Air Quality Index (AQI) Monitoring and Prediction System Report:

Air Quality Index (AQI) Monitoring and Prediction System

A Group Project Report

Submitted in partial fulfilment of the requirements for

Data Visualisation and Analytics

Submitted by:

1. Soham Kadam – 121A1036

2. Sourish Kanna – 122A1098

3. Maheswari Nadar – 121A1062

Under the Guidance of:

Prof. Kranti Bade

Department of Computer Engineering

SIES Graduate School Of Technology

Academic Year: 2025–26

1. Introduction

Air pollution has emerged as a major environmental and public health concern, particularly in metropolitan regions such as Mumbai, Thane, and Navi Mumbai. High concentrations of pollutants like PM2.5 and PM10 adversely affect respiratory and cardiovascular health. Although air quality data is available from various sources, most existing systems either provide delayed information or lack predictive capabilities. There is a growing need for a system that can deliver real-time air quality information, forecast short-term pollution trends, and present insights in an easily interpretable manner.

This group project focuses on the design and development of a real-time Air Quality Index (AQI) monitoring and prediction system. The system integrates live data collection, machine learning-based prediction, and interactive data visualization, while remaining simple, reliable, and deployable within a limited academic timeframe.

2. Project Objectives

The primary objectives of the project are:

- To collect real-time air pollution and weather data for selected urban regions.
- To automate continuous data fetching and storage without manual intervention.
- To apply a machine learning model for short-term AQI prediction.
- To visualize live and predicted pollution trends using an interactive dashboard.
- To generate health advisories based on pollution levels.
- To implement Regional Level Security (RLS) so users can view only city-specific data.

3. System Architecture and Workflow

The system follows a modular end-to-end architecture. Live air pollution and weather data are sourced from the OpenWeatherMap API in structured JSON format. A Python-based backend automation script fetches this data every 15 minutes. The script is deployed on an Oracle Cloud Virtual Private Server (VPS), ensuring 24/7 execution.

After fetching the data, relevant parameters such as AQI level, PM2.5, PM10, gaseous pollutants, temperature, humidity, and wind speed are extracted and structured into a single record for each city. The processed data is then stored in two locations:

- A local CSV file for backup and historical analysis.
- A live Google Sheet, which acts as a cloud-based data source for visualization.

Power BI is connected directly to the Google Sheet, allowing dashboards to refresh automatically as new data arrives. A machine learning module is integrated into the backend to predict short-term air quality trends.

4. Live Data Collection

The OpenWeatherMap platform provides two key APIs used in this project:

- Air Pollution API: Supplies AQI levels and pollutant concentrations such as PM2.5, PM10, CO, and NO_2.
- Weather API: Provides meteorological parameters including temperature, humidity, and wind speed.

The backend script first converts city names into geographical coordinates using the OpenWeatherMap Geocoding API. These coordinates are then used to fetch pollution and weather data. Data is collected for Mumbai, Thane, and Navi Mumbai at 15-minute intervals, creating a continuous stream of real-time information.

5. Machine Learning Model for AQI Prediction

To introduce predictive capability, a machine learning model is used to forecast short-term pollution levels. Linear Regression is selected as the prediction model due to its simplicity, interpretability, and suitability for short-term trend analysis.

The model uses historical PM2.5 values along with time-based indexing to predict the next-hour PM2.5 concentration. If sufficient historical data is not available, a controlled fallback mechanism estimates the value using recent trends. The predicted PM2.5 value is stored alongside the actual data, enabling comparison and trend analysis on the dashboard.

6. Health Advisory Generation

Raw AQI values may not be easily understandable by all users. To address this, a rule-based health advisory system is integrated into the backend. Based on AQI levels, the system generates simple and actionable messages such as safe air conditions, moderate pollution warnings, or high pollution alerts advising the use of protective measures like masks. These advisories are displayed directly on the dashboard.

7. Dashboard Visualization and Scheduling

Power BI is used to develop an interactive dashboard connected to the live Google Sheet. The dashboard includes:

- KPI cards showing average AQI, PM2.5, and PM10 levels.
- Line charts comparing actual and predicted PM2.5 values.
- Scatter plots illustrating the relationship between wind speed and pollution.
- Map visualizations indicating city-wise pollution intensity.

The dashboard is configured to refresh based on scheduled data updates. The backend data pipeline is scheduled to run **once every day at 10:00 AM**, ensuring that the dashboard reflects the most recent consolidated air quality and prediction data without requiring manual refresh or intervention.

8. Results and Observations

The system successfully captures continuous air quality data and reflects real-time changes in pollution levels. Observations indicate that PM2.5 levels remain consistently high in urban regions, while increased wind speed often contributes to reduced pollution concentration. The predicted PM2.5 trends closely follow actual values, validating the effectiveness of the machine learning approach for short-term forecasting.

9. Conclusion and Future Scope

The project demonstrates the successful integration of live data collection, machine learning prediction, cloud deployment, and business intelligence visualization. It fulfills all major academic and technical requirements, including real-time data handling, predictive analytics, dashboard creation, and data security.

Future enhancements may include the use of advanced forecasting models such as LSTM or ARIMA, expansion to additional cities, integration of alert systems via SMS or email, and development of a web or mobile application interface to improve accessibility.

Air Quality Index (AQI) Monitoring and Prediction System

1. Introduction

Air pollution has emerged as a major environmental and public health concern, particularly in metropolitan regions such as Mumbai, Thane, and Navi Mumbai. High concentrations of pollutants like PM2.5 and PM10 adversely affect respiratory and cardiovascular health. Although air quality data is available from various sources, most existing systems either provide delayed information or lack predictive capabilities. There is a growing need for a system that can deliver real-time air quality information, forecast short-term pollution trends, and present insights in an easily interpretable manner.

This group project focuses on the design and development of a real-time Air Quality Index (AQI) monitoring and prediction system. The system integrates live data collection, machine learning-based prediction, and interactive data visualization, while remaining simple, reliable, and deployable within a limited academic timeframe.

2. Project Objectives

The primary objectives of the project are:

- To collect real-time air pollution and weather data for selected urban regions.
- To automate continuous data fetching and storage without manual intervention.

- To apply a machine learning model for short-term AQI prediction.
- To visualize live and predicted pollution trends using an interactive dashboard.
- To generate health advisories based on pollution levels.
- To implement Regional Level Security (RLS) so users can view only city-specific data.

3. System Architecture and Workflow

The system follows a modular end-to-end architecture. Live air pollution and weather data are sourced from the OpenWeatherMap API in structured JSON format. A Python-based backend automation script fetches this data every 15 minutes. The script is deployed on an Oracle Cloud Virtual Private Server (VPS), ensuring 24/7 execution.

After fetching the data, relevant parameters such as AQI level, PM2.5, PM10, gaseous pollutants, temperature, humidity, and wind speed are extracted and structured into a single record for each city. The processed data is then stored in two locations:

- A local CSV file for backup and historical analysis.
- A live Google Sheet, which acts as a cloud-based data source for visualization.

Power BI is connected directly to the Google Sheet, allowing dashboards to refresh automatically as new data arrives. A machine learning module is integrated into the backend to predict short-term air quality trends.

4. Live Data Collection

The OpenWeatherMap platform provides two key APIs used in this project:

- Air Pollution API: Supplies AQI levels and pollutant concentrations such as PM2.5, PM10, CO, and NO_2.
- Weather API: Provides meteorological parameters including temperature, humidity, and wind speed.

The backend script first converts city names into geographical coordinates using the OpenWeatherMap Geocoding API. These coordinates are then used to fetch pollution and weather data. Data is collected for Mumbai, Thane, and Navi Mumbai at 15-minute intervals, creating a continuous stream of real-time information.

5. Machine Learning Model for AQI Prediction

To introduce predictive capability, a machine learning model is used to forecast short-term pollution levels. Linear Regression is selected as the prediction model due to its simplicity, interpretability, and suitability for short-term trend analysis.

The model uses historical PM2.5 values along with time-based indexing to predict the next-hour PM2.5 concentration. If sufficient historical data is not available, a controlled fallback mechanism estimates the value using recent trends. The predicted PM2.5 value is stored alongside the actual data, enabling comparison and trend analysis on the dashboard.

6. Health Advisory Generation

Raw AQI values may not be easily understandable by all users. To address this, a rule-based health advisory system is integrated into the backend. Based on AQI levels, the system generates simple and actionable messages such as safe air conditions, moderate pollution warnings, or high pollution alerts advising the use of protective measures like masks. These advisories are displayed directly on the dashboard.

7. Dashboard Visualization and Scheduling

Power BI is used to develop an interactive dashboard connected to the live Google Sheet. The dashboard includes:

- KPI cards showing average AQI, PM2.5, and PM10 levels.
- Line charts comparing actual and predicted PM2.5 values.
- Scatter plots illustrating the relationship between wind speed and pollution.
- Map visualizations indicating city-wise pollution intensity.

The dashboard is configured to refresh based on scheduled data updates. The backend data pipeline is scheduled to run once every day at 10:00 AM, ensuring that the dashboard reflects the most recent consolidated air quality and prediction data without requiring manual refresh or intervention.

8. Results and Observations

The system successfully captures continuous air quality data and reflects real-time changes in pollution levels. Observations indicate that PM2.5 levels remain consistently high in urban regions, while increased wind speed often contributes to reduced pollution concentration. The predicted PM2.5 trends closely follow actual values, validating the effectiveness of the machine learning approach for short-term forecasting.

9. Conclusion and Future Scope

The project demonstrates the successful integration of live data collection, machine learning prediction, cloud deployment, and business intelligence visualization. It fulfills all major academic and technical requirements, including real-time data handling, predictive analytics, dashboard creation, and data security.

Future enhancements may include the use of advanced forecasting models such as LSTM or ARIMA, expansion to additional cities, integration of alert systems via SMS or email, and development of a web or mobile application interface to improve accessibility.