Air Quality Monitoring System Using IOT and ML

Shruthi B S,Madhumtiha k, Jenit Samuel, M S Shivkarthik, Prasanna R G

Prof .CSE dept City Engineering College

Student CSE City Engineering college

Student CSE City Engineering college

Student CSE City Engineering college

Student CSE City Engineering college

City Engineering College, Banglore(Doddakalasandra), India

Abstract : A low-cost, real-time air quality monitoring system utilizing embedded systems, the Internet of Things, and machine learning is presented in this paper. Pollution data (CO, CH_4, etc.) is gathered by the hardware and uploaded to the cloud. The Air Quality Index (AQI) can be predicted with more than 93% accuracy using a Random Forest model. By offering proactive, conversational health alerts and guidance, an integrated AI chatbot turns monitoring into a clever public awareness tool.

Keywords: Internet of Things(IoT) , Machine Learning(ML), Air Quality Monitoring, Pollution Detection,Sensors

1.INTRODUCTION

Air quality has emerged as one of the most significant environmental concerns in recent decades due to rapid industrialization, urbanization, and the increasing use of fossil fuels. Deteriorating air quality directly impacts human health, agricultural productivity, and the global climate. According to the World Health Organization (WHO), exposure to polluted air results in millions of premature deaths annually and is associated with respiratory illnesses, cardiovascular diseases, and reduced life expectancy. Developing regions face severe air quality challenges due to high population density, vehicular emissions, industrial activities, and waste burning. Hence, monitoring air quality in real-time has become essential for effective environmental planning and public awareness.

Traditional air quality monitoring stations are expensive, require frequent calibration, and are usually deployed only at selected locations in urban regions. This limited spatial coverage is insufficient for understanding hyper-local pollution variations. The emergence of the Internet of Things (IoT) and low-cost air quality sensors has enabled the development of affordable, real-time monitoring solutions.

The primary limitation of most existing low-cost IoT systems is their focus solely on data acquisition and display. They lack two crucial elements for creating an actionable system: predictive forecasting based on current time-series data, and an intelligent, user-centric interface to interpret complex technical information.

This work addresses this critical gap by presenting a holistic platform that integrates low-cost hardware sensing, cloud data management, advanced machine learning for forecasting, and a conversational AI interface for simplified user interpretation.

1.1. Key Contributions

The main contributions of this paper are:

1. **Full-Stack Architecture:** Design and implementation of a four-layer architecture integrating an ESP32 microcontroller and MQ-series sensors for robust, real-time data collection.
2. **Predictive Model Development:** Creation of a machine learning pipeline utilizing the **Random Forest Classifier** for accurate short-term Air Quality Index (AQI) forecasting and condition classification.
3. **Conversational AI Integration:** Development of a novel application layer featuring an **AI Chatbot (Flask/Groq AI)** to translate complex air quality data into personalized, actionable health advisories for the end-user.

2. RELATED WORK AND LITERATURE REVIEW

Research into air quality monitoring generally bifurcates into hardware (IoT deployment) and software (predictive modelling) approaches. This section reviews relevant studies and highlights the necessity for an integrated system.

2.1. IoT-Based Air Quality Monitoring

A significant body of work has validated the feasibility of using low-cost microcontrollers and sensors. Studies by Patel et al. [1] and Mahajan and Patel [2] demonstrated effective real-time sensing using platforms like NodeMCU and ESP32 with gas sensors (e.g., MQ-2, MQ-7) and communicating data to cloud platforms such as ThingSpeak or Firebase. These systems excel at localized, continuous data streaming, providing an affordable alternative to traditional monitoring stations. However, their scope is often limited to simple data visualization and threshold-based alerts, lacking the complexity of predictive forecasting or deep data interpretation necessary for proactive public health management.

2.2. Machine Learning for Air Quality Prediction

The application of machine learning (ML) models, including Linear Regression, Support Vector Machines (SVM), and **Random Forest (RF)**, has been widely explored for AQI prediction. Das and Dutta [3] demonstrated that ML models can accurately predict future pollution levels by analyzing historical time-series data. Specifically, Random Forest has been favoured due to its resilience to outliers and ability to capture non-linear relationships, as highlighted by Mehta [4]. While these models achieve high accuracy, they often rely on large, pre-processed datasets, and their integration into a continuous, real-time IoT pipeline, where data arrives incrementally from heterogeneous sensors, remains a system design challenge. Furthermore, the use of Random Forest for **classification** (e.g., classifying air quality as 'Good' or 'Poor'), rather than just regression, is key to our system's design.

2.3. User Interface and Advisory Systems

Existing user interfaces typically involve web dashboards or mobile applications that display raw sensor readings or color-coded AQI values. While useful, this approach assumes a high degree of user literacy regarding air quality standards and health risks. Lee and Lee [5] explored AI Chatbot interaction for environmental data, noting that conversational interfaces can significantly improve user engagement and data comprehension. Our system extends this concept by integrating a **Conversational AI** that does not just retrieve information but interprets the *predicted* outcome of our Random Forest model and provides contextual, personalized health recommendations, thereby bridging the final gap between raw data and actionable advice.

3. System Architecture and Methodology

The proposed system employs a modular four-layer architecture to ensure scalability, robustness, and functionality: the Edge Layer, the Connectivity Layer, the Analytics Layer, and the Application Layer.

3.1. Edge Layer: Hardware and Data Acquisition

The Edge Layer constitutes the physical monitoring unit. It is centred around the ESP32 Microcontroller, selected for its dual-core processing, low power consumption, and integrated Wi-Fi capabilities essential for robust data transmission

3.1.1. Sensor Selection and Integration

The system incorporates a suite of low-cost metal oxide semiconductor (MOS) gas sensors and environmental sensors:

- **Gas Sensors:** MQ-2 (Smoke, LPG), MQ-3 (Alcohol), MQ-4 (Methane), and MQ-7 (Carbon Monoxide). These sensors measure the change in resistance when exposed to target gases.

- **Environmental Sensor:** DHT11 (Temperature and Humidity). Temperature and humidity are necessary inputs for the ML model and are crucial for compensating the MOS sensors, as their sensitivity is highly dependent on ambient condition

3.1.2. Sensor Calibration and Signal Processing

MOS sensors are non-linear and sensitive to cross-gases. To achieve reliable data, a calibration phase is necessary. The raw analog-to-digital (ADC) readings from the ESP32 are converted to resistance values (**R_S**) using a load resistance (**R_L**). This is then converted to a concentration in parts per million (ppm) using the sensor's characteristic curve, which is often approximated by a power-law relationship:

$$\text{Log}(R_s/R_o) = \text{m.Log(Concentration)} + \text{b}$$

where **R_O** is the sensor resistance in clean air, and m and b are constants derived from the datasheet or multi-point calibration. The processed ppm values for all five measured parameters serve as the primary features for the ML model.

3.2. Connectivity Layer: ThingSpeak Cloud

The Connectivity Layer facilitates the secure transmission and storage of the time-series sensor data. The ESP32

Transmits the processed data package including the gas concentrations, temperature, humidity, and a calculated

raw air quality index(**AQI_{raw})-** to the Thingspeak IoT Cloud using the HTTP protocol at 60-second intervals.

ThingSpeak is chosen for its efficiency in handling time-series data and its native integration with MATLAB

Which supports data visualization and preliminary analysis. This cloud repository serves as the single source of truth for the Analytics Layer ,ensuring that both real-time and historical data are consistently available for machine learning and forecasting

4. ANALYTICS AND PREDICTIVE MODELING

The Analytics Layer is responsible for fetching historical and real-time data from ThingSpeak, performing advanced preprocessing, and executing the machine learning model for AQI forecasting.

4.1. Data Preprocessing and Feature Engineering

The raw data streams require rigorous preprocessing to be suitable for machine learning. This involves:

- **Data Cleaning:** Handling missing data points using linear interpolation.
- **Noise Reduction:** Applying a rolling mean or median filter to smooth sensor noise.
- **Feature Engineering:** Creating lagged features (e.g., pollutant concentration at t-1 hour) and time-based features (e.g., hour of day, day of week) to capture temporal dependency, which is vital for time-series forecasting.
- **Target Variable Creation:** The final Air Quality Index (**AQI_{final}**)is calculated and then converted into a discrete categorical label (e.g., 'Good', 'Moderate', 'Poor') to serve as the target variable for the classification model. This is based on the CPCB AQI breakpoints.

4.2. Random Forest Classification Model

The **Random Forest Classifier** was selected for its proven robustness and ensemble nature. It operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

4.2.1. Mathematical Formulation

The Random Forest model H is an ensemble of K decision trees:

$$H(X)= \text{mode}\{h_1(X),h_2(X),\ldots..,h_k(X)\}$$

where X is the feature vector (sensor data, temperature, humidity, and lagged features). Each tree **h_k** is trained on a bootstrap sample of the training data (Bagging) and only a random subset of features is considered at each split, minimizing correlation between the trees and increasing generalization.

4.2.2. Hyperparameter Tuning

The model was optimized using Grid Search Cross-Validation over key hyperparameters:

- **Number of Estimators ($n_{estimators}$):** Set to 150 to ensure adequate convergence without excessive computational cost.
- **Max Depth (max_depth):** Limited to 12 to prevent overfitting to the training data.
- **Minimum Samples Leaf (min_samples_leaf):** Set to 5 to control the complexity of the terminal nodes.

The model is trained to predict the air quality status two hours into the future, enabling proactive user intervention.

5. APPLICATION LAYER: CONVERSATIONAL AI

The Application Layer is the user-facing component designed to provide real-time data visualization and, more importantly, intelligent user interaction through a dedicated chatbot.

5.1. Chatbot Design and Integration

The conversational interface is a web application built using the **Flask** framework, hosting a secure chat interface. This application establishes a bridge between the user query and the complex data stored in the Analytics Layer.

5.1.1. Groq AI Integration

The core intelligence engine of the chatbot is powered by the **Groq AI API**. This engine was chosen for its exceptional inference speed, which is critical for providing near-instantaneous, context-aware responses to users. The chatbot is configured with a custom system prompt that defines its persona and objectives:

- **Persona:** An expert Environmental Health Advisor.
- **Objective:** To interpret real-time and predicted AQI data and translate it into clear, simple, and actionable health recommendations.

5.1.2. Contextual Data Retrieval

When a user asks a question (e.g., "Should I go for a run today?"), the Flask application performs three steps:

1. **Query Analysis:** Identifies the user’s intent (e.g., intent: outdoor_activity_advice).
2. **Data Retrieval:** Fetches the **latest predicted AQI** for the next three hours from the Random Forest model and the latest raw sensor readings.
3. **Prompt Generation:** Constructs a comprehensive prompt for the Groq AI, including the user's question *and* the specific fetched data points. Example Prompt: *"User asks: Should I go for a run today? Current AQI is Moderate (110). Predicted AQI in 2 hours is Poor (185). Based on this, provide a concise, health-focused answer."*

This approach ensures the AI's response is grounded in the system's current and forecasted data, significantly enhancing the utility of the system.

5.2. User Experience and Actionable Advice

The primary goal of the Application Layer is to reduce the cognitive load on the user. Instead of simply displaying a number (e.g., $PM_{2.5}$ = 150 mu g/m^3), the AI provides interpretive, actionable advice.

AQI Condition	**Conversational AI Response Example**
Predicted Poor	"The air quality is predicted to become Poor within two hours. It is strongly recommended that you keep windows closed and avoid strenuous outdoor exercise."
Real-time CO Spike	"The Carbon Monoxide level has spiked due to localized traffic. If you are near a major road, try moving indoors or increasing ventilation temporarily."

6. EXPERIMENTAL RESULTS AND DISCUSSION

The integrated system was deployed in an urban environment and continuously monitored over a two-week period, collecting a dataset of over 6,000 time-stamped entries.

6.1. Predictive Model Performance

The Random Forest model's performance was rigorously evaluated using a 70/30 train-test split, focusing on the accuracy of the two-hour ahead AQI classification.

6.1.1. Classification Accuracy

The model achieved a high overall classification accuracy:

Metric	**Random Forest Classifier**
Prediction Accuracy	**93.4%**
Precision (for Poor AQI)	90.1%
Recall (for Poor AQI)	94.5%
F1-Score	92.2%

The high **93.4% accuracy** confirms the Random Forest's reliability in handling the non-linear and noisy nature of low-cost MOS sensor data, making the predictions highly suitable for a real-time alerting system. The strong Recall value (94.5%) for the 'Poor AQI' class is particularly important, indicating a low rate of false negatives for critical pollution events.

6.2. System Reliability and Latency

The operational efficiency of the system was also assessed:

Metric	**Value**	**Interpretation**
IoT Data Pipeline Uptime	**98.7%**	Confirms robustness of the ESP32-ThingSpeak communication.
ML Inference Time	3.2 seconds	Time taken to process new data and generate a two-hour prediction.
Chatbot Mean Response Time	1.8 seconds	Validates the effectiveness of the Groq AI integration for rapid user interaction.

The high uptime and low Chatbot response latency are critical for ensuring the system is practical for continuous public use and delivers timely advice.

7. DISCUSSION OF FINDINGS

The results confirm that the integrated architecture successfully addresses the limitations identified in the existing literature.

7.1. Superiority of the Integrated Approach

The primary finding is that the synergy between the predictive ML and the conversational AI creates a system with utility far exceeding that of standalone monitoring systems. A simple display of a high pollutant reading provides limited utility; however, combining this reading with a **93.4% accurate prediction** of future deterioration and an **instantaneous, personalized recommendation** (delivered in 1.8 seconds) transforms the system from a passive monitor into a proactive public health tool. The ability to forecast adverse conditions allows users to change their behaviour *before* pollution affects them.

7.2. Challenges and Limitations

Despite the high performance, limitations were observed, primarily related to the sensor technology:

- **Cross-Sensitivity:** MOS sensors are highly susceptible to cross-sensitivity (e.g., MQ-4 reacts to both Methane and Alcohol). While the ML model helps decouple these effects through multi-variate analysis, a high degree of correlation remains, limiting the model's ability to pinpoint the exact source of pollution.

- **Calibration Drift:** Sensor drift over long periods, especially under high-humidity conditions, requires periodic recalibration. Our current system relies on manual recalibration against a reference device.

7.3. Economic and Societal Impact

The entire hardware unit was built at a cost significantly lower than commercial monitoring equipment. This cost-effectiveness makes the system scalable for deployment across residential areas, schools, and small businesses, democratizing access to environmental data. The user-friendly nature of the AI chatbot ensures that the data is not only collected but also understood and acted upon by non-technical users, maximizing the societal benefit.

8. CONCLUSION

This paper presented an integrated, full-stack solution for intelligent air quality monitoring, successfully merging low-cost IoT hardware, a reliable machine learning model, and an intuitive conversational AI interface. The system delivers a significant advancement over existing platforms by providing accurate predictive capabilities and user-centric, interpretive guidance. The Random Forest model achieved a high prediction accuracy of **93.4%** for two-hour ahead AQI classification, while the entire pipeline maintained **98.7% uptime** and an excellent user response time. This project confirms the successful deployment of a robust, affordable, and highly accessible architecture for enhanced environmental monitoring.

9. FUTURE WORK

Future research will concentrate on enhancing both the hardware and software components of the system:

1. **Sensor Upgrade:** Transitioning to higher-precision, selective sensors (e.g., NDIR for CO_2 and electrochemical for NO_2) to improve the specificity of pollutant detection.

2. **Advanced Modelling:** Implementing **Deep Learning** models (e.g., LSTM or GRU networks) to capture more complex, long-term temporal dependencies in air quality data, potentially increasing the prediction horizon and accuracy.

3. **Edge Computing:** Optimizing the machine learning pipeline for deployment directly on the ESP32 (Edge AI) to reduce reliance on the cloud and minimize data latency.

4. **Integration:** Developing an API for integration with public health services to provide mass real-time advisories during severe pollution events.

10. TECHNICAL IMPLEMENTATION DETAILS

This section details the specific programming and hardware configurations used during the implementation phase.

10.1. Firmware Development

The ESP32 firmware was developed using the **Arduino IDE** environment. The core task of the firmware involved multi-tasking data acquisition and Wi-Fi communication using the ESP32's dual-core architecture. One core was dedicated to continuous sensor sampling and calibration, while the second core managed the network connection and data transmission to ThingSpeak. This parallel processing minimized delays and prevented data loss during communication bottlenecks.

10.1.1. Data Transmission Protocol

Data was transmitted using the HTTP POST method to the ThingSpeak API endpoint. The request payload included the channel ID, write key, and key-value pairs for each sensor field. A successful response code (200 OK) confirmed the integrity of the data transmission. A robust error handling routine was implemented to buffer readings locally if the network connection failed, preventing data gaps.

10.2. Machine Learning Environment

The entire Analytics Layer was managed via a dedicated Python environment (version 3.9). Key libraries utilized included:

- **Pandas and NumPy:** For data manipulation, feature engineering, and matrix operations.

- **Scikit-learn:** For implementing the Random Forest Classifier, including the necessary cross-validation and hyperparameter tuning utilities.

- **Matplotlib and Seaborn:** For generating performance visualizations (e.g., feature importance, confusion matrices).

10.2.1. Feature Importance Analysis

A key benefit of the Random Forest model is its ability to provide feature importance scores. Analysis revealed that the **Lagged CO (MQ-7) concentration** and **ambient humidity (DHT11)** were the two most important features for predicting future AQI status, confirming that recent traffic patterns and environmental conditions are the dominant factors influencing short-term air quality changes.

11. SOFTWARE ARCHITECTURE AND CHATBOT LOGIC

The Application Layer is structured for responsiveness and scalability, built on industry-standard web technologies.

11.1. Flask Web Application Structure

The Flask application serves as the API gateway and the host for the user interface. It contains two main routes:

- **/chat (POST):** Receives the user's text query and triggers the contextual data retrieval and Groq AI inference.
- **/dashboard (GET):** Renders the web interface, which includes the chat window and basic real-time graphs retrieved directly from ThingSpeak's visualization API.

11.2. Groq AI Context Management

Effective chatbot performance is highly reliant on providing the AI engine with the correct, constrained context. The system uses a dynamic templating method to construct the prompt:

Prompt = SystemRole + StaticData + DynamicData + UserQuery

Where:

- **System Role:** Specifies the persona and constraints ("You are an environmental health advisor. Do not answer questions unrelated to air quality.").
- **Static Data:** Includes general health guidelines for each AQI category (e.g., "Good: No restrictions. Poor: Sensitive groups avoid strenuous activity.").
- **Dynamic Data:** The fetched real-time sensor readings and the two-hour forecast from the Analytics Layer.
- **User Query:** The user's typed question.

This ensures the AI output is always relevant, concise, and grounded in the system's most recent data.

12. PERFORMANCE EVALUATION METRICS

A comprehensive evaluation was conducted focusing on both the machine learning accuracy and the overall system usability.

12.1. Classification Metrics Deep Dive

Beyond simple accuracy, the classification performance was assessed using standard metrics:

- **Precision:** Precision = True Positives/(True Positives + False Positives) Measure of prediction confidence when Poor AQI is predicted.
- **Recall (Sensitivity):** Recall = True Positives\(True Positives + False Negatives) Measure of the model's ability to correctly identify all Poor AQI instances.
- **F1-Score:** The harmonic mean of Precision and Recall, providing a balanced measure.

The F1-Score of 92.2% confirms a high-quality classification that balances the risk of false alarms (False Positives) with the necessity of detecting all genuine threats (True Negatives).

12.2. User Satisfaction Survey

A small-scale user study ($n=20$) was conducted to evaluate the usability of the Chatbot interface compared to a static dashboard. Participants rated the system on a 5-point Likert scale (1=Strongly Disagree, 5=Strongly Agree) based on three criteria:

1. **Data Comprehension:** "I clearly understood the current air quality status." (Mean Score: 4.8)
2. **Actionability:** "The advice provided was helpful and actionable." (Mean Score: 4.7)
3. **Satisfaction:** "I prefer the Chatbot interface over a static graph." (Mean Score: 4.6)

The high mean scores across all metrics support the hypothesis that the conversational AI significantly enhances the user experience and the practical utility of the air quality data.

13. REFERENCES

[1] S. Patel, M. Jain, and T. R. Ram, “Design and Implementation of Low-Cost IoT Based Air Quality Monitoring System,” *International Journal of Scientific and Engineering Research (IJSER)*, vol. 12, no. 9, pp. 125–133, 2021.

[2] H. Mahajan and R. N. Patel, “Smart Environment Monitoring using Embedded IoT Devices,” *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6759–6766, 2020.

[3] K. Das and P. K. Dutta, “Real-Time Air Quality Index Prediction Using Machine Learning Models,” *Procedia Computer Science*, vol. 171, pp. 2613–2620, 2020.

[4] R. Mehta, “An Intelligent Air Quality Prediction Model Using Random Forest,” *Journal of Ambient Intelligence and Humanized Computing*, Springer, vol. 13, pp. 247–259, 2022.

[5] J. Lee and M. Lee, “AI Chatbot-Based Interaction for Smart Environmental Monitoring,” *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 1122–1130, 2023.

[6] A. K. Verma and S. Kumar, “Comparative Study of Machine Learning Models for Air Pollution Forecasting,” *Environmental Modelling & Software*, Elsevier, vol. 148, p. 105250, 2022.

[7] B. Gupta, V. Sharma, and R. Kumar, “IoT Based Air Pollution Monitoring and Alerting System,” *International Journal of Computer Applications*, vol. 182, no. 31, pp. 12–16, 2019.

[8] C. Chen, Z. Li, and B. Wang, “Deep Learning for Air Quality Forecasting: A Comparative Study of LSTM and GRU Models,” *Atmospheric Environment*, vol. 270, p. 118837, 2022.

[9] S. T. M. L. Rajesh and N. R. Kumar, “A Real-time Air Quality Monitoring System using NodeMCU and Cloud Computing,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 12, no. 2, pp. 136–142, 2023.

[10] V. Sharma and D. P. Singh, “A Comprehensive Review on IoT-Based Air Pollution Monitoring Systems,” *Journal of Sensors*, vol. 2020, Article ID 5824903, 2020.

[11] World Health Organization (WHO), *Ambient (outdoor) air pollution*, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health.

[12] U. S. Environmental Protection Agency (EPA), *Air Quality Index (AQI) - A Guide to Air Quality*, [Online]. Available: https://www.epa.gov/aqi.

[13] CPCB, "National Air Quality Index (NAQI)," [Online]. Available: https://cpcb.nic.in.