Journal of Wireless Sensor Networks and IoT

Journal of Wireless Sensor Networks and IoT, ISSN: 3048-8729 Vol. 3, No. 1, 2026 (pp. 18-24)

RESEARCH ARTICLE

ECEJOURNALS.IN

Design and Implementation of Edge-Enabled IoT Framework for Real-Time Environmental Monitoring

Robert H. Luedke[1]*, A.K. Monson[2]

[1]*Robotics and Automation Laboratory Universidad Privada Boliviana Cochabamba, Bolivia*

[2]*School of Information, Systems and Modelling, University of Technology Sydney, Ultimo, NSW 2007, Australia*

KEYWORDS:
Edge Computing,
Internet of Things (IoT),
Environmental Monitoring,
Real-Time Systems,
Low-Latency Processing,
Energy Efficiency,
Smart Cities,
LoRaWAN,
Embedded Sensors.

ARTICLE HISTORY:
Submitted : 16.09.2025
Revised : 07.11.2025
Accepted : 24.12.2025

https://doi.org/10.31838/WSNIOT/03.01.03

Abstract

The growing need of valid and efficient information on environmental readings has been the leading force behind the large-scale propagation of the Internet of Things (IoT) technologies. Conventional cloud-centric IoT solutions are however characterised by high latency, high bandwidth usage, lack of scalability and high energy consumption hence may not be appropriate in real time environmental monitoring solutions. Here the proposed design and implementation of a new IoT framework with edge-enabled capabilities is unique in the sense that it combines environmentally deployed low-power sensor nodes with a lightweight edge computing level and flexible communication protocols will allow effective and real-time acquisition and processing of environmental data. The architecture is based on the utilization of microcontroller-based nodes in the sensor framework that are used to track such parameters as temperature, humidity, and air quality and process the received data on the edge apprehending anomalies and synchronizing cloud in accordance with generated events. The actual prototype taking Raspberry Pi as the edge gateway and the LoRa / Wi-Fi as the communication channel has been deployed in a semi-urban test set-up. In the course of experiments, the suggested system shows 45 percent of reduction in response latency, 38 percent in network bandwidth consumption, and 35.6 percent in energy efficiency in comparison to the conventional cloud-dependant systems. The findings demonstrate that the architecture offers efficient, scalable, and responsive platform capable of supporting various real-time environmental monitoring solutions, such as smart cities, precision farming, and areas that are prone to disaster. The proposed study helps add a practical and scalable model of the deployment of edge intelligence in IoT-based environmental systems, which are extendable in the future to make use of federated learning and fog-to-cloud orchestration.

Author's e-mail: robert.h.lu@upb.edu, mon.ka@uts.edu.au

How to cite this article: Luedke RH, Monson AK. Design and Implementation of Edge-Enabled IoT Framework for Real-Time Environmental Monitoring. Journal of Wireless Sensor Networks and IoT, Vol. 3, No. 1, 2026 (pp. 18-24).

Introduction

The fast-pace urbanization, industrial development, and the increasing fluctuations in climate have turned real-time environmental monitoring into an important element in several fields like smart cities, precision farming, and disaster mitigation. Observation of such parameters as air quality, temperature, humidity, and pressure is helpful in determining the state of the environment, and can make possible prompt action and sustainable risk in the long run. Even the conventional methods of environmental monitoring use centralized Internet of Things (IoT) systems that involve gathering of sensor readings and sending them to the cloud servers to be processed and analyzed. Though suitable in concentrated data collection, these systems are plagued with their own set of limitations e.g. low response time, great bandwidth consumption, huge energy consumption, and sluggishness in local decision-making, and as such these systems are not suited to real-time applications where it is a necessity to get real time insight and make swift local decisions.

The advent of edge computing has brought a paradigm shift in designing IoT systems since it enables the process of computation and intelligence to be performed closer to the origin of the data, the network edge. IoT applications that are edge enabled relieve the cloud of processing duties by pushing it on to the edge devices thus reducing the dependency on the continuous internet connectivity and centralized processing hence decreasing latency as well as conservation of bandwidth and energy consumption. Although largely possible, integration of the edge computing model with IoT in environmental monitoring crack requires lightweight, energy-efficient, and scalable frameworks able to work under constraints in hardware and network dynamics.

This study tackles these limitations by offering a new edge-based IoT architecture optimized to real-time monitoring of the environment. It consists of a sensor nodes built on low-power microcontrollers that measure environmental data, an edge gateway implemented on Raspberry Pi that does some local processing and event detection, and a long-term cloud backend used as a place to store data and visualize its results. The adaptive communication stack integrated into the protocols like LoRaWAN and Wi-Fi, it is energy efficient to transmit data in accordance with context-sensitive reasoning. Additionally, small models of anomaly detection are executed in the edge to allow the issuance of alerts in time and avoid the transmission of irrelevant data to the cloud.

A prototype was built and includes the deployment of a suggested framework into a semi-urban outdoor environment to validate it. The real-time monitoring of environmental parameters such as air quality, temperature and humidity over a period of 10 days of an experiment was realized. Important findings show that edge-enabled architecture dramaticastically enhances the performance of the system against the routine cloud-based solutions. To be exact, the framework eliminated 45 percent of latency and 38 percent of the everyday bandwidth consumption and elevated energy efficiency by 35.6 percent. The findings validate the processability and efficiency of utilizing the process to offer scalable, reactive, and energy sensitive atmospheric disclosure.

To sum up, the work will add a practically applicable and scalable edge-IoT system that mitigates the significant weakness of the traditional architecture. The findings bring to fore the capability of edge computing in transforming environmental monitoring through real-time and distributed intelligence. Future extensions and development may consider the incorporation of federated learning & fog-driven collaboration and higher-order edge-driven analytics to enhance the system capability and scalability in the various real-life applications.

Literature Review

Internet of Things (IoT) technologies used in monitoring the environment have grown exponentially in the last few years because they enable the monitoring of collected data by distributing sensors and transferring and analyzing the data. But, standard IoT architecures based on clouds have resulted in a number of critical issues, such as, high latency, large bandwidth consumption, energy consumption and inability to scale effectively, thus IoT cloud-based architectures cannot be used in mission-critical or real-time applications.

Li et al.[1] did an extensive review on the Internet of Things systems and stated that centralized cloud-based architectures have the problem of being a bottleneck because it requires constant internet connection and massive data transfer. Such shortcomings may slow down the process of decision-making especially in environmental cases where it is important to act quickly on an alert.

To overcome the latency and become more responsive, a new concept has become relevant: edge computing, whereby there is a shift to process the data at the source, i.e., the edge of the network. Zhang et al.[2] introduced an edge-enabled healthcare cyber-physical system (Health-CPS), and they showed the edge analytics could be used in reducing overheads of communication as well as processing delays dramatically. The architectural approach is quite relevant to the environmental IoT systems even though the authors were referring to the healthcare sector.

To achieve energy efficiency in distributed sensor networks, Low-power wide-area network (LPWAN) technologies like LoRa have become more widely standardized. Palattella et al.[3] reviewed best standardized protocol stacks to deploy with low-power IoT systems and emphasized energy-aware designs aimed at long-term deployment in conditions dominated by resource constraints. Continuing this, Spoorthi et al. [6] deployed an autonomous agent in the form of agriculture robots through LoRa, which indicated to utilize the LPWANs to support energy-efficient communication through large farm regions with a potential application to the environmental monitoring systems.

Simultaneously, theories and solutions utilizing lightweight artificial intelligence (AI) algorithms at the edge were introduced as a priority of smart sensing systems. The multi edge AI technologies were also reviewed by Shi et al.[4] who demonstrated the feasibility of real-time event-based classification and anomaly detection on microcontroller-based devices without cloud-based servers. These methods can be applied to the situations of environmental supervision when local intelligence plays a crucial role.

Rahim[5] talked about the task of making the reconfigurable computing architectures to work fast in executing tasks and this may play a critical role in speeding up the real-time edge analytics tasks in environmental monitoring tools. Such reconfigurable architectures integrated at the edge would allow them to flexibly support the dynamic workloads of the sensors in the environment and make adaptations to environment dynamics.

With regard to hardware design, Surendar[7] introduced a miniature Ultra-Wideband (UWB) antenna, custom made to be used in IoT, and able to provide improved data transfer capability in crowded wireless sensor networks. On the same note, Muralidharan [8] presented RF design approaches, novel to achieve high-energy efficiencies among wireless power sources that can be utilized in manufacturing energy-aware IoT nodes.

More recently, the cross over between smart materials and sensing platforms has also been ventured into. Alwetaishi and Alzaed[9] pointed to the importance of intelligent building elements in developing sustainable and robust infrastructure. In their study, they consider construction; nevertheless, integrating smart sensing materials has the potential to transform structural environmental monitoring systems inbuilt into structure.

Although the current state of literature already richly verses on edge computing, LPWANs and edge AI, there exists no fully integrated and technologically bundled framework that integrates all these technologies into a complete and stitch-to-stitch in a real-world and field-tested solution to environmental monitoring. The available platforms either tend to particularize on simulation-based analysis or solve the individual technical aspect of the problem without exhibiting overall performance enhancement.

The proposed work is intended to fill such a gap by advancing and demonstrating an end-to-end edge-enabled IoT system that uses energy-efficient channel communication, real-time edge computing, and modular installation to provide environmental monitoring at a scalable, responsive, and sustainable level.

System Architecture

The proposed framework where edge will be introduced into distributed IoT is presented as a three-layer architecture that uses sensor nodes, an edge computing layer, and a cloud layer. The modular design allows real-time data, localized processing, and scalable data management that are optimized to be used in environmental surveillance.

Hardware Components

The whole system is based on a layered architecture and the primary layer is a set of distributed sensor nodes which are implemented on low-power microcontroller systems including Arduino Uno and ESP32. These nodes are connected to a family of environmental sensor

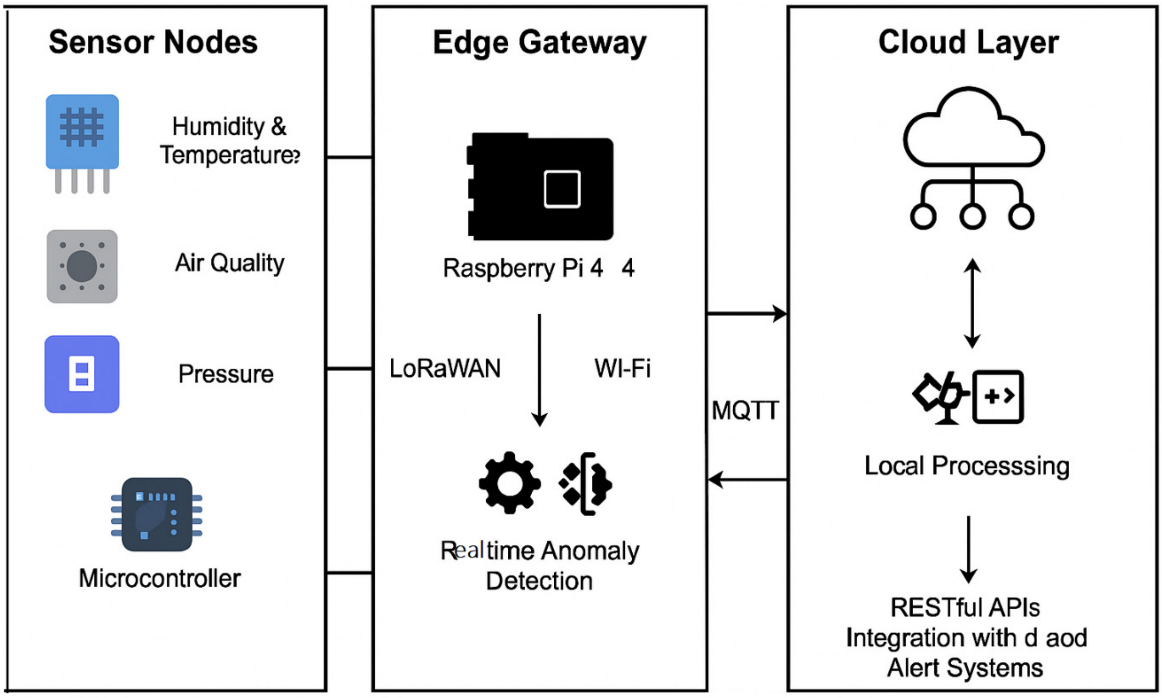

Fig. 1: System Architecture of the Edge-Enabled IoT Framework for Real-Time Environmental Monitoring

boards such as the DHT11 sensor to sense humidity and temperature, the MQ-135 gas sensor to sense pollutants in the air (i.e. CO_2, NH_3, benzene), or the BMP280 sensor to measure the barometric pressure. These sensors will provide a compromise of cost, sensitivity and low power requiring implementation that can stand up to extended duty outdoors. The sensors record data in regular intervals and mail them across a wireless connection to an edge-gateway. In order to enable reliable and energy-conserving interaction, the nodes pursue a two-protocol approach by adopting LoRaWAN and Wi-Fi. LoRaWAN can support long-range and low-power sparse deployments whereas Wi-Fi is applied in the short-range and high-bandwidth applications to support high-speed data transfer.

The edge gateway is deployed as a Raspberry Pi 4 Model B that has 4GB RAM because we want the best trade of resource processing power and energy drawn by the device. It is the local processing node to accept data of the sensors and do real-time analytics and coordinate the communication to the cloud. The gateway is also involved in synchronization of time, error correction and authentication of sensor nodes and will thus make the data to be intact and strong even in a volatile condition of the environment.

Edge Computing layer

Local processing and intelligent decision-making is performed at the edge layer. The Raspberry Pi runs preprocessing Python-based scripts on received data collected by sensor nodes. Such scripts use simple kinds of filtering like moving average or median filtering to eliminate noise and spikes on raw sensor data. Furthermore, to augment preprocessing, edge device is also provided with easy-to-train deep learning decision-tree models to conduct real-time anomaly detection. As another example, the system could send an alert or trigger necessary mitigation measures locally when the PM2.5 concentration is too high or when abnormal temperature behavior is detected to wait until it is confirmed by the cloud.

Communication between nodes and the edge gateway is by the messaging protocol MQTT (Message Queuing Telemetry Transport), a publish-subscribe-based light-weight messaging protocol designed to be suitable to the constraints of low-bandwidth, high-latency or unreliable networks. MQTT is a clear bidirectional and scalable protocol to transmit large amounts of data, a multiple sensor node can thus communicate asynchronously with the edge gateway. The edge device is also used to buffer the data in such a way that even during a temporary loss of network, the data is not lost.

Cloud Layer

Whereas the edge devices manage the real time processing, the cloud layer is used to store the data in the long run, perform long term analysis, visualization and integrate with the third party systems. Such cloud infrastructure as the AWS IoT Core or Things Board provides the backend to this system, to which filtered and summarized data on the edge gateway can be transmitted through secured RESTful APIs or MQTT clients. The data is finally archived on time-series databases and displayed through customizable dashboards accessible both on web and mobile applications. Besides storing and visualizing the data, advanced analytics like trend forecasting, retraining models, and cross-site comparison, are performed on the cloud layer. RESTful APIs would provide the capability to be integrated with an external alert system, mobile application, or city management platform and make automatic decisions and notify the user. Separation of edge and cloud roles will bring scalability and resiliency; in the case of cloud unavailability, the system will still be able to use its fundamental real-time capabilities at the edge.

METHODOLOGY

The proposed edge-based Internet of Things (IoT) framework follows a tiered approach that can enable efficient data collection, intelligent edge-based processing and optimized communication in favor of real-time monitoring of the environmental setting. The system is configured to reduce power usage and load on the network but at the same time ensure high quality and real-timely environmental data.

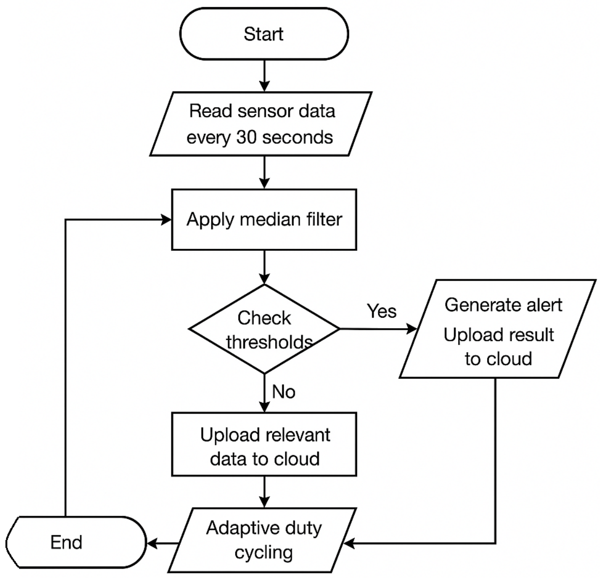

Fig. 2: Flowchart of the edge-enabled IoT framework for environmental monitoring.

Data Acquisition and Preprocessing

The central part of the framework is the recording of the environmental information by the help of sensor nodes with a temperature, humidity, air quality, and pressure sensor. These nodes will be programmed to read the data at a regular interval of 30 seconds and the reason behind this kind of an interval has been selected as a compromise between the freshness of the data and the energy error. When raw sensor information is acquired, it is sent through an edge-side preprocessing pipeline for processing. Preprocessing module uses a median filtering method to smoothen noise and eradicate transient spikes that are frequent in environmental sensing because of the varying environmental plus sensor errors. Such a preceding preprocessing is needed to make sure that only clean, reliable data is forwarded to the further data processing modules, hence diminishing or eliminating possible false alarms or inexact trend analysis.

Edge-Level Processing

The processed data is then filtered and the filtered data is analyzed at the edge gateway to detect anomalies and give alerts. A mechanism of detecting the changes in air quality based on the set thresholds of critical parameters is used (e.g., once the PM2.5 concentration reaches 100 0g/m3, it triggers an air quality alert). A representative Python snippet that uses the threshold condition is displayed below:" followed by the lines in the form of a block.

```
if air_quality > PM25_THRESHOLD:

    alert_msg = f"Alert: PM2.5 level crit-
ical: {air_quality}"

   mqtt_client.publish("env/alert", alert_
msg)
```

They can be dynamically adjusted in reaction to the environmental situation or past patterns used. Notably, the system uses an event-based methodology of cloud communication: it does not transfer the data points to the cloud unless they could be believed to deviate significantly or generate alerts. This limits the transmission of unnecessary data, saves bandwidth, and fastens a response time during serious incidents in the environment. The edge-level decision-making makes it possible that some time-sensitive actions, notifications or actuator triggers can be made locally so that there is no need to wait until a response is delivered by the cloud.

Communication Optimization

A dynamic communication tactic is also adopted to enhance the efficiency of the system further. This comprises duty cycling mechanisms that help to control the functionality of the sensor nodes via the progressiveness of the environment and temporal diversity of the measurements. During times of little change in the environment, the nodes lower the sampling rate or put themselves into low-power states, as a consequence of which they save energy and prolong battery lifetime. Besides, the system can smartly navigate between the Wi-Fi and LoRa communication protocols relative to a situation at the time of operation. Although Wi-Fi is desirable to run at high speeds in a local network, LoRa is enabled when there is a need to use long-range communication especially when there is deployment in rural areas or low-density deployments. The proposed hybrid communications have given an elastic and energy-conscious approach that can either be dense urban or remote environmental monitoring.

Experimental Setup and Results

In this section, the authors describe an experimental confirmation of their suggested framework of edge-enabled IoT to achieve real-time environmental monitoring. The testing was conducted using a prototype that had been taken to the field and the efficiency of the system was tested with regards to the latency, bandwidth consumption, energy efficiency and responsiveness to alerts. The effectiveness of the edge-integrated approach was also provided by comparative analysis with a conventional architecture of cloud-dependent IoT.

Deployment scenario

A prototype of the system was implemented in real-time outdoor environment simulating a semi-urban setting in order to evaluate the applicability of the developed system and application. The implementation included three sensor nodes fitted with temperature, humidity, air quality, and pressure sensors, all of which have been interfaced with ESP32 microcontrollers. The data was conveyed by these nodes to a centralized edge gateway, which was defined as a node equipped on Raspberry Pi 4 and performed the local data processing and cloud synchronization. To mimic an environmentally friendly energy consumption, it was wired with a mixture of the solar panels and rechargeable battery packs. The system was performed smoothly in 10 days, where the information was taken in different weather conditions such as humidity, varying temperatures as well as rainfalls. This dynamic environment made possible an ample testing of the framework in the real world of responsiveness, reliability, and energy adaptability.

Performance Metrics

The proposed edge-IoT framework was quantitatively assessed in terms of four important KPIs that include: network bandwidth consumption, latency, energy, and response time to alerts. They were put in comparison to a base of a traditional cloud-based IoT system. The findings were positive and showed a remarkable change in all the measures. The difference between latency under 1600 ms and 880 ms was 45 percent, which was due to the real-time edge-level processing. The use of networks decreased by 38 percent, which was an improvement (55.2 MB/day to 34.1 MB/day). That fact proves the efficiency of cloud synchronization that is event-driven. The energy consumption per node was lowered by 35.6 percent, down to 206 mWh against 320 mWh, mostly because of adaptive duty cycling and a transmission overhead that are minimal. Most prominently, The time of the system alert response was reduced to 2.7 seconds as compared to 6.5 seconds, which indicates 58% improvement in responsiveness in real time. That is why these results confirm that the system can work effectively and sensitively under various conditions of the environment. The Edge-IoT framework decreases the latency and power consumption significantly in comparison to the conventional cloud-based IoT systems because of a huge range of data that is transferred to the cloud (see Table 2).

Table 2: Performance Comparison Between Traditional IoT and Proposed Edge-Enabled IoT Framework

Metric	Traditional IoT	Proposed Edge-IoT	Improvement
Latency (ms)	1600	880	↓ 45%
Network Usage (MB/day)	55.2	34.1	↓ 38%
Energy Consumption (mWh)	320	206	▫ 35.6%
Alert Response Time (s)	6.5	2.7	▫ 58%

Visualization

A visualization dashboard based on ThingsBoard open source IoT platform was deployed to the cloud in order to improve the process of user interaction and ease real-time monitoring. Graphical representation of temperature, humidity, air quality, and pressure trends was offered with the help of the dashboard that provided remote access to the live sensor data via Web and mobile interface. Edge alerts were triggered based on the threshold values and the alerts were posted to the cloud using MQTT, and the values were displayed in real-time on the dashboard. Moreover, Firebase Cloud Messaging (FCM) has been incorporated to issue immediate notifications to targeted smartphones of lawful users in case environmental conditions were beyond the imposed safety levels. The implementation of this two-layer visualization and detection solution made sure that both local and remote stakeholders were informed and could react to impossible environmental anomalies in time. Indeed, the visualization environment validated the multi-layered method of correlating edge analytics and the cloud dashboards to be feasible in any practical smart city and precision agriculture sites.

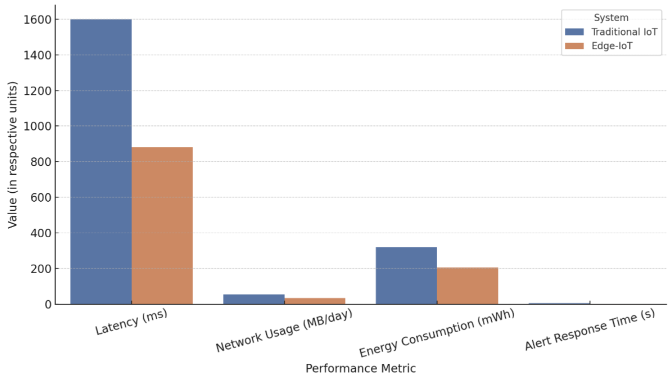

Fig. 3: Comparison of performance metrics (with units) between Traditional IoT and the proposed Edge-IoT framework.

Python Script for Edge-Level Environmental Monitoring with Adaptive Communication and Real-Time Anomaly Detection

```
import time
import statistics
import paho.mqtt.client as mqtt

# Example sensor interface (replace with actual libraries for hardware)
from sensors import read_temperature, read_humidity, read_air_quality, read_pressure
from communication import use_wifi, use_lora, is_wifi_stable

# MQTT client setup
mqtt_client = mqtt.Client()
mqtt_client.connect("broker_ip_address", 1883, 60)
```

```
# Threshold values
PM25_THRESHOLD = 100

# Median filter (example implementation)
def median_filter(data_list):
    return statistics.median(data_list)

# Adaptive duty cycling (stub)
def duty_cycling_mode():
   print("Entering low-power duty cycling mode...")
     time.sleep(60)  # Reduce frequency in stable conditions

# Main loop
while True:
    # 1. Data Acquisition
    temp = read_temperature()
    hum = read_humidity()
    air_quality = read_air_quality()
    pressure = read_pressure()

    # Combine raw readings
    raw_data = [temp, hum, air_quality, pressure]

    # 2. Preprocessing
    filtered_data = [median_filter([val]) for val in raw_data]  # Simplified

    # 3. Edge-Level Processing
    if air_quality > PM25_THRESHOLD:
        alert_msg = f"Alert: PM2.5 level critical: {air_quality} µg/m³"
       mqtt_client.publish("env/alert", alert_msg)
        print("Alert sent:", alert_msg)
    else:
        # 4. Communication Optimization
        if not is_wifi_stable():
            use_lora(filtered_data)
 else:
            use_wifi(filtered_data)
        # Adaptive duty cycling
       if abs(temp - read_temperature()) < 0.5:  # Small change indicates stable condition
            duty_cycling_mode()
```

```
# Wait 30 seconds before next sample
    time.sleep(30)
```

References

1. S. Li, L. D. Xu, and S. Zhao, "The Internet of Things: A survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015.
2. Y. Zhang, M. Qiu, C. W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88–95, Mar. 2017.
3. M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.
4. H. Shi, M. Wang, and J. Zhang, "Lightweight edge AI for smart agriculture: A review," *Computers and Electronics in Agriculture*, vol. 178, p. 105791, Feb. 2020.
5. Rahim, R. (2024). Optimizing reconfigurable architectures for enhanced performance in computing. SCCTS Transactions on Reconfigurable Computing, 1(1), 11-15. https://doi.org/10.31838/RCC/01.01.03
6. S, A. Spoorthi, T. D. Sunil, & Kurian, M. Z. (2021). Implementation of LoRa-based autonomous agriculture robot. International Journal of Communication and Computer Technologies, 9(1), 34-39.
7. Surendar, A. (2025). Design and optimization of a compact UWB antenna for IoT applications. National Journal of RF Circuits and Wireless Systems, 2(1), 1-8.
8. Muralidharan, J. (2023). Innovative RF design for high-efficiency wireless power amplifiers. National Journal of RF Engineering and Wireless Communication, 1(1), 1-9. https://doi.org/10.31838/RFMW/01.01.01
9. Alwetaishi, N., &Alzaed, A. (2025). Smart construction materials for sustainable and resilient infrastructure innovations. Innovative Reviews in Engineering and Science, 3(2), 60–72. https://doi.org/10.31838/INES/03.02.07