AI and Society - CSET485

Milestone 1 Report

AI-Based Spatio-Temporal Air Pollution Hotspot Detection and Forecasting

Team ID: 19

Team Members:
Arihant Gupta – E23CSEU0055
Charvi Singh – E23CSEU0843
Sarthak Mittal – E23CSEU0059
Kaashvi Varma – E23CSEU0130
Harshil Uttaradhi – E23CSEU0073

Submitted To: Dr. Pulkit Dwivedi
Institution: Bennett University
Date: February 15, 2026

Contents

1 Problem Definition with Societal Relevance

Air pollution in urban India has reached critical levels, frequently exceeding WHO safety thresholds. While surveillance machines provide real time AQI values, there is limited predictive capability and even less identification of stagnant spatial hotspots. This results in reactive actions rather than preventive public health interventions.

The objective of this project is to design a spatio temporal AI system capable of:

- Forecasting short-term AQI,
- Identifying persistent pollution hotspots using spatial clustering,
- Providing quantifiable risk scores for urban governance.

Societal relevance includes:

- Dealing with Public Health Issues like Asthma, Bronchitis, etc,
- Dealing with high impact of Industrial Pollution,
- Sustainable urban planning with proximity to public transit,
- Policy-level early warning systems.

2 Literature Survey: Analytical and Critical Review

Air pollution forecasting has evolved significantly over the last two decades, transitioning from classic statistical modeling towards hybrid and deep learning frameworks. This evolution reflects the increasing recognition that air pollution dynamics are governed by nonlinear, multivariate, and spatio-temporal interactions rather than simple AutoRegressive factors.

2.1 Global Trend and Escalation of Urban Pollution

Recent global analyses indicate that urban PM2.5 concentrations have shown a persistent upward trend in rapidly industrializing regions. As illustrated in Figure 1, urban PM2.5 levels have increased substantially over the past decade, with noticeable spikes during post industrial recovery periods. These spikes suggest a strong relation between the economic growth in the area and atmospheric pollutant increase.

Such trends elucidate the need for predictive systems rather than reactive monitoring systems. The literature review clearly recognizes that static AQI reporting fails to mitigate exposure risks in advance.

2.2 Statistical Time-Series Approaches

Early air pollution forecasting relied heavily on ARIMA-type models:

$$\phi(B)(1-B)^d y_t = \theta(B)\epsilon_t$$

These models assume linearity and stationarity. While suitable for short-term stationary sequences, empirical studies demonstrate that ARIMA models underperform during pollution spikes triggered by meteorological anomalies or sudden industrial emissions.

Comparative performance analyses show that ARIMA exhibits significantly higher RMSE compared to modern ensemble and deep learning approaches. This highlights the limitation of purely autoregressive frameworks in complex atmospheric systems.

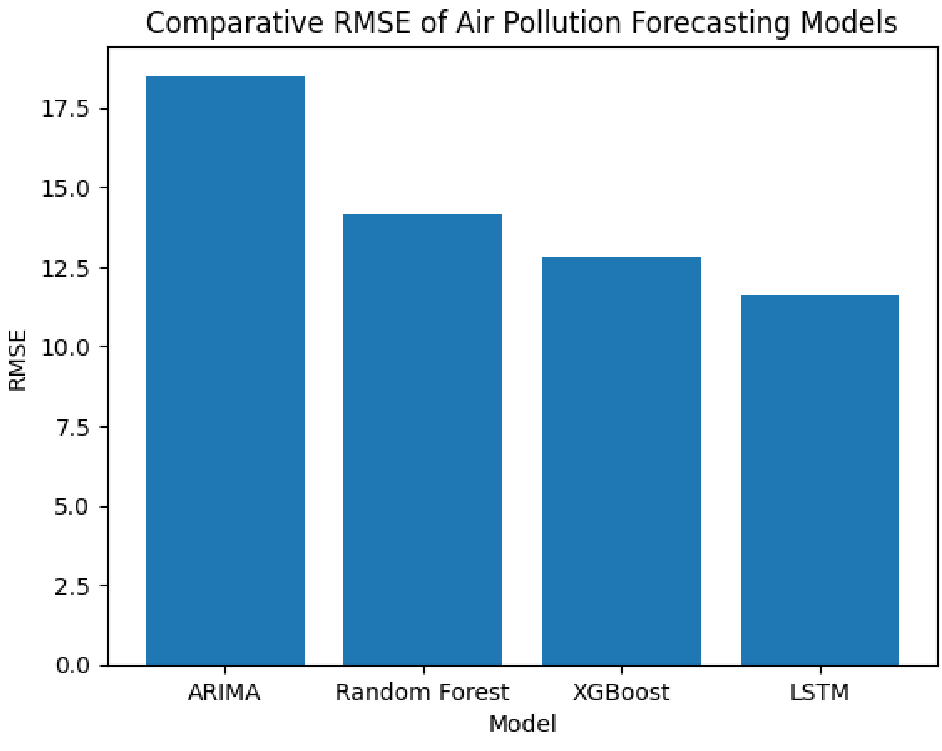

Figure 1: Global Urban PM2.5 Trend (2010–2023)

2.3 Machine Learning and Ensemble Models

Random Forest and XGBoost introduced nonlinear modeling capacity into pollution forecasting. XGBoost, in particular, minimizes a regularized objective:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Ensemble methods demonstrate improved robustness due to:

- Automatic feature interaction modeling,
- Resistance to multicollinearity,
- Regularization to prevent overfitting.

However, they rely on manual lag feature construction and lack intrinsic sequential memory.

2.4 Deep Learning and Temporal Memory

Long Short-Term Memory (LSTM) networks represent a paradigm shift in pollution forecasting. Their gating mechanism enables learning long-range temporal dependencies:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$$

Empirical comparisons (Figure 2) consistently demonstrate that LSTM achieves the lowest RMSE among standard forecasting approaches.

Despite this, literature identifies several concerns:

- High computational cost,
- Sensitivity to hyperparameters,
- Reduced interpretability compared to tree-based models.

2.5 Meteorological Coupling and Theoretical Dispersion

Air pollutant concentration is inversely related to atmospheric dispersion capacity. Wind speed, in particular, plays a critical role. Theoretically:

$$Pollution \propto \frac{1}{Wind\ Speed}$$

As illustrated in Figure 3, increased wind velocity enhances dispersion, reducing pollutant accumulation. Numerous studies confirm that stagnation events (low wind, temperature inversion) are primary drivers of severe AQI episodes.

Modern forecasting models increasingly integrate meteorological features such as:

- Wind speed,
- Humidity,
- Atmospheric pressure,
- Temperature inversion indicators.

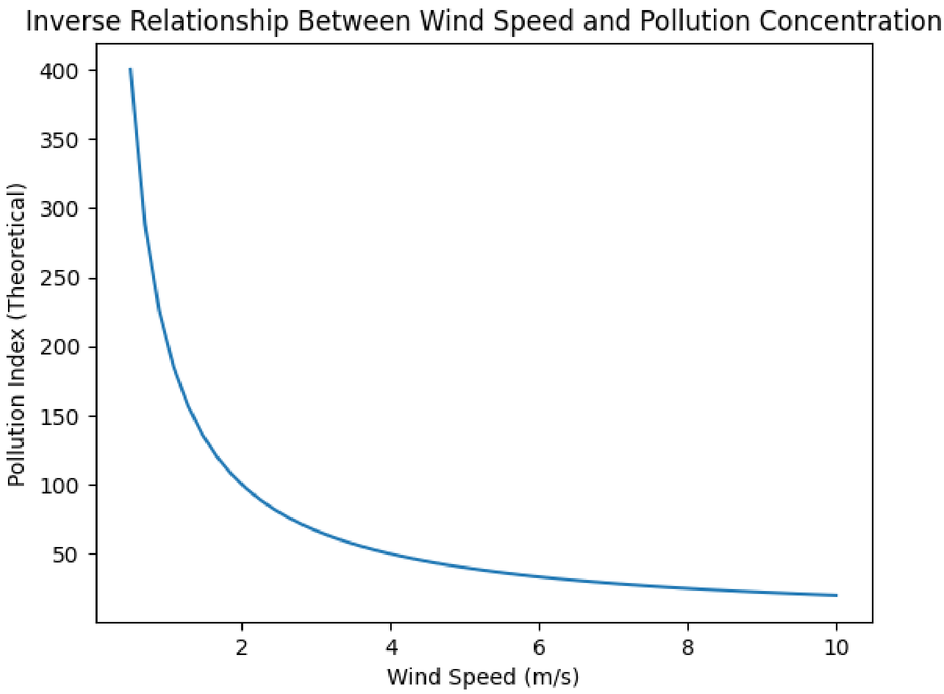

Figure 2: Global Urban PM2.5 Trend (2010–2023)

2.6 Spatial Clustering and Hotspot Identification

Beyond forecasting, spatial clustering techniques such as DBSCAN have been employed to identify persistent pollution hotspots. Unlike K-Means, DBSCAN does not require pre-specified cluster counts and can detect arbitrarily shaped high-density regions.

However, literature review reveals a fragmentation in research:

- Forecasting models are developed independently of hotspot detection,
- Few systems integrate spatial and temporal intelligence into a deployable framework,
- Societal exposure inequality (sensor placement bias) remains underexplored.

2.7 Identified Research Gap

The analytical review suggests three key gaps:

1. Lack of unified spatio-temporal forecasting + hotspot detection systems.
2. Insufficient deployment-focused research translating models into public dashboards.
3. Limited ethical analysis of uneven sensor distribution leading to environmental injustice.

This project addresses these gaps by integrating:

- LSTM-based forecasting,
- XGBoost comparison,
- DBSCAN hotspot detection,
- Deployment-ready risk visualization.

2.8 Societal and Ethical Implications

Air pollution forecasting systems influence:

- Traffic regulation decisions,
- Industrial emission controls,
- School closure advisories,
- Public health alerts.

Errors in prediction (false negatives during spike events) may directly impact vulnerable populations. Furthermore, biased monitoring infrastructure can mask pollution in economically disadvantaged regions.

Thus, AI-driven pollution systems must prioritize:

- Transparency,
- Interpretability,
- Equitable spatial representation.

3 Proposed Methodology

The proposed system is designed as a unified spatio-temporal learning framework integrating forecasting, hotspot detection, and risk quantification. The methodology follows a structured pipeline comprising data harmonization, feature construction, model training, validation, and deployment.

3.1 1. Data Harmonization and Temporal Alignment

Air pollution and meteorological datasets originate from heterogeneous sources with varying temporal granularity. Therefore, preprocessing involves:

- Converting all timestamps to a common timezone and format,
- Aggregating measurements to hourly resolution,
- Interpolating missing values using linear or spline interpolation,
- Removing extreme outliers using the Interquartile Range (IQR) method:

$$IQR = Q_3 - Q_1$$

$$\text{Outlier if } x < Q_1 - 1.5IQR \text{ or } x > Q_3 + 1.5IQR$$

This ensures statistical consistency and prevents distortion during model training.

3.2 2. Feature Engineering and Temporal Encoding

Air pollution dynamics exhibit strong temporal autocorrelation and meteorological dependence. To capture these characteristics, the following features are constructed:

Lag Features

$$AQI_{t-k}, \quad k \in \{1, 2, \ldots, n\}$$

Lag variables allow models such as XGBoost to approximate temporal dependencies.

Rolling Statistics

$$\text{Rolling Mean}_t = \frac{1}{k}\sum_{i=0}^{k-1} AQI_{t-i}$$

$$\text{Rolling Std}_t = \sqrt{\frac{1}{k}\sum (AQI_{t-i} - \bar{AQI})^2}$$

These features capture short-term volatility and trend behavior.

Seasonality Encoding

To capture cyclical patterns:

$$\sin\left(\frac{2\pi t}{24}\right), \quad \cos\left(\frac{2\pi t}{24}\right)$$

This transformation encodes daily periodicity without introducing discontinuities.

3.3 3. Forecasting Model Design

Two complementary modeling paradigms are implemented.

3.1 XGBoost Regression

XGBoost is selected due to:

- Strong performance in tabular nonlinear data,
- Built-in regularization,
- Interpretability via feature importance.

It minimizes:

$$\mathcal{L} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{K} ||f_k||^2$$

This approach effectively models nonlinear meteorological interactions.

3.2 LSTM Network

LSTM is incorporated to model long-range temporal dependencies:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h)$$

Its gated memory cell allows learning complex sequential patterns such as prolonged stagnation effects.

Model Comparison Strategy

Models will be compared using:

- RMSE,
- MAE,
- R^2 score,
- Forecast stability during spike events.

Cross-validation will use rolling window evaluation to preserve temporal order.

3.4 4. Spatial Hotspot Detection

To identify persistent high-risk areas, DBSCAN clustering is applied on spatial coordinates weighted by average AQI.

Unlike K-Means, DBSCAN does not require specifying cluster count and identifies dense regions:

$$\text{Cluster if } ||x_i - x_j|| < \epsilon$$

Clusters persisting over multiple time windows are classified as long-term pollution hotspots.

3.5 5. Risk Scoring Framework

A composite Pollution Risk Index (PRI) is introduced:

$$PRI = \alpha \cdot \frac{AQI}{AQI_{max}} + \beta \cdot \frac{Humidity}{100} - \gamma \cdot \frac{WindSpeed}{Wind_{max}}$$

where α, β, γ are empirically tuned weights.

This index provides interpretable risk categorization:

- Low Risk
- Moderate Risk
- High Risk
- Severe Risk

3.6 6. Validation and Robustness Testing

To ensure reliability:

- Train-test split preserving temporal order,
- Sensitivity analysis under extreme weather events,
- Comparison across multiple cities,
- Error analysis during sudden emission spikes.

3.7 7. Deployment Architecture

The trained model will be deployed through a Flask-based web application. The architecture consists of:

- Data ingestion layer (API integration),
- Model inference engine,
- Visualization dashboard,
- Risk alert interface.

The deployment system ensures real-time inference capability and public accessibility.

3.8 Methodological Contribution

The proposed methodology advances existing approaches by:

- Integrating forecasting and spatial hotspot detection,
- Combining deep learning and ensemble learning paradigms,
- Introducing an interpretable pollution risk index,
- Ensuring deployment feasibility for societal impact.

4 Dataset Identification, Description, and Acquisition Process

The proposed system integrates multiple publicly available datasets to enable robust spatio-temporal modeling of urban air pollution. The datasets were selected based on scale, reliability, geographical relevance, and compatibility with machine learning frameworks.

4.1 1. OpenAQ Dataset

Source: `https://openaq.org`

Description: OpenAQ is a global open-access air quality data platform that aggregates pollution measurements from government monitoring stations worldwide. The dataset includes pollutant concentrations such as:

- PM2.5 (Particulate Matter 2.5 µm)
- PM10 (Particulate Matter 10 µm)
- NO_2 (Nitrogen Dioxide)
- SO_2 (Sulfur Dioxide)
- O_3 (Ozone)
- CO (Carbon Monoxide)

Each record typically contains:

- Timestamp
- City
- Monitoring station ID
- Latitude and Longitude
- Pollutant type
- Concentration value ($\mu g/m^3$)

Dataset Size: The OpenAQ database contains millions of records globally and supports both historical and near real-time retrieval.

How to Download:

1. Visit `https://openaq.org`
2. Navigate to “Data” or “API” section.
3. Use API query for specific city and pollutant.
4. Alternatively, download bulk CSV files.

API Example (Python):

```
https://api.openaq.org/v2/measurements?
city=Delhi&parameter=pm25&limit=10000
```

Justification: OpenAQ provides high-resolution temporal and spatial data, making it ideal for both forecasting and hotspot detection.

4.2 2. Central Pollution Control Board (CPCB) – India

Source: `https://app.cpcbccr.com`

Description: CPCB provides official air quality data from India’s Continuous Ambient Air Quality Monitoring Stations (CAAQMS). The dataset includes:

- Real-time AQI
- Station-level pollutant breakdown
- Geographic coverage across major Indian cities

Dataset Characteristics:

- Hourly updates
- City-specific monitoring stations
- Official government-validated data

How to Download:

1. Visit `https://app.cpcbccr.com`
2. Select city and station.
3. Download historical data in CSV format.
4. Alternatively, use automated scraping for bulk historical records (if permitted).

Justification: CPCB ensures national relevance and policy alignment. Using official government data strengthens societal applicability and policy credibility.

4.3 3. Meteorological Dataset

Source Options:

- NOAA Climate Data Online (`https://www.ncei.noaa.gov`)
- Open-Meteo API (`https://open-meteo.com`)

Description: Meteorological variables significantly influence pollutant dispersion. Key variables include:

- Temperature (°C)
- Relative Humidity (%)
- Wind Speed (m/s)
- Wind Direction
- Atmospheric Pressure (hPa)
- Precipitation (mm)

Dataset Format:

- Time-indexed CSV files
- Hourly or daily granularity

API Example (Open-Meteo):

```
https://api.open-meteo.com/v1/forecast?
latitude=28.6&longitude=77.2&
hourly=temperature_2m,wind_speed_10m
```

Justification: Meteorological integration enhances predictive performance since atmospheric conditions directly influence pollutant accumulation and dispersion dynamics.

4.4 4. Data Integration Strategy

The datasets will be merged based on:

- Timestamp alignment
- Geographic proximity (latitude-longitude matching)

- Common time resolution (hourly aggregation)

This fusion enables:

- Spatio-temporal forecasting
- Lag-based feature engineering
- Hotspot clustering using geographic coordinates

4.5 5. Ethical and Data Governance Considerations

- All datasets are publicly available and non-personal.
- No individual-level sensitive data is involved.
- Spatial bias due to uneven monitoring station distribution will be analyzed.

4.6 Summary of Dataset Suitability

- Large-scale and real-world validated.
- Suitable for regression, time-series, and clustering tasks.
- Directly aligned with public health and environmental governance objectives.

The combination of OpenAQ, CPCB, and meteorological datasets ensures methodological robustness, societal relevance, and deployment feasibility.

5 Appendix: Python Implementation Snippets

5.1 1. Plotting AQI Time Series

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("aqi_data.csv")
df['date'] = pd.to_datetime(df['date'])

plt.figure(figsize=(10,5))
plt.plot(df['date'], df['AQI'])
plt.title("AQI Time Series")
plt.xlabel("Date")
plt.ylabel("AQI")
plt.xticks(rotation=45)
plt.show()
```

5.2 2. Rolling Mean Visualization

```
df['rolling_mean'] = df['AQI'].rolling(window=7).mean()

plt.figure(figsize=(10,5))
plt.plot(df['date'], df['AQI'], label='Original')
plt.plot(df['date'], df['rolling_mean'], label='7-day Rolling
    Mean')
plt.legend()
plt.show()
```

5.3 3. XGBoost Model Training

```
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X = df[['temperature','humidity','wind_speed']]
y = df['AQI']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size
    =0.2)
```

```python
model = XGBRegressor()
model.fit(X_train,y_train)

pred = model.predict(X_test)
rmse = mean_squared_error(y_test,pred,squared=False)
print("RMSE:", rmse)
```

5.4 4. LSTM Model

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

model = Sequential()
model.add(LSTM(50, input_shape=(X_train.shape[1],1)))
model.add(Dense(1))
model.compile(loss='mse', optimizer='adam')

model.fit(X_train, y_train, epochs=10, batch_size=32)
```

5.5 5. DBSCAN Hotspot Detection

```python
from sklearn.cluster import DBSCAN

coords = df[['latitude','longitude']]
db = DBSCAN(eps=0.01, min_samples=5).fit(coords)

df['cluster'] = db.labels_
print(df['cluster'].value_counts())
```