# Air Quality Prediction and Monitoring System in Delhi NCR

**N. Nalini Krupa ,Thota Navya Sri , Upputuri Sri Satviki, S N Venkata Jahnavi, Pulla John Stephen**

Assistant Professor, Department of CSE-AI&ML , Vasireddy Venkatadri Institute of Technology,Guntur,A.P.,India.
Department of CSE-AI&ML , Vasireddy Venkatadri Institute of Technology,Guntur,A.P.,India.
Department of CSE-AI&ML , Vasireddy Venkatadri Institute of Technology,Guntur,A.P.,India.
Department of CSE-AI&ML , Vasireddy Venkatadri Institute of Technology,Guntur,A.P.,India.
Department of CSE-AI&ML , Vasireddy Venkatadri Institute of Technology,Guntur,A.P.,India.

***Abstract :*** The Delhi NCR region faces hazardous air pollution, but existing monitoring systems are reactive, lacking the predictive capabilities needed for timely public health action. To address this, the project developed a three-tier full-stack application using React.js and FastAPI. The methodology integrates real-time data from the WAQI API with an XGBoost ensemble model for 24, 48, and 72-hour AQI forecasting, alongside a Multi-output Random Forest model for pollution source attribution. The results demonstrate high technical efficacy, with the forecasting model achieving an $R^2$ score of 0.91 and an overall RMSE of 12.4. The system successfully passed functional and security testing and was containerised using Docker for scalable cloud deployment. This outcome solves the core problem by bridging the gap between raw data and actionable insights. By providing accurate forecasts, identifying major pollution contributors (such as traffic or industry), and offering AI-powered health advisories, the system empowers citizens and authorities to move from passive observation to proactive health preparedness.

## INTRODUCTION

The primary problem is that air pollution in the Delhi NCR region has reached hazardous levels, yet existing monitoring systems are reactive rather than proactive. While government stations provide real-time AQI readings, they are often limited in spatial coverage and lack predictive capabilities. Consequently, citizens and authorities only become aware of dangerous pollution levels after they have already peaked, leaving little time for effective mitigation or health precautions.

This issue is critical because air pollution is a major contributor to millions of premature deaths globally and causes severe health problems, including respiratory distress and cardiovascular diseases, for the millions of people living in Delhi NCR. Current delays in implementing measures like traffic restrictions or public advisories mean that large populations are frequently exposed to hazardous spikes without warning. Moving to a predictive, proactive system is essential for improving public health preparedness and enabling data-driven decision-making.

The project, named Air Quality Prediction & Monitoring System, is bridging the gap between raw data and actionable insights by using an XGBoost ensemble model to provide accurate short-term AQI forecasts for 24, 48, and 72-hour intervals. Implementing a Multi-output Random Forest model to estimate the percentage contribution of different sources, such as traffic, industry, construction, and stubble burning.

Building a secure, scalable web application using a React.js frontend and a Fast API backend to visualize real-time data, trend analysis, and pollution heatmaps. Integrating Google Gemini AI to deliver personalized health advisories and precautionary measures based on predicted pollution levels.The entire system is containerised using Docker and deployed on cloud infrastructure to ensure it is stable, scalable, and accessible to the public via modern browsers.

## RELATED WORK

### 2.1 FIRST GENERATION

The earliest approaches relied on fixed monitoring stations that measured real-time pollutant concentrations (PM2.5, PM10, $NO_2$, etc.). These systems provided reliable data for reporting current conditions but lacked predictive capabilities, using only simple threshold-based rules for analysis. Government-operated platforms, such as those by the CPCB, fall into this category; while they offer high-quality real-time data, they provide limited insights and no forecasting.

### 2.2 SECOND GENERATION

The second generation introduced predictive modeling using techniques like Linear Regression, ARIMA (AutoRegressive Integrated Moving Average), Support Vector Machines (SVM), and Random Forests. ARIMA-based models are used for short-term forecasting but often struggle with the non-linear patterns inherent in environmental data. SVM and Random Forest models achieved moderate performance, typically between 70–85% accuracy, but were limited in their ability to model complex temporal dependencies over time.

### 2.3 THIRD GENERATION

The current generation focuses on Long Short-Term Memory (LSTM) networks and hybrid deep learning models. These systems are highly effective at learning seasonal variations and complex temporal patterns, achieving accuracy rates between 85–

95%. However, these models remain primarily in the research or prototype stage and are highly sensitive to data inconsistencies or missing values in real-world settings.

### 2.4 EXISTING APPLICATIONS

Commercial platforms like AirVisual aggregate data and provide basic machine learning forecasts. While these apps are user-friendly, they often lack robust AI-driven prediction models and fail to provide personalized, context-aware health recommendations. General AI models (LLMs) have also been explored but generally lack the domain-specific accuracy and real-time integration required for precise monitoring.

### IDENTIFICATION

Most deployed systems focus only on real-time reporting, leaving a gap in accessible. Existing tools provide AQI values but lack user-centric guidance on health impacts and preventive measures. Much of the advanced research remains confined to experimental stages, lacking fully integrated, cloud-based platforms that combine data collection, prediction, and user interaction for the general public.

### METHODOLOGY

The project utilizes a three-tier client-server architecture comprising a React.js frontend, a FastAPI backend, and a PostgreSQL or SQLite database. The methodology initiates with real-time data ingestion from external sources such as the WAQI API, which is then preprocessed using Pandas and NumPy to handle missing values and engineer up to 28 features, including temporal data, pollutant ratios, and historical AQI lags. Core intelligence is provided by two modular machine learning components: an XGBoost ensemble model (utilizing five boosters with different seeds) for 24, 48, and 72-hour AQI forecasting[1], and a Multi-output Random Forest regressor for identifying the percentage contribution of various pollution sources[2]. Actionable insights are enhanced through Google Gemini AI for health recommendations and presented via interactive dashboards featuring Recharts and Leaflet-based heatmaps. For robust deployment, the system is containerized with Docker and hosted on cloud platforms like Railway using GitHub Actions for CI/CD automation.

### 4.1 DATASET

The dataset used in this project is a structured time-series environmental dataset specifically scoped to the Delhi NCR region. It aggregates information from multiple primary sources, including the World Air Quality Index (WAQI) API and the Central Pollution Control Board (CPCB)[5]. The dataset comprises a total of 28 input features. Real-time concentrations for PM2.5, PM10, $NO_2$, $SO_2$, CO, and $O_3$. Environmental variables such as temperature, humidity, and wind speed. Time-based indicators including hour, day, month, day of the week, and a weekend indicator. Sine and cosine transformations for month and hour to capture periodic seasonal and daily patterns. Spatial attributes consisting of latitude and longitude for the region. Historical "lag" features (AQI levels at t-1, t-6, t-12, and t-24 hours)[6] and rolling means for 24-hour and 72-hour intervals. Engineered features such as the PM Ratio (PM10/PM2.5) and the Traffic Ratio ($NO_2$/CO).

The project utilizes two distinct subsets of historical data for different functional requirements. One of which is Forecasting Dataset which consists of CPCB and WAQI historical data covering 2019-2015 and other is Source attribution Dataset which utilizes labelled CPCB data from 2015-2024 specifically indexed for identifying pollution sources like traffic, industry and construction. The data is maintained as a structured numerical dataset and includes contextual attributes to capture both short-term fluctuations and long-term environmental trends.

### 4.2 PREPROCESSING

The preprocessing workflow for the ADPSI system is a structured pipeline designed to transform raw environmental data into a format suitable for machine learning inference. Initially, the data is cleaned using Python libraries like Pandas and NumPy to address inconsistencies and handle missing values from external API sources. In this tabular and time-series context, the system effectively resizes the feature space through extensive feature engineering, expanding the raw pollutant data into 28 specific input features for the forecasting model and 10 features for the source attribution model. These features include temporal indicators, cyclic transformations (sine/cosine), and historical lag values to provide the models with necessary context.
Feature scaling and normalization are applied consistently across the entire dataset to ensure that the diverse ranges of pollutants and meteorological variables are processed uniformly by the machine learning models. This normalization logic is parameterized within a configuration module in the backend service layer. By using the same normalization parameters for both the initial model training and real-time inference, the system avoids discrepancies that could otherwise degrade the accuracy of the AQI predictions.

### 4.3 TRAINING

The dataset is divided using a structured train-test-validation split, specifically allocating 70% of the data for training, 15% for validation, and 15% for testing. A critical requirement of this split is the preservation of temporal order, which prevents "data leakage" in time-series forecasting. By ensuring that the training data strictly precedes the testing data in time, the system is evaluated on its genuine ability to forecast future conditions based on past patterns, mirroring real-world operational scenarios.

During this process, the validation set is used to monitor for overfitting and to tune hyperparameters like learning rate and tree depth.

### 4.4 MODEL ARCHITECTURE

The ADPSI system is built on a three-tier client-server architecture utilizing a React.js presentation layer, a FastAPI application layer, and a PostgreSQL or SQLite data tier. The system adopts a monolith-first approach with modular layers, where machine learning logic is encapsulated within service classes like ForecastService and SourceAnalysisService that are loaded into memory at startup for efficient real-time inference. The primary AQI forecasting engine is an XGBoost Gradient Boosting Ensemble consisting of five boosters trained with unique random seeds (42, 53, 64, 75, and 86) [1] to ensure high accuracy and generalization. This model processes 28 input features—including real-time pollutant levels, cyclic temporal data, and historical lag values—to generate reliable 24, 48, and 72-hour forecasts with confidence levels derived from ensemble agreement.For identifying pollution causes, the system integrates a Multi-output Random Forest Regression model that estimates percentage contributions from five sources: traffic, industry, construction, stubble burning, and others. This model focuses on a streamlined set of 10 features, specifically leveraging key pollutant ratios like the PM ratio (PM10/PM2.5) and the $NO_2$/CO ratio to pinpoint dominant environmental drivers. The entire application is containerised using Docker to provide a consistent runtime environment across development and production stages. The FastAPI backend serves as the communication hub, delivering results via RESTful APIs in structured JSON format, which allows the XGBoost and Random Forest modules to be independently scaled or updated while remaining part of a unified full-stack platform.

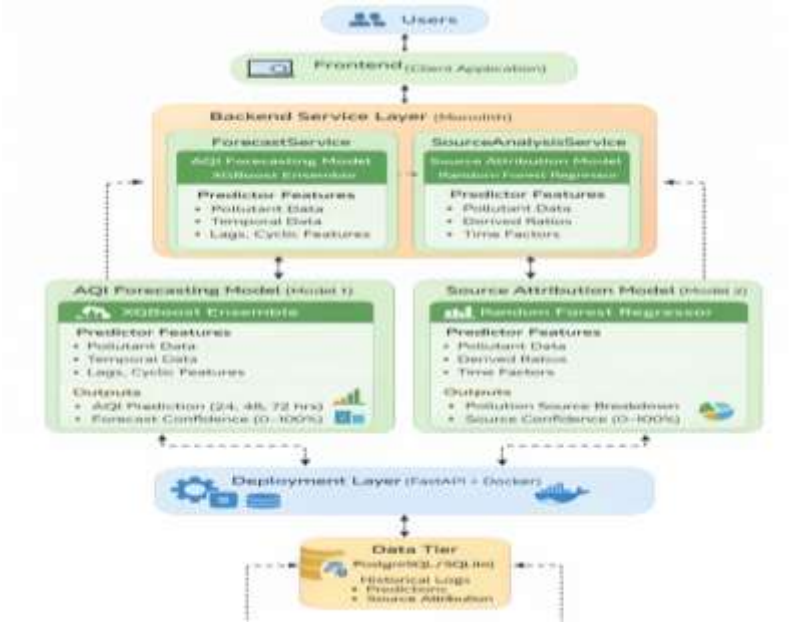


Figure 1: Model Architecture

## RESULTS

The primary XGBoost forecasting ensemble achieved high predictive accuracy with an overall $R^2$ score of 0.91, an overall Mean Absolute Error (MAE) of 8.7, and a Root Mean Squared Error (RMSE) of 12.4. The model's performance is strongest in the 24-hour window, reaching an $R^2$ of 0.93, while maintaining reliable scores of 0.91 and 0.88 for 48-hour and 72-hour horizons, respectively. While most predictions remain within a tight ±10 AQI range, the sources note that deviations are more common during sudden, non-linear pollution spikes caused by external factors such as seasonal stubble burning or intense traffic congestion.

For deeper insights, the Multi-output Random Forest model identifies the root causes of pollution by estimating the percentage contributions of traffic, industry, and construction, providing a dominant source identification. The system's technical integrity was verified through 15 structured test cases that all returned a PASS result, confirming its security against risks like SQL injection and unauthorised data access. Additionally, user testing with ten participants validated the platform's accessibility, leading to a 100% success rate for core tasks and the implementation of key UI improvements such as interactive tooltips and AQI category indicators to enhance data interpretability.
Now fully containerised with Docker and deployed on the Railway cloud, the application maintains a median latency of 2.1 seconds, ensuring high performance for real-world users. By bridging the gap between monitoring and prediction, the ADPSI platform successfully delivers actionable, AI-powered public health intelligence to the residents of the Delhi NCR region.

| Prediction Horizon | MAE | RMSE | $R^2$ Score | Description |
|---|---|---|---|---|
| 24 Hours | 7.9 | 11.2 | 0.93 | Short-Term prediction Accuracy |

| | | | | |
|---|---|---|---|---|
| 48 Hours | 8.5 | 12.1 | 0.91 | Mid – Term Prediction Accuracy |
| 72 Hours | 9.7 | 13.8 | 0.88 | Longer-Term Prediction Accuracy |
| Overall Performance | 8.7 | 12.4 | 0.91 | Total Model Accuracy |

Table 1: The XGBoost Ensemble Model was evaluated using Mean Absolute Error(MAE),Root Mean Squared Error(RMSE), and the $R^2$ Score acroos different time Horizons.

## CONCLUSION

This project successfully bridged the significant gap between the availability of raw air quality data and the practical accessibility of actionable predictive insights for the Delhi NCR region. The core achievement is the development of a high-accuracy forecasting engine using an XGBoost-based ensemble model, which achieved an overall $R^2$ score of 0.91 and an average Mean Absolute Error (MAE) of 8.7 across 24, 48, and 72-hour intervals. Unlike traditional monitoring systems that are purely reactive, this system empowers users to move toward proactive health preparedness by identifying pollution trends before they reach hazardous levels.

A secondary technical milestone was the implementation of a Multi-output Random Forest model for Dominant Source Identification. This module provides meaningful insights into the major contributors of pollution—such as traffic, industry, and construction—allowing both citizens and authorities to understand the root causes of air quality degradation. This analytical capability was integrated into a production-ready, three-tier full-stack application featuring a responsive React.js frontend and a high-performance FastAPI backend, which includes over 13 unique features such as pollution heatmaps, AI-powered health recommendations via Google Gemini, and a citizen reporting portal.

The project also achieved a high standard of engineering excellence and reliability through structured testing and modern deployment practices. Systematic functional and security testing resulted in a 100% pass rate across 15 critical scenarios, including protection against OWASP Top 10 risks. User testing with ten participants further validated the system's usability, leading to a 100% success rate for core tasks and the incorporation of key interface improvements like interactive tooltips and AQI category indicators. Finally, by containerising the entire system with Docker and deploying it to the Railway cloud with automated CI/CD pipelines, the project demonstrated a scalable, end-to-end solution ready for real-world impact and future expansion to other urban regions.

The future scope for the ADPSI system focuses on expanding its geographical reach and functional capabilities to maximize its public health impact. A primary priority for Version 2.0 is the expansion to other major Indian cities, such as Mumbai, Kolkata, and Bengaluru, which will require regional model calibration and enhanced infrastructure to handle increased data loads. To improve accessibility, the system aims to integrate multi-language voice input and output in regional languages and develop an offline-capable Progressive Web App (PWA) for users with intermittent connectivity. Additionally, the roadmap includes IoT sensor integration, personalized health recommendations, and the use of advanced orchestration tools like Kubernetes alongside monitoring platforms like Prometheus and Grafana to ensure system reliability and scalability as the user base grows.

We are excited for the future which can implement more systems for different cities that can have good impact and transform the existing models with higher efficiency to a greater cause. Code is available here : https://github.com/navyasrithota2006/aqi

## REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," 2016.
[2] L. Breiman, "Random Forests," Machine Learning, 2001.
[3] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," 2001.
[4] World Health Organization, "Ambient (Outdoor) Air Pollution," 2023.
[5] Central Pollution Control Board, "National Air Quality Monitoring Programme (NAMP)," 2022.
[6] G. Box and G. Jenkins, "Time Series Analysis: Forecasting and Control," 1970.
[7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997.