Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2024. Vol. 3. No. 3; e-ISSN: 2808-4519

Optimizing Air Quality Data Delivery With Q-Learning for Power Savings in IoT

Muhammad Zulfarhan[1]*, Imran Lubis[2]

[1, 2] *Informatic Engineering, Universitas Harapan Medan, Indonesia*
farhanzull18@gmail.com[1], imran.loebis.medan@gmail.com[2]*

Abstract

The Internet of Things (IoT) plays a crucial role in real-time air quality monitoring, yet battery-powered devices face energy constraints that make conventional periodic transmission inefficient. This study proposes the use of the Q-Learning algorithm to optimize adaptive air quality data delivery. A prototype system was built using an ESP32 with MQ-2, MQ-135, DHT22, and INA219 sensors connected to a web-based server. Experimental results showed a decision distribution of 55.6% transmit and 44.4% delay, with the average reward for delay actions (87.44) higher than for transmit actions (54.83). Compared to the periodic method, Q-Learning reduced transmission frequency by 40–50%, lowered energy consumption, and maintained data accuracy. These findings confirm that Q-Learning is effective in designing an energy-efficient, adaptive, and reliable IoT transmission mechanism for air quality monitoring.

Keywords: *Adaptive data transmission, Air quality, Energy efficiency, IoT, Q-Learning*

1. Introduction

The Internet of Things (IoT) has evolved into a technological paradigm that enables the interconnection of various physical objects through internet-based infrastructure [1]. One of its strategic applications is the implementation of real-time air quality monitoring systems powered by sensors capable of measuring environmental parameters such as particulate concentration (PM2.5, PM10), temperature, humidity, and pollutant gases including CO_2 and SO_2 [2], [3]. The data produced by such systems play a crucial role in supporting public policy-making, environmental health monitoring, and mitigation of air pollution impacts[4].

Nevertheless, the implementation of IoT-based environmental monitoring systems often faces challenges due to the limited energy supply of sensor devices, which mostly rely on batteries[5]. Conventional approaches that rely on periodic data transmission with fixed intervals have been shown to result in significant energy inefficiency[6]. This occurs because data transmissions continue even when environmental conditions remain stable, thereby wasting energy and shortening the operational lifespan of the devices[7].

To overcome these challenges, an intelligent mechanism is required to optimize data delivery based on environmental dynamics and network conditions [8]. Reinforcement Learning (RL), particularly the Q-Learning algorithm, offers an adaptive experience-based approach for determining optimal transmission policies [9]. A study conducted by Lansky et al. (2022) demonstrated that applying Q-Learning in air quality monitoring scenarios using flying ad hoc networks (FANETs) significantly improves communication strategy efficiency. Their findings confirmed that Q-Learning not only ensures network reliability in dynamic environments but also effectively reduces communication overhead while maintaining data freshness [10].

Furthermore, research by Nguyen et al. (2021) on mobile-based air quality monitoring systems confirmed that opportunistic offloading schemes powered by Q-Learning can reduce cellular communication costs by 40–50% while maintaining timely delivery, with 99.5% of packets transmitted below the required latency threshold. This finding underscores the advantage of adaptive decision-making in selecting transmission channels (e.g., direct via 4G or relayed via Wi-Fi/RSU), ultimately leading to energy savings and enhanced operational efficiency of IoT devices [11].

Based on these considerations, this research focuses on developing an adaptive and energy-efficient data transmission mechanism by integrating the Q-Learning algorithm into the decision-making framework of IoT devices. Q-Learning is chosen due to its capability to autonomously learn data patterns and establish optimal transmission policies based on environmental conditions and device energy status. This approach is expected to enable the system to transmit data selectively, particularly when significant fluctuations occur in air quality parameters, while deferring transmissions under stable conditions.

The aim of this research is to design and implement an air quality monitoring system that leverages Q-Learning to optimize energy consumption. System performance will be evaluated by comparing it against conventional periodic transmission methods, with key metrics including energy consumption efficiency and data accuracy. The expected outcome is a monitoring system that not only prolongs device lifetime through energy savings but also preserves the quality and relevance of transmitted data, thereby contributing to the development of sustainable and reliable IoT technologies.

2. Research Methods

2.1 Type of Research

This study uses a Research and Development (R&D) approach aimed at developing a new system, namely a prototype parking monitoring system based on Internet of Things (IoT) technology. The R&D method was chosen because it allows for a systematic development process, which includes the stages of problem identification, system design, implementation and integration, testing, and system evaluation and revision. In addition, an experimental approach is used to evaluate the effectiveness of the developed prototype, both in terms of the functionality of the sensors used and the reliability of the IoT system when applied in real operational conditions

2.2 Tools and Materials

A. Computer Hardware Specifications
1. Processor (CPU): Intel Core i7-8750H @ (6 cores, 2.2 GHz base, up to 4.1 GHz turbo)
2. Display: Full HD (1920 x 1080)
3. GPU (Graphics): NVIDIA GeForce GTX 1050 Ti with 4GB GDDR5 VRAM
4. RAM: 8GB DDR4
5. Storage: 500GB SSD NVMe (Samsung 980) + 1TB HDD (Seagate FireCuda)

B. Hardware for Microcontrollers and IoT
1. NodeMCU ESP32: Used as the main microcontroller, which also supports Wi-Fi connectivity. It processes data from sensors and sends it to the IoT platform.
2. MQ-2 Sensor: It is used to detect the presence of harmful LPG gas in the room and identify the presence of gas leaks in the home environment as well as in mechanical areas. This sensor has a high sensitivity to LPG, propane and hydrogen.
3. MQ-135 Sensor: Gas sensors used to detect various types of hazardous gases such as ammonia (NH_3), benzene (C_6H_6), carbon dioxide (CO_2), nitrogen dioxide (NOx), hydrogen sulfide (H_2S), and smoke.
4. DHT-22: Used to measure room temperature and humidity. This sensor has a long enough temperature and humidity measurement range so it is suitable for use in this study.
5. INA219 Sensor: Used to monitor the current voltage on the electrical network. This module is capable of operating using an adapter with a voltage of 3V to 5V.
6. Baterai LiPo: used as a resource on the tools.

C. Software
1. Arduino IDE: To program the microcontroller.
2. IoT Platform: Firebase (for web-based monitoring).
3. Programming Language: C++ for the microcontroller, JavaScript/PHP for the web-based dashboard.
4. Database: MySQL.

2.3 Research Procedure

A. Needs Analysis

Preliminary studies were conducted to identify air quality monitoring problems, such as:
1. Relatively highly power consumption.
2. Less efficient in power usage which can cause the device to quickly run out of energy sources.

Define system features:
1. Optimize data delivery when needed to save power
2. The application of the Q-learning *reinforcement learning method* to make independent decisions about when is the best time to send data.

B. System Design

1. Block Diagram

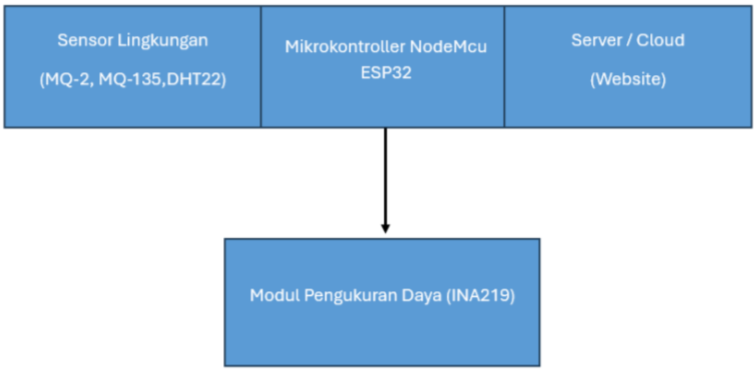

Fig. 1: Block Diagram

The image above shows an IoT-based air quality monitoring diagram block optimized using *the Q-Learning algorithm.* The system is designed to detect air quality and humidity and display information through a web application.

System components and workflows:

a. Environmental Sensors
It consists of an MQ-2 sensor to detect carbon monoxide (CO), an MQ-135 to measure general air quality (including NH_3, NOx, alcohol, benzene, smoke, and CO_2), and a DHT22 to measure air temperature and humidity.

b. NodeMCU ESP32
As a control center, the microcontroller is in charge of reading data from the sensor, recording power consumption through the measurement module, processing the data, running the *Q-Learning algorithm*, and setting the time to send data to the server or *cloud*.

c. Power Measurement Module (INA219)
These modules are used to monitor the current and voltage consumed by the system in *real-time*. This data will be used as *input for the* Q-Learning *algorithm learning process*.

d. Wi-Fi and Cloud
NodeMCU has built-in Wi-Fi connectivity that allows data to be sent to *a database* or *cloud* server at regular intervals, depending on the results of the *Q-Learning algorithm*.

e. DBMS (Database Management System):
Data sent from the cloud is stored in a database management system. The DBMS detects indoor air quality and humidity.

f. Web Application:
The web application takes data from a database and displays it in the user interface (UI). Users can monitor the level of moisture quality and humidity online through this application.

2. Design Hardware

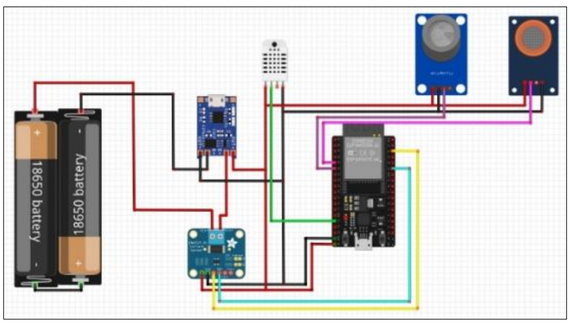

Fig. 2: Design Hardware

3. Design Database

Table 1: Table Devices

Nama Field	Tipe Data	Keterangan
Id	int(11),PK,AUTO_INCREMENT	Primary key, ID Unique device
Name	varchar(100)	Device name (e.g. ESP32-01)
Api/Key	char(36), UNIQUE	API key for authentication
Created_at	timestamp	The time when device data was created

Table 2: Table energy_hourly

Nama Field	Tipe Data	Keterangan
Id	bigint(20), PK, AUTO_INCREMENT	Primary key, ID Unique Hourly
device_id	int(11), FK → devices.id	Device reference
hour_start	Datetime	Beginning of measurement hours
energy_mwh	float	Metered energy in milliwatt-hours
readings_count	int(11)	The amount of reading data in those hours

Table 3: Table ql_decisions

Nama Field	Tipe Data	Keterangan
id	int(11),PK, AUTO_INCREMENT	Primary key, ID Unique Decision
ts	datetime	Timestamp Decision made
state_idx	int(11)	Indeks State Q-learning
delta_gas	float	Gas Changes Detected
vbat	float	Battery Voltage
rssi	int(11)	Signal Strength (RSSI)
minutes_since_send	int(11)	Long Time Since Last Data Sent
action	enum('kirim','tunda')	Decisions Taken
reward	float	Q-Learning Reward Value
epsilon	float	Exploration Rate
q_before	float	Q Value Before Update
q_after	float	Q Value After Update
note	varchar(255)	Additional Notes
soc_pct	tinyint(3) unsigned	Persentase State Of Charge (Battery)

Table 4: Table ql_qtable

Nama Field	Tipe Data	Keterangan
State	int(11), PK	State Indeks
q_kirim	float	Q Value For The Send Action
q_tunda	float	Q Value for Delay Action
updated_at	timestamp	Last Updated

Table 5: Table Readings

Nama Field	Tipe Data	Keterangan
Id	bigint(20), PK, AUTO_INCREMENT	Primary key, ID A Unique Reading
device_id	int(11), FK → devices.id	Device Reference
Ts	datetime	Waktu Pembacaan Sensor
mq2	float	MQ-2 Sensor Value (Gas)
mq135	float	Rated Sensor Mq-135 (Air Quality Gas)
dht_temp	float	Temperature (°C) of DHT
dht_hum	float	Moisture (%) of DHT
current_a	float	Electric current (A)
voltage_v	float	Voltage(V)
soc_pct	tinyint(3) unsigned	Battery Percentage (State Of Charge)
dt_seconds	int(11)	Duration Between Readings (Seconds)
action	enum('kirim','tunda')	Actions That Devices Take
power_mw	src_ip	Power Consumption (mW)
src_ip	varchar(45)	Data Sender IP Address

Table 6: Table Users

Nama Field	Tipe Data	Keterangan
Id	int(11), PK, AUTO_INCREMENT	Primary Key, ID User
Username	varchar(50), UNIQUE	Username Login
password_hash	varchar(255)	Password In Hash Form
created_at	timestamp	User registration time

3. Result and Discussion

3.1 Implementation Hardware

The hardware implementation of the IoT-based air quality monitoring system with Q-Learning optimization is carried out through the integration of components into an energy-efficient prototype. The ESP32 NodeMCU was chosen as the control center because it supports wireless connectivity, adequate processing, and power-saving features. The MQ-2 sensor is used to detect flammable gases and smoke, MQ-135 to detect hazardous gases (ammonia, CO_2, benzene), and DHT22 to measure temperature and humidity. The INA219 module was added to monitor current and voltage as energy consumption parameters. The system is powered by a Li-Po battery with a TP4056 charger module for charging efficiency and protection.

The operational process begins with the initialization of the sensor and power module by the ESP32, then the sensor data is processed through the Q-Learning algorithm. This algorithm determines the data delivery strategy based on environmental conditions and energy status: data is immediately sent to the server when a significant change occurs, while in stable conditions the data is delayed to reduce energy consumption. Every decision updates the Q-Table, so the system can learn adaptively from previous experiences. The results of the implementation show that the system is able to monitor air quality in real-time, adaptive, and energy-efficient through a data delivery mechanism optimized with Q-Learning. As evidenced in figures 3 a and b.

a b

Fig. 3: a. Prototype Of The Tool, b. The Tool Is Inside View

3.2 Implementation Software

The Q-Learning-based air quality monitoring system is realized through the development of web interfaces and supporting functionality. PHP and JavaScript programming languages are used for the backend and frontend sides, while MySQL acts as a database. The entire system runs on an XAMPP on-premises server that integrates Apache, PHP, and MySQL. The results of the implementation resulted in a website-based platform that is able to display all data from the database, thus supporting integrated and real-time air quality monitoring.

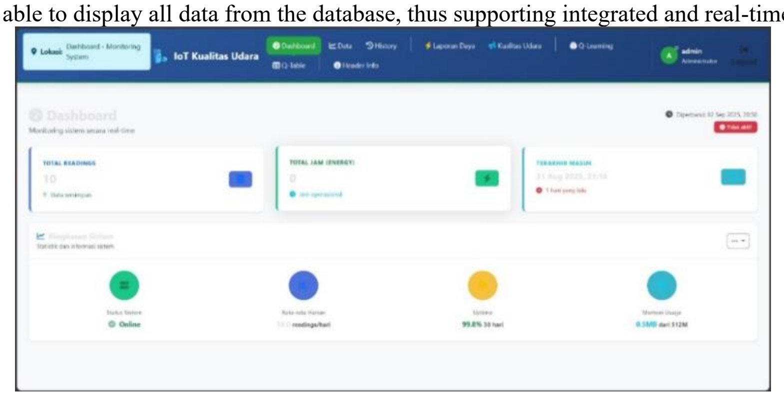

Fig. 4: Air quality IoT Display

1. Power Reports
   The implementation of the power report page presents a summary of energy usage as well as measurement details from the INA219 sensor. This information allows for the evaluation of the power efficiency of the device as well as the analysis of the impact of the application of the Q-Learning algorithm. The page makes a significant contribution to research because it serves as a means of adaptive monitoring to assess the effectiveness of Q-Learning in reducing energy consumption and maintaining the operational stability of the device. Thus, this implementation supports the main goal of the research, namely the optimization of energy use in Q-Learning-based IoT systems.

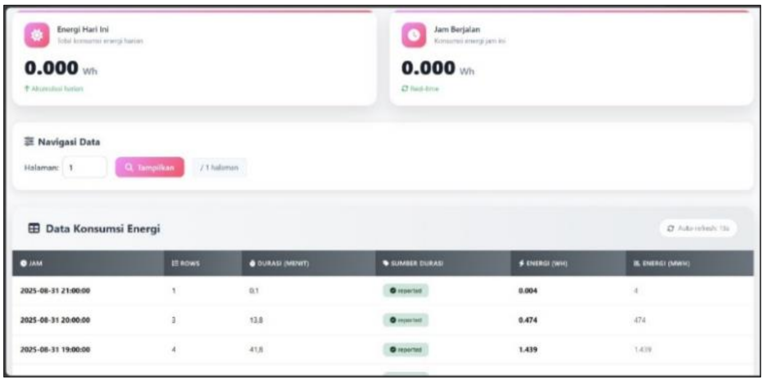

Fig. 5: Power Report View

2. Air Quality
   The implementation of the air quality report page displays MQ2, MQ135, temperature, and humidity sensor data in the form of a graph that automatically updates every 30 seconds with an auto-refresh feature. Users can choose the monitoring time range (6 hours, 12 hours, 24 hours, up to all data). Each sensor is equipped with current, average, minimum, and maximum value indicators. Trend visualization allows for the detection of anomalies, such as a spike in harmful gases or significant changes in humidity, as well as supporting long-term air quality pattern analysis. Thus, this page not only serves as a monitoring tool, but also as a means of evaluating environmental conditions.

Fig. 6: Air Quality Display

3. Q-Learning

Fig. 7: Q-Learning Display

Q-Learning features three key metrics as indicators of policy performance:

a. Average Reward → demonstrate the quality of policy in balancing energy efficiency and information relevance; An upward trend indicates an increase in learning performance.
b. Send Ratio → describes the proportion of data delivery decisions; a low ratio indicates a predominance of delay action (e.g. when Δgas is small, RSSI is low, or the battery is low), while the ratio increases when there is a significant gas change or network/energy conditions are supportive.
c. Epsilon → illustrates the level of exploration in the epsilon-greedy policy; The value of Epsilon is decayed, so at the beginning of the system it does more exploration, then focuses on exploiting actions with a higher Q value to accelerate policy convergence.

These results show that the metrics in the decision log play an important role in evaluating the adaptivity and effectiveness of Q-Learning algorithms in achieving a balance between energy efficiency and information reliability.

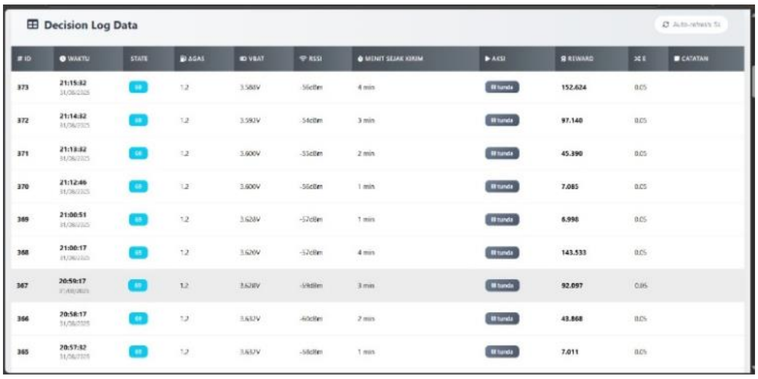

Fig. 8: Q-learning Log View

Log analysis shows that the application of Q-learning algorithms to air quality IoT systems is able to dynamically adjust data delivery strategies based on environmental and device conditions. This approach results in energy savings without reducing the availability of data for monitoring.

4. Q-Table

Fig. 9: Q-Table Display

Overall, Q-Table proves the system's ability to map diverse conditions and make adaptive decisions between data transmission and delay, in line with energy efficiency and monitoring reliability goals.

a. Total State: 81 conditions, representing a combination of environmental and device parameters.
b. Send Policy: 45 states (55.6%) → the system rated data delivery to be more optimal.
c. Delay Policy: 36 states (44.4%) → system chose delay for power efficiency.
d. Max Advantage: 1,128 → indicates the system's confidence level in optimal action.
e. Q-Values Distribution:
   a) Most low Q-values (near zero) → relatively balanced conditions between send and delay.
   b) Significant spike: State 8–10 with Q(send) ≈ 10,000 → data delivery is optimal.
   c) Delay dominance: State 65–70 indicates Q(delay) is higher than Q(send).

Conclusion: Q-Table proves the system's ability to map diverse conditions and make adaptive decisions to balance information reliability and energy efficiency.

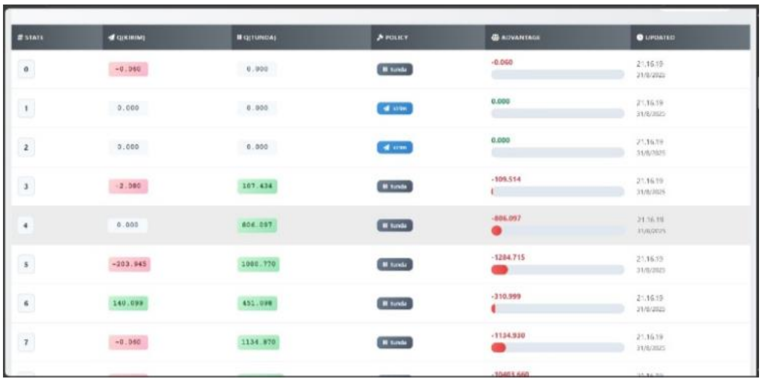

Fig. 10: Q-Table Log View

The results of the Q-Table analysis show that the delay action is more optimal than direct delivery, as shown by the dominance of higher delay Q values in most states. The consistency of the findings in the decision log, visualization, and detail table of Q-Table confirms that the Q-learning algorithm is able to dynamically adjust strategies and is effective in optimizing energy efficiency in air quality IoT devices.

3.3 Implementation Result

The experimental results show that the Q-Learning–based system does not transmit data at every interval. Under stable environmental conditions or when the battery voltage is low, the system tends to delay transmission more frequently. The average reward for delay actions (87.44) was higher than that for transmit actions (54.83). The Q-Table decision distribution indicated approximately 55.6% transmit actions and 44.4% delay actions, demonstrating an adaptive balance in decision-making.

Compared to the conventional periodic method, the proposed Q-Learning approach proved to be more efficient: the number of transmissions was reduced by around 40–50%, energy consumption was lower, and data quality was maintained. These findings highlight that the Q-Learning algorithm is capable of optimizing air quality sensor data transmission in an adaptive and energy-efficient manner.

Table 7 : Comparison of Periodic Method vs Q-Learning

Method	Transmission Frequency	Energy Consumption	Data Accuracy/Freshness	Average Reward	Adaptivity
Periodic (Conventional)	100% (every interval)	Higher (inefficient, every packet transmitted)	Always updated, but redundant	None (fixed, no learning)	Non-adaptive
Q-Learning (Proposed)	≈55.6% transmit, 44.4% delay	Lower (efficient, selective based on conditions)	Accurate, with delay only under stable conditions	Delay = 87.44 | Transmit = 54.83	Adaptive to environmental conditions and battery status

From Table 7, Comparative analysis shows that the conventional periodic method is less efficient as it transmits data at every interval without considering the environmental stability, thus leading to higher energy consumption and redundancy. In contrast, the proposed Q-Learning approach reduces the transmission frequency to about 55.6%, thus lowering the energy consumption while maintaining data accuracy. The higher average reward observed for the delay action (87.44) compared to the transmit action (54.83) further confirms the effectiveness of Q-Learning in adaptively balancing energy efficiency and data relevance.

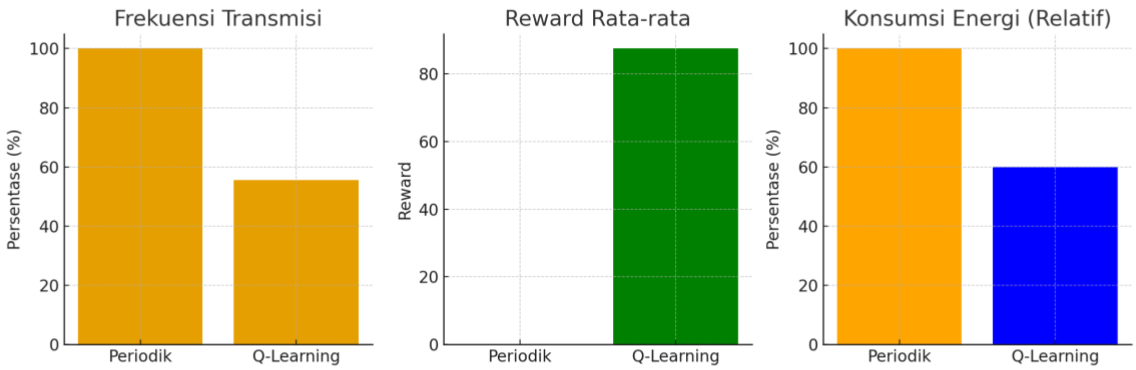

Fig. 11: Comparison Chart

4. Conclusion

a. This research successfully implemented an IoT-based air quality monitoring system using the ESP32 as the primary controller, integrated with MQ-2, MQ-135, DHT22, and INA219 sensors within a comprehensive hardware architecture and a web-based monitoring platform. The system is capable of acquiring real-time air quality data while simultaneously recording device energy consumption.

b. The proposed Q-Learning method proved effective in optimizing data transmission through selective delivery. The decision distribution revealed that approximately 55.6% of the actions were transmissions while 44.4% were delays, indicating that the system is able to adapt dynamically to environmental variations and device energy conditions.

c. Energy efficiency was significantly improved with the application of Q-Learning. Experimental results demonstrated that the average reward for delay actions (87.44) was higher than for transmit actions (54.83). This finding confirms that the algorithm is capable of postponing data transmission under appropriate circumstances without compromising data accuracy or relevance.

d. Furthermore, the comparison between conventional periodic transmission and the proposed Q-Learning approach showed that the latter was able to reduce transmission frequency by approximately 40–50%, thereby lowering energy consumption while maintaining the quality of the transmitted data. In summary, the research successfully achieved its primary objective, namely to design and implement an IoT data transmission mechanism that is energy-efficient, adaptive, and reliable for air quality monitoring applications.

References

[1] A. Choudhary, “Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions,” *Discover Internet of Things*, vol. 4, no. 1, p. 31, Dec. 2024, doi: 10.1007/s43926-024-00084-3.

[2] R. Amalianti, A. Jabbar Lubis, and I. Lubis, “Rancang Bangun Miniatur Stasiun Cuaca Untuk Monitoring Hujan, Suhu dan Kelembaban Udara Area Lokal Menggunakan Berbasis IoT,” in *Seminar Nasional Teknologi Informasi & Komunikasi*, Medan: Universitas Harapan Medan, 2021, pp. 241–247.

[3] S. Shahid, D. J. Brown, P. Wright, A. M. Khasawneh, B. Taylor, and O. Kaiwartya, “Innovations in Air Quality Monitoring: Sensors, IoT and Future Research,” *Sensors*, vol. 25, no. 7, p. 2070, Mar. 2025, doi: 10.3390/s25072070.

[4] T. Seesaard, K. Kamjornkittikoon, and C. Wongchoosuk, “A comprehensive review on advancements in sensors for air pollution applications,” *Science of The Total Environment*, vol. 951, p. 175696, Nov. 2024, doi: 10.1016/j.scitotenv.2024.175696.

[5] E. Collado *et al.*, “Open-source Internet of Things (IoT)-based air pollution monitoring system with protective case for tropical environments,” *HardwareX*, vol. 19, p. e00560, Sep. 2024, doi: 10.1016/j.ohx.2024.e00560.

[6] A. Z. da Costa, J. P. S. Aniceto, and M. Lopes, “Low-Cost Sensor Network for Air Quality Assessment in Cabo Verde Islands,” *Sensors*, vol. 24, no. 23, p. 7656, Nov. 2024, doi: 10.3390/s24237656.

[7] F. O. Adunola, V. N. Omeke, A. A. Ahmad, A. R. Bunu, A. J. Nkohon, and E. Obiajulu, “Energy-efficient wireless sensor network for real-time environmental monitoring: Simulink-based design,” *Discover Electronics*, vol. 2, no. 1, p. 63, Jul. 2025, doi: 10.1007/s44291-025-00104-8.

[8] N. Yu *et al.*, “Data-driven control, optimization, and decision-making in active power distribution networks,” *Appl Energy*, vol. 397, p. 126253, Nov. 2025, doi: 10.1016/j.apenergy.2025.126253.

[9] S.-Y. Kim and H. Ko, “Energy-Efficient Cooperative Transmission in Ultra-Dense Millimeter-Wave Network: Multi-Agent Q-Learning Approach,” *Sensors*, vol. 24, no. 23, p. 7750, Dec. 2024, doi: 10.3390/s24237750.

[10] J. Lansky *et al.*, “A Q-learning-based routing scheme for smart air quality monitoring system using flying ad hoc networks,” *Sci Rep*, vol. 12, no. 1, p. 20184, Nov. 2022, doi: 10.1038/s41598-022-20353-x.

[11] T. T. Nguyen, T. Thao Nguyen, T. A. Nguyen Dinh, T.-H. Nguyen, and P. Le Nguyen, “Q-learning-based Opportunistic Communication for Real-time Mobile Air Quality Monitoring Systems,” in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, IEEE, Oct. 2021, pp. 1–7. doi: 10.1109/IPCCC51483.2021.9679398.