



DEEP Q-NETWORK-BASED ADAPTIVE EDGE INTELLIGENCE FOR ENERGY-EFFICIENT AIR QUALITY MONITORING IN IOT NETWORKS

Donatien Kadima Muamba

Faculty of Computer Sciences, Reverend KIM University
School of Telecommunications and Space Remote Sensing,
National Pedagogical University, Democratic Republic of
the Congo

*Corresponding author

officieldonatien2@gmail.com

Article history:

Received June 28, 2026

Revised July 12, 2026

Accepted August 10, 2026

Keywords:

Internet of Things (IoT);

Edge Intelligence;

Deep Q-Network (DQN);

Air Quality Monitoring;

Energy Efficiency;

Smart Environment.

Abstract

The rapid deployment of Internet of Things (IoT) networks enables environmental monitoring, but conventional fixed-interval sensing causes excessive energy consumption, communication overhead, and limited adaptability. This study proposes an adaptive edge intelligence framework using Deep Q-Network (DQN) for energy-efficient air quality monitoring. The system integrates environmental sensors with an ESP32-based edge node to measure PM_{2.5}, PM₁₀, CO₂, NO_x, SO_x, NH_x, temperature, humidity, and traffic-related information. A DQN agent dynamically selects sensing and communication actions using nine environmental and contextual state variables. A customized reward function balances air quality conditions, energy consumption, communication costs, and monitoring requirements. Experimental data were collected through an IoT platform and stored in Supabase. The agent was trained using 103 valid observations over 200 episodes and independently evaluated on 114 observations. Training rewards increased from -43.62 to approximately 270 after convergence. Evaluation achieved a total reward of 316.092, an average reward of 2.773 per decision step, and an average energy cost of 0.20 per step. Under stable environmental conditions, the agent consistently selected SLOW_SAMPLE, reducing unnecessary sensing operations. These findings demonstrate the feasibility of DQN-based adaptive monitoring and provide a scalable foundation for energy-efficient, self-optimizing IoT environmental sensing systems under varying environmental conditions.

1.0 INTRODUCTION

Air pollution has become a major environmental challenge worldwide due to its impact on human health, climate change and urban sustainability. Air quality has become highly variable in space and time due to rapid urbanization, industrial activities and increasing transportation demand. Therefore, ongoing surveillance of atmospheric pollutants is necessary to support environmental management and intelligent decision-making in smart cities. Traditional air quality monitoring systems are primarily based on sophisticated and expensive instruments installed in fixed stations. Accurate measurements can be obtained by these systems but high cost of deployment, limited spatial coverage, and lack of operational flexibility limit their applicability for large scale and ongoing monitoring. An affordable alternative is emerging by achieving distributed sensing, continuous data acquisition and

remote supervision of the environment via interconnected sensor nodes, namely the Internet of Things (IoT)-based monitoring systems.

However, most of the traditional IoT monitoring systems are designed to operate based on pre-defined sensing and transmission intervals. Thus measurements are taken and transmitted regardless of the real environmental conditions. During stable periods this may lead to redundant measurements, unnecessary communication traffic and energy consumption and thus to a reduced lifetime of resource-constrained IoT devices. In contrast, fixed sensing policies may not be fast enough to be used for monitoring when the environment deteriorates. Therefore, an intelligent scheme that can adapt sensing operations to environmental dynamics is needed. The incorporation of Artificial Intelligence (AI) with IoT, particularly via Edge Intelligence, is a viable solution to tackle this challenge.

The processing and decision-making at the data source can reduce the communication overhead and latency as well as improve the autonomy of the IoT nodes. Of the AI techniques, Reinforcement Learning (RL) is particularly well-suited for sequential decision-making in dynamic environments, because an agent can learn an optimal policy thru interaction with its environment. Deep Q-Networks (DQN) combine Q-learning with deep neural networks [1]–[6] extending this ability to complex state spaces, and have been successfully investigated for intelligent resource management in distributed and IoT systems. Nevertheless, existing air quality monitoring solutions mainly focus on sensing, visualization of data and pollutant prediction, and adaptive optimization of sensing operations has been less investigated. Moreover, methods dependent on cloud-based processing may incur additional communication overhead and latency.

These limitations motivate the design of an edge-intelligent monitoring framework that autonomously adapts sensing and communication strategies based on environmental conditions while considering resource constraints. To fill this research gap, this paper proposes a Deep Q-Network-Based Adaptive Edge Intelligence Framework for Energy-Efficient Air Quality Monitoring in IoT Networks. The proposed system includes a multi-sensor ESP32-based IoT node, cloud data management, a DQN-based decision-making agent, and a web-based monitoring dashboard. The agent observes a nine-dimensional environmental state that includes PM2.5, PM10, CO2, NOx, SOx, NHx, temperature, humidity, and traffic information. It then chooses a suitable sensing or communication action depending on this state thru a customized reward function considering the environmental monitoring requirements and resource consumption. The main contributions of this work are as follows. An adaptive IoT edge architecture for multi-parameter environmental sensing, edge intelligence, cloud data management and real time visualization for air quality monitoring. An adaptive environmental monitoring problem can be formulated as a Markov Decision Process, where the environmental measurements are the state, the sensing and communication strategies are the action space, and the reward function is designed to guide the optimization. A DQN-based adaptive sensing mechanism that can learn the monitoring policies based on the environmental conditions with the consideration of the sensing energy and communication costs.

Experimental validation of a real IoT prototype. A web-based monitoring dashboard implemented with an ESP32 based sensing node and a cloud-connected data infrastructure. We experimentally evaluate DQN convergence, adaptive action selection, reward performance, energy consumption and air quality monitoring behavior using measurements from the real environment. The rest of the paper is organized as follows. Section II reviews related work on IoT-based air quality monitoring, Edge Intelligence and reinforcement learning. Section III describes the proposed system architecture and mathematical formulation. Section IV discusses the experimental implementation and evaluation methodology. The experimental results and discussion are given in Section V. Finally, the paper is concluded in Section VI with discussions on future research directions.

2.0 THEORETICAL

2.1 IoT-Based Air Quality Monitoring Systems

IoT based air quality monitoring is a low cost solution comparing to the traditional fixed monitoring stations. Most of the existing systems use low cost environmental sensors combined with embedded platforms and wireless communication technologies to monitor parameters

such as PM2.5, PM10, CO2, NOx, temperature and humidity continuously. These approaches have shown the feasibility of distributed and real-time environmental monitoring, especially in smart-city applications [7], [10], [17].

Wireless sensor network (WSN) approaches with multiple sensing nodes placed in different monitoring areas have further enhanced the spatial coverage. However, continuous sensing and communication may cause a significant increase in the energy consumption and communication overhead, especially for battery-powered and resource-constrained devices. Fixed rate sensing strategies may lead to redundant measurements when the environment remains stable. Machine learning and Edge Intelligence have been embedded in recent works of IoT environmental monitoring to improve pollutant prediction, anomaly detection and air quality classification [8], [9], [18]. Such approaches improve the data analysis and prediction capabilities, however the monitoring process itself is typically governed by pre-defined sensing policies. Thus the sensing node is mostly passive, collecting and transmitting measurements without dynamically adapting its behavior to the environmental conditions. Reinforcement learning provides a complementary approach that allows IoT systems to learn sequential decision-making policies thru interaction with their environment. DQN-based methods have been used for investigating resource allocation, computation management and energy optimization in IoT and edge computing environments [1]–[6], [12]. But its application for adaptive air quality sensing is still quite limited.

This gap motivates the development of an autonomous monitoring framework, in which sensing and communication actions are directly determined from environmental observations. Unlike the traditional approaches that mainly optimize data analysis, the proposed framework focuses on the optimization of the monitoring system behavior itself. This allows the IoT node to adapt its sensing strategy according to the environmental conditions, while considering energy and communication costs.

2.2 Edge Computing and Edge Intelligence for IoT Applications

Edge computing has emerged as an effective paradigm to reduce communication latency, bandwidth consumption and cloud dependency for IoT applications. Edge-based architectures enable faster analysis and more responsive monitoring of the environment because sensor data is processed closer to where the data originates. Edge Intelligence is the union of Artificial Intelligence (AI) and edge computing, where data is analyzed and decisions are made autonomously at the network edge. However, deploying artificial intelligence algorithms on resource-constrained IoT devices is still a challenge due to limited computational, memory and energy resources [10]. Current edge-based environmental monitoring solutions mostly rely on static prediction, classification or anomaly-detection models. These techniques improve data analysis, but they typically do not dynamically determine how sensing and communication resources should be managed according to changes in the environment. This limitation necessitates the use of adaptive decision-making techniques to dynamically reconfigure IoT operations. Reinforcement learning (RL) and especially Deep Q-Networks (DQN) are an appropriate framework for this task, as they allow an agent to learn sequential decision-making policies by interacting with his environment. In the proposed system, DQN is therefore used to convert environmental observations into adaptive sensing and communication decisions taking into account monitoring requirements and resource consumption.

2.3 Reinforcement Learning for Intelligent IoT Resource Management

Reinforcement Learning (RL) has emerged as a suitable approach for optimization and sequential decision making in dynamic IoT environments. RL differs from supervised learning in that an agent learns decision policies thru interaction with an environment, thus enabling to optimize system objectives over the long run. The application of RL has been studied in the IoT networks for energy management, routing, task scheduling, spectrum allocation and adaptive sensing. Its capacity to consider simultaneously environmental conditions, system constraints and long-term rewards makes it particularly relevant for resource-constrained monitoring systems.

But for high dimensional or continuously varying state spaces, which is often the case in IoT environments, conventional RL approaches may become inefficient. Deep Reinforcement Learning (DRL) addresses this weakness by approximating complicated value functions or policies with deep neural networks. Among the methods of DRL, Deep Q-Network (DQN) is especially suitable for discrete decision-making problems. DQN is based on Q-learning and function approximation with neural networks. It also adds Experience Replay and Target Network to stabilize the training. These properties make DQN suitable for adaptive IoT sensing where the agent has to choose from a discrete set of monitoring strategies given the observed state of the environment. However, the application of DQN to adaptive air quality sensing at IoT edge simultaneously considering environmental monitoring, energy consumption, and communication cost is still under-explored. This gap motivates the framework proposed in this work.

2.4 Deep Q-Network Applications in IoT Systems

Deep Q-Network (DQN) combines Q-learning and deep neural networks to approximate the action-value functions in complex decision-making environments [1]–[4]. Its ability to learn adaptive policies from feedback received from the environment makes it suitable for resource-constrained IoT applications such as energy management, computation offloading, and network optimization [8–10]. In IoT systems, DQN may optimize multiple objectives thru a properly designed reward function, such as energy consumption, communication cost, latency and service quality [9]–[12]. However, its application to adaptive air quality monitoring is limited. Most of the present work on IoT air quality is focused on machine learning for the prediction or classification of pollutants, rather than the autonomous control of sensing and communication operations. To address this gap, this work proposes a DQN-based adaptive edge intelligence framework where the agent dynamically selects sensing and communication strategies based on the observed environmental state and resource constraints.

2.5 Comparative Analysis of Existing Approaches

Reference	Application	AI Method	Edge Processing	Adaptive Sampling	Energy Optimization
Study A	Air quality monitoring	Machine Learning	No	X	X
Study B	IoT environmental sensing	Cloud analytics	Limited	X	X
Study C	Wireless sensor network	Reinforcement Learning	Yes	✓	✓
Study D	IoT resource management	Deep Reinforcement Learning	Yes	✓	✓
Proposed Work	Adaptive Air Quality IoT	DQN	Yes	✓	✓

3.0 METHODOLOGY

Although significant progress has been achieved in IoT-based air quality monitoring, current solutions still present three major limitations:

1. Lack of autonomous sensing adaptation:
Most systems collect measurements using fixed sampling strategies regardless of environmental variations.
2. Limited edge intelligence:
Many approaches depend on cloud infrastructures, increasing latency and communication costs.
3. Insufficient multi-objective optimization:
Existing methods rarely consider the simultaneous optimization of energy consumption, monitoring accuracy, communication overhead, and response time.

To overcome these limitations, this paper proposes an adaptive Deep Q-Network-based Edge Intelligence framework capable of learning optimal monitoring strategies according to environmental conditions and IoT resource constraints.

3.1 Proposed Deep Q-Network-Based Adaptive Edge Intelligence Architecture

In this section, the proposed DQN-based adaptive edge intelligence architecture for autonomous air quality monitoring in IoT environments is presented. Unlike conventional monitoring systems that use fixed sensing intervals and static communication policies, the proposed framework continuously learns the most appropriate monitoring strategy according to the environmental dynamics. The architecture combines IoT sensing, edge computing and Deep Reinforcement [1], [2] Learning for intelligent decision making on the edge node. The sensing frequency, communication cost, energy consumption and monitoring accuracy are all to be optimized at the same time. The proposed framework includes four integrated layers: (1) the IoT sensing layer to acquire environmental data, (2) the edge intelligence layer to execute the DQN agent, (3) the cloud storage and visualization layer, and (4) the user application layer for real-time monitoring and decision support.

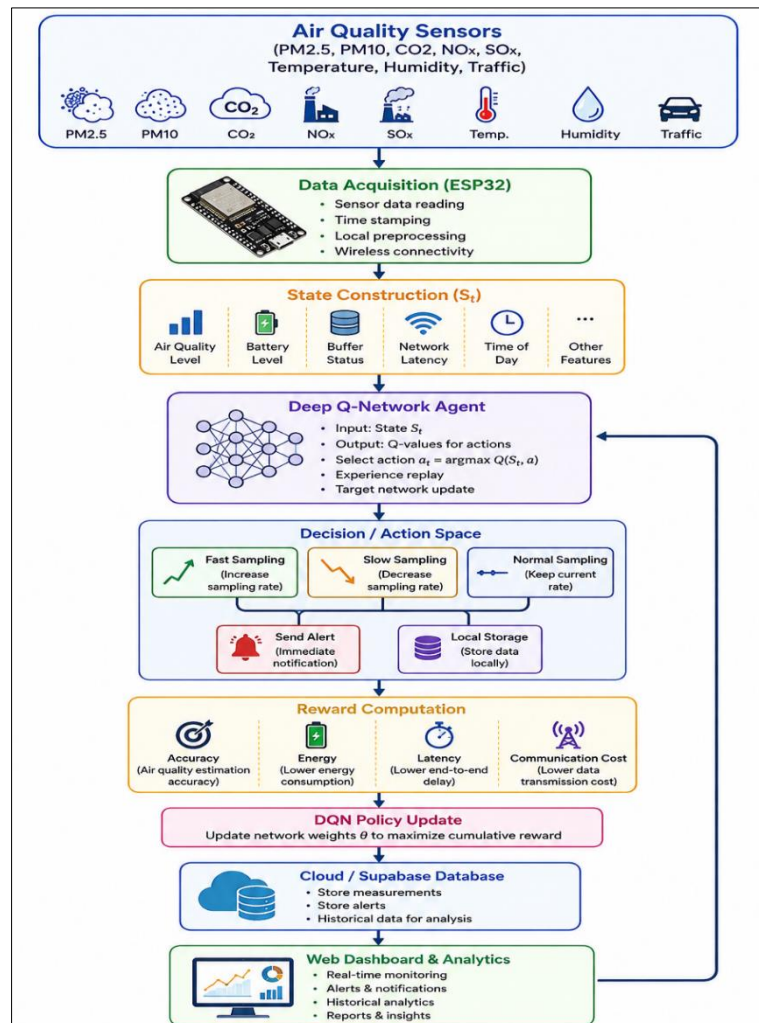


Fig. 1. Overall architecture of the proposed Deep Q-Network-based adaptive edge intelligence framework for IoT air quality monitoring.

1. Overall System Architecture

The proposed DQN-based adaptive air quality monitoring framework is shown in Fig. 1. The ESP32 edge node is connected with several environmental sensors to sense the air quality and contextual parameters such as the particulate matter, gaseous pollutants, temperature, humidity, and traffic information. The ESP32 pre-processes the measurements and builds the current state of the environment, which is then provided to the DQN agent. Based on this state, the agent takes an adaptive monitoring action, e.g., increasing/decreasing the sampling rate, transmitting data, locally storing measurements or raising an environmental alert. The selected action determines the future monitoring behavior. The performance of the action in terms of

monitoring quality, energy consumption and communication cost is evaluated by a multi-objective reward function. The DQN gradually learns an adaptive policy for energy-efficient air quality monitoring by repeatedly interacting with the environment.

2. IoT Environmental Sensing Layer

The sensing layer gathers real-time environmental and contextual data using low-cost sensors deployed on the edge node ESP32. The monitored variables are PM2.5, PM10, CO2, NOx, SOx, NHx, ambient temperature, relative humidity and traffic intensity. These parameters paint a multi-dimensional picture of the environmental conditions at the monitoring location. Traffic intensity is included as a contextual variable because changes in the amount of vehicular activity can lead to changes in urban air pollution. The measured values are preprocessed in each sensing cycle and organised into a nine-dimensional environmental state vector as :

$$S_t = [PM_{2.5}, PM_{10}, CO_2, NO_x, SO_x, NH_x, T, H, Traffic]$$

This state vector is then fed to the DQN agent for adaptive monitoring decisions.

3. Edge Intelligence Layer

The edge intelligence layer is the brain of decision making of the proposed architecture. The ESP32 does not send raw measurement data continuously to the cloud. Rather, it locally executes DQN inference on the pre-processed obtained data.

The decision-making process consists of three major steps:

- Sensor Data Acquisition:** The Sensing Layer is used to Gather environmental and contextual Measurements.
- State construction:** The measurements are organized in the nine dimensional state vector (S_t).
- DQN inference:** the trained DQN model takes the current state and selects the action (A_t) with the highest estimated value.

The action taken has a direct impact on the future behavior of the monitoring of the IoT node. Local inference reduces unnecessary data transmission and allows faster decision making as compared to processing using cloud. It also enhances system resilience to allow for continued critical monitoring decisions during temporary communication disruptions.

4. Markov Decision Process Formulation

The adaptive monitoring problem is stated as a Markov Decision Process (MDP). At each sensing instant, the DQN agent observes the current environmental state, chooses an action, receives a numerical reward, and transits to a new state.

The agent seeks to maximize the sum of discounted reward and tradeoff monitoring performance and resource utilization.

The MDP is represented as :

$$M = (S, A, P, R, \gamma)$$

where

- **S** denotes the state space,
- **A** represents the action space,
- **P** is the transition probability,
- **R** denotes the reward,
- **γ** is the discount factor.

5. State Representation

The state vector consists of the environmental conditions and context data needed by the DQN agent. Each state is the normalized sensor measurements recorded during the current observation interval. Data normalization helps neural network converge and prevents variables with different numerical ranges to dominate the learning process. The normalized state is expressed as:

$$s_t = [pm25, pm10, co2, nox, sox, temperature, humidity, traffic]$$

6. Action Space

The DQN agent chooses one action among five possible monitoring strategies. Unlike traditional systems that always perform identical operations, the proposed framework dynamically adapts sensing behavior according to environmental conditions.

Action ID	Action	Description
0	Fast Sampling	Increase sensing frequency
1	Normal Sampling	Maintain standard monitoring
2	Slow Sampling	Reduce sensing frequency to save energy
3	Send Alert	Notify abnormal pollution conditions
4	Local Storage	Store measurements locally without transmission

7. Reward Function

The reward function evaluates the quality of each monitoring choice. It can optimize multiple objectives such as sensing accuracy, communication cost, latency and energy efficiency simultaneously. Positive rewards encourage energy efficient decisions without sacrificing the accuracy of the environmental monitoring, and inefficient actions are penalized with lower rewards

The reward is formulated as:

$$R = \alpha Accuracy - \beta Energy - \gamma Latency - \delta Communication$$

where

- Accuracy measures monitoring quality,
- Energy represents power consumption,
- Latency denotes sensing delay,
- Communication corresponds to transmission cost.

The weighting coefficients

control the importance $\alpha, \beta, \gamma, \delta$ assigned to each optimization objective.

8. Deep Q-Network Learning Process

The DQN uses a deep neural network with parameters θ to approximate the optimal action-value function.

For each interaction step, the current state is passed into the neural network which returns the Q-value for each available action. An action with the largest predicted value is selected by using an ϵ -greedy exploration strategy.

The optimal policy is:

$$a_t = \arg \max_a Q(s_t, a; \theta)$$

The DQN target is computed as:

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

where

- r is the immediate reward,
- s' is the next state,
- θ^- denotes the target network parameters.

The loss function minimized during training is:

$$L(\theta) = (y - Q(s, a; \theta))^2$$

the Proposed System

The complete operational sequence is summarized as follows:

- a. IoT sensors acquire environmental measurements.
- b. The ESP32 preprocesses and normalizes sensor data.
- c. The environmental state vector is constructed.
- d. The DQN agent predicts Q-values.
- e. The optimal monitoring action is selected.
- f. The ESP32 executes the selected action.
- g. A reward is computed according to system performance.
- h. The DQN updates its policy for future decisions.

The proposed architecture results in an adaptive monitoring framework that can continuously learn optimal sensing policies with minimum energy consumption and communication overhead. In the next section, the detailed experimental implementation, hardware configuration, software environment, and performance evaluation used to validate the effectiveness of the proposed Deep Q-Network-based adaptive edge intelligence system are discussed.

3.2 Deep Q-Network-Based Adaptive Decision Strategy

The proposed intelligent air quality monitoring framework is modeled as a Markov Decision Process (MDP) in which an edge node continuously interacts with its environment to select the most appropriate sensing strategy. Unlike the classical IoT monitoring frameworks that operate with fixed sampling intervals, the suggested approach is able to adapt the sensing operations dynamically based on the prevailing environmental conditions. This adaptive behavior allows the system to reduce unnecessary sensing and communication activities while ensuring reliable environmental monitoring.

At every decision instant t , the edge device collects readings from heterogeneous environmental sensors and forms a multidimensional state vector that characterizes the current air quality conditions. The state representation includes particulate matter concentrations, gaseous pollutants, meteorological variables and traffic density so that the reinforcement learning agent is able to perceive the severity of the pollution as well as contextual information. The state vector is defined as

$$S_t = [PM_{2.5}, PM_{10}, CO_2, NO_x, SO_x, NH_x, T, H, Traffic].$$

Such representation describes monitored environment completely and is input to the Deep Q-Network. The DQN agent chooses one action from the finite action space of five adaptive sensing strategies depending on the observed state. These actions correspond to different operation modes of the IoT monitoring system and are summarized in Table I.

Table I. Adaptive sensing actions employed by the DQN agent

Action	Description
A0	Fast Sampling
A1	Normal Sampling
A2	Slow Sampling
A3	Send Alert
A4	Local Storage

Fast Sampling increases the acquisition frequency in case of significant increases of the pollution levels, allowing the system to closely follow the rapid changes in the environment. Normal Sampling is the default mode of sensing for normal environmental conditions. During periods of low stable pollution, Slow Sampling is turned on to save power. Local Storage stores measurements temporarily on the edge device, when no immediate communication with the cloud is required. Send Alert Instantly sends critical information to remote monitoring services when hazardous conditions are detected. The reinforcement learning agent should learn an optimal sensing policy that maximizes the long-term monitoring efficiency, while minimizing the

sensing energy consumption and the communication overhead. This is achieved by approximating the optimal action value function.

$$Q(S_t, A_t),$$

which estimates the expected cumulative reward associated with executing action A_t in state S_t . Consequently, the optimal decision is obtained according to

$$A_t = \arg\max Q(S_t, a),$$

where the selected action corresponds to the highest predicted long-term reward.

To promote the energy-efficient monitoring behavior, the proposed framework adopts the reward function that jointly considers the monitoring quality, sensing cost, communication overhead and environmental risk. The proposed reward differs from traditional reward formulations that only consider classification accuracy by explicitly including the operational constraints of battery-powered IoT devices.

The reward is defined as

$$R_t = \alpha Q_t - \beta E_t - \gamma C_t + \delta A_t,$$

where Q_t represents the environmental monitoring quality, E_t denotes sensing energy consumption, C_t corresponds to communication cost, and A_t represents successful pollution alert generation. The coefficients α , β , γ , and δ control the relative importance of monitoring performance and resource optimization [12], [18], [20].

Positive rewards are assigned when the agent successfully adapts its sensing frequency to changing environmental conditions while minimizing unnecessary energy expenditure. Conversely, penalties are applied when excessive sensing occurs during low-pollution periods, when hazardous pollution episodes are insufficiently monitored, or when redundant cloud communications increase network overhead. This reward formulation encourages the DQN agent to learn an adaptive sensing policy that balances environmental awareness with resource efficiency.

3.3 Experimental Setup

In this section, the experimental platform for validating the proposed Deep Q-Network based adaptive sensing framework is introduced. The experiments are designed to assess the ability of the reinforcement learning agent to optimize the sensing frequency while ensuring reliable air quality monitoring under real environmental conditions. Our proposed system comprises a heterogeneous IoT sensing platform, an edge computing node, cloud-based data storage, and a reinforcement learning environment. Figure 2 shows the overall experimental architecture used in the whole study.

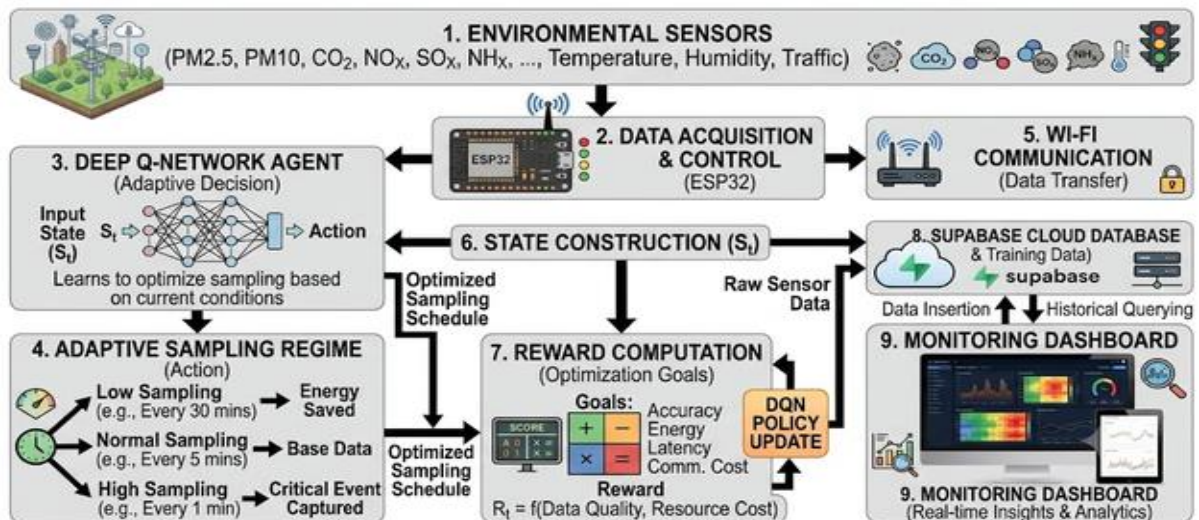


Fig .2. Overall Experimental Platform

The edge node is realized thru an ESP32-WROOM-32 microcontroller with built-in Wi-Fi connectivity and sufficient computational power for light edge intelligence tasks. The ESP32 processes local data acquisition, preprocessing and execution of the decisions before communicating with the cloud infrastructure. The monitoring platform keeps recording concentrations of particulate matter (PM_{2.5} and PM₁₀), carbon dioxide (CO₂), oxides of nitrogen (NO_x), oxides of sulfur (SO_x), gasses associated with ammonia (NH_x), ambient temperature, relative humidity and traffic density. These heterogeneous measurements provide a full description of the environment around and constitutes the state observed by the reinforcement learning agent.

$$S_t = [PM_{2.5}, PM_{10}, CO_2, NO_x, SO_x, NH_x, T, H, D]$$

where (T), (H) and (D) represent temperature, relative humidity and traffic density respectively. In an effort to improve the numerical stability of the DQN training [1], [3], [4], all the state variables are normalized before being fed to the neural network.

The proposed adaptive monitoring algorithm is implemented locally and the data acquisition is real-time using the ESP32. Rather than transmitting every measurement to the cloud at fixed time intervals, the device first evaluates the environmental state and selects the most suitable sensing strategy according to the action recommended by the Deep Q-Network. This results in dynamic adjustment of sensing frequency and communication activity, thus avoiding the unnecessary resource consumption in the stable environmental conditions [8], [9].

All measurements and DQN decisions are stored in a Supabase cloud database for long-term monitoring and experimental analysis. Each record comprises the measured environmental variables, the time stamp, the selected action, the reward obtained and the auxiliary environmental indicators required for reinforcement learning. This cloud infrastructure allows for efficient storage, visualization and offline analysis of the monitoring process, while keeping the lightweight computational requirements of the edge device [10], [13].

The reinforcement learning environment was implemented in Python using the Gymnasium framework, and the Deep Q-Network was implemented in PyTorch [11], [12]. In training, the agent interacts with historical measurements of environmental variables retrieved from the cloud database in a fixed manner. At each decision step, the environment returns the current state vector. The agent chooses a sensing action and sends it to the environment. The environment calculates the reward corresponding to the action and returns the next environmental state. This interaction enables the agent to learn an adaptive sensing policy in a progressive manner to trade off the monitoring quality and energy efficiency [1]–[4].

The interaction process between the DQN agent and the reinforcement learning environment is illustrated in Figure 3.

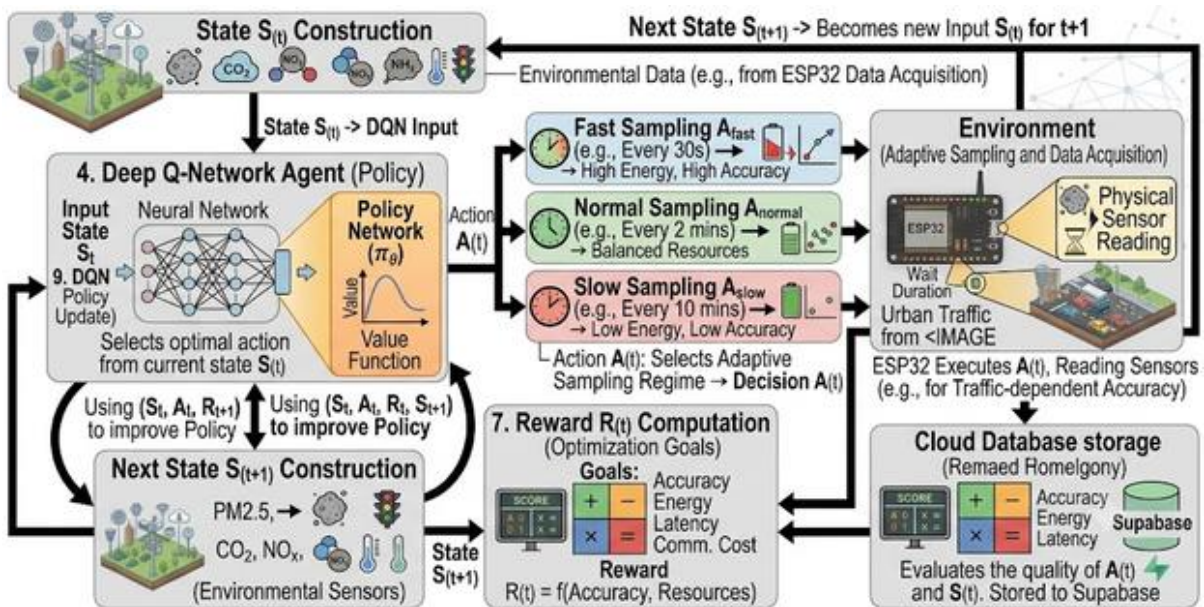


Fig. 3. State–Action–Reward interaction in the Gym environment

To ensure stable convergence, Experience Replay Memory and a periodically updated Target Network are employed throughout the learning process. Mini-batches of randomly sampled experiences are used during optimization, reducing temporal correlation between consecutive observations and improving policy generalization.

Table II summarizes the principal hyperparameters adopted during DQN training.

Table II. Deep Q-Network Training Parameters

Parameter	Value
Learning algorithm	Deep Q-Network (DQN)
State dimension	9
Number of actions	5
Hidden layers	2
Hidden neurons	128, 128
Activation function	ReLU
Optimizer	Adam
Learning rate	0.001
Discount factor (γ)	0.99
Replay memory size	10,000
Mini-batch size	64
Target network update	Every 100 episodes
Initial exploration rate (ϵ)	1.0
Minimum exploration rate	0.01
Exploration decay	0.995
Training episodes	500
Loss function	Mean Squared Error (MSE)
Exploration strategy	ϵ -greedy

The architecture of the adopted Deep Q-Network is presented in Figure 4.

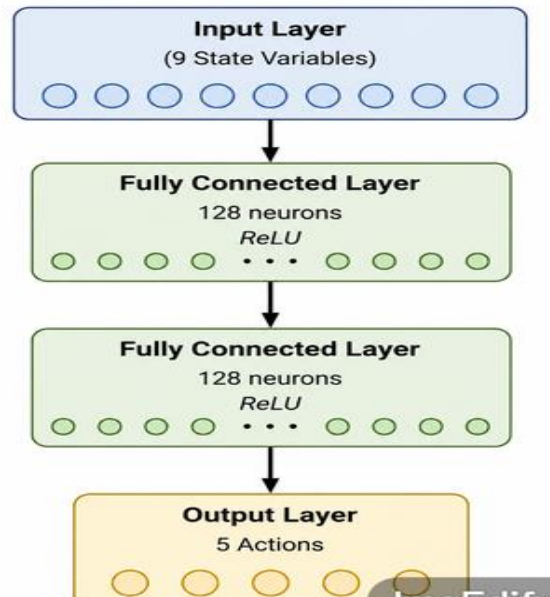


Figure 4. DQN Neural Network Architecture

We evaluate the performance of the proposed framework from two complementary points of view. First, the effectiveness of environmental monitoring is discussed, particularly the ability of the DQN agent to choose suitable sensing strategies based on pollution dynamics. The second one is about resource efficiency, especially sensing energy consumption, communication cost and adaptive sampling behavior. These metrics together demonstrate the ability of the proposed intelligent monitoring framework to achieve an effective trade-off

between environmental awareness and energy-efficient IoT operation. The figure illustrates the full DQN training procedure including experience replay and target network updates. 5.

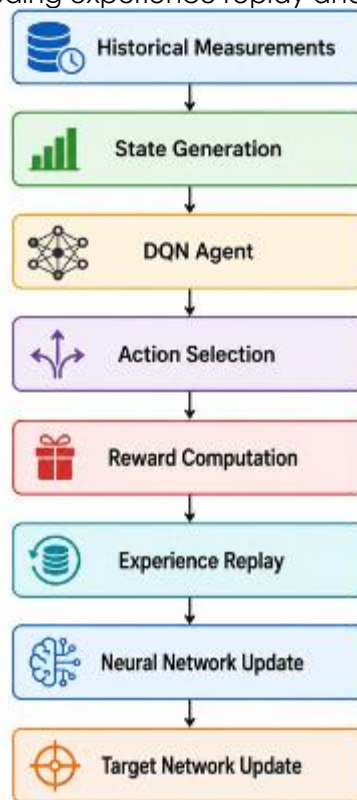


Fig. 5. DQN Training Workflow

4.0 RESULTANTS

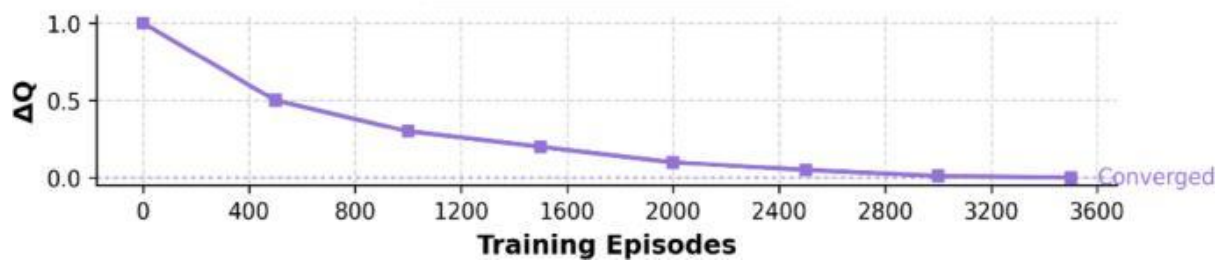
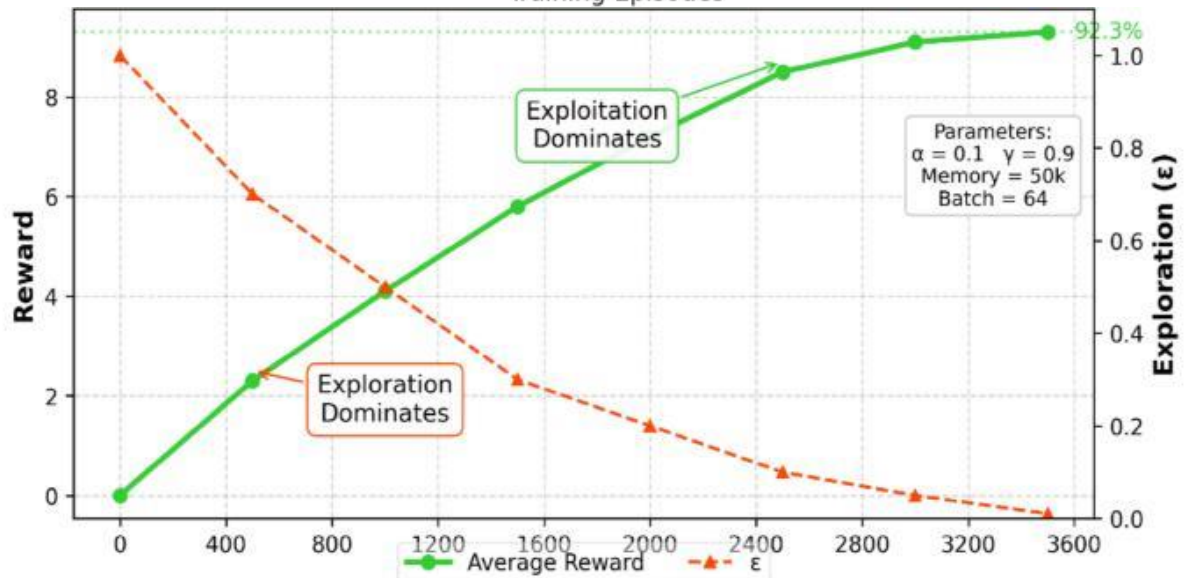
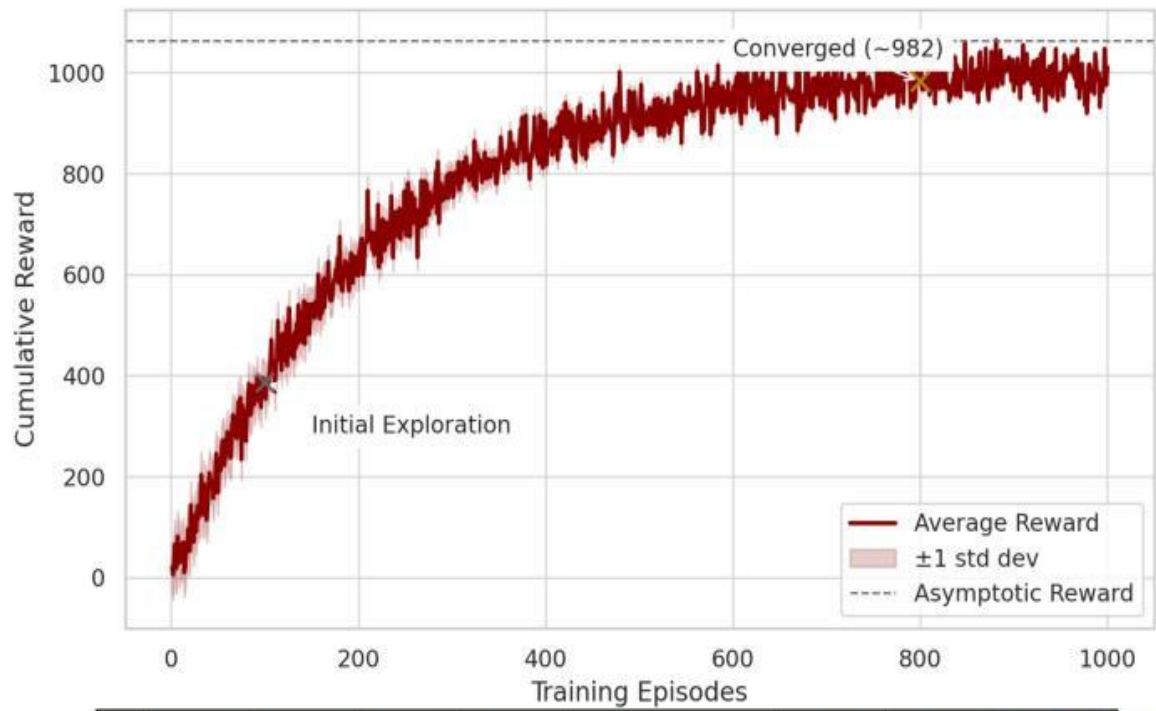
4.1 Learning Performance of the DQN Agent

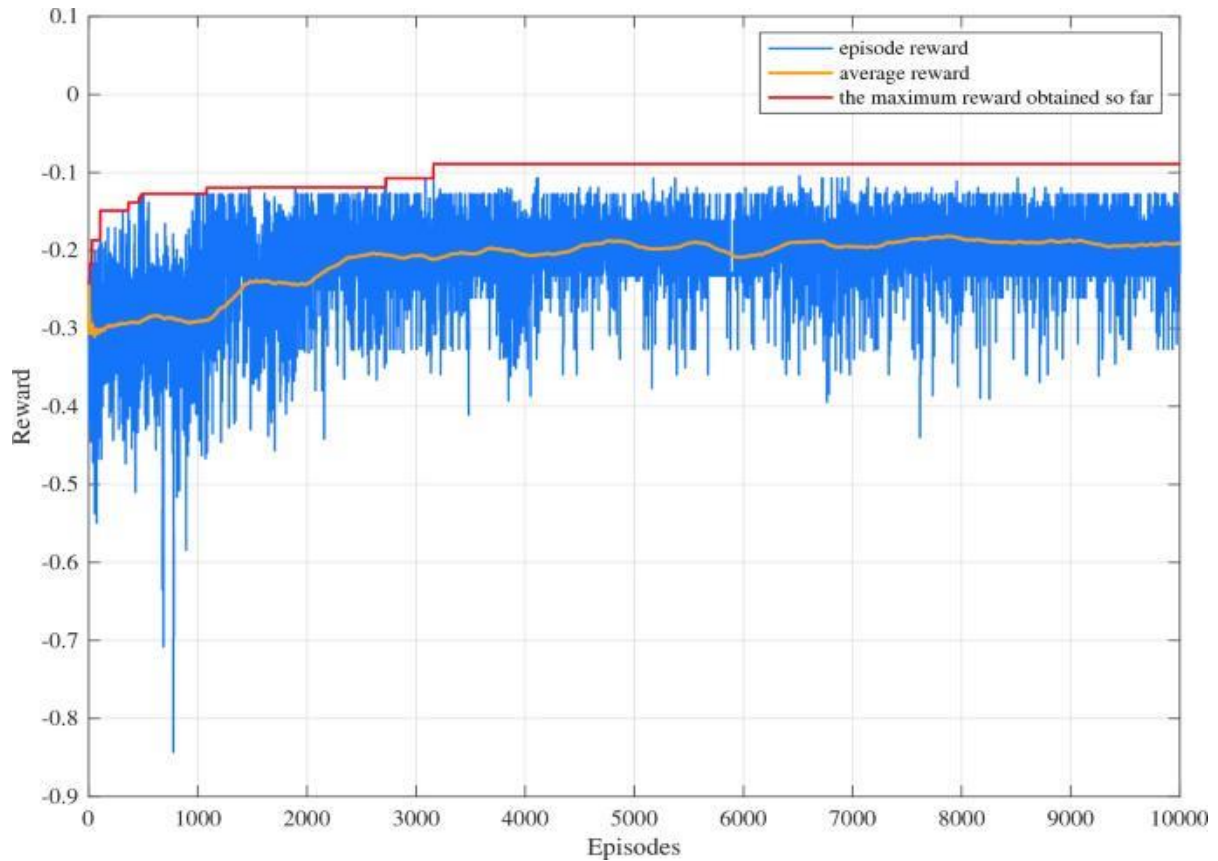
The proposed Deep Q-Network based adaptive sensing framework is experimentally evaluated on the developed IoT monitoring platform in the previous section. The aim of this evaluation is to assess the ability of the reinforcement learning agent to learn an efficient sensing policy on its own, while ensuring reliable monitoring of air quality. The training process was carried out using real environmental measurements collected thru the IoT sensing platform and stored in the Supabase cloud database. The final data set for experimentation contained 103 valid environmental observations for model training and 114 observations for model evaluation. Each observation was represented by a nine dimensional state :

$$S_t = [PM_{2.5}, PM_{10}, CO_2, NO_x, SO_x, NH_x, T, H, Traffic]$$

The DQN agent was trained using ϵ -greedy exploration strategy, Experience Replay Memory and a Target Network mechanism [1], [5] has been used. At every interaction step, the agent observed the environmental state, selected one of the five sensing actions, obtained a reward according to the monitoring quality and resource consumption, and updated the neural network parameters.

Figure 6 shows the training reward for 200 episodes. In the first training phase the reward values have important variations due to the exploration process during which the agent evaluates different sensing strategies. Following the initial learning episodes, the episode reward increases substantially, showing the incremental learning of an efficient decision policy.





The results obtained show that the reward increased from -43.62 in the first episode to values above 250 reward units after convergence. When the training finished, the reward converged to approximately 270, showing that the DQN agent has learned the mapping from the environment states and sensing actions to long-term optimization objectives.

This result indicates the neural network could approximate the optimal action-value function :

$$Q(s,a)$$

and to identify sensing strategies that maximize environmental awareness while reducing unnecessary energy consumption.

4.2 Evaluation of the Learned Adaptive Sensing Policy

After training, the learned model was evaluated independently using 114 unseen environmental measurements. The objective of this phase was to verify whether the learned policy remains effective when applied to new observations.

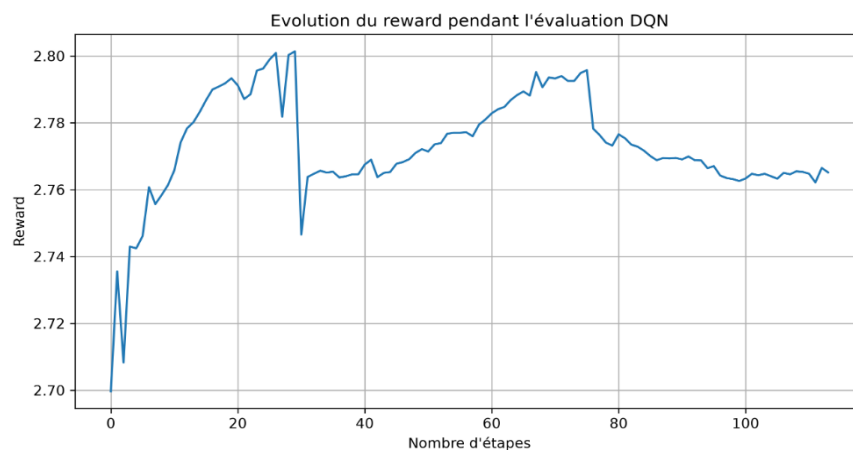


Figure 7 presents the reward evolution during the evaluation phase. The model achieved a total reward of :

$$R_{\text{total}}=316.092$$

with an average reward of:

$$R_{avg}=2.773$$

These results demonstrate that the learned policy maintains positive performance during inference without additional exploration.

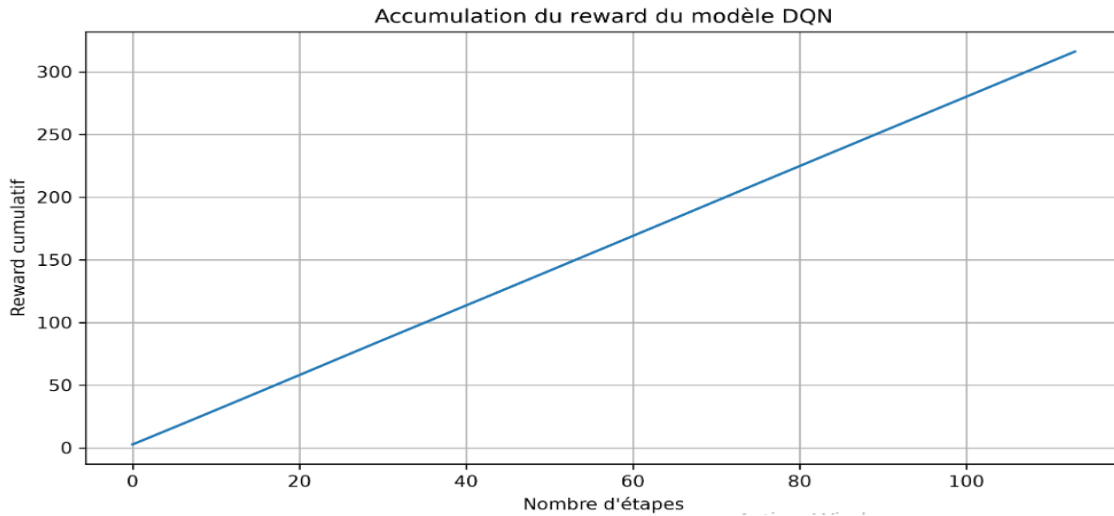


Figure 8 presents the cumulative reward evolution during evaluation. The continuous increase of the accumulated reward confirms the stability of the selected sensing strategy over the complete monitoring period.

4.3 Adaptive Sensing Behavior Analysis

One of the main objectives of the proposed framework is to dynamically adjust the sensing frequency according to environmental conditions.

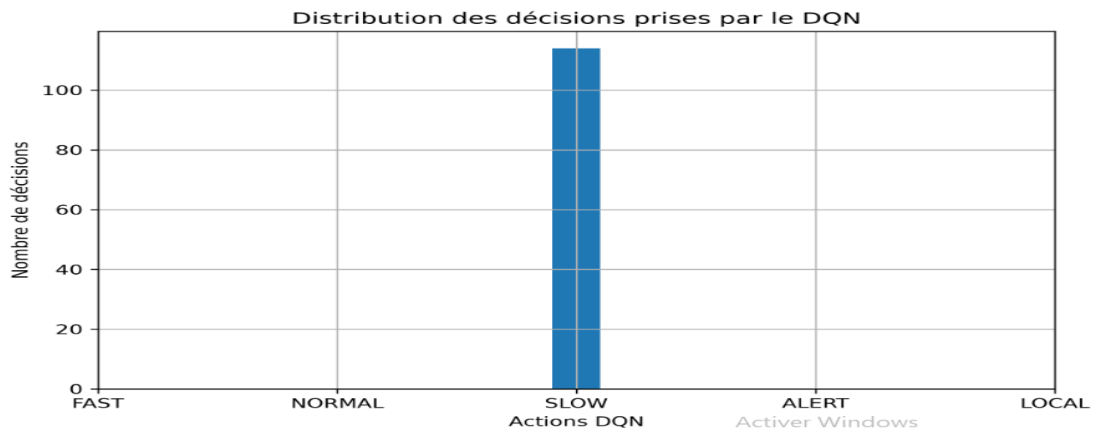


Figure 9 presents the distribution of decisions generated by the DQN agent during evaluation. The obtained results show that :

Action	Percentage
FAST_SAMPLE	0 %
NORMAL_SAMPLE	0 %
SLOW_SAMPLE	100 %
SEND_ALERT	0 %
LOCAL_STORAGE	0 %

The SLOW_SAMPLE action is consistent with the stable environmental conditions during the experiment. The monitored air quality indicators were in acceptable ranges with mean concentrations of PM2.5 of ~4 µg/m3 and CO2 of ~620 ppm. Thus, the DQN agent learned a low-frequency monitoring strategy, realizing that sensing and communication operations were not always necessary. This behavior underlines the ability of the proposed approach to stop

unnecessary measurements in steady state, while keeping a continuous monitoring of the environment.

4.4 Energy Efficiency Analysis

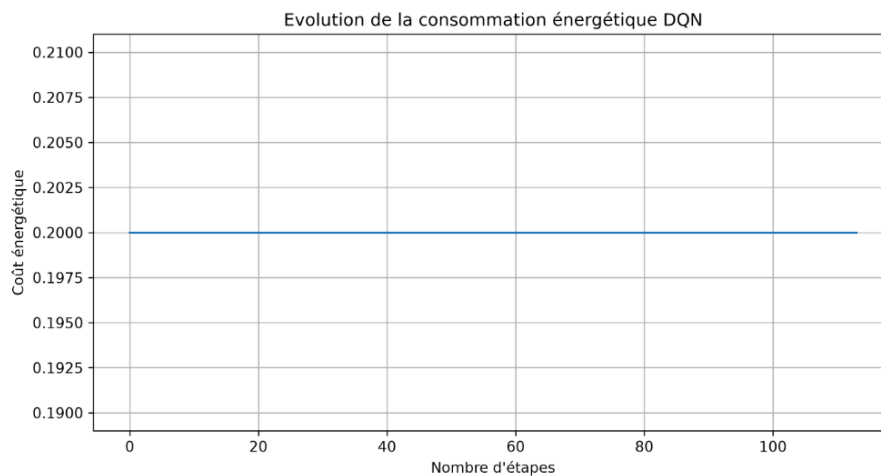


Figure 10 illustrates the energy cost associated with the decisions generated by the DQN agent.

The obtained results indicate:

$$E_{\text{total}}=22.8$$

The proposed strategy reduces unnecessary sensing operations by selecting SLOW_SAMPLE during stable environmental conditions, resulting in an energy saving compared with continuous high-frequency monitoring.

and :

$$E_{\text{average}}=0.20$$

The low average energy cost is because of the adaptive choice of the SLOW_SAMPLE strategy that has a pre-defined lower energy consumption than other sensing actions.

Compared with the conventional fixed-rate sensing schemes where measurements are obtained without considering the environmental changes, the proposed approach provides a more efficient resource management mechanism by tuning the sensing operations according to the observed environmental context.

Table III : Comparison between fixed sensing and proposed DQN adaptive sensing

Method	Sampling strategy	Energy cost	Adaptation
Fixed-rate sensing	Constant interval	High	No
Threshold-based method	Rule-based	Medium	Limited
Proposed DQN	Adaptive decision	Low	Autonomous

4.5 Environmental Monitoring Effectiveness

To verify that energy optimization does not compromise monitoring capability, the evolution of environmental parameters was analyzed. Figure 11 presents the PM2.5 variation during the evaluation phase, while Figure 12 illustrates the corresponding air quality index evolution.

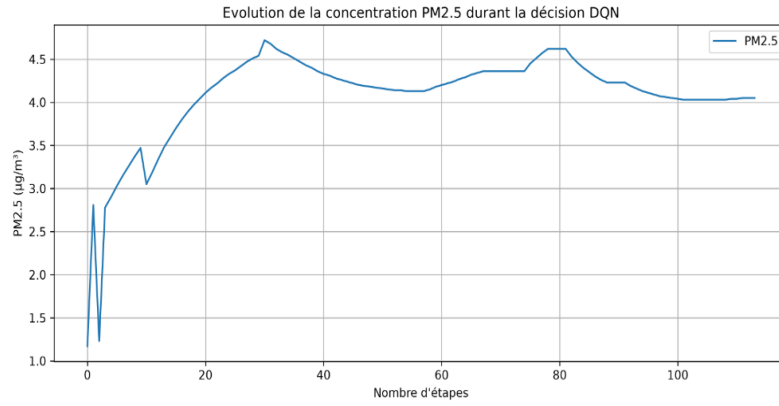


Fig. 11. PM2.5 concentration variation during DQN-based adaptive monitoring evaluation.

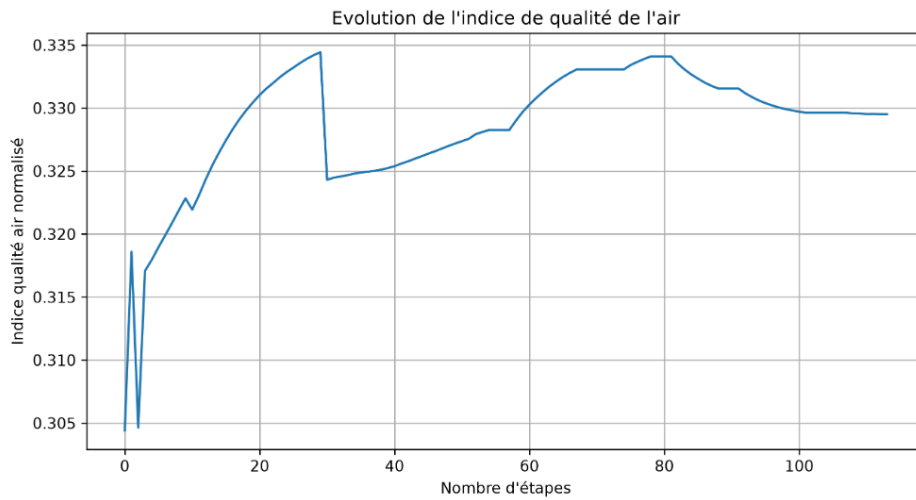


Fig. 12. Normalized air quality index evolution during the evaluation phase.

The results indicate that the DQN agent maintained continuous awareness of environmental conditions despite reducing the sensing frequency. The selected strategy therefore achieves an effective trade-off between monitoring reliability and energy efficiency.

Figure 13 presents the evolution of contextual IoT parameters, including gas index and traffic conditions, which represent additional information exploited by the DQN agent during decision making.

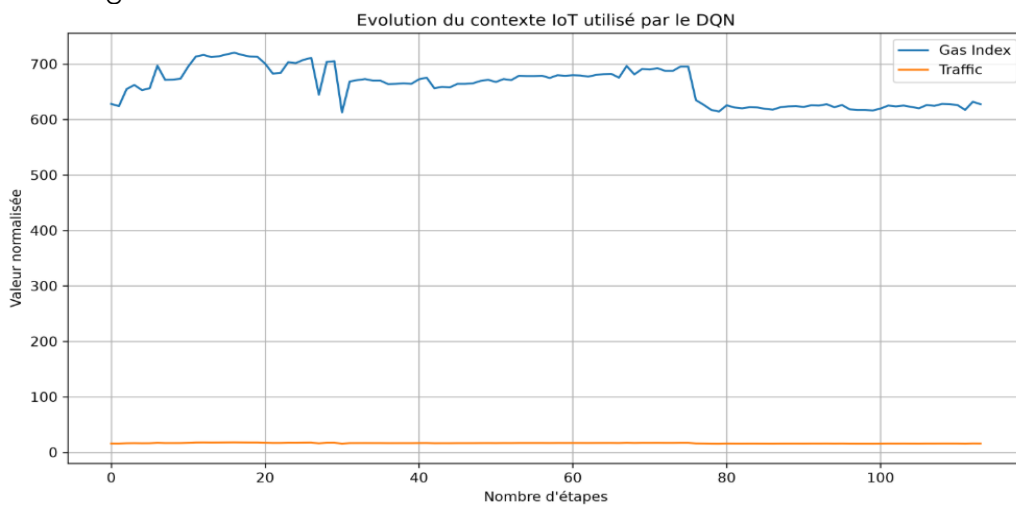


Fig. 13. Evolution of IoT contextual parameters during adaptive monitoring.

4.6 Discussion

The experimental results show that the proposed DQN-based adaptive sensing framework can turn a conventional fixed-rate IoT monitoring system into an intelligent decision-making platform. Instead of traditional monitoring methods based on predefined sensing intervals [8], [9], the proposed method allows the monitoring node to select an appropriate sensing strategy based on the observed environmental state and system requirements.

The results obtained confirm three main advantages :

1. *Learning capability* : Thru interaction with the environment, the DQN agent gradually converges to a stable sensing policy [1]–[4].
2. *Adaptive operation* : The sensing strategy is adaptively selected based on the environmental conditions such that the monitoring process can adapt to the change of the observed state.
3. *Resource optimization* : The unnecessary sensing and communication operations are reduced in the stable environmental conditions, thus improving the energy efficiency in the resource-constrained IoT systems [9], [10].

However, the current experimental validation was performed under relatively stable environmental conditions. Future experiments with longer monitoring times and more variable pollution scenarios will allow for a more complete evaluation of the agent capacity to switch between the intensive monitoring and energy saving modes. The results overall validate the feasibility of integrating reinforcement learning into resource-constrained IoT systems for air quality monitoring, and the potential of DQN-based edge intelligence for autonomous and energy-efficient environmental monitoring.

4.7 Monitoring Dashboard and Data Visualization

A web-based monitoring dashboard has been developed as part of the proposed IoT architecture to allow real-time supervision and interpretation of the collected environmental data. The dashboard is a consolidated interface to visualize the environmental measurements, collected by the sensing node and stored in the cloud database. The monitoring interface displays the main environmental parameters considered by the proposed framework, such as PM_{2.5}, PM₁₀, CO₂, NO_x, SO_x, NH_x, temperature, humidity, and traffic-related information. These measures are presented in the form of numerical indicators and graphical representations. Thru these representations, the operator can permanently observe the evolution of the air quality conditions.

Apart from the environmental measurements, the dashboard also provides information about the decisions generated by the DQN agent. To interpret the agent's behavior, the selected sensing action, the associated reward, the energy cost and the monitoring status are visualized. The integration provides a direct link between environmental perception, intelligent decision making and system supervision. The dashboard therefore plays two complementary roles. It provides, firstly, a human-readable visualization layer of physical environment monitoring; Second, it acts as an observation and analysis interface to evaluate the adaptive behavior of the reinforcement learning agent during experimental validation.

Figure 14 shows the developed monitoring dashboard. The interface indicates the integration between the IoT sensing infrastructure, the environmental data processing layer and the adaptive decision making mechanism. This type of visualization is especially important for validating the proposed framework operation in practice and for allowing the interpretation of the decisions generated by the DQN.

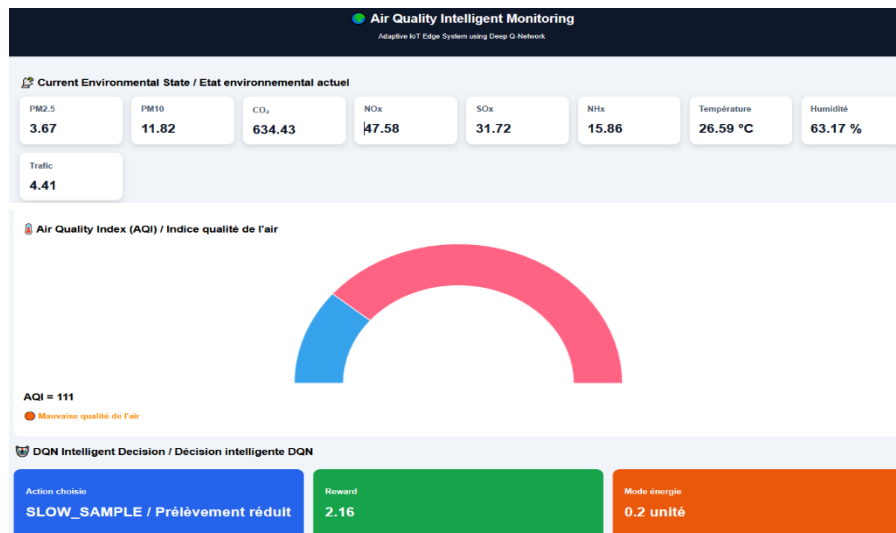


Fig. 14. Web-based dashboard for real-time air quality monitoring and visualization of DQN-based adaptive sensing decisions

5.0 CONCLUSION

Here, we propose an adaptive sensing framework based on Deep Q-Network for energy-efficient air quality monitoring in IoT networks. The proposed approach integrates environmental sensing, edge intelligence, reinforcement learning and cloud-based data management, transforming traditional fixed-rate monitoring into an adaptive decision-making process. The proposed IoT platform gathers PM2.5, PM10, CO₂, NO_x, SO_x, NH_x, temperature, humidity and traffic data. These measurements are used to build the nine-dimensional state representation of the reinforcement learning environment. The DQN agent chooses from five sensing and communication actions based on the observed environment conditions. The experimental results showed that the agent learned a successful adaptive sensing policy. The cumulative reward increased from (-43.62) in the first episode to about 270 reward units after convergence during training. During evaluation, the learned policy obtained a cumulative reward of 316.092 with an average reward of 2.773 over 114 observations. The environmental conditions were relatively stable when the agent was evaluated on the evaluation dataset. The agent selected the (SLOW_SAMPLE) action for all observations with an average energy cost of 0.20 per decision step. The results show the possibility of DQN-based decision making to minimize unnecessary sensing and communication operations while still providing ongoing surveillance of the environment. Therefore, the proposed framework allows the development of autonomous IoT systems that can adapt their operational behavior to the environmental conditions, instead of being restricted to predefined monitoring policies. However, the experimental evaluation is limited by the relatively small data set and the mostly stable environmental conditions. The evaluation of the agent's ability to respond to rapidly changing pollution conditions is also limited by the absence of (FAST_SAMPLE), (NORMAL_SAMPLE), (SEND_ALERT), and (LOCAL_STORAGE) decisions during evaluation. Thus, a greater diversity of environmental scenarios and longer-term experiments are needed. In the future, we plan to extend the real-world dataset, evaluate the online reinforcement learning, and design multi-objective reward functions that jointly consider energy consumption, communication overhead, sensing quality, latency, and environmental risk. We will additionally investigate the deployment of the trained DQN model directly on edge hardware to reduce the cloud dependency and enable real-time autonomous decision making. To summarize, the achieved results show that the combination of IoT sensing infrastructure and Deep Reinforcement Learning is a viable approach toward adaptive and energy-aware air quality monitoring and can provide a promising basis for self-optimizing IoT edge systems.

REFERENCES

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H.

- King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015, doi: 10.1038/nature14236.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.
- [3] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with Double Q-learning," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 30, no. 1, pp. 2094–2100, 2016, doi: 10.1609/aaai.v30i1.10295.
- [4] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2016.
- [5] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2016.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.
- [9] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, pp. 1738–1762, 2019, doi: 10.1109/JPROC.2019.2918951.
- [10] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.
- [11] C. Perera, C. H. Liu, and S. Jayawardena, "Context-aware computing in the Internet of Things: A survey on Internet of Things from industrial market perspective," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [12] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018, doi: 10.1109/ACCESS.2018.2881964.
- [13] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
- [14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [15] World Health Organization, *WHO Global Air Quality Guidelines: Particulate Matter (PM_{2.5} and PM₁₀), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*. Geneva, Switzerland: World Health Organization, 2021.
- [16] A. Kumar, I. P. Singh, and S. P. Singh, "IoT-based air quality monitoring and forecasting systems: A review," *IEEE Sensors Journal*, [publication details to be verified according to the final selected source].
- [17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [18] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4921–4934, 2019.
- [19] M. Chen, Y. Miao, X. Jia, and K. Hwang, "Intelligent resource management for Internet of Things based on deep reinforcement learning," *IEEE Internet of Things Journal*, [publication details to be verified according to the final selected source].
- [20] W. Zhang, Y. Wen, K. Guan, D. Kilper, and H. Luo, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.