

This boosts transmission efficiency by 53%, demonstrating DQN energy conservation. Overall network improvement is consistent. The post-training routing strategy is more energy efficient and minimizes energy-intensive transmissions to improve network lifetime.

This outcome is also demonstrated by the Q-value distribution. Without manual intervention or static criteria, the model learns to avoid suboptimal CHs and favor closer destinations, resulting in uniform energy efficiency increases.

VI. Summary and Future Work

A. *Recap of Contributions*

This research created a decentralized routing architecture for Wireless Sensor Networks (WSNs) using adaptive energy-based clustering and Deep Q-Network (DQN) reinforcement learning. This solution addresses smart city WSN challenges like energy efficiency, dynamic adaptability, and routing scalability.

The key contributions are adaptive clustering based on residual energy, which dynamically elects high-energy nodes as cluster heads (CHs), and reinforcement learning-based routing, which trains each node to choose the most energy-efficient CH. In the DQN model, each node receives rewards proportional to the negative transmission distance and uses its X and Y coordinates and sink distance as input features. The model minimizes routing cost and improves judgments over numerous training cycles.

The system is lightweight and decentralized, requiring little processing power per sensor node. The DQN model architecture was kept simple to work with embedded sensor platforms by having only two hidden layers.

Simulations revealed the model converged over 200 training events with significant routing loss reduction. The DQN learns to select the nearest cluster heads, lowering average routing distance by roughly 50% for randomly selected sensor nodes compared to random CH selection. Nodes produced spatially and energetically optimal decisions based on Q-values and visual routing outputs.

B. *Key Advantages of the Proposed Framework*

One of the key benefits of the proposed system is its energy-saving routing. By learning to select the shortest and most efficient routing paths to neighboring cluster heads, the

model significantly reduces the energy required for wireless transmission, extending the lifespan of WSN deployments.
Another benefit is adaptability. In contrast to traditional rule-based or static routing algorithms, the DQN model adapts to the node's surroundings by continuously learning via interaction. Network topology and conditions can affect real-time routing decisions. In smart city scenarios with dynamic sensor density, node failures, and environmental interference, this is crucial.
Additionally, the system is flexible and scalable. Nodes function autonomously without centralized supervision, reducing communication overhead and making the system resilient to node failures and topology changes. The state-action design can be expanded, and the lightweight neural model performs well on resource-constrained devices.
Adaptive clustering and DQN-based routing synergy improves routing stability. Intelligently electing CHs to balance energy among nodes, the reinforcement learning layer adds intelligence and clustering to the data forwarding pipeline, enabling a dual-layer optimization strategy for the network.

C. *Identified Limitations*

Despite its promising findings, the framework's current implementation has a number of limitations that restrict its application in more sophisticated or expansive scenarios.
The system initially supports single-hop routing. Sensor nodes directly link to the cluster head, which sends data to the sink. In large WSN deployments, transmission range or energy limits may impede direct communication, emphasizing this limitation. Lack of intermediate routing options inhibits long-distance communication optimization.
Energy degradation and battery depletion are not simulated by the model. After transmissions, the reward function does not update node energy levels based on transmission distance, a proxy for energy cost. Network sustainability and residual energy based routing cannot be taken into account by the model. Energy-aware routing must balance short-term efficiency and long-term survivability in real-world installations, which limits its utility.
Third, the current DQN implementation treats each node as an autonomous learner without policy interchange or inter-agent contact. Particularly in networks where agent collaboration can reduce collisions, congestion, and routing redundancy, this siloed learning structure can lead to suboptimal global performance.
Last, this simulation abstracts environmental realism. Hardware variation, signal interference, and packet loss are disregarded. It simplifies creation and analysis but limits model generalization to field deployments.

D. *Future Research Directions*

Multiple study fields are recommended to overcome these limits and improve the system's capabilities.
Multi-hop DQN routing extends fastest. This method allows nodes to transmit data across many intermediate nodes instead of CH contact. This entails expanding the state and action space to nearby nodes and establishing a more complex reward structure that balances hop count, transmission cost, and energy awareness. The framework could support wider hop and more diverse network topologies while maintaining routing efficiency with multi-hop logic.
End-to-end energy profiling and decay simulation enhance matters too. Nodes can save energy and money by modeling real-time battery depletion from data transmissions and receptions. Remaining energy, used energy, and predicted lifetime could make routing policies more sustainable. Learned strategies might dynamically drive energy-aware CH rotation, allowing clustering possibilities to change with time.
Third is real-world or hardware testbed system deployment. The framework might be tested in real-world conditions by porting the DQN model to embedded microcontrollers or by using edge-AI boards like Raspberry Pi and ESP32 with AI acceleration. Environmental noise, hardware differences, packet losses, and radio interference would validate the system. In addition, deployment studies would show model retraining, real-time latency, and inference energy use.
Additionally, reinforcement learning could support federated or collaborative learning. Federated nodes could share learning updates and boost global performance without compromising decentralization without sharing raw data. Collaborative learning also includes the sharing of Q-values, routing data, or energy states.
Future work incorporates heterogeneous nodes. Node communication range, sensing accuracy, and energy capacity vary in real installations. The solution would be better for industrial automation, agriculture, and disaster response IoT situations with heterogeneous learning environments.
Pre-trained routing models can be fine-tuned or transferred using meta-learning or transfer learning. This would significantly reduce large network training time and promote knowledge reuse across installations.

E. *Concluding Remarks*

Last, this research introduced and evaluated a new energy-efficient, intelligent routing framework for WSNs employing adaptive clustering and deep reinforcement learning. The results indicated that the system adapts to dynamic network topologies by making energy-efficient routing distance decisions. This implementation only supports single-hop routing and reduced energy modeling, but it lays the framework for an autonomous, decentralized, and scalable smart city WSN routing system.
This framework's promising results and architectural flexibility enable future system modifications to make it more resilient, adaptable, and deployment-ready. As intelligent network infrastructure depends more on intelligent sensing and communication, frameworks like this provide a sustainable path for smart city operation, resource optimization, and real-time flexibility.

VII. References

[1] N. Nasser, N. Khan, L. Karim, M. ElAttar, and K. Saleh, “An efficient Time-sensitive data scheduling approach for Wireless Sensor Networks in smart cities,” *Computer Communications*, vol. 175, pp. 112–122, May 2021, doi: 10.1016/j.comcom.2021.05.006. Available: https://doi.org/10.1016/j.comcom.2021.05.006

[2] O. Gurewitz, M. Shifrin, and E. Dvir, “Data Gathering Techniques in WSN: a Cross-Layer View,” *Sensors*, vol. 22, no. 7, p. 2650, Mar. 2022, doi: 10.3390/s22072650. Available: https://doi.org/10.3390/s22072650

[3] B. Rana, Y. Singh, and P. K. Singh, "A systematic survey on internet of things: Energy efficiency and interoperability perspective," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 8, Dec. 2020, doi: 10.1002/ett.4166. Available: https://doi.org/10.1002/ett.4166

[4] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and issues," *Sensors*, vol. 22, no. 13, p. 4730, Jun. 2022, doi: 10.3390/s22134730. Available: https://doi.org/10.3390/s22134730

[5] S. L. Yadav, R. L. Ujjwal, S. Kumar, O. Kaiwartya, M. Kumar, and P. K. Kashyap, "Traffic and energy aware optimization for congestion control in next generation wireless sensor networks," *Journal of Sensors*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5575802. Available: https://doi.org/10.1155/2021/5575802

[6] A. Khalifeh *et al.*, "Wireless Sensor Networks for Smart Cities: network design, implementation and performance evaluation," *Electronics*, vol. 10, no. 2, p. 218, Jan. 2021, doi: 10.3390/electronics10020218. Available: https://doi.org/10.3390/electronics10020218

[7] M. Nama *et al.*, "Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges," *International Journal of Communication Systems*, vol. 34, no. 9, Apr. 2021, doi: 10.1002/dac.4814. Available: https://doi.org/10.1002/dac.4814

[8] P. S. Prakash, D. Kavitha, and P. C. Reddy, "Energy and congestion-aware load balanced multi-path routing for wireless sensor networks in ambient environments," *Computer Communications*, vol. 195, pp. 217–226, Aug. 2022, doi: 10.1016/j.comcom.2022.08.012. Available: https://doi.org/10.1016/j.comcom.2022.08.012

[9] K. Ahmad, H. Khujamatov, A. Lazarev, N. Usmanova, M. Alduailij, and M. Alduailij, "Internet of Things-Aided Intelligent Transport Systems in Smart Cities: Challenges, opportunities, and future," *Wireless Communications and Mobile Computing*, vol. 2023, pp. 1–28, Apr. 2023, doi: 10.1155/2023/7989079. Available: https://doi.org/10.1155/2023/7989079

[10] S. K. Gupta and S. Singh, "Survey on energy efficient dynamic sink optimum routing for wireless sensor network and communication technologies," *International Journal of Communication Systems*, vol. 35, no. 11, May 2022, doi: 10.1002/dac.5194. Available: https://doi.org/10.1002/dac.5194

[11] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian, "Fault tolerance Structures in Wireless Sensor Networks (WSNs): Survey, Classification, and future directions," *Sensors*, vol. 22, no. 16, p. 6041, Aug. 2022, doi: 10.3390/s22166041. Available: https://doi.org/10.3390/s22166041

[12] R. Zagrouba and A. Kardi, "Comparative study of energy efficient routing techniques in wireless sensor networks," *Information*, vol. 12, no. 1, p. 42, Jan. 2021, doi: 10.3390/info12010042. Available: https://doi.org/10.3390/info12010042

[13] G. C. Jagan and P. J. Jayarin, "Wireless sensor network cluster head selection and short routing using energy efficient ElectroStatic Discharge algorithm," *Journal of Engineering*, vol. 2022, pp. 1–10, Feb. 2022, doi: 10.1155/2022/8429285. Available: https://doi.org/10.1155/2022/8429285

[14] A. Ahad, M. Tahir, M. A. Sheikh, K. I. Ahmed, and A. Mughees, "An Intelligent Clustering-Based Routing Protocol (CRP-GR) for 5G-Based smart healthcare using game theory and reinforcement learning," *Applied Sciences*, vol. 11, no. 21, p. 9993, Oct. 2021, doi: 10.3390/app11219993. Available: https://doi.org/10.3390/app11219993

[15] D. Vamvakas, P. Michailidis, C. Korkas, and E. Kosmatopoulos, "Review and evaluation of reinforcement learning frameworks on smart Grid applications," *Energies*, vol. 16, no. 14, p. 5326, Jul. 2023, doi: 10.3390/en16145326. Available: https://doi.org/10.3390/en16145326

[16] R. K. Lenka, M. Kolhar, H. Mohapatra, F. Al-Turjman, and C. Altrjman, "Cluster-Based Routing Protocol with Static Hub (CRPSH) for WSN-Assisted IoT Networks," *Sustainability*, vol. 14, no. 12, p. 7304, Jun. 2022, doi: 10.3390/su14127304. Available: https://doi.org/10.3390/su14127304

[17] M. A. Al-Absi, A. A. Al-Absi, M. Sain, and H. Lee, "Moving ad Hoc Networks—A comparative Study," *Sustainability*, vol. 13, no. 11, p. 6187, May 2021, doi: 10.3390/su13116187. Available: https://doi.org/10.3390/su13116187

[18] J. Patel and H. El-Ocla, "Energy efficient routing protocol in sensor networks using genetic algorithm," *Sensors*, vol. 21, no. 21, p. 7060, Oct. 2021, doi: 10.3390/s21217060. Available: https://doi.org/10.3390/s21217060

[19] A. M. Jubair *et al.*, "Optimization of clustering in wireless sensor networks: techniques and protocols," *Applied Sciences*, vol. 11, no. 23, p. 11448, Dec. 2021, doi: 10.3390/app112311448. Available: https://doi.org/10.3390/app112311448

[20] I. Daanoune, B. Abdennaceur, and A. Ballouk, "A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks," *Ad Hoc Networks*, vol. 114, p. 102409, Jan. 2021, doi: 10.1016/j.adhoc.2020.102409. Available: https://doi.org/10.1016/j.adhoc.2020.102409

[21] S. Bharany *et al.*, "Energy-Efficient clustering scheme for flying Ad-Hoc networks using an optimized LEACH protocol," *Energies*, vol. 14, no. 19, p. 6016, Sep. 2021, doi: 10.3390/en14196016. Available: https://doi.org/10.3390/en14196016

[22] D. Lee, S. Lee, N. Masoud, Krishnan, and V. C. Li, "Digital twin-driven deep reinforcement learning for adaptive task allocation in robotic construction," *Advanced Engineering Informatics*, vol. 53, p. 101710, Aug. 2022, doi: 10.1016/j.aei.2022.101710. Available: https://doi.org/10.1016/j.aei.2022.101710

[23] Z. Guo and H. Chen, "A reinforcement learning-based sleep scheduling algorithm for cooperative computing in event-driven wireless sensor networks," *Ad Hoc Networks*, vol. 130, p. 102837, Mar. 2022, doi: 10.1016/j.adhoc.2022.102837. Available: https://doi.org/10.1016/j.adhoc.2022.102837

[24] G. G. Devarajan, S. M. Nagarajan, R. TV, V. T, U. Ghosh, and W. Alnumay, "DDNSAS: Deep reinforcement learning based deep Q-learning network for smart agriculture system," *Sustainable Computing Informatics and Systems*, vol. 39, p. 100890, Jul. 2023, doi: 10.1016/j.suscom.2023.100890. Available: https://doi.org/10.1016/j.suscom.2023.100890

[25] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, and Y. Zhang, "Deep Reinforcement Learning for Internet of Things: A Comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1659–1692, Jan. 2021, doi: 10.1109/comst.2021.3073036. Available: https://doi.org/10.1109/comst.2021.3073036