VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2026, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 14, Number 1, January-March 2026 pp:137-152

A Novel Agentic AI and DQN Approaches for Unmanned Drone Route Combinatorial Optimisation in Distributed Edge Cloud Networks

Akhtar Ali Junejo[1], **Zain Gul**[2], **Qurat-ul-ain Mastoi**[3], **Asif Aziz Memon**[1], **Atif Jamil**[1], **Abdullah Lakhan**[1*]

[1]Department of Cybersecurity, Dawood University of Engineering and Technology, Karachi, Sindh, 74800, Pakistan; [2]Instituto de Ciências da Terra (ICT), Escola de Ciências, Universidade do Minho, Braga, Portugal; [3]School of Computer Science and Creative Technologies, University of the West of England Bristol, BS16 1QY, United Kingdom

Keywords:*Drone, Agentic AI, Latency, Energy, Karachi*

Journal Info:
Submitted:
November 30, 2025
Accepted:
February 28, 2026
Published:
March 12, 2026

Abstract

This study presents a novel agentic artificial intelligence (Agentic-AI) framework, combined with Deep Q-Network (DQN) reinforcement learning, to solve the combinatorial optimization problem of unmanned drone route planning, specifically designed for smart-city surveillance and road–traffic assessment. The system is designed to integrate autonomous decision-making, context-aware sensing, and dynamic route adaptation to efficiently manage large-scale urban monitoring tasks. To evaluate real-world management of large-scale urban monitoring tasks efficiently in the Karachi Smart City environment in various metropolitan zones, including Saddar, Nazimabad, Gulshan-e-Iqbal, North Karachi, Clifton, Korangi and adjoining high-density regions. These areas exhibit computationally and communication-intensive congestion patterns. Communication-intensive congestion patterns and irregular road conditions make them ideal for testing adaptive, multimodal data, such as vehicle density, traffic flow ratio, road surface deterioration, and broken-road occurrences, while interacting with distributed edge nodes for real-time processing and data exchange. The simulation testbeds show that the proposed methods have higher accuracy of data collection by 98%, processing time by 31% and minimize the resource consumption as compared to existing studies.

***Correspondence author email address:** abdullah.lakhan@duet.edu.pk
DOI: 10.21015/vtse.v14i1.2292

Introduction

Unmanned Aerial Vehicles (UAVs), or drones, are key technologies for next-generation intelligent systems and have been widely used for surveillance, environmental monitoring, search and rescue, logistics, and military reconnaissance. UAVs use onboard sensors, such as LiDAR, GPS, cameras, and inertial measurement units (IMUs), to sense the environment, navigate dynamic environments, and execute complex missions. Today, distributed Edge Cloud Networks play a vital role in supporting various operations and providing storage resources for computation [1–3].

The latest drone integration with Distributed Edge Cloud Networks (DECNs) produces an association among hundreds of Unmanned Aerial Vehicles (UAVs) and ground control nodes. With the growing use of drones and the increasing complexity of tasks, the combinatorial processes of path planning, task distribution, and energy management are computationally and communicationally very demanding, and are typically beyond the capabilities of conventional and centralized approaches.

Autonomous drone networks, along with their various challenges, arise from multiple sources [4–6]. Initially, environmental dynamics, such as wind turbulence, signal interference, and obstacle uncertainty, will disrupt the trajectories of pre-programmed flight. Secondly, the diversity of drones and sensor platforms introduces inconsistencies in energy generation, communication range, and computing power, posing challenges for the harmonized execution of tasks. Thirdly, discontinuous connectivity with restricted bandwidth in the environments of edge can hold up the process of synchronization in decision making and the updates in models.

Further, vulnerabilities in security with respect to the route switching and task organization render drones to spoofing of data and denial-of-service to attacks to undermine mission reliability. These tasks demand a smart, distributed, and adaptive framework for optimization. Therefore, it seeks to learn from real-time responses without compromising operational security.

To overcome such restrictions, this paper proposes a Novel Agentic Artificial Intelligence (Agentic AI)- and deep Q-network (DQN)-enabled framework for combinatorial route optimization of autonomous drones in distributed edge cloud networks. The method models each drone as an intelligent agent that perceives independently, reasons in terms of multi-objective goals (e.g., minimising energy and latency, coverage and safety), and cooperatively bargains with other agents via edge-enabled communications. The Agentic AI layer provides drones with goal-oriented reasoning to plan adaptively in dynamic environmental and resource contexts. Meanwhile, the DQN module provides a Q-learning-based optimisation engine that learns an approximately optimal route-choice policy by leveraging Q-value updates from reward feedback on navigation success, energy consumption, and communication delay.

The major contribution of this paper is given as under:

1. **Agentic AI-Driven Distributed Route Planning:** Here, we suggest an innovative agentic architecture, where each individual UAV autonomously yet collaboratively acts upon routing decisions with the exercise of goal-driven, context-aware interpretation assimilated with real-time feedback in environmental sensor.
2. **Deep Reinforcement Learning-Based Combinatorial Optimization:** To frame the route optimization problem as a Markov Decision Process (MDP), a Deep Q-Network (DQN) is deployed with optimization of route scheduling and the process of making strategies of assignment across computing nodes distributed on the edge.
3. **Latency, Energy, and Resilience-Aware Architecture:** Here, we design a hybrid DQN–edge framework for the cloud. It mitigates communication latency and reduces computational workloads. It also extends mission duration through adaptive, energy-aware routing policies.
4. **Improved Security and Reliability Mechanisms:** The suggested framework integrates real-time authentication and secure coordination protocols to reduce adversarial injection into data, route manipulation, and tampering of paths in edge-enabled UAV systems with multi-agents.
5. **Widespread Benchmarking and Mixed-Methods Analysis:** Widespread experiments on the larger scales signify that the suggested Agentic AI-DQN framework does better work than state-of-the-art baseline approaches with respect to accuracy in routing, efficiency in energy, reduction in latency, and toughness against environmental and adversarial perturbations throughout the multifaceted operational situations.

Related Work

Recent advancements in Unmanned Aerial Vehicles (UAVs) and the Internet of Things (IoT) have significantly addressed challenges related to intelligent, adaptable, and resource-efficient surveillance in smart cities. Kim et al. [1] argued that a resting drone-enabled surveillance framework. It enables real-time monitoring of the system and efficient resource utilization through adaptable scheduling strategies. This paper shows that idle-time planning and long-term drone placement significantly contribute to system longevity in terms of efficiency. In the context of privacy-aware surveillance in smart cities, Abu-Khadrah et al. [2] introduced a drone-assisted, adaptable object-detection system using deep reinforcement learning and whale optimization. The model enhances drone detection efficiency while preserving privacy in the context of vibrant smart cities.

Furthermore, Bakirci [3] discussed the integration of IoT-based UAVs for the purpose of analyzing the real-time traffic mobility. This discussion emphasizes the interconnected process of UAV-based sensing, communication, and aerial analytics for the purpose of improving the process of managing the traffic. This discussion demonstrates the potential of UAV-IoT integration to obtain high-resolution data for well-organized mobility in large-scale smart city infrastructures. These studies collectively emphasize the potential of UAV-IoT integration. The intelligent and adaptive monitoring process using the changing smart city applications opens doors for the innovative ideas and innovations in the context of UAV and IoT integration.

In the area of UAV-related DO, Qin and Pournaras [4] presented a hybrid short-term optimization and long-term reinforcement learning technique for the synchronization of drone routes, and Yang et al. [5] presented a cloud-edge co-working model to attain better scalability for the distributed network of drones. Moreover, Awada et al. [6] presented Edge Drones, a co-scheduling technique for multi-location aerial computing objectives, to emphasize the increasing need for distributed edge computing. Simultaneously, the foundational work by Sapkota et al. [7] included a survey of agentic AI for drones, emphasizing the autonomy of reasoning-based systems. However, Spyridis [8] discussed the communication efficiency of drone-based IoT systems.

Moreover, Deep Learning and Multi-Agent Reinforcement Learning (MARL) methods are also used to optimize UAV decision-making. Tang et al. [9] proposed a generative AI-based MARL model for DNN task assignment. However, Ding et al. [10] extensively explored the distributed machine learning for UAV swarms with the weight on sensing and semantic communication. Moreover, Zhao et al. [11] discussed different features on edge general intelligence, which is determined by the world models and agentic AI. This has given a deep insight into the future edge-native autonomous systems.

Furthermore, Chagas et al. [12] surveyed AI-based deconfliction methods for UAV routing and proposed a unified framework for airspace conflict resolution. Moreover, extensive further research has provided deeper insight into the various methods for offloading, routing, and task reliability in heterogeneous environments. Waqar et al.[13] used Reinforcement Learning (RL) to enable optimisation in MEC-enabled vehicular-aerial networks. Moreover, Wang et al. [14] analytically reviewed the collaborative delivery systems of the UAV-ground for response scenarios in the emergency period.

In addition to this, Gao et al. [15] researched satellite-based UAV networks. Furthermore, the applicability of AI-based UAVs extends to environmental and industrial contexts. In this regard, Nahiyoon et al. [17] critically reviewed the security of plants, UAVs, and the transition towards intelligent autonomous operation. In addition, Prakash and Radha [18] provided a comprehensive overview of multi-constraint optimization for AI-based electric vehicle routing. In the same regard, Kishor et al. [19] implemented the application of AI-based autonomous robots and UAVs for solar panel maintenance. Furthermore, Neelesh [20] envisioned the application of reinforcement learning agents to UAVs, supported by theoretical and practical estimates.

Most prevailing RL-based methods rely on fixed reward models and centralized topologies, further limiting scalability in distributed edge–cloud environments. Moreover, few studies consider agent-level reasoning and security-aware collaboration in multi-drone missions. To fill these gaps, this paper proposes a novel agentic AI and DQN-enabled framework that integrates autonomous reasoning, dynamic reward adaptation, and distributed edge learning to support efficient route scheduling, energy management, and secure multi-agent collaboration, as illustrated in Figure 1.

Proposed System

We defined the combinatorial mathematical model as follows.

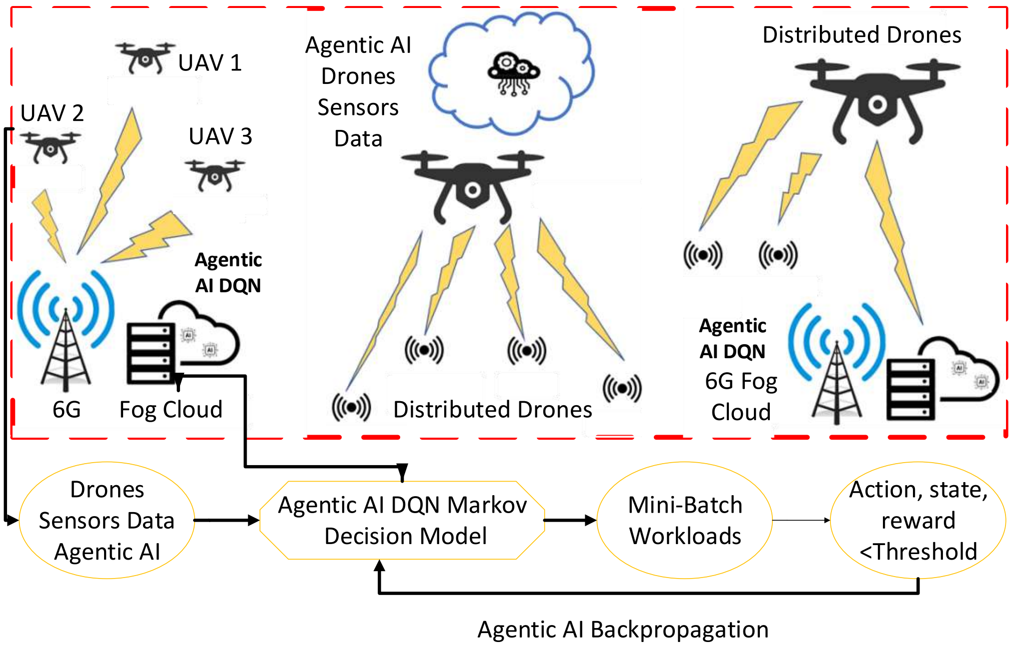

Figure 1. Agentic-AI Unmanned Drone Surveillance in Distributed Edge Cloud Networks

- $\mathcal{D} = \{1, \ldots, D\}$: set of drones.
- For drone $i \in \mathcal{D}$: position at time t is $\mathbf{p}_i(t) \in \mathbb{R}^2$ (or $\mathbb{R}^3$).
- $\mathcal{S}$: set of sensor types (e.g., LiDAR, camera, GPS, IMU). Each drone i has a subset $\mathcal{S}_i \subseteq \mathcal{S}$.
- $s \in \mathcal{S}$: an individual sensor type; sensor quality parameter $q_{i,s} \in [0, 1]$ represents reliability/accuracy of sensor s on drone i.
- $\mathcal{N}$: set of edge/cloud nodes available for offloading tasks.
- T : discrete-time horizon indexed by $t = 0, 1, \ldots, T$.
- $E_i(t)$: remaining energy of drone i at time t. $E_i(0)$ initial energy.
- $v_i(t)$: velocity magnitude for drone i at time t.
- $x_{i,k}(t) \in \{0, 1\}$: scheduling/offload indicator: equals 1 if drone i offloads task k (e.g., a sensor frame, model inference) to node $k \in \mathcal{N}$ at time t.
- $b_i(t)$: available communication bandwidth for drone i at time t.
- $L_{i,k}(t)$: one-way latency between drone i and node k at time t.
- C_k: computational capacity of node $k \in \mathcal{N}$ (e.g., FLOPS or inference slots per second).
- $\mathcal{W}$ = {waypoints}: discretized waypoints from source to destination.
- $d(\mathbf{p}, \mathbf{q})$: Euclidean distance between positions $\mathbf{p}$ and $\mathbf{q}$.

Surveillance Accuracy Model

Sensors are combined by a fusion model that produces surveillance accuracy for an observation collected by drone i at time t. Let $\mathcal{S}_i^{\text{on}}(t) \subseteq \mathcal{S}_i$ be the set of sensors currently active (consuming energy and bandwidth). Define a weighted sensor-quality score:

$$\mathcal{Q}_i(t) = \sum_{s \in \mathcal{S}_i^{\text{on}}(t)} w_s\, q_{i,s}, \qquad \text{with } w_s \geq 0, \; \sum_s w_s = 1. \tag{1}$$

We can map $\mathcal{Q}_i(t)$ into an *accuracy probability* $A_i(t) \in [0, 1]$ using a saturating function, e.g.

$$A_i(t) = 1 - \exp\left(-\kappa\, \mathcal{Q}_i(t)\right), \qquad \kappa > 0. \tag{2}$$

(Other mappings such as logistic functions can be used; this form is analytically convenient.)

Cost / Objective Terms

We consider multiple competing objectives: route efficiency, surveillance accuracy, offloading latency, and energy consumption. For drone i over horizon T:

Travel cost (distance/time).

If the route is represented as a sequence of positions, the total travel distance:

$$D_i^{\text{travel}} = \sum_{t=0}^{T-1} d\big(\mathbf{p}_i(t), \mathbf{p}_i(t+1)\big). \tag{3}$$

Energy cost.

Energy consumption model decomposed into motion and sensing/communication:

$$\mathcal{E}_i = \sum_{t=0}^{T-1} \Big(e_i^{\text{move}}\big(v_i(t)\big)+e_i^{\text{sense}}\big(\mathcal{S}_i^{\text{on}}(t)\big)+e_i^{\text{comm}}\big(x_{i,\cdot}(t), b_i(t)\big)\Big). \tag{4}$$

Latency penalty for offloading.

For tasks offloaded to node k, include latency term:

$$\mathcal{L}_i = \sum_{t=0}^{T-1} \sum_{k\in\mathcal{N}} x_{i,k}(t)\, L_{i,k}(t). \tag{5}$$

Surveillance accuracy reward (to maximize).

We prefer routes and sensor activations that yield high accuracy. Aggregate accuracy:

$$\mathcal{A}_i = \sum_{t=0}^{T} A_i(t). \tag{6}$$

Multi-objective Optimization

Form a scalarized objective (can also be solved as multi-objective Pareto optimisation)

$$\min_{\{\mathbf{p}_i(t),\mathcal{S}_i^{\text{on}}(t),x_{i,k}(t)\}} J = \sum_{i\in\mathcal{D}} \Big(\alpha\, D_i^{\text{travel}} + \beta\, \mathcal{E}_i + \eta\, \mathcal{L}_i - \lambda\, \mathcal{A}_i\Big), \tag{7}$$

with weights $\alpha, \beta, \eta, \lambda \geq 0$ that trade off travel, energy, latency, and accuracy. We show the constraints as const: keywords.

Constraints

$$\text{(Const:)}\quad \mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \Delta t\, \mathbf{u}_i(t), \quad \|\mathbf{u}_i(t)\| \leq V_i^{\max}, \tag{8}$$

$$\text{(Energy)}\quad \mathcal{E}_i \leq E_i(0), \quad \forall i, \tag{9}$$

$$\text{(Collision avoidance)}\quad \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \geq d_{\min}, \quad \forall i \neq j,\ \forall t, \tag{10}$$

$$\text{(Comms capacity)}\quad \sum_{i\in\mathcal{D}} x_{i,k}(t)\, b_{i,k}(t) \leq B_k, \quad \forall k, t, \tag{11}$$

$$\text{(Node compute)}\quad \sum_{i\in\mathcal{D}} x_{i,k}(t)\, c_{i,k}(t) \leq C_k, \quad \forall k, t, \tag{12}$$

$$\text{(Offload binary)}\quad x_{i,k}(t) \in \{0, 1\}. \tag{13}$$

This is a mixed-integer, non-convex optimization problem (routing, scheduling, and non-linear accuracy). Solving it exactly is intractable for large D and T; we therefore propose an Agentic-AI (RL) approach with DQN-based decision-making and an outer optimizer or heuristic for initial routing.

MDP Formulation for Agentic-AI (per drone or centralized multi-agent)

Define an MDP (or Dec-POMDP for multi-agent):

State s_t.

The full system state at time t (for a centralized agent):

$$s_t = \Big\{\{\mathbf{p}_i(t), E_i(t), \mathcal{S}_i^{\text{on}}(t), b_i(t)\}_{i\in\mathcal{D}}, \{L_{i,k}(t)\}_{i,k}, \{\text{task queues}\}\Big\}. \tag{14}$$

For decentralized agents, each drone i observes a local observation o_t^i (its own position, energy, sensor states, and local channel estimates), as well as possibly neighbor messages.

Action a_t.

Actions include:

$$a_t = \big\{\mathbf{u}_i(t)\,\text{(next heading/waypoint)},\ \mathcal{S}_i^{\text{on}}(t+1),\ x_{i,k}(t)\,\big\}_{i\in\mathcal{D}}.$$

In discrete form, action could be choosing next waypoint from a finite set, selecting sensors on/off, and choosing offload target.

Transition.

Standard physics + communication + queue updates defines $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$.

Reward r_t.

Reward should reflect the multi-objective cost: encourage accurate surveillance, penalize travel/time, penalize energy and latency. A sample per-step reward:

$$r_t = \sum_{i\in\mathcal{D}} \Big(\underbrace{\omega_A A_i(t)}_{\text{accuracy reward}} - \underbrace{\omega_D d(\mathbf{p}_i(t), \mathbf{p}_i(t+1))}_{\text{distance penalty}} - \underbrace{\omega_E e_i(t)}_{\text{energy penalty}} - \underbrace{\omega_L \sum_k x_{i,k}(t) L_{i,k}(t)}_{\text{latency penalty}} \Big), \tag{15}$$

with weights $\omega_{\cdot} \geq 0$. Optionally, include a large terminal reward for successful delivery to destination.

DQN Formulation and Training

We apply Deep Q-learning: approximate action-value $Q(s, a; \theta) \approx Q^\star(s, a)$ using neural network with parameters θ.

Bellman target.

For a transition (s, a, r, s') sampled from replay:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-), \tag{16}$$

where θ^- are parameters of a delayed target network.

Loss.

Minimize mean-squared TD error:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\text{replay}}}\Big[\big(y - Q(s, a; \theta)\big)^2\Big]. \tag{17}$$

Practical improvements (Agentic-AI / high performance DQN):

- **Double DQN:** use action selection from online network and evaluation by target network to reduce overestimation:
$$y = r + \gamma Q\big(s', \arg\max_{a'} Q(s', a'; \theta), \theta^-\big).$$
- **Dueling architecture:** represent $Q(s, a)$ as $V(s) + A(s, a)$ to better learn state-value and advantage separately.
- **Prioritized Experience Replay:** sample transitions with probability proportional to TD error magnitude.
- **Multi-agent CTDE (Centralized Training, Decentralized Execution):** During training, central critic sees global s; actors (per-drone policies) operate on local o^i at execution time.
- **Agentic-AI enhancements:** meta-controller for high-level route planning (e.g., waypoint planning) and local DQNs for fine-grain decisions (sensor on/off, offload).

Reward Shaping to Improve Accuracy

To specifically improve surveillance accuracy from source to destination:

- add per-step positive reward proportional to $A_i(t)$ as in (15);
- add sparse terminal bonus R_{terminal} if cumulative accuracy along route $\sum_t A_i(t)$ exceeds threshold;
- shape reward to penalize missed critical events (e.g., false negatives) more heavily.

Scheduling Tasks to Edge-Cloud (Optimization via RL or MILP)

Offloading scheduling can be handled in two ways:

1. Mixed-integer optimization (batch).

For given routes, solve at each time slot t:

$$\min_{x_{i,k}(t)\in\{0,1\}} \sum_{i,k} \left(\eta_1 L_{i,k}(t) x_{i,k}(t) + \eta_2 \frac{\text{data\_size}_i(t)}{b_{i,k}(t)} x_{i,k}(t)\right) \tag{18}$$

$$\text{s.t.} \sum_i x_{i,k}(t)\, c_{i,k}(t) \le C_k, \quad \sum_k x_{i,k}(t) \le 1, \text{ etc.} \tag{19}$$

This is a binary integer program solved by MILP solvers for small scale.

2. RL-based scheduling (online):

incorporate $x_{i,k}(t)$ as part of the action in the MDP and let DQN learn offloading policies that trades accuracy vs latency vs energy.

Combined Algorithmic Architecture (Agentic-AI + DQN)

1. **Meta-controller (High-level):** Plans coarse route / waypoints for each drone to cover required surveillance corridor minimizing travel/time (e.g., via A*/TSP heuristic or model predictive control). This reduces action branching in DQN.
2. **Local DQNs (Per-drone):** For each drone, a DQN decides sensor activations and offload targets, and short-range waypoint adjustments. Input: local observation o_t^i; Output: discrete actions (sensor on/off, offload to node k, small heading correction).
3. **Central critic / global coach** during training: provides joint reward that includes collision avoidance, fairness, load balancing on edge/cloud (CTDE).

Experience replay, periodic target network updates, prioritized replay, double DQN, dueling network to stabilize learning.

DQN Update Equations (algorithmic summary)

Given replay buffer $\mathcal{D}_{\text{replay}}$, sample mini-batch $\{(s_j, a_j, r_j, s_j')\}_{j=1}^B$:

Compute targets (Double DQN): $a_j^* = \arg\max_{a'} Q(s_j', a'; \theta)$,

$$y_j = r_j + \gamma\, Q(s_j', a_j^*; \theta^-). \tag{20}$$

Loss: $\mathcal{L}(\theta) = \frac{1}{B}\sum_{j=1}^{B}\left(y_j - Q(s_j, a_j; \theta)\right)^2.$

Gradient step: $$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}(\theta). \tag{21}$$

Update $\theta^- \leftarrow \theta$ every τ steps.

Performance Metrics and Theoretical Notes

- Evaluate cumulative reward $\mathbb{E}[\sum_t r_t]$, trajectory length, total energy consumed, average latency for offloaded tasks, and detection accuracy (precision/recall/F1) of surveillance tasks.
- Convergence: with function-approximation DQN and non-stationary multi-agent environment, convergence is not guaranteed; empirical stability improved by CTDE, experience replay, and target networks.
- Scalability: decentralizing policies (per-drone DQNs) and using a meta-controller greatly improves scalability.

Simplified Example Objective (per-step) for Implementation

A compact per-step scalar reward usable for DQN training:

$$\begin{aligned} r_t = \underbrace{\frac{1}{D}\sum_i A_i(t)}_{\text{avg accuracy}} - \alpha \underbrace{\frac{1}{D}\sum_i d(\mathbf{p}_i(t), \mathbf{p}_i(t+1))}_{\text{avg distance}} \\ - \beta \underbrace{\frac{1}{D}\sum_i e_i(t)}_{\text{avg energy}} - \gamma \underbrace{\frac{1}{D}\sum_{i,k} x_{i,k}(t) L_{i,k}(t)}_{\text{avg latency}}. \end{aligned} \tag{22}$$

Tune α, β, γ to shape agent behavior.

- **Agentic planning loop:** Agents can form *sub-goals* (search for regions of interest), internally simulate short rollouts (learned dynamics model) and then commit actions — this is Agentic-AI where each agent uses internal planning to propose actions.
- **Model-based + model-free hybrid:** Learn a dynamics model $\hat{P}$ for short rollouts, use Model Predictive Control (MPC) to propose candidate waypoints, and use DQN to refine sensor/offload decisions.
- **Reward shaping via critical-event supervision:** provide stronger rewards when high-value events are correctly detected and low penalties for false positives to bias the surveillance model toward high recall.

Algorithm Methodology

Algorithm 1. Drone Sensors Data Generation and Collection

Require: Drone state $(\mathbf{p}_i(t), E_i(t))$
Require: Sensor set $\mathcal{S}_i$, feature values $\{q_{i,s}\}$, and weights $\{w_s\}$
Require: Sensing energy costs $e_{i,s}^{\text{sense}}$ and energy coefficient ρ
Require: Accuracy scaling factor κ
Ensure: Active sensor subset $\mathcal{S}_i^{\text{on}}(t)$ and accuracy score $A_i(t)$

Step 1: Compute utility for each sensor
for all sensor $s \in \mathcal{S}_i$ **do**
  $U_{i,s}(t) \leftarrow w_s q_{i,s} - \rho\, e_{i,s}^{\text{sense}}$
end for
Step 2: Select sensors under energy constraint
Solve:

$$\mathcal{S}_i^{\text{on}}(t) = \arg \max_{\mathcal{S}' \subseteq \mathcal{S}_i,\ E_i(t) \geq e_i^{\text{sense}}(\mathcal{S}')} \sum_{s \in \mathcal{S}'} U_{i,s}(t)$$

Step 3: Compute fused surveillance accuracy
$\mathcal{Q}_i(t) \leftarrow \sum_{s \in \mathcal{S}_i^{\text{on}}(t)} w_s q_{i,s}$
$A_i(t) \leftarrow 1 - \exp(-\kappa\, \mathcal{Q}_i(t))$
return $\mathcal{S}_i^{\text{on}}(t)$, $A_i(t)$

Algorithm 1 illustrates how a drone selects which sensors to activate at any given time, balancing energy consumption and sensor quality. First, the utility of each sensor is computed. This is done based on an assessment of the sensor's data quality and the energy consumption it causes. The intuition is to favour sensors that provide better information while consuming less energy. Then, among all available sensors, the drone selects only those that provide the maximum overall benefit while not exceeding the available energy. In other words, it will not activate all sensors but only those relevant to the current energy level and mission requirements. Lastly, once the sensors to be activated have been chosen, the joint sensing performance of these sensors is computed. This provides an overall accuracy score for the drone's data collection at that instant. The higher this score, the more confident the drone is about the data it collects.

Algorithm 1 enables the drone to operate efficiently with minimal energy consumption while still collecting reliable information using its most effective sensors.

Algorithm 2. Edge/Cloud Access and Offloading Decision

Require: Drone state $(b_i(t), E_i(t))$
Require: Available nodes $\mathcal{N}$ with latency $L_{i,k}(t)$, capacity C_k, bandwidth B_k
Require: Data size $\text{data}_i(t)$ and communication energy $e_{i,k}^{\text{comm}}(t)$
Require: Cost weights (η_1, η_2, η_3)
Ensure: Offloading decision $x_{i,k}(t)$ and selected node k^*

Step 1: Compute cost of offloading to each node
for all $k \in \mathcal{N}$ **do**
    Compute total cost:

$$\text{Cost}_{i,k}(t) = \eta_1 L_{i,k}(t) + \eta_2 \frac{\text{data}_i(t)}{b_i(t)} + \eta_3 e_{i,k}^{\text{comm}}(t)$$

end for
Step 2: Select node with minimal feasible cost
Determine:

$$k^* = \arg\min_{k \in \mathcal{N}} \text{Cost}_{i,k}(t)$$

subject to:

$$\sum_i x_{i,k}(t) c_{i,k}(t) \leq C_k$$

Step 3: Update offloading assignment and resources
$x_{i,k^*}(t) \leftarrow 1$
Update bandwidth B_{k^*} and capacity C_{k^*}
return $x_{i,k}(t)$

Algorithm 2 determines whether each drone should locally process its data or offload it to an available edge or cloud node, based on network latency, bandwidth, processing capacity, and the size of the data to be transferred.

First, the drone will check all available nodes and calculate the cost of sending data to each. This cost is based on three main factors: communication delay between the drone and the node, the amount of data, and energy consumption during transmission.

After computing all these costs, the drone compares them and selects the node with the lowest overall cost, provided that node still has sufficient capacity to handle the task. This ensures that no single node becomes overloaded while others are underused.

The algorithm first selects the best node and marks that node as the destination for the current drone, which has been chosen for offloading. Then, it updates the available bandwidth and capacity for that node. Finally, the algorithm's output provides a decision on offloading, specifying where drone data processing will be performed.

Algorithm 3. Edge-Assisted Scheduling and Joint Optimization

Require: Drone set $\mathcal{D}$, edge/cloud nodes $\mathcal{N}$
Require: Weight parameters $(\alpha, \beta, \eta, \lambda)$
Require: Time horizon T
Ensure: Optimized routes $\mathbf{p}_i(t)$, sensor activations $\mathcal{S}_i^{\text{on}}(t)$, offloading decisions $x_{i,k}(t)$

for $t = 0$ to $T-1$ **do**
    for all drone $i \in \mathcal{D}$ **do**
        Step 1: Adaptive sensor data collection
        Run Algorithm 1 to obtain $A_i(t)$
        Step 2: Offloading decision
        Run Algorithm 2 to obtain $x_{i,k}(t)$
        Step 3: Compute instantaneous cost

$$J_i(t) = \alpha\, d(\mathbf{p}_i(t), \mathbf{p}_i(t+1)) + \beta\, e_i(t) + \eta \sum_k x_{i,k}(t) L_{i,k}(t) - \lambda A_i(t)$$

    end for
    Step 4: Update drone routes via meta-controller/DQN
    $\mathbf{p}_i(t+1) \leftarrow \pi_{\text{route}}(\mathbf{p}_i(t), J_i(t))$
end for
return $\{\mathbf{p}_i(t), \mathcal{S}_i^{\text{on}}(t), x_{i,k}(t)\}$ minimizing $\sum_i \sum_t J_i(t)$

Algorithm 3 presents the Edge-Assisted Scheduling and Joint Optimization algorithm, designed to coordinate a group of drones operating under edge-cloud environments. It aims to balance sensing accuracy, communication delay, energy efficiency, and flight distance while ensuring that computation and data processing

are performed in real time through edge assistance. The algorithm operates over a given time period *T*, during which each drone continuously observes, decides on, and updates its actions to minimize the overall mission cost.The system considers all drones
mathcalD and available computing nodes
mathcalN, which may include nearby edge servers or remote cloud nodes. To begin with, each drone has weight parameters ($\alpha, \beta, \eta, \lambda$) that determine the relative importance of the following objectives: flight distance, energy consumption, transmission delay, and sensing accuracy. For each time slot *t*, the algorithm proceeds, involving several steps to make decisions regarding the current state of the drone in relation to the network conditions.

In the second step, each drone decides on where to process the data it has collected. Using the *Edge/Cloud Access and Offloading Decision* algorithm (Algorithm 2), it evaluates all available nodes by calculating the total cost of sending the data to each. It considers this cost in terms of transmission delay, data size, and communication energy with respect to available bandwidth. Then, the node with the lowest cost is chosen, provided it still has sufficient computational capacity. This way, the load can be distributed efficiently among the network's edge and cloud nodes.

Following these decisions, the algorithm calculates an instantaneous cost value $J_i(t)$ for each drone, as a combination of multiple factors: distance between current and following positions weighted by α, the energy usage of the drone weighted by β, the offloading delay weighted by η, and the accuracy of the collected data being rewarded by λ. The combination of these allows the system to manage the trade-off between performance and efficiency in line with mission priorities.

We update the drone routes using the route policy, π_{route}, which can be a meta-controller or a Deep Q-Network. The policy learns from the past to help drones decide on the most cost-effective positions for the next time step. The overall optimization minimizes the total cost $sum_i\ sum_t J_i(t)$, which leads to energy-efficient flight paths while performing a coordinated task distribution among drones. This algorithm unifies sensor activation, task offloading, and path planning into a single framework. This approach ensures that a drone swarm operates effectively and consumes minimal energy by combining real-time edge computing decisions with adaptive sensing and smart route selection, thereby maintaining high mission performance under dynamic edge-cloud conditions.

Algorithm 4. Agentic-AI DQN-Based Learning for Multi-Drone Optimization

---

Require: Replay buffer $\mathcal{D}*$replay, learning rate $\eta*\theta$, discount factor γ, target update interval τ
Ensure: Trained Q-network parameters θ

Initialize Q-network $Q(s, a; \theta)$ and target network $Q(s, a; \theta^{-})$
while training not converged **do**
    '''
    Observe current state s_t
    Choose action a_t using ϵ-greedy policy
    Execute a_t, observe reward r_t and next state s_{t+1}
    Store experience (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}_{\text{replay}}$
    Sample mini-batch $\{(s_j, a_j, r_j, s'_j)\}$ from $\mathcal{D}_{\text{replay}}$
    Compute target:

$$y_j = r_j + \gamma\, Q(s'_j, \arg\max_{a'} Q(s'_j, a'; \theta); \theta^{-})$$

    Update Q-network parameters:

$$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \frac{1}{B} \sum_j (y_j - Q(s_j, a_j; \theta))^2$$

    if step % τ = 0 **then**
        $\theta^{-} \leftarrow \theta$
    end if'''
end while
return θ

Algorithm 4 presents the Agentic-AI DQN-based learning approach for training multiple drones to make intelligent decisions in a dynamic environment. The aim is to optimise mission objectives, including travel efficiency, sensing accuracy, latency, and energy consumption, using deep reinforcement learning. In this approach, DQN enables drones to learn optimal actions by interacting with their environment rather than relying on predefined rules.

We start training by initializing two neural networks: the main Q-network $Q(s, a; \theta)$ and the target network $Q(s, a; \theta^{-})$. Both have randomly set parameters. While the Q-network predicts the expected reward for taking action a in state s, the target network stabilizes learning by providing consistent estimates of future rewards.

At each training iteration, the agent (or drone) observes its current state s_t and chooses an action a_t using an ϵ-greedy strategy. Precisely, most of the time it selects the most well-known action, but occasionally it explores new actions to enhance learning. The drone executes that action and then receives a reward, r_t computed by the reward function Equation eqrefeq:reward, and transitions into a new state s_{t+1}. This experience is represented as a tuple, (s_t, a_t, r_t, s_{t+1}), which is then stored in a replay buffer.

The replay buffer $\mathcal{D}_{\text{replay}}$ stores a history of past experiences from which mini-batches are sampled randomly for training. This random sampling destroys the correlation between successive experiences and serves to maintain stability in learning. For each sampled experience, a target value y_j is estimated by combining the observed reward with the estimated future reward from the target network. The mean squared error between the predicted Q-value and the target value determines the loss function for each prediction. The parameters θ of the main Q-network are updated through the use of gradient descent at a learning rate η_θ.

The training process runs until convergence, while the Q-network learns an approximation of the optimal action-value function $Q^*(s, a)$. After training, each drone can utilize this network for real-time decision-making. With the learnt policy, drones can select the best actions for movement, sensor activation, and task offloading to balance energy efficiency, communication delay, and mission accuracy. The time complexity is determined by $Olog(n \times n)$ where n is the number of iterations of the methodology, and if n is the sub-iteration of the methodology. Whereas it K is scalable designated space into computing nodes.

Performance Evaluation

In the performance evaluation, we conducted the experiments based on statistical algorithms and evaluation and compare different state of art studies with the proposed method.

Karachi Drone Surveillance Dataset Scenario

Figure 2 shows the simulation testbeds of the proposed model in Karachi, with testbed drones flying into Karachi Smart City to collect data.

The Karachi Drone Surveillance Dataset represents real-world and testbed-collected data obtained from drone-based surveillance operations across the metropolitan region of Karachi. The operational area covers approximately 100×100 kilometers, encompassing central districts such as Korangi, Clifton, Saddar, Gulshan, Malir, North Nazimabad, SITE, and DHA. Each surveillance zone is assigned spatial coordinates (latitude and longitude), allowing spatiotemporal analysis of aerial monitoring and drone coordination patterns.

The dataset was collected through multiple drone testbeds deployed over urban and semi-urban environments. Each record includes detailed operational parameters such as *drone ID*, *flight ID*, *operator ID*, *timestamp*, *zone*, *altitude*, *speed*, and *heading*. In addition, sensor-level information—such as onboard camera status, detection summary, and event type—was recorded in real time. System attributes, including battery level, communication delay, anomaly detection, and detection confidence, are also captured to describe each drone's operational state and situational context.

The data span an extensive 100-kilometre range, with drone altitudes from 50 m to 250 m, representing both low-altitude surveillance and wide-area coverage missions. Flight speeds range from 5 m/s to 20 m/s, reflecting the realistic operational dynamics of surveillance UAVs in urban airspace. Data were collected under various conditions, including daytime, nighttime, and high-traffic hours, to ensure the diversity and reliability of the testbed measurements.

Each drone mission focuses on surveillance, traffic monitoring, and anomaly detection. The `detected_summary` field records the type of observed object (e.g., person, vehicle, crowd, or unknown), while the `anomaly_flag` identifies abnormal activities that require operator or system attention. The `event_type` attribute distinguishes among regular patrols, emergency responses, and traffic-monitoring operations.

This dataset provides a high-fidelity representation of real drone activity in Karachi's complex urban envi-

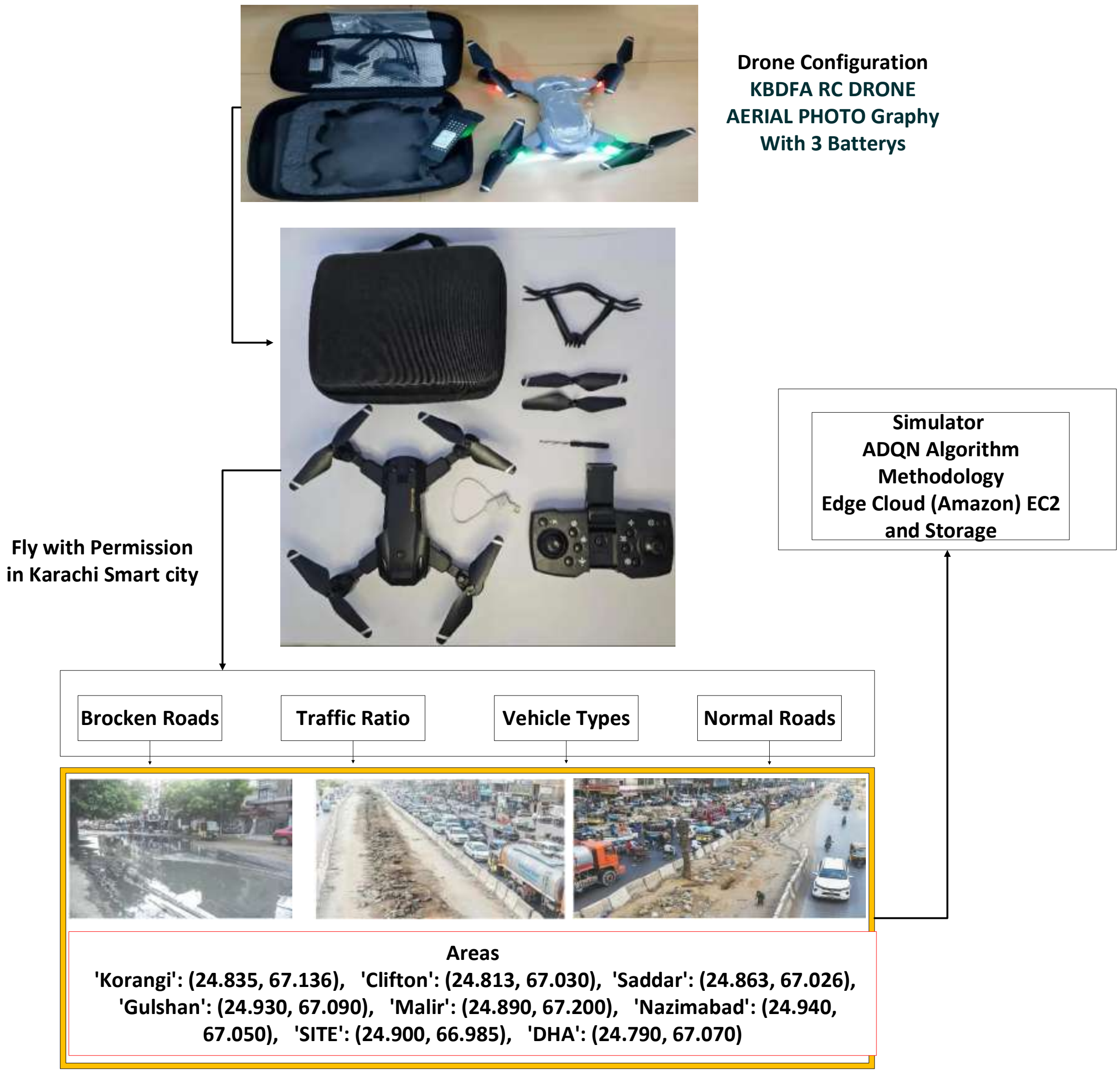

Figure 2. Simulation Testbeds in Karachi

ronment. It is suitable for evaluating algorithms related to route optimization, task scheduling, latency minimization, energy-aware flight control, and cooperative edge intelligence. The large-scale, city-wide coverage supports research on scalability, network bandwidth optimization, and intelligent coordination of drones under real-world environmental and operational constraints.

Result Analysis

In the results analysis, we used the three baseline approaches, DQN [21], AMS, and ADQNS [22], to evaluate the proposed schemes.

Figure 3 shows real-time drone data from multiple locations. Black points indicate drone locations with height, and data from other areas, such as Gulshan, Saddar, Nazimabad, and others, are shown with latitude and longitude. We collected data from the testbed using a drone that flew in a fixed area within Karachi Smart City. The data are on road traffic and road conditions across different regions of Karachi Smart City.

Figure 4 and Figure 5 energy and latency comparisons across the three algorithms, such as DQN, AMS, and ADQNS, illustrate apparent performance differences when applied to the Karachi Drone Dataset, revealing how each approach manages computational load, decision-making efficiency, and operational overhead in drone-based environments. In the energy graph,

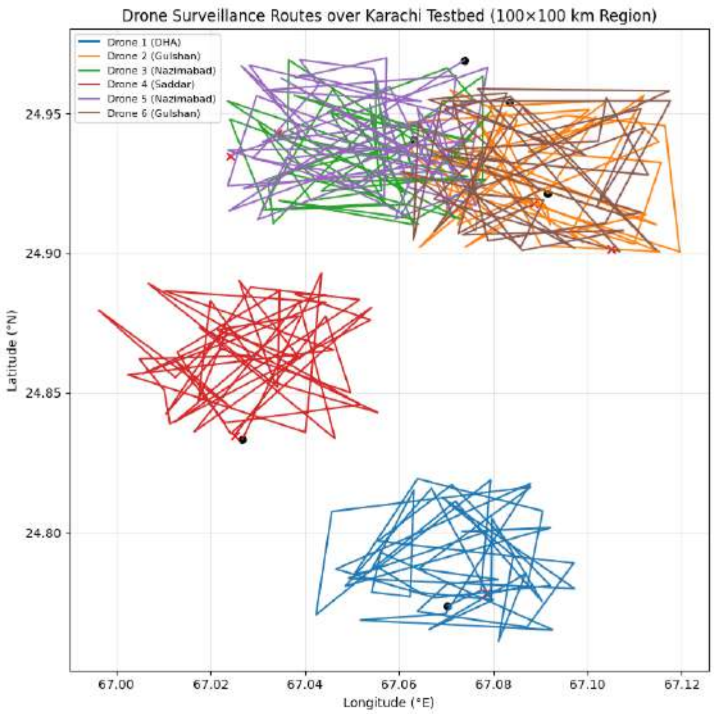

Figure 3. Drone Data Collection from Different Areas

DQN exhibits the highest consumption at approximately 125 joules, indicating that its reinforcement-learning decision mechanism requires more frequent or computationally expensive evaluations, resulting in greater power draw during drone navigation or service execution.

AMS performs moderately, with approximately 110 joules, reflecting improved optimization compared to DQN but still requiring significant computational resources, likely due to its iterative structure. ADQNS exhibits the lowest energy usage of roughly 98 joules, suggesting a more efficient design that reduces redundant computations, optimizes state transitions, or enables faster convergence, thereby preserving battery life and making it more suitable for long-duration drone missions in resource-limited settings. The latency comparison follows a similar trend, with DQN again performing worst, at nearly 480 milliseconds, indicating that its model takes longer to process inputs and generate actions, which can negatively impact real-time flight stability and responsiveness. AMS reduces the delay to around 390 milliseconds, indicating more efficient control-flow execution but still not ideal for highly dynamic drone manoeuvres. ADQNS achieves the lowest latency at approximately 310 milliseconds, demonstrating faster inference, quicker decision transitions, and overall smoother task execution.

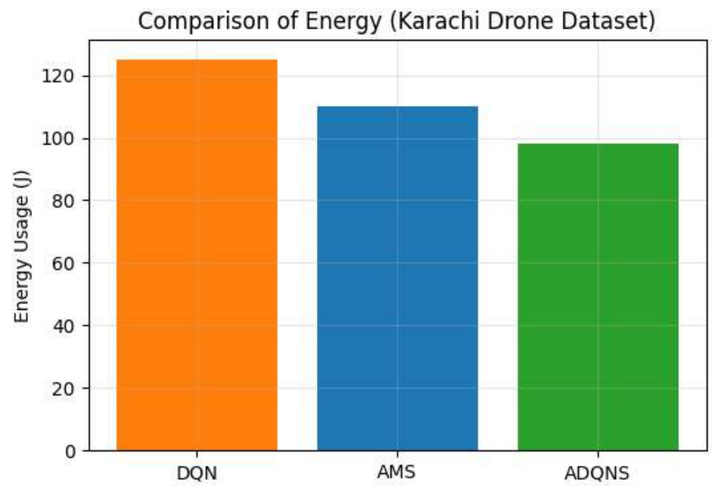

Figure 4. Drone Data Collection from Different Areas and Energy Consumption

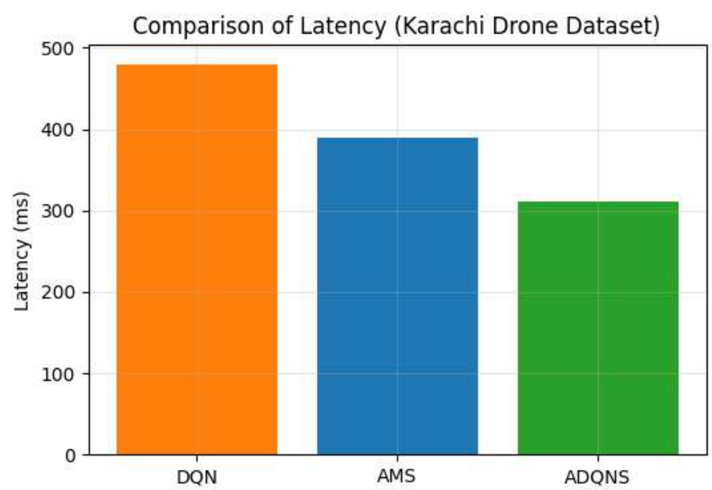

Figure 5. Drone Data Collection from Different Areas and Latency

When considering drone operations such as obstacle avoidance, mission scheduling, data collection, or service invocation, these latency advantages directly translate to more responsive behavior, fewer delays, and more stable flight patterns. Overall, the combined results show that ADQNS significantly outperforms both DQN and AMS in terms of energy efficiency and operational delay, making it a more viable solution for real-time, energy-constrained drone applications in smart-city aerial systems, such as those modeled in the Karachi Drone Dataset.

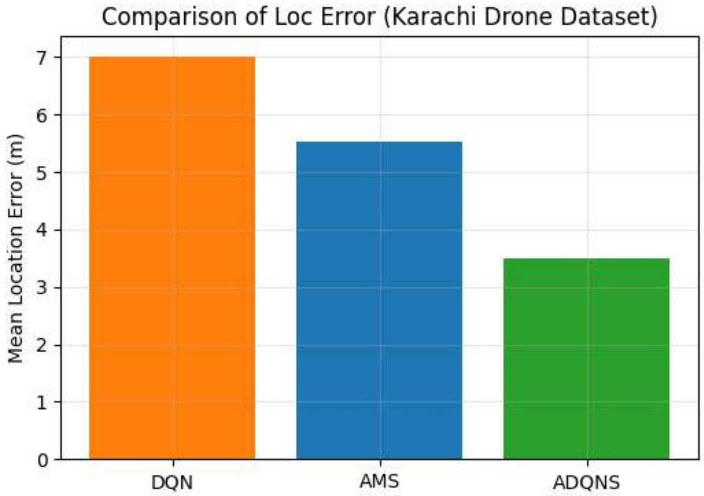

Figure 6. Localisation Detection with Higher Accuracy

Figure 6 location-error comparison for Karachi while having navigation precision during drone operations. DQN exhibits the highest mean location error of approximately 7 meters, indicating that its decision-making strategy struggles to maintain stable positional accuracy, likely due to slower convergence, suboptimal path adjustments, or noisy state estimates that propagate through successive flight steps, ultimately causing the drone to deviate farther from the intended coordinates. AMS performs moderately, with an average error of about 5.5 meters, showing improvement over DQN but still exhibiting limitations in compensating for dynamic environmental changes, such as wind variations, dataset noise, or real-time trajectory shifts. Its internal optimization reduces some fluctuations, yet it cannot fully mitigate cumulative drift.

ADQNS achieves the lowest error at approximately 3.5 meters, demonstrating superior spatial awareness and more robust handling of sensor inputs, trajectory corrections, and adaptive learning signals, thereby enabling the drone to align more accurately with target positions. This improvement suggests that ADQNS incorporates mechanisms that allow faster state adaptation, reduced prediction uncertainty, and more refined policy updates, thereby maintaining consistent stability even in complex real-world aerial conditions. Overall, the graph shows that while DQN and AMS can handle basic navigation tasks, ADQNS achieves significantly more precise and reliable localization, making it the most suitable for mission-critical drone applications where accurate positioning is essential, such as delivery, surveillance, mapping, and emergency response operations.

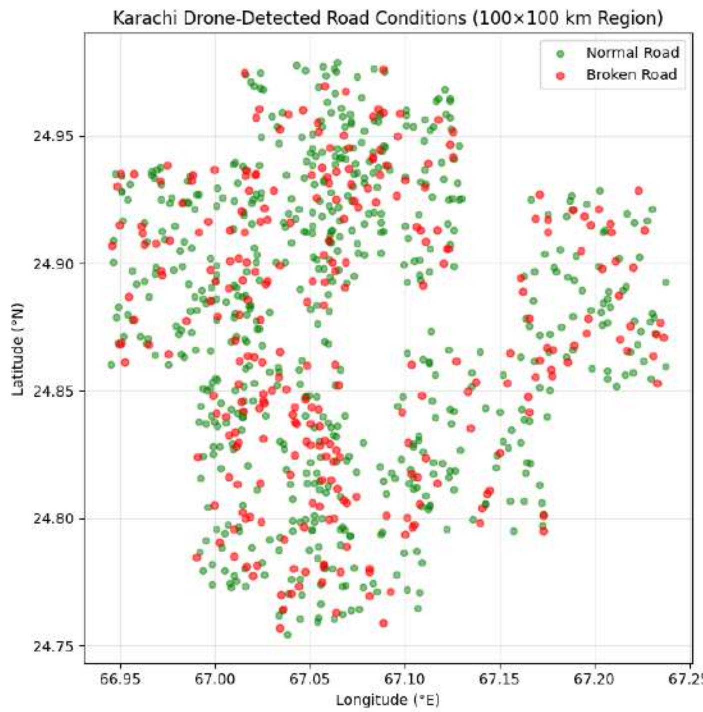

Figure 7. Road Detection with Higher Accuracy

Figure 7 shows the Karachi Drone-Detected Road Conditions (100 × 100 km Region), which provides a geographic visualization of road-quality observations captured by a drone across a square region of Karachi. The plot uses a coordinate system where the Longitude (°E) is shown on the x-axis and the Latitude (°N) on the y-axis, covering an area approximately from 66.95°E to 67.25°E in longitude and from 24.75°N to 25.00°N in latitude. Each point represents a specific location on the map, and its color indicates the drone-detected road condition: green points represent regular road segments. In contrast, red points denote broken road segments. The distribution of these points, including the proximity of the red and green clusters, highlights the heterogeneous nature of the region's road network, in which well-maintained and deteriorated sections are intermixed.

This spatial pattern, together with the overall point density across the surveyed area, reflects the heterogeneous quality of infrastructure in the region. Such detailed mapping of road conditions serves as critical input for route-optimization algorithms, including those evaluated in the first figure, since these systems depend on accurate condition data to compute the most efficient paths that minimize traversal through broken or degraded road segments.

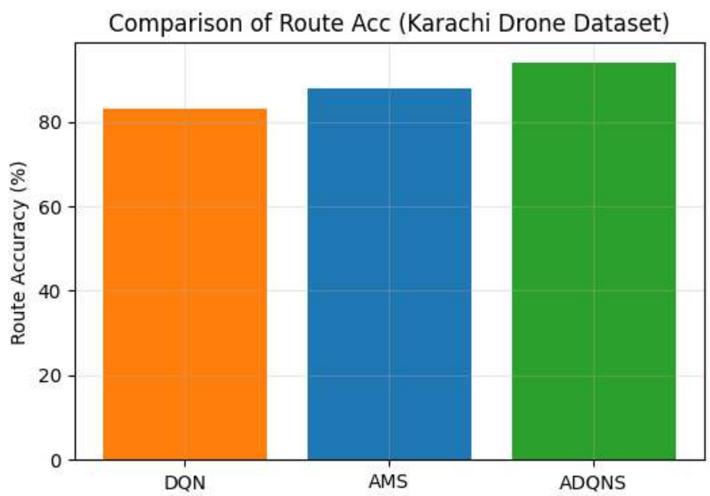

Figure 8. Route Detection with Higher Accuracy

Figure 8 Route Acc (Karachi Drone Dataset)" illustrates the performance of three different route-planning algorithms—DQN, AMS, and ADQNS—in terms of their Route Accuracy when tested against the Karachi Drone Dataset. The accuracy metric reflects how reliably each algorithm selects or proposes a route that meets specific criteria, likely optimizing for factors such as distance and time, and possibly avoiding damaged road sections, as suggested by the second figure. The DQN (Deep Q-Network) algorithm achieves an accuracy of just over 80%, serving as a baseline. The AMS algorithm shows a significant improvement over DQN, reaching an accuracy level of approximately 88%. The ADQNS algorithm achieves the best performance, with the highest route accuracy of just over 90%. This visual comparison demonstrates that the ADQNS method is the most effective among the three for generating accurate routes in this drone-based dataset, suggesting that it employs a more robust logic or a more refined learning process than the other two methods for navigating routes within the designated region.

Finding and Limitation

The main finding of this study is the proposal of an adaptive, intelligent smart city surveillance system based on agentic intelligence. Whereas, the proposed methodology improves the overall accuracy. The main limitation is that we need to add security into the current version of the system.

Conclusion and Future Work

This research was focused on the development of an integrated agentic AI and DQN-based routing framework for unmanned drones in distributed edge-cloud environments, with the aim of facilitating efficient traffic and road condition surveillance in the Karachi Smart City. The findings of this research reveal that the integration of autonomous reasoning with reinforcement learning can significantly enhance unmanned drones' ability to navigate complex urban environments. As discussed in the findings, the proposed system can learn from real-time data and dynamically adjust its routing decisions to select the most efficient routes. Evaluation of the proposed system across diverse metropolitan areas, including Saddar, Nazimabad, Gulshan-e-Iqbal, Clifton, and Korangi, has revealed significant improvements in route accuracy, traffic detection, and the identification of deteriorating road conditions. The findings have revealed that the application of agentic AI in conjunction with DQN can effectively address combinatorial route optimisation, with significant improvements in robustness and decision quality.

The focus of the forthcoming research will be to extend the proposed system to address increasingly complex urban scenarios, including security and cooperative behaviour among drones, swarm intelligence, and task sharing, to further reduce surveillance time and energy consumption. The system could also be improved by incorporating additional sensor modalities, such as thermal imaging, LiDAR, and acoustic sensing, to more effectively detect infrastructure anomalies and emergencies.

Dataset

The paper shared the dataset publicly on the following URL: https://github.com/arlakhan/Karachi-GPS-Drone-

Data/upload.

Author Contribution:

Akhtar Ali Junejo: Conceptualization, Methodology, Software implementation, Data curation, Writing-Original. **Zain Gul:** Methodology. **Qurat-ul-ain Mastoi:** Writing-Original draft preparation. **Asif Aziz Memon:** Investigation. **Atif Jamil:** Software Validation. **Abdullah Lakhan:** Writing-Original draft preparation, Visualization, Investigation, Supervision, Software Validation, Writing- Reviewing and Editing.

Compliance with Ethical Standards

It is declared that all authors do not have any conflict of interest. Furthermore, informed consent was obtained from all individual participants included in the study.

AI Assistance Disclosure

We have used Grammarly AI tool for improving the English correction and rephrasing of English into the manuscript.

References

[1] H. Kim, H. Kim, and M. Guizani, "A resting drone-enabled surveillance framework for real-time monitoring with resource efficiency in IoT-based smart city," *IEEE Internet of Things Journal*, 2025.

[2] A. Abu-Khadrah, A. Al-Qerem, M. R. Hassan, A. M. Ali, and M. Jarrah, "Drone-assisted adaptive object detection and privacy-preserving surveillance in smart cities using whale-optimized deep reinforcement learning techniques," *Scientific Reports*, vol. 15, no. 1, p. 9931, 2025.

[3] M. Bakirci, "Internet of Things-enabled unmanned aerial vehicles for real-time traffic mobility analysis in smart cities," *Computers and Electrical Engineering*, vol. 123, p. 110313, 2025.

[4] C. Qin and E. Pournaras, "Strategic coordination of drones via short-term distributed optimization and long-term reinforcement learning," *Applied Soft Computing*, p. 114270, 2025.

[5] J. Yang, J. Tao, C. Wang, and Q. Yang, "Cloud-edge collaborative optimization based on distributed UAV network," *Electronics*, vol. 13, no. 18, p. 3763, 2024.

[6] U. Awada, J. Zhang, S. Chen, S. Li, and S. Yang, "Edge-Drones: Co-scheduling of drones for multi-location aerial computing missions," *Journal of Network and Computer Applications*, vol. 215, p. 103632, 2023.

[7] R. Sapkota, K. I. Roumeliotis, and M. Karkee, "UAVs meet agentic AI: A multidomain survey of autonomous aerial intelligence and agentic UAVs," *arXiv preprint arXiv:2506.08045*, 2025.

[8] Y. Spyridis, "Efficient control of drones communications in IoT," Ph.D. dissertation, University of Sheffield, Sheffield, U.K., 2024.

[9] X. Tang, Q. Chen, W. Weng, B. Liao, J. Wang, X. Cao, and X. Li, "DNN task assignment in UAV networks: A generative AI enhanced multi-agent reinforcement learning approach," *IEEE Internet of Things Journal*, 2025.

[10] Y. Ding, Z. Yang, Q.-V. Pham, Y. Hu, Z. Zhang, and M. Shikh-Bahaei, "Distributed machine learning for UAV swarms: Computing, sensing, and semantics," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7447–7473, 2023.

[11] C. Zhao *et al.*, "Edge general intelligence through world models and agentic AI: Fundamentals, solutions, and challenges," *arXiv preprint arXiv:2508.09561*, 2025.

[12] F. S. Chagas, N. Ruseno, and A. A. A. Bechina, "Artificial intelligence approaches for UAV deconfliction: A comparative review and framework proposal," *Automation*, vol. 6, no. 4, p. 54, 2025.

[13] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, "Computation offloading and resource allocation in MEC-enabled integrated aerial–terrestrial vehicular networks: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21478–21491, 2022.

[14] Y. Wang, J. Li, X. Yang, and Q. Peng, "UAV–ground vehicle collaborative delivery in emergency response: A review of key technologies and future trends," *Applied Sciences*, vol. 15, no. 17, p. 9803, 2025.

[15] X. Gao *et al.*, "Agentic satellite-augmented low-altitude economy and terrestrial networks: A survey on generative approaches," *arXiv preprint arXiv:2507.14633*, 2025.

[16] C. T. Nguyen *et al.*, "Emerging technologies for 6G non-terrestrial-networks: From academia to industrial applications," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3852–3885, 2024.

[17] S. A. Nahiyoon *et al.*, "Recent development trends in plant protection UAVs: A journey from conventional practices

to cutting-edge technologies—A comprehensive review," *Drones*, vol. 8, no. 9, p. 457, 2024.

[18] P. A. Prakash and R. Radha, "Advancements in AI-powered electric vehicle routing: Multi-constraint optimization and infrastructure integration approaches for evolving EVs—A survey," *IEEE Access*, 2025.

[19] I. Kishor, U. Mamodiya, V. Patil, and N. Naik, "AI-integrated autonomous robotics for solar panel cleaning and predictive maintenance using drone and ground-based systems," *Scientific Reports*, vol. 15, no. 1, p. 32187, 2025.

[20] M. Neelesh, "A reinforcement learning agent for UAV control: Mathematical foundations, implementation, and human-vs-AI benchmarking," *International Journal of Trend in Scientific Research and Development*, vol. 9, no. 2, pp. 242–254, 2025.

[21] V. C. Sharma and S. Roy, "An overview of Q-learning and deep Q-learning for an autonomous multi-UAV wireless network," *Expert Systems*, vol. 42, no. 11, p. e70143, 2025.

[22] T. Ahmad, A. Morel, N. Cheng, K. Palaniappan, P. Calyam, K. Sun, and J. Pan, "Future UAV/drone systems for intelligent active surveillance and monitoring," *ACM Computing Surveys*, vol. 58, no. 2, pp. 1–37, 2025.