PREDICTING AIR QUALITY USING ADVANCED MACHINE LEARNING ALGORITHM

- ***STUDENT NAME:*** *MANISHA. A*
- ***REGISTER NUMBER:*** *422223106022*
- ***COLLEGE NAME:*** *Surya Group of Institutions*
- ***DATE OF SUBMISSION:*** *05/05/2025*
- *GitHub Link: https://GitHub.com/MANISHA2906-GM/Predicting-air-quality_using-advanced-machine-learning-algorithm-git*

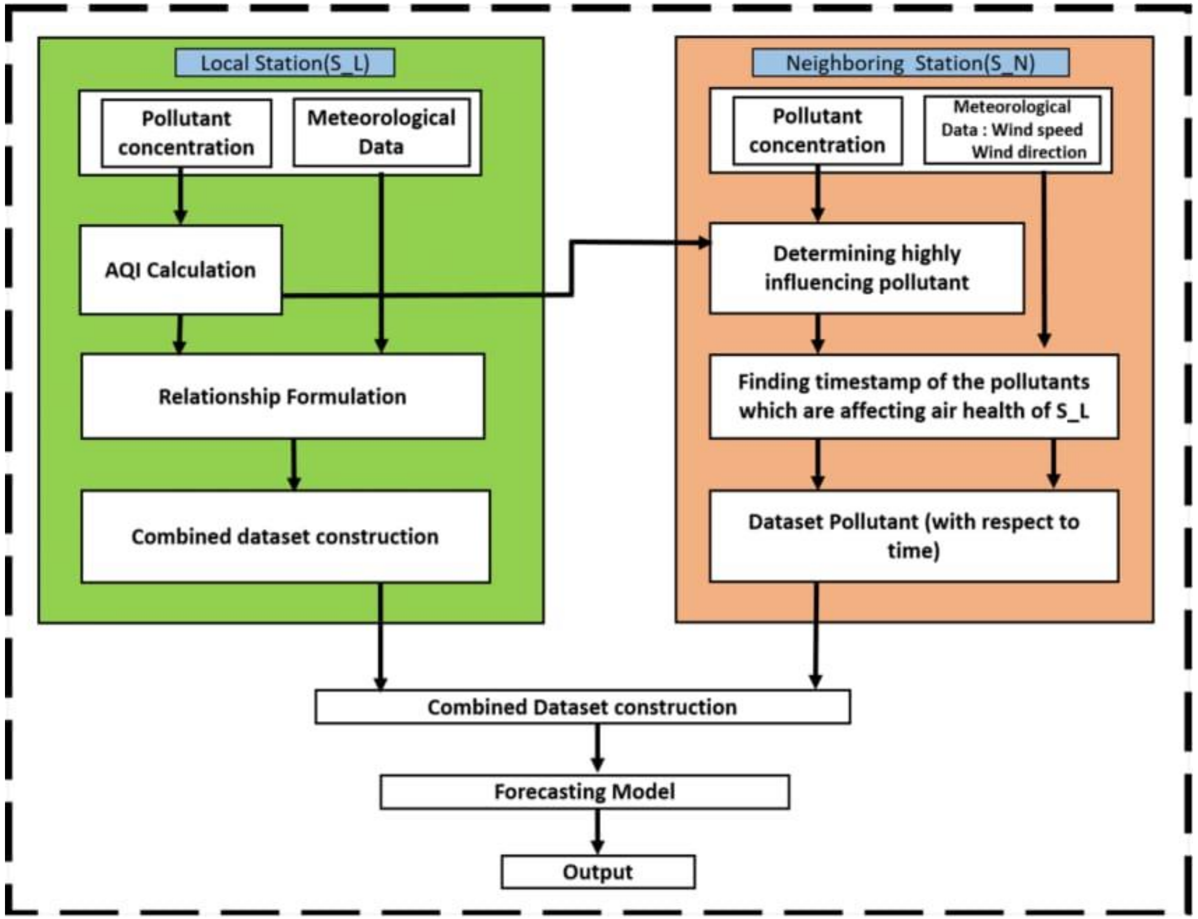
PROBLEM STATEMENT

- *Air pollution is a growing concern globally, affecting human health, ecosystem, and climate. The need for real-time and accurate prediction of air quality has become crucial. The challenge lies in processing complex environmental data to forecast the Air Quality Index (AQI). This project aims to address this issue by applying machine learning algorithms to predict AQI levels based on environmental and meteorological parameters.*

PROJECT OBJECTIVE

- *To develop a predictive model: Create a machine learning-based model that accurately forecast AQI using environmental datasets.*
- *To support decision making: Provide actionable insights for government agencies and citizens to take preventive measure.*
- *To analyze pollutant impact: Identify key pollutants and meteorological features that significance influence air quality.*
- *To compare algorithm: Evaluate different machine learning models and identify the most effective one for AQI prediction.*

FLOWCHART OF THE PROJECT WORKFLOW


Local Station(S_L)
Pollutant concentration
Meteorological Data
AQI Calculation
Relationship Formulation
Combined dataset construction
Neighboring Station(S_N)
Pollutant concentration
Meteorological Data : Wind speed Wind direction
Determining highly influencing pollutant
Finding timestamp of the pollutants which are affecting air health of S_L
Dataset Pollutant (with respect to time)
Combined Dataset construction
Forecasting Model
Output

DATA DESCRIPTION

- *To predict air quality using advanced machine learning, data sets typically include historical air quality data from monitoring stations, meteorological conditions, and potentially other relevant factors like traffic data.*
- *This datasets are used to train models that can predict future air quality, often using advanced algorithms like random forests or deep learning.*

- *Date Description link:*

DATA PREPROCESSING

- *Missing value handling: Imputation using mean, median, or interpolation*
- *Data Formatting: Converting timestamp to datetime objects*
- *Outline Detection: Using IQR method or Z-score to filter extreme values*
- *Normalization/Scaling: StandardScaler or MixMaxScaler to scale features*
- *Encoding Time Features: Extracting month, day, weekday for seasonal patterns*

EXPLORATORY DATA ANALYSIS (EDA)

- *Univariate Analysis: Distribution plots of individual pollutants*
- *Bivariate Analysis: Scatter plots and correlation matrices to study relationship between features and AQI*
- *Temporal Trends: Line plots showing pollutant variation over time*
- *Heatmaps: Correlation matrix showing dependencies among variables*
- *AQI Distribution: Histogram and box plots for AQI values*

FEATURE ENGINEERING

- *Time Features: Adding features like hour of day, day of week, or season*
- *Rolling Average: Computing moving averages for pollutant levels to capture trends*
- *Lag Features: Including values from previous time steps to capture temporal dependencies*
- *Pollutants Ratios: Creating ratios like PM2.5/PM10 to indicate pollution types*
- *Dimensionality Reduction: Using PCA if required for high-dimensional data*

MODEL BUILDING

- *Algorithms used:*
- *Linear Regression (as a baseline)*
- *Random Forest Regressor*
- *XGBoost*
- *LSTM (for time-series data)*
- *Training Process: Splitting the data into train/test sets, cross-validation*
- *Evaluate Metrics: RMSE, MAE,R2 score for model performance*

VISUALIZATION OF RESULTS AND MODEL

- *Predicted vs. Actual AQI: Line or scatter plots to compare model predictions*

- *Features Importance: Bar chart from Random Forest or XGBoost*

- *Error Distribution: Histogram of residuals (errors)*

- *Model Comparison: Graphical comparison of performance metrics across models*

TOOLS AND TECHNOLOGIES USED

- *Programming Language: Python*
- *Libraries and Frameworks: Pandas, Numpy for data handling*
- *Matplotlib, Seaborn for visualization*
- *Scikit-learn for machine learning*
- *XGBoost and TensorFlow/Keras for advanced modeling*
- *Development Environment: Jupyter Notebook, Google Colab*
- *Version Control: Git/GitHub*
- *Data sources: CPCB API, OpenAQ API, Kaggle datasets*

TEAM MEMBERS AND CONTRIBUTIONS

- Data analyst: Collected datasets, performed data cleaning, and conducted EDA – KAVITHA. S
- ML Engineer: Designed and trained machine learning models, features engineering – ROOBINI. S
- Project Manager: Coordinated tasks, documented the project, and conducted stakeholder review – MANISHA. A
- Visualization Lead: Created plots, dashboards, and presented results visually – TAMIZHVANI. R