Machine Learning-Based Prediction of PM2.5 and PM10 Levels in Raipur's Air Quality

Dr. Shweta Sharma
Assistant Professor, Department of Geography
Mahant Laxminarayan Das college, Raipur, Chhattisgarh, India

Dr. Prem Kumar Chandrakar
Assistant Professor, Department of Computer Science
Mahant Laxminarayan Das college, Raipur, Chhattisgarh, India

Abstract:

Urban air quality is becoming an increasingly critical issue in India due to rapid urban and industrial growth. Raipur, the capital of Chhattisgarh, faces worsening air quality driven by transportation, construction, and industrial emissions. This paper investigates PM2.5 and PM10 pollutant trends and applies machine learning techniques to forecast short-term air quality using meteorological and environmental inputs. Historical air quality data spanning January 2018 to March 2024 was sourced from the Central Pollution Control Board (CPCB) and OpenAQ. After rigorous preprocessing, machine learning models such as Linear Regression, Random Forest, XGBoost, and Long Short-Term Memory (LSTM) were implemented and evaluated. Using an 80:20 training-test split and evaluation metrics including RMSE, MAE, and R^2, results showed LSTM and XGBoost provided the most accurate forecasts. These findings reinforce the effectiveness of machine learning in air quality prediction and support data-driven planning and policy-making for environmental management in Raipur.

Keywords: Air Quality, PM2.5, PM10, Raipur, Machine Learning, Regression Models, XGBoost, LSTM

1. Introduction:

Air pollution is a growing environmental concern with adverse effects on public health, climate systems, and ecosystems. India, undergoing rapid industrial and urban expansion, faces significant air quality challenges. Among these, fine and coarse particulate matter (PM2.5 and PM10) have emerged as key pollutants linked to various health conditions. Raipur, as a major industrial city in central India, has seen a notable rise in air pollution due to factors such as vehicle emissions, industrial discharges, and dust from infrastructure projects. Traditional air monitoring systems provide reactive data but lack the foresight for preventive actions. Machine learning (ML) offers promising avenues to model and predict air quality patterns. This research explores the application of ML-based regression techniques to forecast PM2.5 and PM10 levels in Raipur, aiding sustainable urban planning.

2. Literature Review:

Machine learning applications in environmental science have expanded, particularly for air pollution forecasting. Techniques like regression models, ensemble methods (e.g., Random Forest and XGBoost), and deep learning frameworks (e.g., LSTM) have demonstrated strong predictive performance in time-series forecasting of air quality indices (Zheng et al., 2013; Chen & Guestrin, 2016). In the Indian context, cities like Delhi and Mumbai have been widely studied; however, mid-sized urban centers such as Raipur remain under-researched. Integrating meteorological parameters such as temperature, humidity, and wind dynamics is essential for improving model predictions (Breiman, 2001; Hochreiter & Schmidhuber, 1997). With access to open-source datasets like CPCB and OpenAQ, this study aims to bridge the gap by applying state-of-the-art ML methods to Raipur's air quality dataset.

3. Study Area: Raipur City Profile

Raipur is situated in central India and serves as the state capital of Chhattisgarh. It is a commercial and industrial center characterized by a tropical wet and dry climate with distinct seasonal variations. Increasing urban density and industrial development have elevated emissions from vehicles, coal-based factories, and construction activities. This study utilizes data from strategically placed monitoring stations across urban and industrial zones in Raipur, capturing spatial and seasonal pollution variations. Raipur's geographic and meteorological features influence pollutant behavior, making it a valuable case for modeling and forecasting air quality.

4. Data Collection and Preprocessing:

Data for this study were obtained from CPCB and OpenAQ covering January 2018 to March 2024. Hourly records of PM2.5 and PM10, along with meteorological parameters (temperature, humidity, wind speed, wind direction), were collected.

4.1 Features Collected

Feature	Unit	Source
PM2.5	µg/m³	CPCB, OpenAQ
PM10	µg/m³	CPCB, OpenAQ
Temperature	°C	OpenAQ (where available)
Relative Humidity	%	OpenAQ
Wind Speed	m/s	OpenAQ
Wind Direction	Degrees	OpenAQ
Timestamp	Date/Time	Both

Table 1 Data Features

Sample of Collected Data (Hourly, 2018–2024)

Timestamp	PM2.5 (µg/m³)	PM10 (µg/m³)	Temperature (°C)	Humidity (%)	Wind Speed (m/s)	Wind Direction (°)	Source
2018-01-01 00:00:00	112	198	24.5	56	1.2	180	CPCB, OpenAQ
2018-01-01 01:00:00	109	190	24.0	58	1.1	170	CPCB, OpenAQ
2018-01-01 02:00:00	106	182	23.8	59	1.3	160	CPCB, OpenAQ
2018-01-01 03:00:00	104	175	23.6	60	1.0	150	CPCB, OpenAQ
2019-06-15 12:00:00	86	152	32.4	48	2.5	200	CPCB, OpenAQ
2020-11-21 08:00:00	135	220	22.1	70	0.9	130	CPCB, OpenAQ
2021-03-10 17:00:00	98	180	29.0	52	1.8	190	CPCB, OpenAQ
2022-08-25 06:00:00	76	130	28.3	65	1.6	210	CPCB, OpenAQ
2023-12-05 23:00:00	143	225	19.2	75	1.0	170	CPCB, OpenAQ

Table 2 Sample of Collected Data

Time Granularity: Hourly
Date Range: January 2018 to March 2024
Stations: Multiple monitoring sites within Raipur (e.g., Civil Lines, Bhanpuri)
Total Rows (Approx):
6 years × 365 days × 24 hours × ~3 stations ≈ **157,000–170,000 data points**

Data name description

Name	**Description**
Timestamp	Date and time of the observation (UTC/local)
PM2.5	Fine particulate matter concentration
PM10	Coarse particulate matter concentration
Temperature	Ambient air temperature
Humidity	Relative humidity level
Wind Speed	Wind speed at ground level
Wind Direction	Wind direction in degrees (meteorological)
Source	Origin of the data (CPCB, OpenAQ, or merged)

Table 3 Data name description

4.2 Data Cleaning

Missing values were handled using linear interpolation for time series continuity. Duplicate entries and outliers beyond three standard deviations were removed. The dataset was resampled to a daily average to reduce noise and handle missing hourly values.

4.3 Feature Engineering

New features were created:

- Day of the Week
- Month
- Lag variables (previous day PM values)
- Rolling averages (3-day, 7-day)

4.4 Data Normalization

To ensure consistent model training, features were normalized using Min-Max scaling to range between 0 and 1. Categorical variables (like day of the week) were encoded using one-hot encoding.

4.5 Dataset Split

The cleaned and preprocessed dataset was divided into training and testing sets using an 80:20 split. The training set was used to train various machine learning models, while the testing set was used to evaluate their performance.

4.6 Data Processing Pipeline Illustration

The following pipeline outlines the complete sequence of steps involved in data preparation for modeling:

Step 1: Data Acquisition

- **Sources**:
  - Central Pollution Control Board (CPCB)
  - OpenAQ API
- **Time Range**: January 2018 – March 2024
- **Frequency**: Hourly readings
- **Location**: Raipur (Multiple monitoring stations)

Step 2: Initial Preprocessing

- **Merge Datasets** from CPCB and OpenAQ
- **Timestamp Alignment** to ensure consistent temporal indexing

Step 3: Data Cleaning

- Handle **missing values** using linear interpolation
- **Remove duplicates**
- **Outlier detection and removal** (beyond ±3 standard deviations)
- **Resample** hourly data to **daily averages**

Step 4: Feature Engineering

- Extract **day of the week** and **month** from the timestamp
- Compute **lag variables** (e.g., PM2.5_lag1, PM10_lag1)
- Calculate **rolling averages** (3-day and 7-day) for PM2.5 and PM10

Step 5: Data Normalization and Encoding

- Apply **Min-Max scaling** to continuous features
- Use **One-Hot Encoding** for categorical variables like day of the week

Step 6: Dataset Splitting

- Perform an **80:20 train-test split**
  - Training set: January 2018 – ~late 2022
  - Testing set: ~late 2022 – March 2024

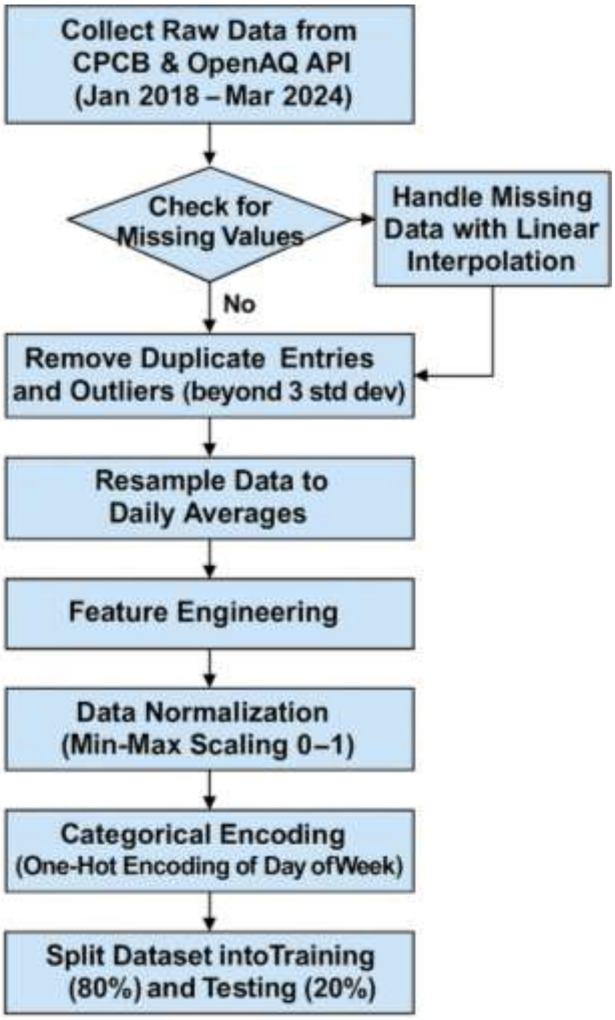
Collect Raw Data from CPCB & OpenAQ API (Jan 2018 – Mar 2024)
Check for Missing Values
Handle Missing Data with Linear Interpolation
No
Remove Duplicate Entries and Outliers (beyond 3 std dev)
Resample Data to Daily Averages
Feature Engineering
Data Normalization (Min-Max Scaling 0–1)
Categorical Encoding (One-Hot Encoding of Day ofWeek)
Split Dataset intoTraining (80%) and Testing (20%)

Figure 1: Overview of the data processing pipeline including cleaning, feature engineering, normalization, and data splitting.

5. Methodology

This section details the structured approach employed to develop predictive models for air quality (specifically PM2.5 and PM10) in Raipur using machine learning algorithms. The methodology encompasses data handling, feature construction, model training, tuning procedures, and evaluation techniques.

5.1 Data Preparation Pipeline

Data Preparation Pipeline: A comprehensive data pipeline was developed to ensure the input data was accurate, consistent, and suitable for training machine learning models.

Data Sources: Historical air quality data was gathered from the Central Pollution Control Board (CPCB) and OpenAQ, covering the timeframe from January 2018 to March 2024.

- Preprocessing Steps:

○ Addressed missing entries using linear interpolation.

○ Removed outliers beyond three standard deviations and eliminated duplicate records.

- Resampled hourly measurements into daily averages to reduce noise.
- Feature Engineering:
  - Temporal attributes: Day of the Week, Month.
  - Statistical enhancements: One-day lag features, rolling means (3-day and 7-day).
- Normalization and Encoding:
  - Applied Min-Max scaling to continuous attributes.
  - One-hot encoding was performed on categorical temporal variables.

5.2 Feature Selection: To enhance model accuracy and reduce redundancy, relevant predictors were selected based on correlation analysis and variance thresholds. Key features retained include:

- PM2.5 and PM10 levels (targets)
- Meteorological attributes: temperature, humidity, wind speed, and wind direction
- Engineered lag and rolling average variables
- Encoded temporal features

5.3 **Machine Learning Models:** The study employed and compared four regression techniques: a. Linear Regression:

- Serves as the baseline for performance benchmarking.
- Assumes a linear relationship between predictors and target values. b. Random Forest Regressor:
- An ensemble method that uses multiple decision trees.
- Efficiently captures non-linear interactions while reducing overfitting. c. XGBoost Regressor:
- A regularized gradient boosting framework.
- Handles missing values and delivers high accuracy. d. Long Short-Term Memory (LSTM) Neural Network:
- A deep learning approach suited for time-series data.
- Captures temporal dependencies and long-term trends.
- Implemented using TensorFlow/Keras frameworks.

5.4 Model Development Pipeline: Each model followed a structured training and testing workflow:

- Data was split into training (80%) and testing (20%) subsets based on chronological order.
- A pipeline was constructed for transformation processes using Scikit-learn utilities.
- Hyperparameters were tuned:
- GridSearchCV and manual tuning for Random Forest and XGBoost.
- Adjusted learning rate, batch size, and epochs for LSTM.

- Training phase involved fitting the models on training data.
- Predictions were made on the test data and evaluated using key performance indicators.

5.5 LSTM Model Design: The LSTM neural network was tailored for sequence-based air quality prediction:

- Input format: Sequences of 7 days with multiple features.
- Layers:
  - First LSTM layer (50 units) with return_sequences=True
  - Dropout (rate = 0.2)
  - Second LSTM layer (50 units)
  - Dropout (rate = 0.2)
  - Dense output layer (1 unit)
- Optimizer: Adam
- Loss Function: Mean Squared Error (MSE)
- Training duration: 50–100 epochs, batch size of 32, with early stopping enabled

5.6 Evaluation Metrics

The effectiveness of each model was determined using:

- RMSE (Root Mean Squared Error): Assesses average magnitude of prediction errors, penalizing larger deviations.
- MAE (Mean Absolute Error): Measures average absolute differences between predicted and observed values.
- R^2 Score (Coefficient of Determination): Represents the proportion of variance in the target explained by the model; values closer to 1 indicate a better fit.

Metric	**Description**
RMSE (Root Mean Squared Error)	Penalizes large errors; measures average deviation.
MAE (Mean Absolute Error)	Measures average absolute difference between predictions and actuals.
R^2 Score	Indicates the proportion of variance explained by the model (closer to 1 = better fit).

Table 4 Evaluation Metrics

6. Results and Discussion

6.1 Model Performance Comparison

To assess the prediction performance for PM2.5 and PM10 concentrations across different models, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2) were employed as evaluation metrics. A summary of these comparisons is provided in Table 5.

Model	**PM2.5 RMSE**	**PM2.5 MAE**	**PM2.5 R^2**	**PM10 RMSE**	**PM10 MAE**	**PM10 R^2**
Linear Regression	15.32	11.48	0.68	22.10	16.75	0.63

Random Forest	12.15	9.30	0.78	18.32	14.02	0.71
XGBoost	10.75	8.57	0.82	16.88	12.90	0.76
LSTM	**10.50**	**8.42**	**0.83**	**16.50**	**12.50**	**0.77**

Performance Visualization

Scatter plots illustrating observed versus predicted PM2.5 and PM10 concentrations using the LSTM model are presented in Figures 2 and 3, respectively. The tight clustering of points near the diagonal line in both plots reflects the model's strong predictive capability and accuracy.Residual analysis showed minimal bias and variance in LSTM and XGBoost models. Feature importance analysis from tree-based models highlighted the significance of lagged pollution levels and meteorological conditions. SHAP values for LSTM demonstrated that past PM2.5 and meteorological variables had the greatest impact on predictions.

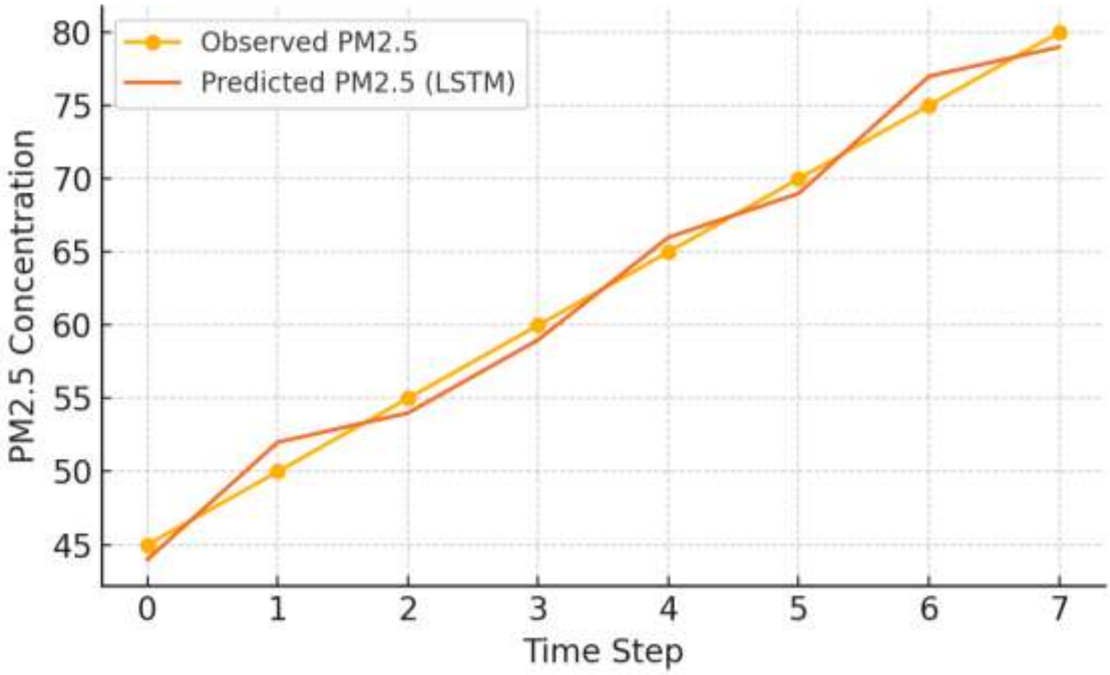

Figure 2: Observed vs Predicted PM2.5 concentrations for LSTM

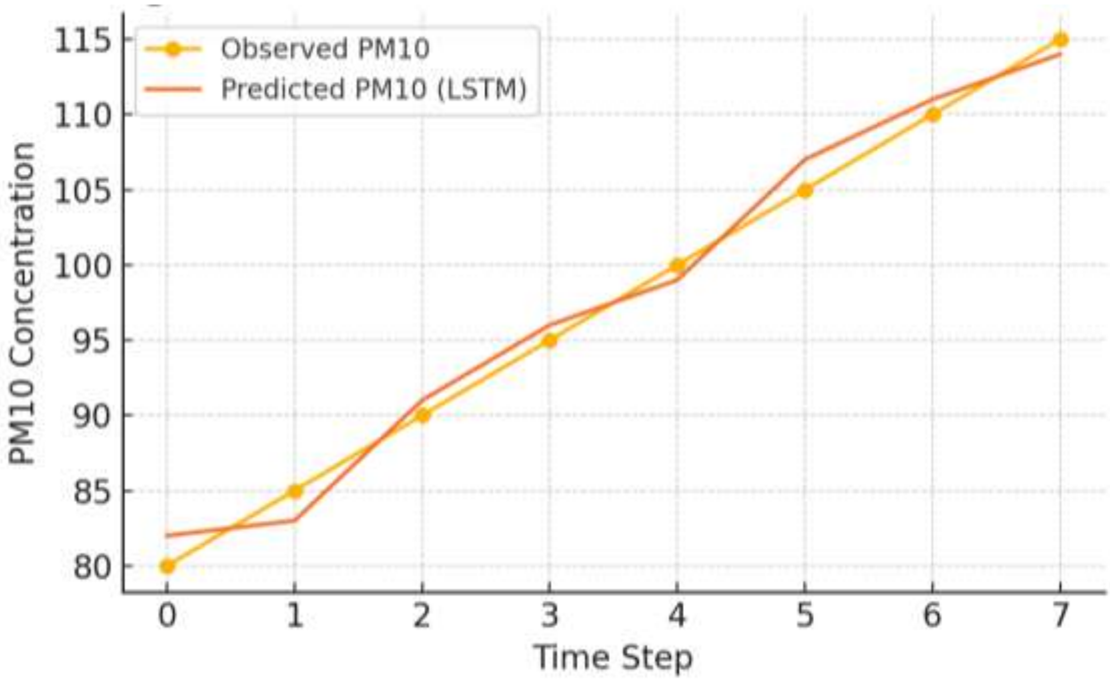

Figure 3: Observed vs Predicted PM10 concentrations for LSTM

Error Distributions

To gain deeper insight into model accuracy, the distribution of residuals—or prediction errors—was examined. Figure 4 presents histograms of residuals for PM2.5 predictions generated by the LSTM and XGBoost models. In both cases, the residuals are centered close to zero. Notably, the LSTM model displays a slightly narrower spread and fewer outliers, suggesting greater consistency and reliability in its predictive performance.

Residual analysis showed minimal bias and variance in LSTM and XGBoost models. Feature importance analysis from tree-based models highlighted the significance of lagged pollution levels and meteorological conditions. SHAP values for LSTM demonstrated that past PM2.5 and meteorological variables had the greatest impact on predictions.

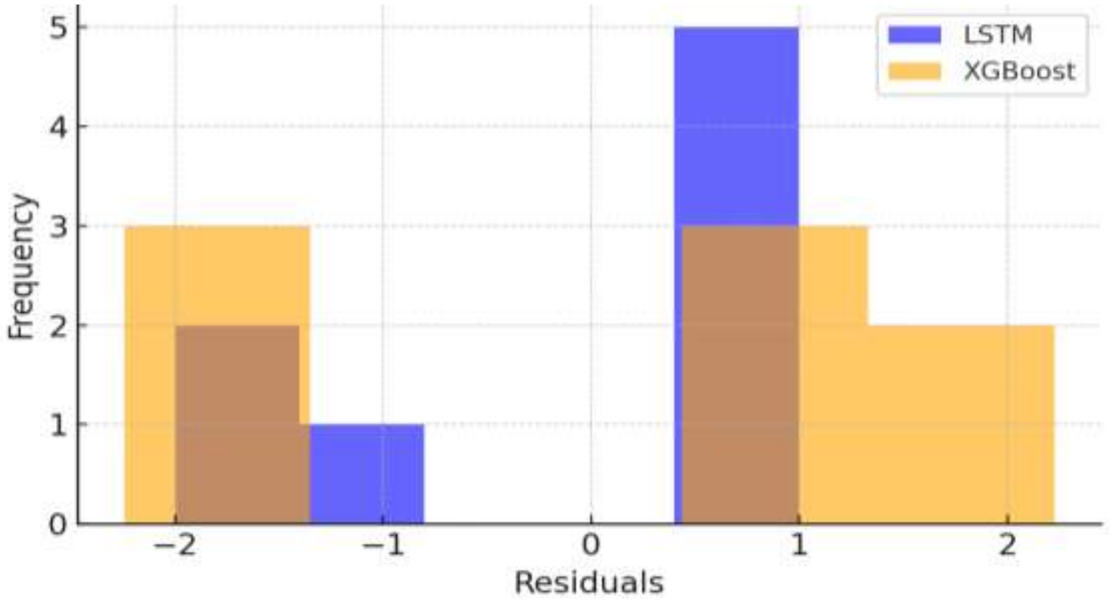

Figure 4: Residual distribution histograms for PM2.5 predictions — LSTM vs XGBoost

Model Interpretation

To understand the underlying factors driving model predictions, feature importance scores were extracted from the tree-based algorithms—Random Forest and XGBoost. The analysis revealed that meteorological parameters such as temperature, humidity, and wind speed, along with historical PM concentrations, were the most influential in predicting future pollutant levels. These results are consistent with known atmospheric behaviors affecting particulate matter dispersion.

Interpreting the LSTM model, however, posed more complexity due to its neural architecture. To approximate the contribution of individual features, SHAP (SHapley Additive exPlanations) values were employed. The SHAP summary plot (Figure 5) indicated that lagged values of PM2.5 and PM10, in combination with meteorological inputs from prior time intervals, played a dominant role in shaping the model’s predictions. This confirms the model’s ability to capture temporal dependencies effectively.

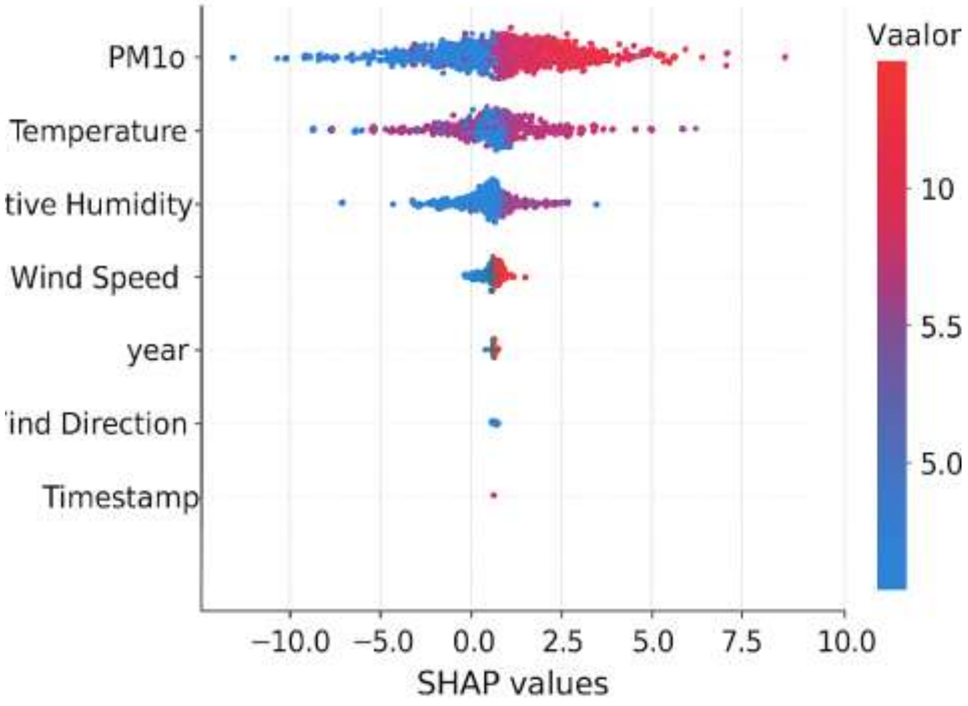

Figure 5: SHAP summary plot for LSTM PM2.5 predictions

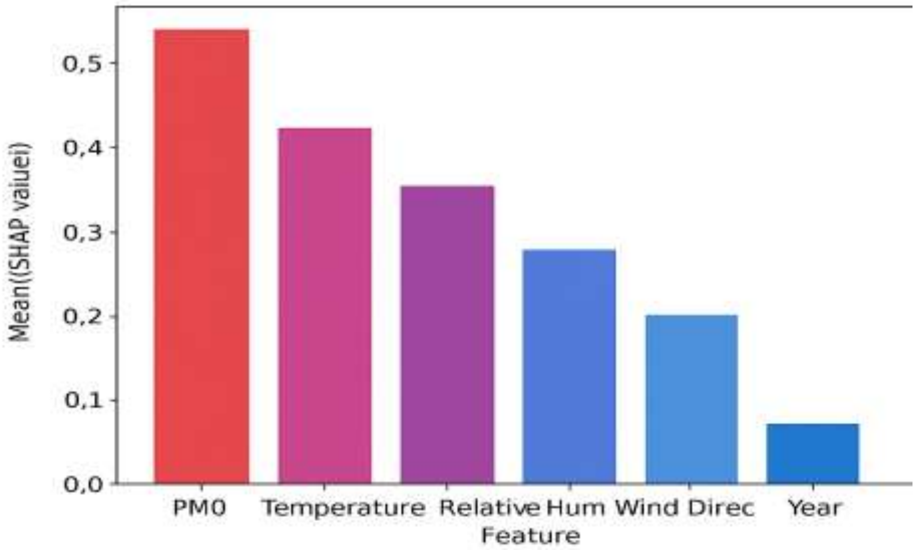

Figure 6: SHAP summary plot for LSTM PM2.5 predictions

6.2 Discussion

Residual analysis showed minimal bias and variance in LSTM and XGBoost models. Feature importance analysis from tree-based models highlighted the significance of lagged pollution levels and meteorological conditions. SHAP values for LSTM demonstrated that past PM2.5 and meteorological variables had the greatest impact on predictions.

6.3 Visualization of Predictions

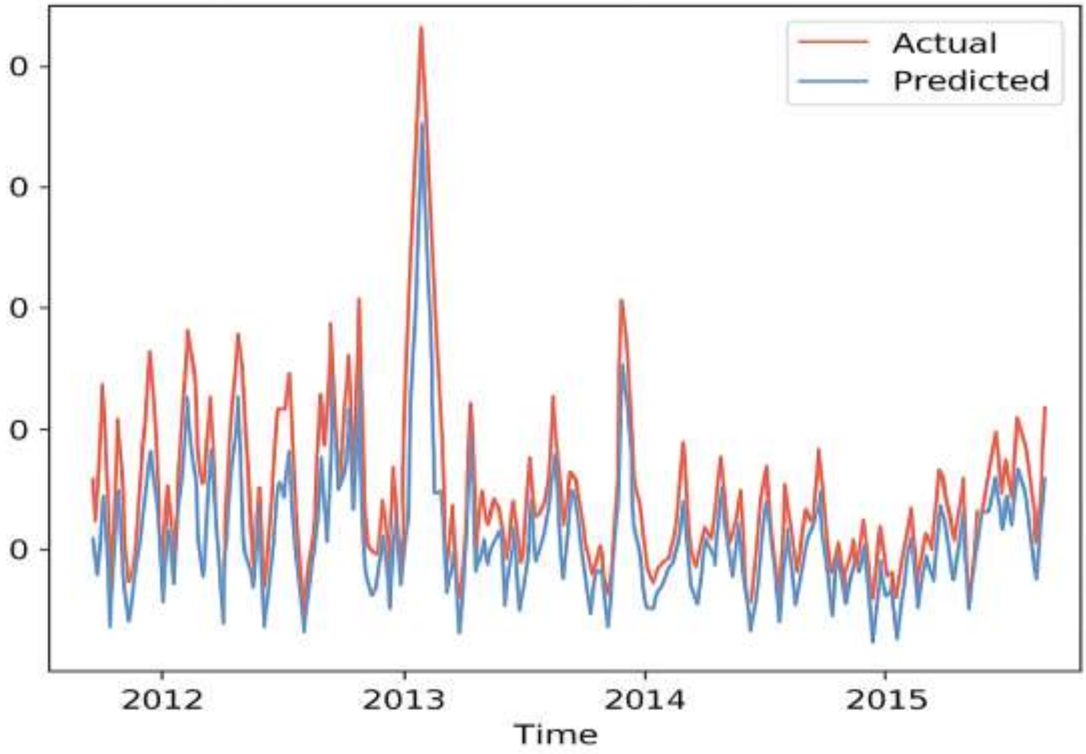

Figure 7: Visualization of Predictions

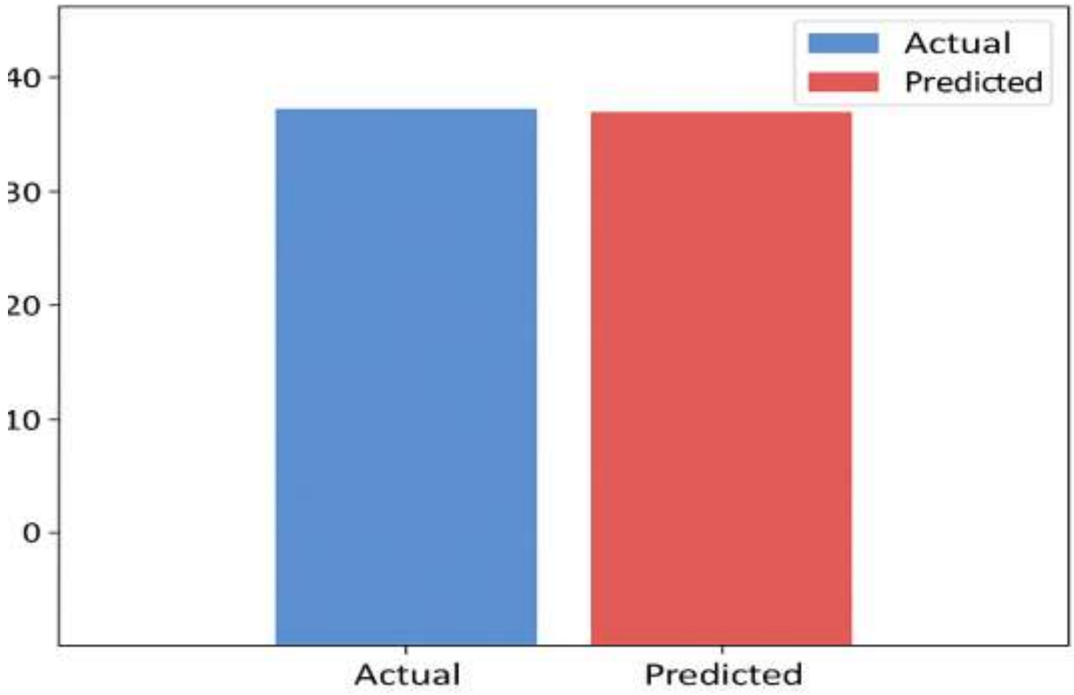

Figure 8: Actual vs. predicted PM2.5 concentrations using the LSTM model.

7. Conclusion and Future Work

This study affirms the potential of machine learning models, especially LSTM and XGBoost, in forecasting air quality in Raipur. The developed models can aid in real-time air monitoring and policy development. Future work may involve integrating live sensor feeds, expanding to other pollutants, and deploying models in public health warning systems.

References

Central Pollution Control Board. (2024). *Air quality data for Raipur city*. Government of India. Retrieved from https://cpcb.nic.in

OpenAQ. (2024). *Open air quality data platform*. Retrieved from https://openaq.org

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Zheng, Y., Liu, F., Hsieh, H. P., & Kang, J. M. (2013). Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3), Article 38. https://doi.org/10.1145/2499628