Phase -III

Predicting Air Quality Levels Using Advanced Machine Learning Algorithms

Student Name : S.Kokila

Register Number : 620123106055

Institution: AVS ENGINEERING COLLEGE

Department: ECE

Date of submission: 17/05/2025

GitHub Repository Link :
http://https://github.com/kokila2410/Kokila.S-Naan-Mudhalvan-project-.git

1.Problem Statement

Air pollution poses a significant risk to public health and the environment. Traditional methods of air quality monitoring are often reactive and limited in scope. There is a need for predictive systems that can forecast air quality levels using real-time data and machine learning to enable timely interventions.

2. Abstract

This project focuses on predicting air quality levels using advanced machine learning algorithms. Leveraging data from environmental sensors and meteorological sources, we develop models to forecast pollutant concentrations such as PM2.5, NO2, and O3. The system includes data preprocessing, model training, evaluation, and deployment of a web-based dashboard for real-time prediction. The goal is to assist policymakers and the public in making informed decisions about environmental safety.

3. System Requirements

Hardware:
- Intel i5/i7 processor
- 8GB+ RAM
- 100GB+ storage

Software:
- Python (NumPy, Pandas, Scikit-learn, TensorFlow/Keras)
- Jupyter Notebook
- Docker (for deployment)
- Web framework (Flask/Streamlit)
- Database (SQLite or PostgreSQL)
- Cloud (optional: AWS/GCP/Azure)

4. Objectives

- Predict AQI levels based on environmental and weather features.
- Build a robust, scalable ML pipeline.
- Develop an intuitive interface for real-time AQI visualization.

- Support government agencies and individuals in pollution control planning.

5. Project Workflow (Flowchart)

[Data Collection] --> [Data Preprocessing] --> [Exploratory Data Analysis] --> [Feature Engineering] --> [Model Building] --> [Model Evaluation] --> [Deployment] --> [Monitoring & Maintenance]

6. Dataset Description

Source: Government agencies (e.g., EPA, CPCB), Kaggle, OpenAQ, AQICN.
Features:

- Timestamp
- Location (Latitude/Longitude)
- Pollutant levels (PM2.5, PM10, NO2, SO2, O3, CO)
- Temperature, humidity, wind speed
- AQI levels (target)

7. Data Preprocessing

- Handling missing data using imputation.
- Encoding timestamps to extract day, hour, season.
- Scaling numeric features using StandardScaler/MinMaxScaler.
- Handling outliers using IQR or Z-score.

8. Exploratory Data Analysis (EDA)

- Visualize trends in pollutant levels.
- Correlation matrix to identify key predictors.
- Seasonal and hourly patterns of pollution.
- Detect anomalies or sensor errors.

9. Feature Engineering

- Create lag features for time-series predictions.
- Aggregate pollutant levels by hour/day.
- Introduce interaction terms (e.g., wind * pollutant level).
- Use PCA or feature selection to reduce dimensionality.

10. Model Building

- Algorithms used:
  - Linear Regression, Random Forest
  - XGBoost, LightGBM
  - LSTM (for time-series predictions)
- Train-test split (or TimeSeriesSplit)
- Hyperparameter tuning with GridSearchCV

11. Model Evaluation

- Metrics:
  - MAE, RMSE, R^2 for regression
  - Accuracy/F1 if predicting AQI categories
- Cross-validation to assess generalization

I

12. Deployment

- Deploy model using Flask or Streamlit as a web app.
- Docker containerization for portability.
- Optionally host on AWS EC2 or Heroku.
- Provide UI for users to input environmental data and view AQI prediction.

13. Source Code

GitHub Repository: http://https://github.com/kokila2410/Kokila.S-Naan-Mudhalvan-project-.git

Includes:

Data preprocessing scripts

Jupyter notebooks for EDA and model training

Deployment files (Flask app, Dockerfile)

README and documentation

14. Future Scope

- Integrate satellite imagery and IoT sensor data.
- Extend to global cities and forecast long-term air quality.
- Include health advisories based on AQI predictions.
- Mobile app for real-time alerts.

15. Team Members and Roles

Team Member	Role
Kokila.S	Project Lead – Managed the project lifecycle, ML design, and environmental insights.
Gomathi.T	ML Engineer – Model training, tuning, and evaluation.
Kamali.M	Data Engineer – Data acquisition and preprocessing pipeline.