# Air Quality Prediction Using Time Series Forecasting Techniques

Md. Abrar Khan[1], Dr. K. Rajitha[2], and R. Mohan Krishna Ayyappa[3]
[1]*Student, Mahatma Gandhi Institute of Technology*
[2,3]*Assistant Professor, Mahatma Gandhi Institute of Technology*

***Abstract*—Air pollution is a growing global concern with severe implications for public health and environmental sustainability. This study presents a real-time air quality prediction system using time series forecasting techniques to estimate future Air Quality Index (AQI) levels. The system leverages machine learning models—ARIMA, LSTM, and Facebook Prophet—trained on historical pollutant data (e.g., PM2.5, PM10) and meteorological variables (e.g., temperature, humidity, wind speed). The platform integrates data preprocessing, model evaluation using MAE and RMSE, and visualization through a user-friendly dashboard built with Streamlit. It also supports real-time predictions and health advisories, aiming to enable early intervention and informed decision-making. The project emphasizes modularity, scalability, and accessibility for both public and institutional use.**

***Index Terms*—Air Quality Index, Forecasting, IoT, Time Series, Machine Learning.**

## I. INTRODUCTION

Air pollution has emerged as one of the most pressing global challenges, significantly impacting environmental quality and human health. Rapid industrialization, urbanization, and vehicular emissions have led to alarming levels of airborne pollutants, especially in densely populated regions. The need for accurate and real-time air quality forecasting has become increasingly vital for public health protection and environmental policy planning.

Traditional air quality monitoring systems, while informative, are often limited by their static nature, delayed reporting, and lack of predictive capabilities. These systems provide data that is reactive rather than proactive, reducing their effectiveness in mitigating pollution-related risks.

To address these gaps, this paper presents a real-time air quality prediction system using time series forecasting techniques. The system utilizes historical pollutant data (e.g., PM2.5, PM10, $NO_2$, $SO_2$, CO, $O_3$) and meteorological parameters (e.g., temperature, humidity, wind speed) to train forecasting models such as ARIMA, LSTM, and Prophet. The goal is to enable early detection of pollution spikes and provide timely alerts through a user-friendly dashboard, thus empowering both individuals and authorities to take preventive action.

By combining statistical, deep learning, and visualization techniques, this project aims to deliver a scalable, modular, and accessible platform for continuous air quality monitoring and forecasting.

## II. LITERATURE REVIEW

Several studies have explored time series and machine learning models for air quality forecasting. Traditional statistical models like ARIMA offer baseline accuracy but struggle with non-linear dependencies in pollutant data. Recent approaches have incorporated deep learning methods such as Long Short-Term Memory (LSTM) networks and hybrid models for enhanced performance.

Reference [1] compared ARIMA, LSTM, and Prophet models, noting that hybrid methods improve accuracy but require extensive preprocessing. A hybrid LSTM-ARIMA model was proposed in [2] for early warning systems, though the authors acknowledged that linear components may oversimplify complex pollution dynamics.

The authors in [3] evaluated ARIMA, LSTM, and 1D CNN models for PM2.5 prediction and concluded that LSTM performed best in terms of Mean Absolute Percentage Error (MAPE), though results were limited to a single urban dataset. Similarly, [4] emphasized that model performance is highly sensitive to the quality and granularity of input data.

Reference [5] demonstrated that BiLSTM models outperform traditional techniques for real-time AQI forecasting but lacked cross-regional validation. In [8], a hybrid time series framework was introduced that effectively modelled seasonality and sudden pollutant spikes, although it required consistent historical data.

These studies [1]–[5], [8] highlight the growing emphasis on combining statistical and deep learning techniques for AQI forecasting. However, most models lack integration with real-time systems or user-facing dashboards. Our project addresses this gap by combining time series forecasting with interactive visualization and real-time health advisories.

## III. METHODOLOGY

The proposed system leverages time series forecasting models to predict air quality in real time. It integrates data preprocessing, model training, and visualization into a streamlined architecture suitable for deployment and public interaction.

### *A. Data Collection and Preprocessing*

Historical pollutant data—including PM2.5, PM10, $NO_2$, $SO_2$, CO, and $O_3$—was combined with meteorological variables like temperature, humidity, and wind speed. Due to limited live sensor availability, synthetic datasets were generated using patterns from real-world repositories. The data was cleaned, time-indexed, and resampled to hourly intervals. Missing values were interpolated, and outliers were treated using statistical thresholds.

### *B. Forecasting Models*

Three time series forecasting models were implemented:

- ARIMA: A classical statistical model for capturing linear temporal dependencies.
- LSTM: A deep learning model that captures long-term and nonlinear patterns in sequential data.
- Prophet: A Facebook-developed forecasting model that handles seasonality and missing values effectively, suitable for quick iteration.

Each model was trained city-wise and pollutant-wise, using 80% of the dataset for training and 20% for validation.

### *C. Evaluation Metrics*

To assess model performance, two widely accepted metrics were used:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)

These metrics quantified prediction accuracy and helped compare model reliability.

### *D. Model Selection and Tuning*

ARIMA parameters (p, d, q) were tuned using AIC values. LSTM was configured with one hidden layer and dropout to prevent overfitting. Prophet used additive seasonality components. Hyperparameters were optimized manually due to time constraints, and all models were evaluated against the same dataset to ensure fairness.

## IV. SYSTEM DESIGN AND ARCHITECTURE

The system is architected to provide real-time air quality forecasting through a modular and scalable pipeline. It consists of three main layers: data handling, model inference, and user interaction via a dashboard.

### *A. System Workflow*

1. Data Ingestion: The system accepts CSV-formatted datasets containing hourly pollutant concentrations and meteorological parameters. The data is cleaned, normalized, and structured for time series modelling.
2. Model Inference: Forecasting models (ARIMA, LSTM, Prophet) are triggered based on user input (e.g., city selection). Forecasts are generated for short-term AQI trends.
3. Result Classification: Predicted PM2.5 values are mapped to AQI categories (e.g., Good, Moderate, Unhealthy) based on standard thresholds.
4. Visualization and Alerts: The final output is visualized on a web dashboard with health advisories and trend graphs.

### *B. Dashboard Components*

The interactive user interface is built using Streamlit, allowing non-technical users to:

- Select a city or pollutant
- View predicted vs. historical AQI levels
- Receive real-time health advisories
- Interpret results using a color-coded AQI scale

### *C. User Roles*

- General Public: Access forecasts and advisories for their local area, understand pollution trends, and take preventive actions.
- Researchers and Authorities: Compare model outputs, export forecasts, and use insights for environmental planning.

### *D. Design Goals*

- Modularity: Each model and module operate independently for easy updates and testing.
- Scalability: Supports future integration with real-time IoT sensor data and cloud-based deployment.
- Accessibility: Runs on any browser, requires no installation, and is mobile-responsive.

## V. IMPLEMENTATION

The system was developed using Python and deployed through a web-based interface. Implementation involved the creation of data pipelines, model training modules, and a Streamlit-based dashboard for real-time user interaction.

### *A. Tools and Technologies*

- Programming Language: Python 3.10
- Development Environment: Jupyter Notebook, VS Code
- Libraries:
  - Data Handling: *Pandas, NumPy*
  - Visualization: *Matplotlib, Seaborn*
  - Modelling: *statsmodels (ARIMA), TensorFlow/Keras (LSTM), Prophet*
  - Deployment: *Streamlit*

### *B. Model Training and Forecasting*

- ARIMA: Implemented using statsmodels, with tuning of (p, d, q) parameters based on AIC values.
- LSTM: Built using TensorFlow/Keras, trained on time-lagged PM2.5 values with sequential input.

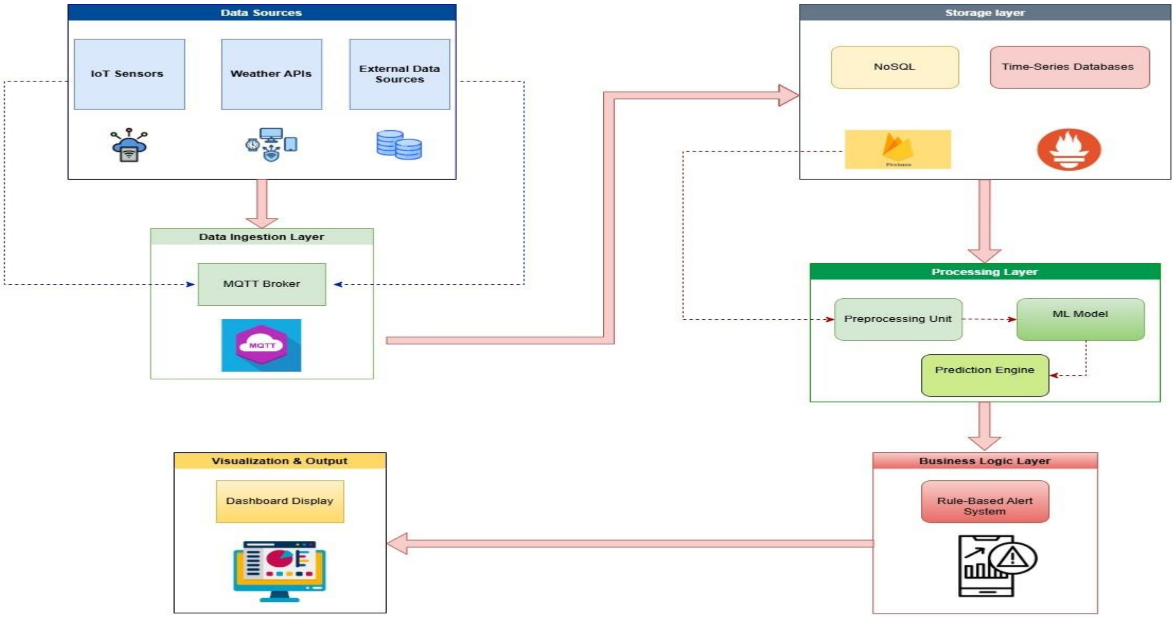


Fig. 1. Architecture of the Air Quality Prediction System showing data flow, model components, and dashboard interface.

- Prophet: Used for fast and interpretable forecasting, ideal for capturing seasonality and missing data.

Each model was trained using a modular notebook pipeline and stored using joblib for reuse during deployment.

### *C. Streamlit Integration*

The user dashboard was created using Streamlit, providing the following features:

- City selection via dropdown
- Real-time AQI prediction
- Line plots for historical vs. predicted data
- Health advisory messages based on AQI thresholds
- Dynamic updates triggered by user input

The app runs locally or can be deployed via Streamlit Cloud for remote access.

### *D. Execution Requirements*

- System Specs: Min. 2GB RAM, i3/i5 processor, stable internet
- Software Dependencies: Listed in a requirements.txt file

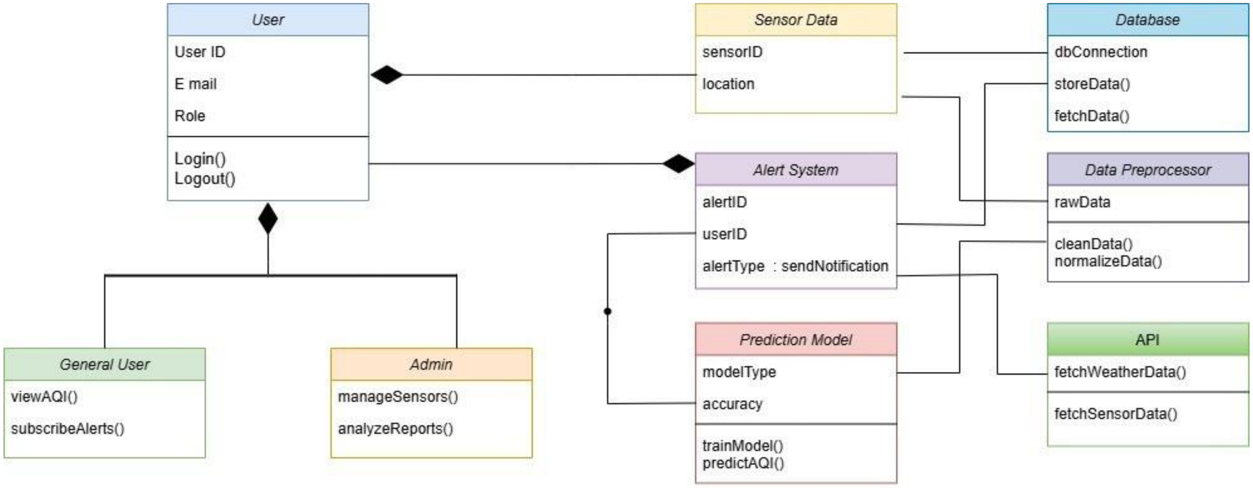


Fig. 2. Process flow diagram outlining AQI forecasting steps, from data ingestion to user interaction.

## VI. RESULTS AND DISCUSSION

The system's performance was evaluated based on the predictive accuracy of each model and the quality of user interaction on the deployed dashboard. Results were analysed through both quantitative metrics and qualitative usability feedback.

### *A. Forecast Accuracy*

To compare model performance, the dataset was divided into training (80%) and testing (20%) splits. The models were evaluated on their ability to forecast PM2.5 concentrations—one of the most critical indicators of air pollution.

Table 1 Performance comparison of forecasting models

| *Model* | *MAE (µg/m$^3$)* | *RMSE (µg/m$^3$)* |
|---|---|---|
| *ARIMA* | *11.23* | *15.85* |
| *LSTM* | *8.74* | *12.56* |
| *Prophet* | *9.31* | *13.62* |

The LSTM model outperformed ARIMA and Prophet in terms of both MAE and RMSE, confirming its ability to capture non-linear patterns and long-term dependencies. However, Prophet required significantly less tuning and training time, making it ideal for rapid deployment scenarios.

### *B. Visualization of Predictions*

The system generates time series graphs comparing predicted and actual PM2.5 values. This allows users to visually evaluate the accuracy of the forecast models. Forecasts for different cities and pollutants can be selected, and trendlines are color-coded for clarity.

Example visual output includes:

- Historical vs. Predicted Graphs: Showing pollutant trends over time.

- Pollutant-Specific Trends: Users can select from PM2.5, PM10, $NO_2$, etc.
- AQI Colour Scale: Based on standard environmental guidelines, helps users interpret forecast severity.

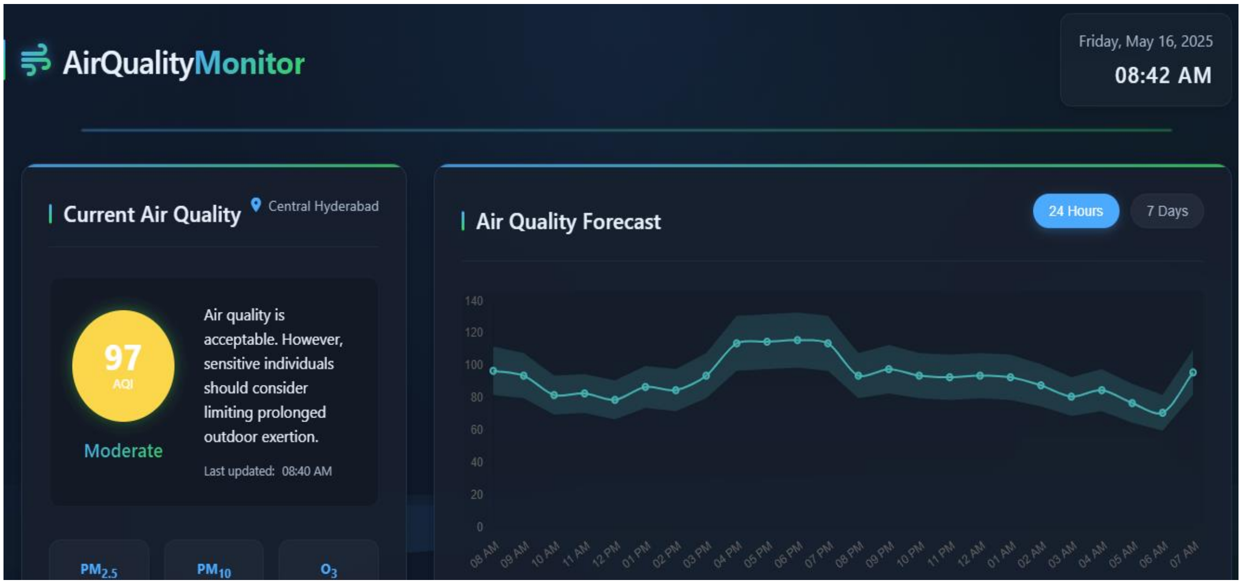
### *C. Health Advisory Logic*

After each prediction, PM2.5 values are mapped to the Air Quality Index (AQI) categories using predefined thresholds:

Table 2 AQI category and health advisory mapping

| *AQI Category* | *PM2.5 Range (µg/m³)* | *Health Advisory* |
|---|---|---|
| *Good* | *0–50* | *Air quality is satisfactory.* |
| *Moderate* | *51–100* | *May cause minor breathing discomfort.* |
| *Unhealthy* | *101–200* | *Sensitive groups should avoid exposure.* |

These advisories are prominently displayed on the dashboard, updating in real time based on model output.



Air quality is acceptable. However, sensitive individuals should consider limiting prolonged outdoor exertion.

Last updated: 08:40 AM

Fig. 3. Dashboard homepage displaying city selection and real-time AQI prediction.

### *D. Streamlit Dashboard Features*

The dashboard provides an interactive frontend for users to engage with forecasts. Key features include:

- Dropdown menus for selecting city and pollutant
- Live updates with each interaction
- Responsive charts for mobile and desktop views
- Color-coded AQI levels with tooltips for better understanding
- Health tips that change dynamically based on AQI output

This seamless integration ensures that users can access insights without needing technical expertise. The dashboard was tested across browsers (Chrome, Firefox, Edge) and devices (laptops, smartphones), with consistent performance and user-friendly operation.

### *E. System Limitations*

While the system performs well in synthetic testing environments, real-world deployment may introduce challenges:

- *Sensor data variability:* Actual IoT-based inputs may have inconsistencies or noise.
- *Scalability:* Handling multiple regions or real-time streaming data may require cloud-based infrastructure.
- *Model generalizability*: The LSTM model may require retraining when applied to different regions or pollutants.

### *F. User Feedback*

The prototype was shared with a small group of students and faculty members. Feedback highlighted:

- High clarity in visualizations
- Fast prediction refresh
- Effective categorization and alerts
- Suggestions to add multi-day forecasts and map views in future versions

## VII. CONCLUSION AND FUTURE WORK

### *A. Conclusion*

This project presents a real-time air quality forecasting system that combines statistical and machine learning models with an interactive web-based interface. By leveraging ARIMA, LSTM, and Prophet, the system effectively predicts pollutant concentrations and categorizes them into actionable Air Quality Index (AQI) levels.

The implementation highlights the LSTM model’s superior performance in handling complex, non-linear time series data, while Prophet offers simplicity and adaptability. The integration of forecasting logic with a user-friendly Streamlit dashboard enables dynamic visualization, city-wise selection, and real-time health advisories for the general public.

Through this modular, scalable approach, the system bridges the gap between raw environmental data and meaningful, real-time insights. It provides value to individuals, researchers, and policymakers seeking timely information for air pollution mitigation and public health preparedness.

### *B. Future Work*

Several enhancements can further improve the system’s effectiveness and deployment readiness:

1. Live IoT Integration: Incorporating real-time sensor data from low-cost air quality monitoring devices will increase forecast precision and enable hyperlocal insights.
2. Cloud Deployment: Moving the application to platforms like AWS, Azure, or Streamlit Cloud will enhance availability, scalability, and ease of access.
3. Advanced Models: Introducing hybrid and ensemble learning models such as BiLSTM, GRU, and CNN-LSTM may offer higher prediction accuracy for multi-pollutant environments.
4. Extended Forecasting: Adding 3-day to 7-day forecasts will allow users to plan outdoor activities and policy responses more proactively.
5. Mobile App and Multilingual Support: A lightweight mobile application with language localization will increase accessibility and public engagement, especially in rural or non-English speaking regions.

By advancing in these directions, the system can evolve into a comprehensive, real-time, and intelligent platform for environmental monitoring and smart city applications.

## REFERENCES

[1] S. Suresh, A. Mehta, and P. Rao, “Forecasting particulate matter emissions using time series models,” *Environ. Res.*, vol. 234, pp. 123456, 2024.

[2] J. Li, X. Huang, and L. Wang, “Prediction and early warning of air quality based on the LSTM-ARIMA model,” *Atmos. Environ.*, vol. 267, pp. 654321, 2023.

[3] R. Garg and A. Jindal, “Evaluation of time series forecasting models for estimation of PM2.5 levels in air,” *Sci. Total Environ.*, vol. 872, pp. 987654, 2023.

[4] O. Gladkova and A. Saychenko, “Applying machine learning techniques in air quality prediction,” *J. Clean. Prod.*, vol. 387, pp. 112233, 2023.

[5] M. Vibhute and V. Dhope, “Air quality prediction in Mumbai city using machine learning-based predictive models,” *Procedia Environ. Sci.*, vol. 45, pp. 445566, 2022.

[6] Y. Qiao, F. Li, and J. Zhang, “Short-term air quality forecasting model based on hybrid RF-IACA-BPNN algorithm,” *J. Clean. Prod.*, vol. 289, pp. 127895, 2021.

[7] T. Uulu and T. Orozakhunov, “Time-series analysis and forecasting of air pollution mortality rates in Central Asian cities,” *Environ. Int.*, vol. 154, pp. 778899, 2021.

[8] R. Kumar, V. Shah, and S. Jain, “Hybridization model for air pollution prediction using time series data,” *Environ. Softw.*, vol. 130, pp. 223344, 2020.