

# Air Quality Index (AQI) Prediction Model Using Time Series Analysis

Armaan Shah  
Ziv Barretto  
Ashoka University

December 2, 2024

## Abstract

This report presents the development of an Air Quality Index (AQI) prediction model utilizing time series analysis. The project encompasses data collection, data cleaning, data distribution and correlation analysis, exploratory data analysis, model training, evaluation, and deployment.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Data Collection</b>	<b>4</b>
2.1	Data Sources . . . . .	4
2.2	Data Structure . . . . .	4
<b>3</b>	<b>Data Cleaning</b>	<b>4</b>
3.0.1	Data Availability by Parameter . . . . .	4
3.0.2	Data Completeness by Year . . . . .	5
3.1	Handling Missing Values . . . . .	5
3.1.1	Primary Method: Hourly Mean Imputation . . . . .	5
3.1.2	Secondary Method: Four-Day Rolling Window Average . . . . .	5
3.1.3	Implementation Sequence . . . . .	6
3.2	Test Run on a Single Site . . . . .	6
3.2.1	Initial Data at CRRI Station . . . . .	6
3.2.2	Data Cleaning on CRRI Station . . . . .	6
3.2.3	Results After Cleaning at CRRI Station . . . . .	7
3.3	Data Cleaning on the Entire Dataset . . . . .	7
3.3.1	Removal of Irrelevant Columns . . . . .	7
3.3.2	Data Availability After Cleaning . . . . .	7
3.4	Addition of Weather Data . . . . .	7

<b>4</b>	<b>Data Distribution and Correlation Analysis</b>	<b>8</b>
4.1	Box Plot Distribution Analysis . . . . .	8
4.1.1	Concentration Ranges and Outliers . . . . .	8
4.2	Correlation Matrix Analysis . . . . .	9
4.2.1	Strong Positive Correlations . . . . .	10
4.2.2	Notable Negative Correlations . . . . .	10
4.2.3	Weather Parameter Influences . . . . .	10
4.3	Environmental Implications . . . . .	10
<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>Feature Engineering</b>	<b>11</b>
2.1	Lagged Features . . . . .	11
2.2	Rolling Window Features . . . . .	11
2.3	Time-based Features . . . . .	11
2.4	Combining Features and Targets . . . . .	11
<b>3</b>	<b>Data Imputation</b>	<b>12</b>
<b>4</b>	<b>Outlier Detection</b>	<b>12</b>
<b>5</b>	<b>Feature Scaling</b>	<b>12</b>
<b>6</b>	<b>Data Splitting</b>	<b>12</b>
<b>7</b>	<b>Evaluation Metrics</b>	<b>13</b>
<b>8</b>	<b>Feature Selection</b>	<b>13</b>
<b>9</b>	<b>Retraining with Selected Features</b>	<b>13</b>
<b>10</b>	<b>Final Evaluation</b>	<b>13</b>
<b>11</b>	<b>AQI Calculation</b>	<b>13</b>
<b>12</b>	<b>Results</b>	<b>14</b>
<b>13</b>	<b>Conclusion</b>	<b>15</b>
<b>1</b>	<b>Introduction</b>	<b>15</b>
<b>2</b>	<b>Data Preprocessing Strategy</b>	<b>15</b>
2.1	Feature Engineering . . . . .	16
2.2	Data Transformation Techniques . . . . .	16
2.3	Comparative Analysis . . . . .	17
<b>3</b>	<b>Transformer Model Architecture</b>	<b>17</b>
3.1	Model Components . . . . .	17
3.2	Why Transformer? . . . . .	17

<b>4</b>	<b>Training Strategy</b>	<b>17</b>
4.1	Loss and Optimization . . . . .	17
<b>5</b>	<b>Evaluation Metrics</b>	<b>18</b>
<b>6</b>	<b>Results</b>	<b>18</b>
<b>7</b>	<b>Conclusion</b>	<b>22</b>
<b>1</b>	<b>Introduction</b>	<b>23</b>
<b>2</b>	<b>Training Results</b>	<b>24</b>
<b>3</b>	<b>Evaluation Results</b>	<b>25</b>
<b>1</b>	<b>Results</b>	<b>28</b>

# 1 Introduction

Air pollution poses significant risks to public health and the environment. The Air Quality Index (AQI) is a standardized indicator used to communicate how polluted the air currently is or how polluted it is forecasted to become. Accurate prediction of AQI levels is crucial for informing the public and aiding policymakers in implementing effective air quality management strategies.

This report outlines the development of an AQI prediction model using time series analysis techniques. The study focuses on data collected from various monitoring stations in Delhi, covering multiple pollutants over several years.

## 2 Data Collection

### 2.1 Data Sources

The dataset was obtained from the Central Pollution Control Board (CPCB) of India and includes measurements from 40 monitoring stations across Delhi from 2017 to 2023. The parameters collected encompass various pollutants and meteorological factors:

- **Pollutants:** PM<sub>2.5</sub>, PM<sub>10</sub>, NO, NO<sub>2</sub>, NO<sub>x</sub>, NH<sub>3</sub>, SO<sub>2</sub>, CO, O<sub>3</sub>, Benzene, Toluene.
- **Meteorological Factors:** Temperature, Humidity, Wind Speed, Wind Direction, Solar Radiation, Pressure.

### 2.2 Data Structure

The raw dataset contained:

- **245,376** records.
- **628** features, including pollutant concentrations and meteorological parameters from different stations.
- Timestamped measurements at 15-minute intervals.

## 3 Data Cleaning

### 3.0.1 Data Availability by Parameter

Data availability analysis:				
	Total Readings	Non-null Count	Null Count	Null Percentage
temperature	8342784.0	4977685.0	3365099.0	40.34
pressure	8588160.0	5125252.0	3462908.0	40.32
benzene	9324288.0	5707334.0	3616954.0	38.79
humidity	9324288.0	6014953.0	3309335.0	35.49
toluene	8588160.0	5546138.0	3042022.0	35.42
nh3	7852032.0	5646538.0	2205494.0	28.09
so2	7852032.0	5650537.0	2201495.0	28.04
co	9569664.0	6966775.0	2602889.0	27.20
pm10	9569664.0	6970477.0	2599187.0	27.16
ozone	9569664.0	7002360.0	2567304.0	26.83
no2	9569664.0	7234808.0	2334856.0	24.40
no	9324288.0	7114124.0	2210164.0	23.70
pm25	9569664.0	7336787.0	2232877.0	23.33
nox	9324288.0	7236747.0	2087541.0	22.39

Figure 1: Data Availability Analysis Before Cleaning

### 3.0.2 Data Completeness by Year

Figure 2 presents the completeness percentages per year.

Data completeness analysis by year:			
	Total Readings	Missing Data (%)	Complete Data (%)
2017	35040.0	84.39	15.61
2018	35040.0	31.54	68.46
2020	35136.0	21.67	78.33
2023	35040.0	21.59	78.41
2019	35040.0	19.92	80.08
2021	35040.0	19.23	80.77
2022	35040.0	18.56	81.44
Year with most incomplete data: 2017			
Missing data percentage: 84.39%			

Figure 2: Data Completeness Analysis by Year Before Cleaning

The year **2017** exhibited the highest percentage of missing data at **84.39%**, indicating significant data quality issues during that period.

## 3.1 Handling Missing Values

The dataset cleaning process employed two sequential imputation methods to handle missing values in the air quality measurements.

### 3.1.1 Primary Method: Hourly Mean Imputation

The first imputation method calculated hourly means using the following formula:

$$\bar{x}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{h,i} \quad (1)$$

Where:

- $\bar{x}_h$  is the mean value for hour  $h$ .
- $n_h$  is the number of non-null measurements for hour  $h$ .
- $x_{h,i}$  represents the  $i$ -th measurement in hour  $h$ .

This calculation was performed for each of the 24 hours in a day, creating a set of reference values for each pollutant parameter. Missing values were then replaced with their corresponding hourly mean.

### 3.1.2 Secondary Method: Four-Day Rolling Window Average

For remaining missing values, we implemented a four-day window average using the formula:

$$\bar{x}_t = \frac{1}{N} \sum_{i=-48}^{48} x_{t+i} \quad (2)$$

Where:

- $\bar{x}_t$  is the imputed value at time  $t$ .
- $N$  is the number of non-null values in the window.
- $x_{t+i}$  represents values at  $i$  time steps before/after  $t$ .
- 48 represents 2 days (in hours) before and after the missing value.

The window spans 96 hours in total, taking measurements from two days before and two days after the missing value.

### 3.1.3 Implementation Sequence

The cleaning process followed this order:

1. Apply hourly mean imputation to all missing values.
2. For any remaining gaps, apply the four-day window average.
3. Record the number of values imputed by each method for quality assessment.

## 3.2 Test Run on a Single Site

Before applying the cleaning methods to the entire dataset, we conducted a test run on a single monitoring station, the **CRRI Mathura Road Delhi** site, to evaluate the effectiveness of our imputation strategies.

### 3.2.1 Initial Data at CRRI Station

Figure 3 shows the initial percentage of missing values for key pollutants at the CRRI station.

CRRI Station Analysis:		
	NaN_Count	NaN_Percentage \
Timestamp	0	0.00
site_103_crri_mathura_road_delhi_co	7279	3.46
site_103_crri_mathura_road_delhi_no	21057	10.01
site_103_crri_mathura_road_delhi_no2	10517	5.00
site_103_crri_mathura_road_delhi_nox	9436	4.49
site_103_crri_mathura_road_delhi_ozone	8642	4.11
site_103_crri_mathura_road_delhi_pm10	14815	7.04
site_103_crri_mathura_road_delhi_pm25	10323	4.91
site_103_crri_mathura_road_delhi_solar_radiation	209025	99.38

Figure 3: CRRI Station Missing Data Before Cleaning

### 3.2.2 Data Cleaning on CRRI Station

We applied the two-step imputation process:

1. **Hourly Mean Imputation:** Replaced missing values with the mean value of that specific hour across all days.
2. **Four-Day Rolling Window Average:** For any remaining missing values, applied a rolling window average considering two days before and after the missing timestamp.

### 3.2.3 Results After Cleaning at CRRI Station

Post-cleaning, the percentage of missing values significantly decreased, as shown in Figure 4.

CRRI Station Analysis:		
	NaN_Count	NaN_Percentage \
Timestamp	0	0.00
site_103_crri_mathura_road_delhi_co	336	0.16
site_103_crri_mathura_road_delhi_no	6820	3.24
site_103_crri_mathura_road_delhi_no2	836	0.40
site_103_crri_mathura_road_delhi_nox	460	0.22
site_103_crri_mathura_road_delhi_ozone	412	0.20
site_103_crri_mathura_road_delhi_pm10	5372	2.55
site_103_crri_mathura_road_delhi_pm25	380	0.18
site_103_crri_mathura_road_delhi_solar_radiation	207744	98.77

Figure 4: CRRI Station Missing Data After Cleaning

The test run confirmed the effectiveness of our cleaning methodology, resulting in a substantial reduction of missing values.

## 3.3 Data Cleaning on the Entire Dataset

Encouraged by the successful test run, we applied the cleaning process to the entire dataset.

### 3.3.1 Removal of Irrelevant Columns

We removed **255 columns** that had excessively high percentages of missing data or were irrelevant to the AQI prediction, retaining **374 columns** for further analysis.

### 3.3.2 Data Availability After Cleaning

Figure 5 summarizes the data availability after the cleaning process.

## 3.4 Addition of Weather Data

To the above, we added the following variables: Cloud Base Height, Evaporation, Surface Pressure, 2M Temperature, Total Cloud Cover, Total Precipitation, u10m, and v10m components of wind speed. This was retrieved from the "ERA5 hourly data on single levels from 1940 to present" dataset. As there were no missing values, no imputation or data filling methods were implemented. The collected data was hourly, while the main dataset was every 15 minutes. Thus, we have duplicated each row 3 times to fill the hour in the main dataset.

Data availability analysis after cleaning:				
	Total Readings	Non-null Count	Null Count	Null Percentage
pm10	8203104.0	7302344.0	900760.0	10.98
nh3	6730752.0	6039444.0	691308.0	10.27
ozone	8203104.0	7456876.0	746228.0	9.10
co	8203104.0	7548476.0	654628.0	7.98
no2	8203104.0	7561148.0	641956.0	7.83
so2	6730752.0	6219916.0	510836.0	7.59
pm25	8203104.0	7617044.0	586060.0	7.14
no	7992768.0	7475124.0	517644.0	6.48
nox	7992768.0	7514140.0	478628.0	5.99

Figure 5: Data Availability Analysis After Cleaning

The cleaned dataset, containing **210,336** rows and **374** columns, served for the subsequent data distribution analysis, exploratory data analysis, and model training phases.

## 4 Data Distribution and Correlation Analysis

### 4.1 Box Plot Distribution Analysis

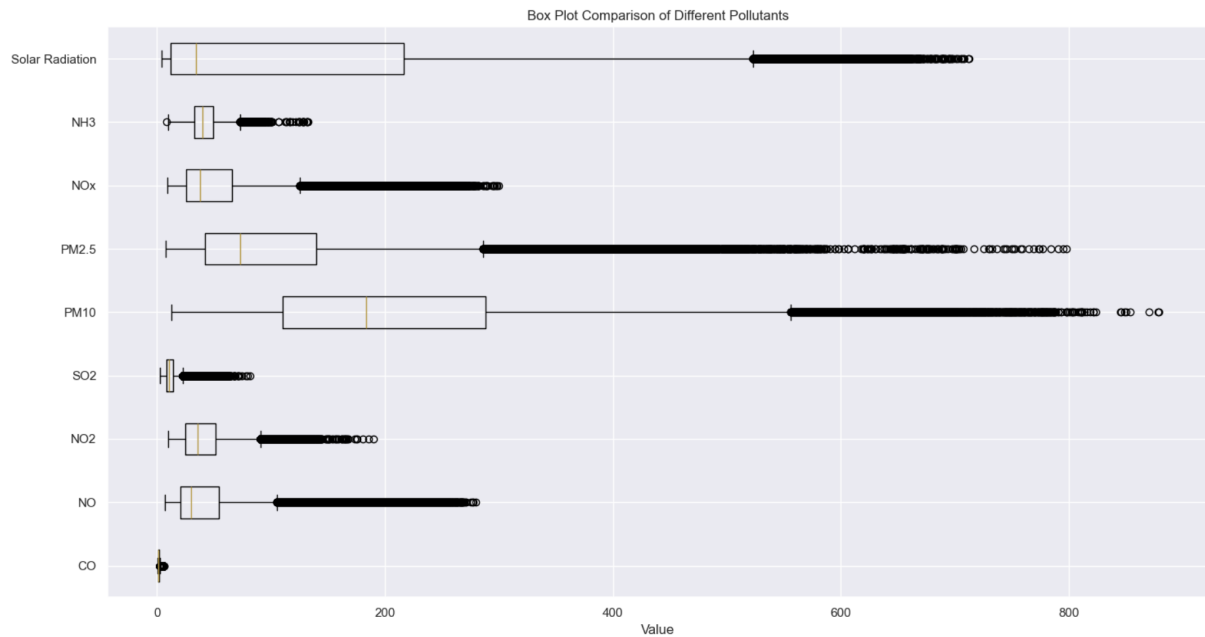


Figure 6: Distribution of Pollutant Concentrations

The box plots in Figure 6 illustrate the distribution of various pollutant concentrations:

#### 4.1.1 Concentration Ranges and Outliers

- **Particulate Matter:**

- **PM<sub>10</sub>** exhibits the widest interquartile range (IQR), spanning approximately 100 to 300 units, with numerous high-value outliers extending beyond 800 units.



- **PM<sub>2.5</sub>** shows a narrower distribution but a significant number of outliers exceeding 600 units.
- Both **PM<sub>10</sub>** and **PM<sub>2.5</sub>** have right-skewed distributions, indicating frequent episodes of severe particle pollution.
- **Gaseous Pollutants:**
  - **CO** has the most compact distribution, concentrated near zero with minimal spread and few outliers.
  - **NO** and **NO<sub>2</sub>** display moderate IQRs with several high-value outliers, suggesting variable emission sources.
  - **SO<sub>2</sub>** maintains relatively low concentrations with occasional spikes, indicating sporadic sources.
- **Other Parameters:**
  - **NH<sub>3</sub>** exhibits a relatively symmetric distribution with moderate outliers.
  - **Solar Radiation** shows a right-skewed distribution due to natural variations between day and night cycles.

## 4.2 Correlation Matrix Analysis

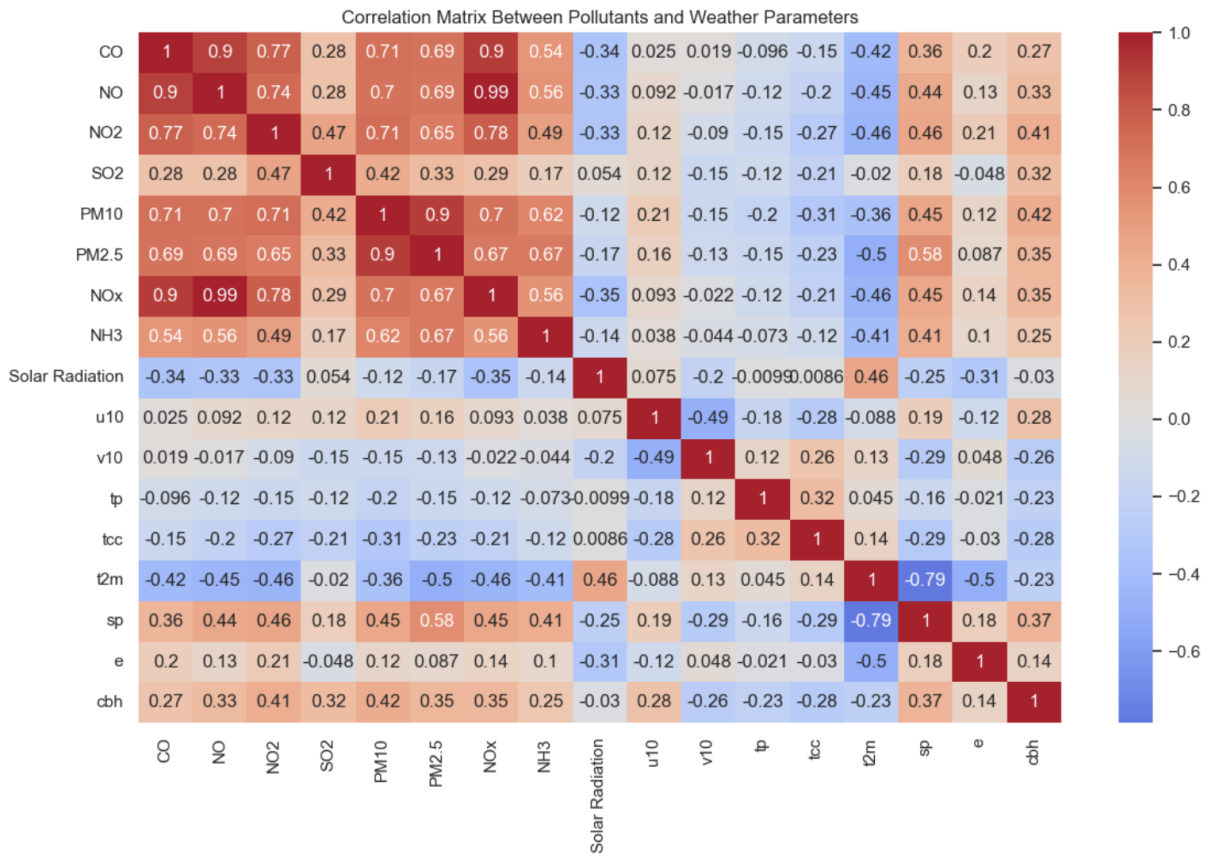


Figure 7: Correlation Matrix Between Pollutants and Weather Parameters

The correlation matrix in Figure 7 highlights the relationships between pollutants and meteorological factors:

#### 4.2.1 Strong Positive Correlations

- **NO and NO<sub>x</sub>** ( $r = 0.99$ ): An almost perfect correlation indicating they share common emission sources, primarily vehicular traffic.
- **CO and NO** ( $r = 0.90$ ): Suggests that these pollutants often originate from similar combustion processes.
- **PM<sub>10</sub> and PM<sub>2.5</sub>** ( $r = 0.90$ ): Reflects the similar behavior and sources of particulate matter.

#### 4.2.2 Notable Negative Correlations

- **Temperature with Pollutants** ( $r = -0.42$  to  $-0.50$ ): Indicates higher pollutant concentrations during colder periods, possibly due to temperature inversions trapping pollutants near the ground.
- **Solar Radiation with NO<sub>x</sub>** ( $r = -0.35$ ): Suggests that increased sunlight may enhance photochemical reactions that reduce NO<sub>x</sub> levels.

#### 4.2.3 Weather Parameter Influences

- **Wind Components** (u10, v10): Generally weak correlations with pollutants, indicating limited direct influence on pollution levels.
- **Surface Pressure** (sp): Shows a moderate positive correlation with NO<sub>2</sub> ( $r = 0.46$ ), suggesting that higher pressure conditions may be associated with pollutant accumulation.

### 4.3 Environmental Implications

- The strong correlations between NO, NO<sub>2</sub>, and NO<sub>x</sub> emphasize the impact of traffic emissions on air quality.
- Negative correlations with temperature and solar radiation highlight the influence of meteorological conditions on pollutant dispersion and chemical reactions.
- The distribution patterns of PM<sub>10</sub> and PM<sub>2.5</sub> suggest frequent occurrences of high particulate matter levels, raising health concerns.

## Model Wise Analysis:

### XGBoost

# 1 Introduction

The following sections detail our attempts at using a XGBoost regression model for the purpose of AQI prediction. Our dataset consists of 10 stations that track CO, NO, NO2, NOX, Ozone, PM10, PM2.5, and Solar Radiation.

In addition to this, we are also tracking Cloud Base Height, Evaporation, Surface Pressure, 2M Temperature, Total Cloud Cover, Total Precipitation, u10m, and v10m components of wind speed. The model was trained on this dataset for 100 iterations, with a depth of 8, and learning rate of 0.05, . This took around 30 minutes on an Nvidia RTX 3060.

The columns are split into features and targets based on the pollutants.

The `Timestamp` column is removed from the features.

## 2 Feature Engineering

Additional features are created to capture temporal dependencies and trends.

### 2.1 Lagged Features

Lagged features are created to include previous observations, by taking the values of a specific column from 1 day, 7 days, 30 days, and 1 year ago.

### 2.2 Rolling Window Features

Rolling mean and standard deviation are calculated over different window sizes to capture local trends:

```
window_sizes = [7, 14, 30]
for target in targets:
    for window in window_sizes:
        data[f'{target}_rolling_mean_{window}'] = data[target].rolling(window).mean()
        data[f'{target}_rolling_std_{window}'] = data[target].rolling(window).std()
```

### 2.3 Time-based Features

Day of the week and month features are extracted to account for seasonal patterns:

```
data['day_of_week'] = data['Timestamp'].apply(lambda x: pd.to_datetime(x).dayofweek)
data['month'] = data['Timestamp'].apply(lambda x: pd.to_datetime(x).month)
```

### 2.4 Combining Features and Targets

All features are combined into X, and targets into y:

```
X = data[features + [f'{target}_t-1' for target in targets] + ... ]
y = data[targets]
```

## 3 Data Imputation

Missing values are imputed using the median strategy:

```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='median')
X_imputed = imputer.fit_transform(X)
y_imputed = imputer.fit_transform(y)
```

## 4 Outlier Detection

An Isolation Forest is used to detect and remove outliers:

```
from sklearn.ensemble import IsolationForest

outlier_detector = IsolationForest(contamination=0.05, random_state=42)
outliers = outlier_detector.fit_predict(X_imputed)
X_imputed = X_imputed[outliers == 1]
y_imputed = y_imputed[outliers == 1]
```

## 5 Feature Scaling

Features are standardized using StandardScaler:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)
```

## 6 Data Splitting

A time-based split is performed to respect the temporal order of data:

```
train_size = int(len(X_scaled) * 0.8)
X_train_full = X_scaled[:train_size]
X_test = X_scaled[train_size:]
y_train_full = y_imputed[:train_size]
y_test = y_imputed[train_size:]

val_size = int(len(X_train_full) * 0.2)
X_train = X_train_full[:-val_size]
X_val = X_train_full[-val_size:]
y_train = y_train_full[:-val_size]
y_val = y_train_full[-val_size:]
```

## 7 Evaluation Metrics

Mean Absolute Percentage Error (MAPE) and R-squared ( $R^2$ ) score are used:

## 8 Feature Selection

Features are selected based on the model's importance scores:

```
from sklearn.feature_selection import SelectFromModel

selector = SelectFromModel(model, threshold="mean", max_features=30)
X_train_selected = selector.fit_transform(X_train, y_train)
X_val_selected = selector.transform(X_val)
X_test_selected = selector.transform(X_test)
```

## 9 Retraining with Selected Features

The model is retrained using the selected features:

```
model.fit(
    X_train_selected, y_train,
    eval_set=[(X_val_selected, y_val)],
    verbose=True,
)
```

## 10 Final Evaluation

The model is evaluated on the test set:

```
y_test_pred = model.predict(X_test_selected)

for i, target in enumerate(targets):
    mape = mean_absolute_percentage_error(y_test[:, i], y_test_pred[:, i])
    r2 = r2_score(y_test[:, i], y_test_pred[:, i])
```

## 11 AQI Calculation

Air Quality Index is calculated using CPCB standards:

The AQI is computed for actual and predicted values, followed by the Mean Squared Error (MSE)

## 12 Results

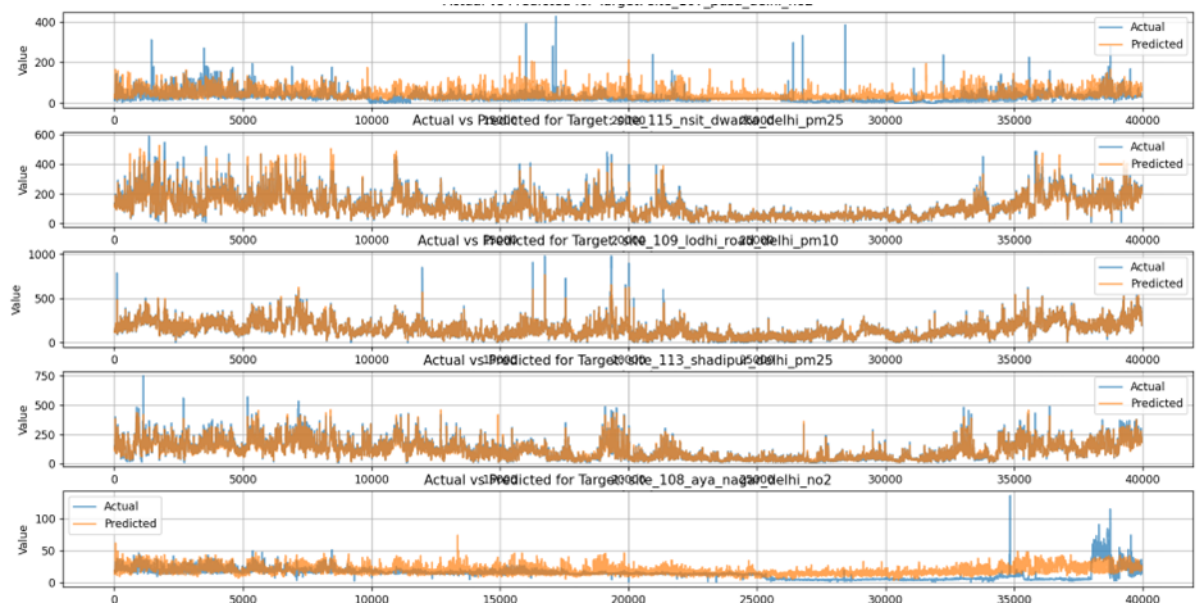


Figure 8: Actual Vs Predicted Features

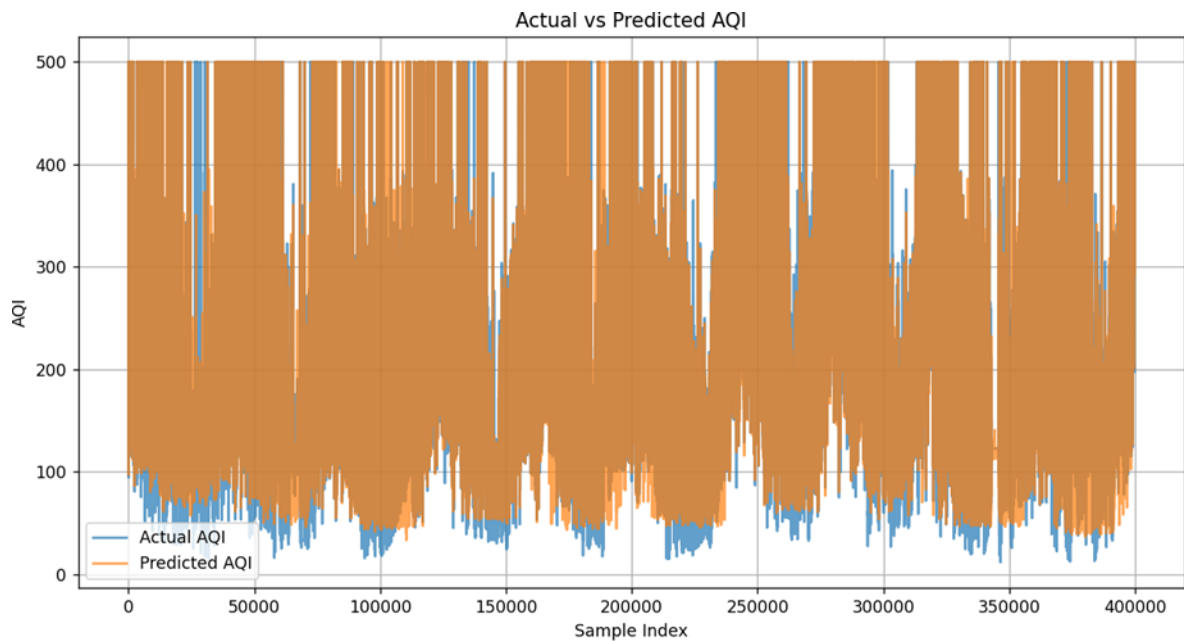


Figure 9: Actual Vs Predicted AQI

MSE for AQI prediction: 1169.9296

Location	Pollutant	MAPE (%)	R <sup>2</sup> Score
CRRRI Mathura Road	CO	9.62e+09	-0.3195
	NO	993.03	-1.1645
	O	119.14	0.0052
	PM	19.00	0.9560
	PM.	24.30	0.9349
Burari Crossing	CO	1.05e+11	-0.1857
	NO	190.08	-0.2535
	O	202.46	-0.2245
	PM	28.46	0.7847
	PM.	56.27	0.9184
North Campus DU	CO	3.71e+10	-0.1780
	NO	254.44	-0.6648
	O	209.78	-0.1198
	PM	15.19	0.9597
	PM.	21.60	0.9697
IGI Airport (T3)	CO	7.66e+09	-0.0647
	NO	313.36	-5.2892
	O	214.87	-0.0099
	PM	9.97	0.9648
	PM.	20.36	0.9720
PUSA	CO	8.12e+09	-0.0276
	NO	137.58	-0.0414
	O	221.73	-0.3936
	PM	13.46	0.9520
	PM.	32.24	0.9306
Aya Nagar	CO	3.36e+09	0.0495
	NO	96.12	-0.3306
	O	135.72	0.0067
	PM	13.56	0.9635
	PM.	33.45	0.9339
Lodhi Road	CO	9.61e+09	0.1840
	NO	88.83	-0.0922
	O	70.98	0.0885
	PM	15.97	0.9677
	PM.	34.98	0.9724

Table 1: Performance metrics for air quality prediction models at key Delhi monitoring stations

## 13 Conclusion

The XGBoost regression model effectively predicts the air quality indices by leveraging extensive feature engineering and proper data preprocessing techniques. Feature selection further enhances the model’s performance by reducing overfitting and improving generalization.

# Transformer

## 1 Introduction

The following sections detail our attempts at using a Transformer model for the purpose of AQI prediction. Our dataset consists of 10 stations that track CO, NO, NO<sub>2</sub>, NO<sub>x</sub>, Ozone, PM<sub>10</sub>, PM<sub>2.5</sub>, and Solar Radiation.

In addition to this, we are also tracking Cloud Base Height, Evaporation, Surface Pressure, 2M Temperature, Total Cloud Cover, Total Precipitation, u<sub>10m</sub>, and v<sub>10m</sub> components of wind speed. The model was trained on this dataset with a batch size of 32, for 50 epochs. This took around 4 hours on an Nvidia RTX 3060.

## 2 Data Preprocessing Strategy

Effective data preprocessing is critical for developing a robust air quality prediction model. Our strategy incorporates feature engineering, imputation, normalization, and careful train-test splitting. Below, we detail each step and provide mathematical justification for the choices made while contrasting alternative methods.

## 2.1 Feature Engineering

The preprocessing pipeline constructs features and targets that capture temporal dependencies and integrate relevant data modalities:

- **Lagged Features:** Given a lookback period of 96 time intervals (approximately 1 day for data of 15-minute resolution), lagged pollutant measurements are incorporated to provide historical context for the prediction task. Formally, for a pollutant  $P_t$ , we construct lagged feature  $P_{t-96}$ . This approach models the time-series dependency:

$$\text{Features at time } t = \{P_{t-i}\}.$$

This method outperforms alternative techniques like rolling averages, which may smooth out critical fluctuations essential for understanding temporal trends.

- **Multiple Pollutant Tracking:** Pollutants such as  $\text{PM}_{2.5}$ ,  $\text{PM}_{10}$ , CO, NO,  $\text{NO}_2$ ,  $\text{NO}_x$ , Ozone, and  $\text{SO}_2$  are treated as interrelated. By including all these pollutants as features, we leverage cross-pollutant correlations:

$$\mathbf{X}_t = [P_t^{(1)}, P_t^{(2)}, \dots, P_t^{(n)}],$$

where  $P_t^{(i)}$  represents pollutant  $i$  at time  $t$ . Alternative approaches, such as predicting pollutants independently, fail to exploit these interdependencies, leading to reduced prediction accuracy.

- **Weather Integration:** Weather parameters (e.g., temperature, humidity, wind speed) are crucial for predicting pollutant dispersion and reaction rates. Thus, we append weather variables  $W_t$  to the feature set:

$$\text{Feature vector at time } t = [P_t, W_t].$$

Ignoring weather data, as in simplistic univariate time-series models, results in poor generalization, especially under changing meteorological conditions.

## 2.2 Data Transformation Techniques

1. **Imputation:** Missing data in pollutant and weather measurements are common. We use mean imputation:

$$\hat{x}_i = \frac{1}{N} \sum_{j=1}^N x_j, \quad \text{if } x_i \text{ is missing.}$$

Alternatives such as median or mode imputation fail to consider the distribution symmetry present in our dataset, and advanced methods like k-nearest neighbors (KNN) are computationally expensive for the size of dataset we are working with.

2. **Normalization:** StandardScaler is applied to transform features and targets to zero mean and unit variance:

$$\hat{x} = \frac{x - \mu}{\sigma}.$$

This standardization ensures numerical stability in model training, particularly for the transformer architecture, which can be sensitive to unscaled inputs. Alternatives like Min-Max scaling can compress data, losing the contribution of outliers, which are often critical in air quality data.



3. **Train-Test Split:** An 80%-20% train-test split ensures a sufficient training set for learning patterns while preserving a large enough test set for evaluation. A more imbalanced split (e.g., 90%-10%) risks overfitting, while stratified sampling is unnecessary here as pollutant data is not categorical.

## 2.3 Comparative Analysis

Our methodology is optimized for air quality prediction tasks, balancing computational efficiency and predictive power. By incorporating historical pollutant data, weather conditions, and robust preprocessing techniques, the model is well-positioned to outperform simpler models such as univariate ARIMA or naive baselines that ignore weather data and cross-pollutant relationships.

# 3 Transformer Model Architecture

## 3.1 Model Components

The Transformer model consists of three primary layers:

- **Embedding Layer:** Linear transformation reducing input dimension to 32
- **Transformer Encoder:** Self-attention mechanism with configurable:
  - Number of heads (default: 4)
  - Number of layers (default: 2)
  - Dropout rate (default: 0.2)
- **Output Layer:** Linear layer mapping to pollutant prediction dimensions

## 3.2 Why Transformer?

Advantages over traditional models:

- Captures complex non-linear relationships
- Handles multiple input features simultaneously
- Inherent attention mechanism learns intricate temporal dependencies

# 4 Training Strategy

## 4.1 Loss and Optimization

- **Loss Function:** Mean Absolute Error (MAE)
- **Optimizer:** Adam with weight decay
- **Learning Rate Schedule:** Step learning rate reduction

## 5 Evaluation Metrics

Multiple metrics provide comprehensive model assessment:

- Mean Absolute Percentage Error (MAPE)
- Symmetric Mean Absolute Percentage Error (SMAPE)
- Comprehensive AQI Calculation
- Confusion Matrix for AQI Categories

## 6 Results

Site	MSE	MAPE (%)	SMAPE (%)
site_103_crri_mathura_road_delhi_co	3.1706	27212412.00	68.28
site_103_crri_mathura_road_delhi_no	5520.9951	484.99	70.01
site_103_crri_mathura_road_delhi_no2	734.6662	452.25	51.30
site_103_crri_mathura_road_delhi_nox	5129.6509	174867792.00	57.44
site_103_crri_mathura_road_delhi_ozone	524.4279	69.61	45.55
site_103_crri_mathura_road_delhi_pm10	14340.3438	117.41	44.01
site_103_crri_mathura_road_delhi_pm25	4091.4817	84.93	44.79
site_104_burari_crossing_delhi_co	0.8770	67661792.00	44.19
site_104_burari_crossing_delhi_no	2156.2156	60.71	44.89
site_104_burari_crossing_delhi_no2	730.9974	147.60	64.90
site_104_burari_crossing_delhi_nox	2138.7451	110942304.00	39.09
site_104_burari_crossing_delhi_ozone	775.6478	156.66	65.76
site_104_burari_crossing_delhi_pm10	17056.8926	93.39	43.35
site_104_burari_crossing_delhi_pm25	5646.0664	116.27	52.24
site_105_north_campus_du_delhi_co	4.7776	620704704.00	73.41
site_105_north_campus_du_delhi_no	4952.4858	288.57	96.08
site_105_north_campus_du_delhi_no2	305.5740	131.49	43.07
site_105_north_campus_du_delhi_nox	5008.7295	37507180.00	60.33
site_105_north_campus_du_delhi_ozone	221.2104	104.49	41.18
site_105_north_campus_du_delhi_pm10	10989.6709	88.25	39.07
site_105_north_campus_du_delhi_pm25	6469.1362	96.30	48.02
site_106_igi_airport_(t3)_delhi_co	1.9693	11630871.00	46.17
site_106_igi_airport_(t3)_delhi_no	8169.2842	74.39	79.54
site_106_igi_airport_(t3)_delhi_no2	168.4816	92.61	35.42
site_106_igi_airport_(t3)_delhi_nox	7520.1978	4905972.50	51.87
site_106_igi_airport_(t3)_delhi_ozone	150.3093	74.31	46.70
site_106_igi_airport_(t3)_delhi_pm10	14275.5889	54.60	40.58
site_106_igi_airport_(t3)_delhi_pm25	5256.0942	88.59	48.71
site_107_pusa_delhi_co	2.3300	72739560.00	38.41
site_107_pusa_delhi_no	818.3304	78.07	62.57
site_107_pusa_delhi_no2	461.5032	49.56	40.94
site_107_pusa_delhi_nox	1893.2133	1684597.88	40.78
site_107_pusa_delhi_ozone	123.3067	76.26	34.13

Site	MSE	MAPE (%)	SMAPE (%)
site_107_pusa_delhi_pm10	9742.3115	64.79	44.39
site_107_pusa_delhi_pm25	4608.8643	111.18	56.59
site_108_aya_nagar_delhi_co	1.1014	144262736.00	33.18
site_108_aya_nagar_delhi_no	367.7643	74.53	63.66
site_108_aya_nagar_delhi_no2	31.7549	33.07	25.96
site_108_aya_nagar_delhi_nox	349.9889	3759977.25	25.22
site_108_aya_nagar_delhi_ozone	331.3746	91.24	32.92
site_108_aya_nagar_delhi_pm10	12953.5254	66.12	43.07
site_108_aya_nagar_delhi_pm25	4191.9048	109.66	52.81
site_109_lodhi_road_delhi_co	0.7940	10424999.00	37.89
site_109_lodhi_road_delhi_no	1843.8599	119.16	69.52
site_109_lodhi_road_delhi_no2	172.0291	59.98	35.98
site_109_lodhi_road_delhi_nox	1736.7333	93551216.00	47.37

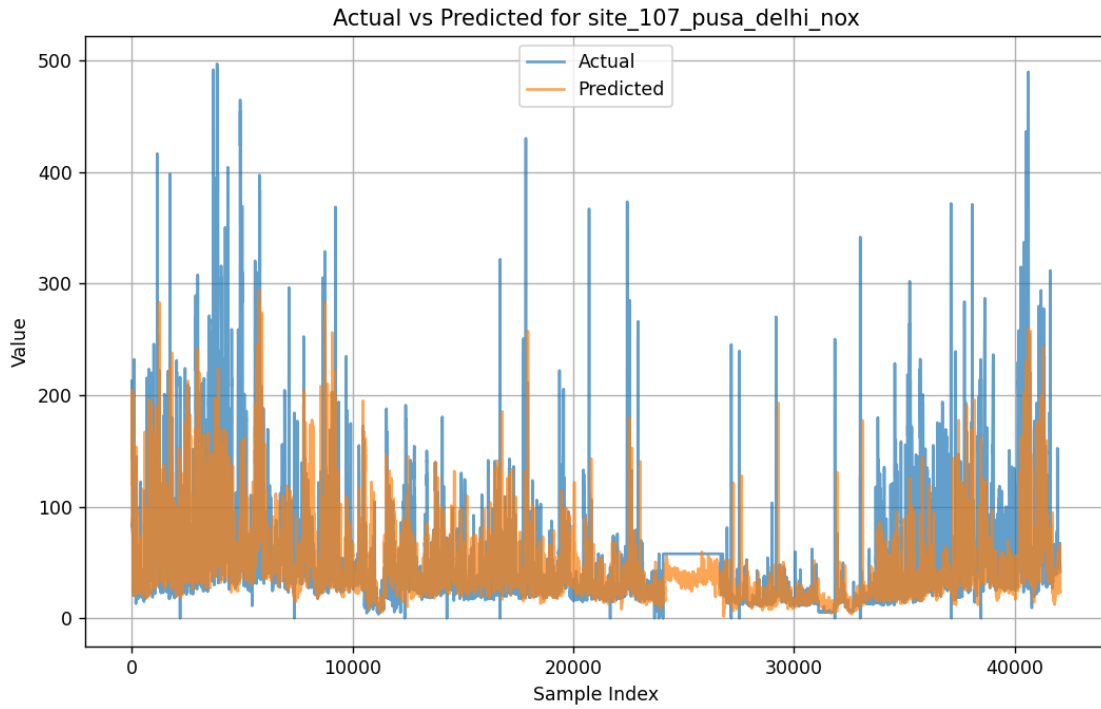


Figure 10: Feature Data Comparison 1

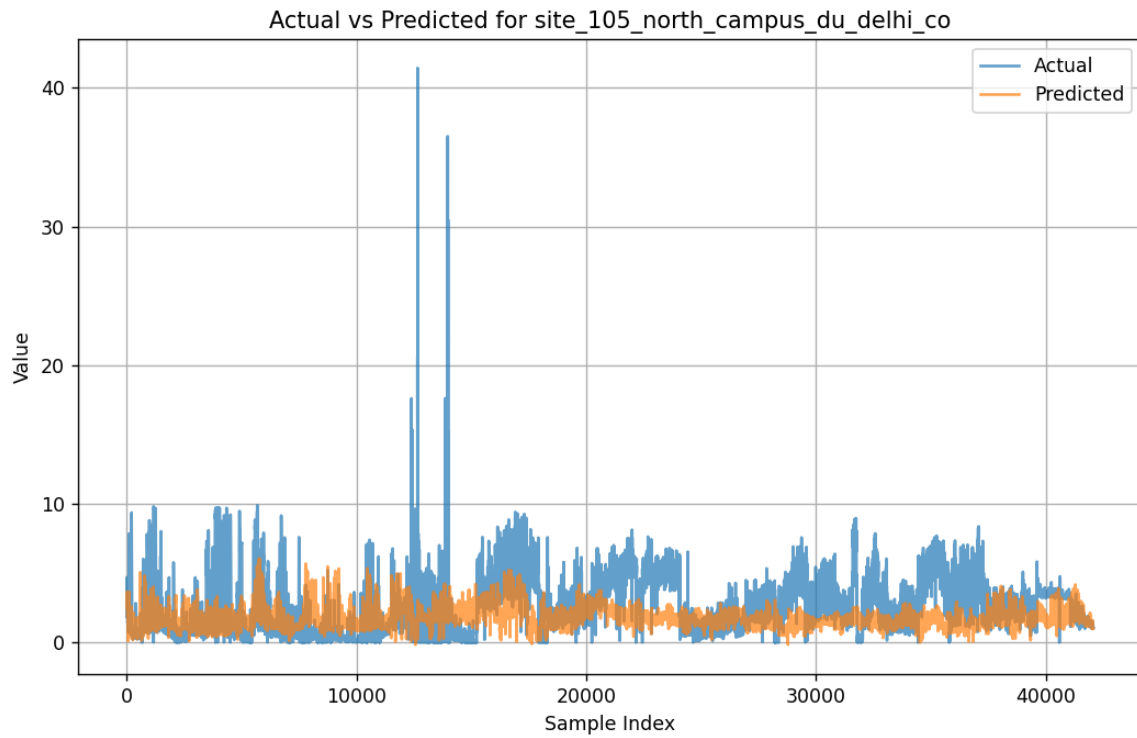


Figure 11: Feature Data Comparison 2

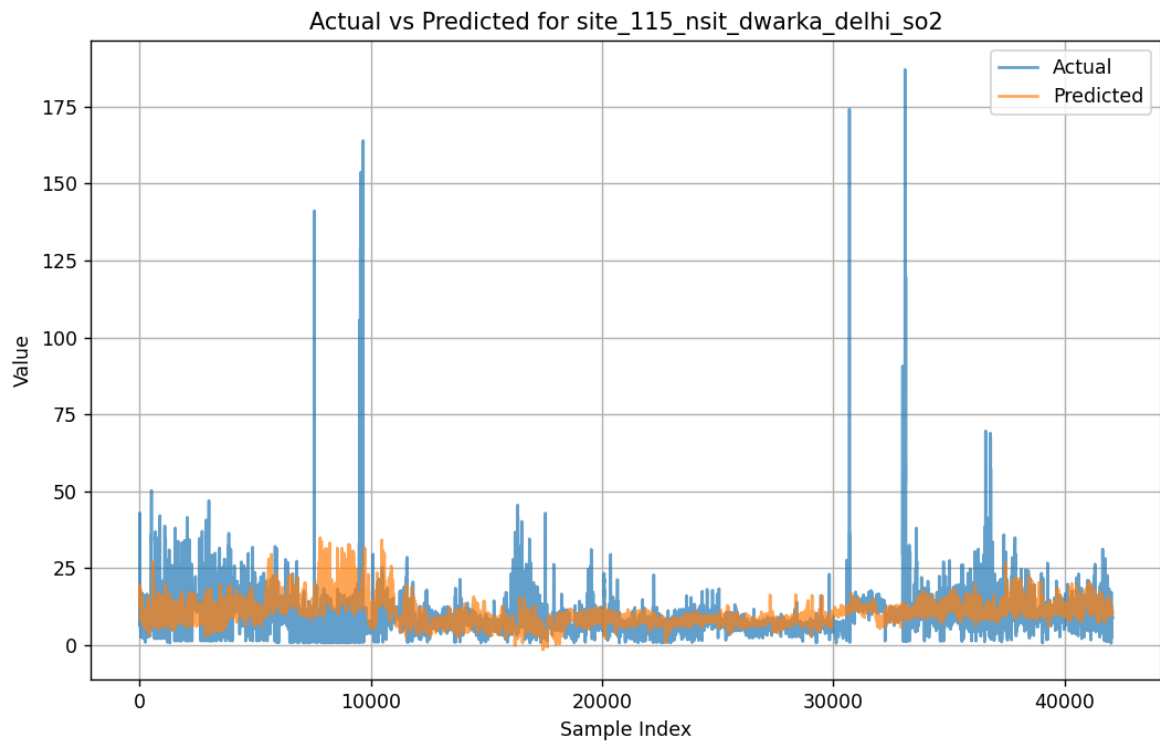


Figure 12: Feature Data Comparison 3

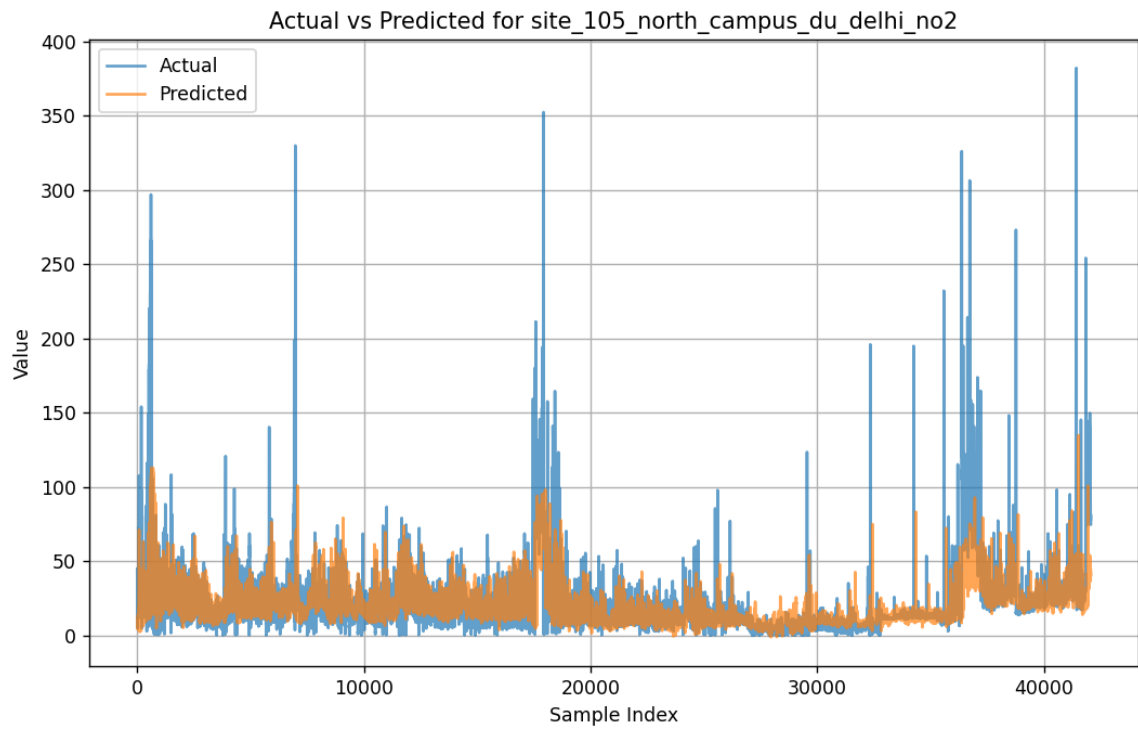


Figure 13: Feature Data Comparison 4

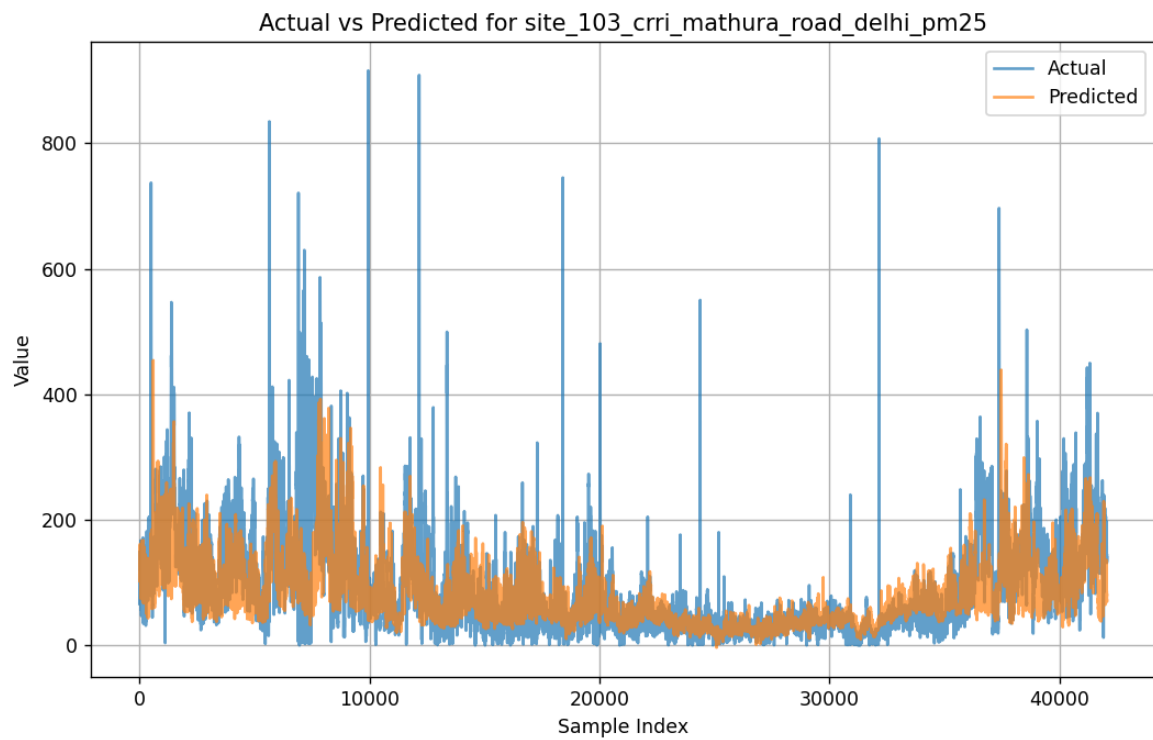


Figure 14: Feature Data Comparison 5

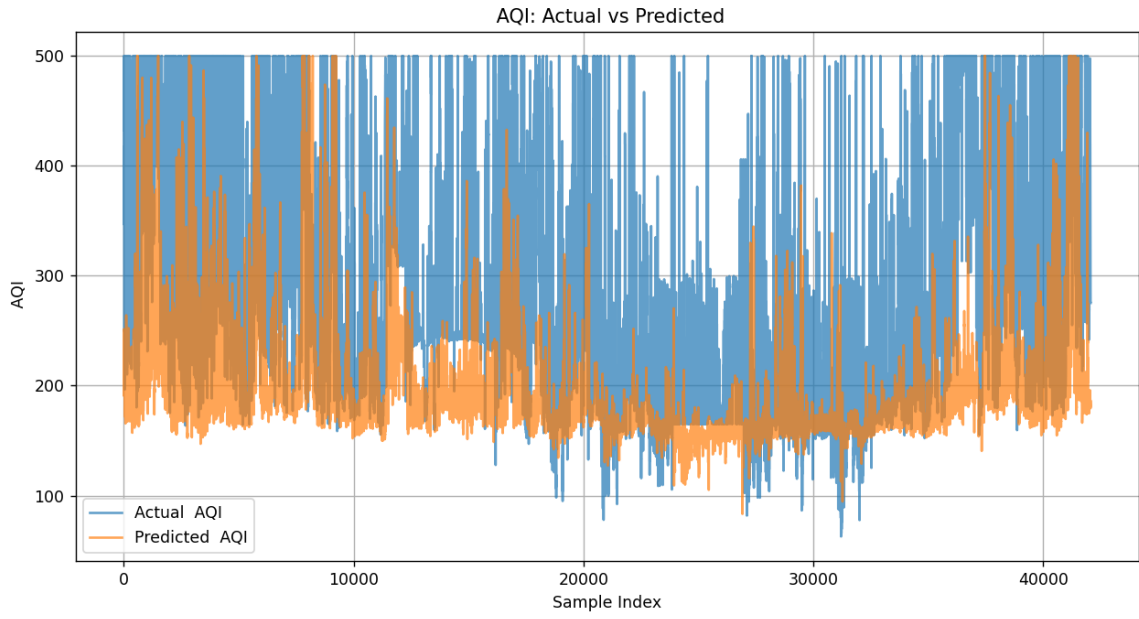


Figure 15: Calculated AQI Actual vs Predicted Values

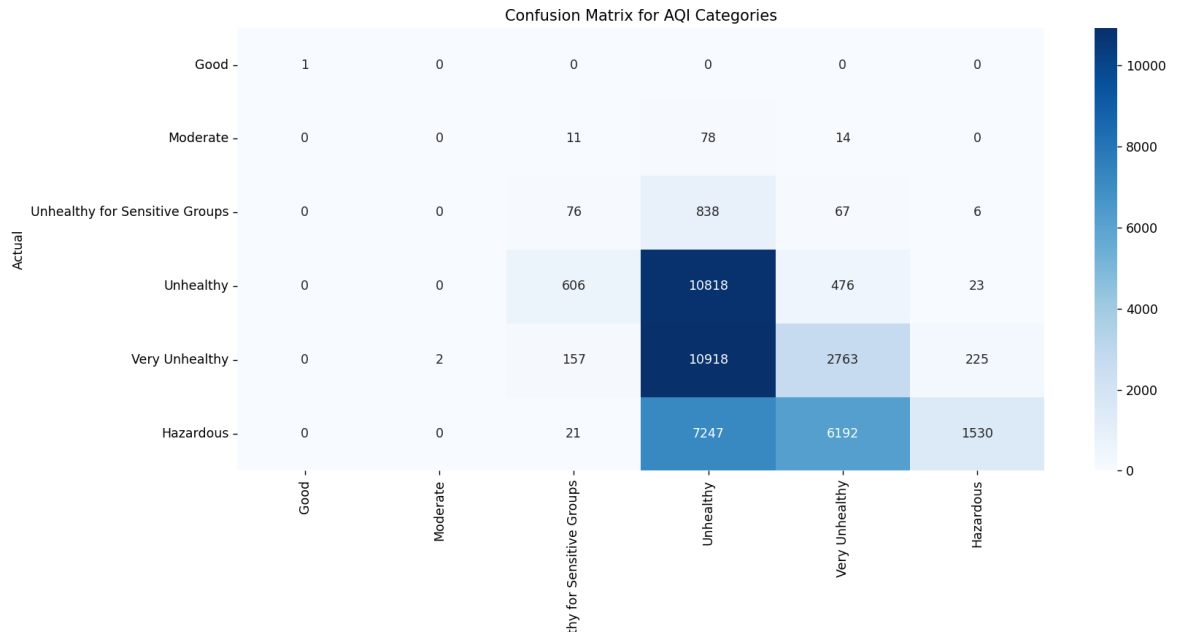


Figure 16: Confusion Matrix

## 7 Conclusion

This Transformer-based approach represents a sophisticated method for multivariate air quality prediction, demonstrating the power of deep learning in environmental monitoring.

Class	Precision	Recall	F1-Score	Support
Good	1.00	1.00	1.00	1
Moderate	0.86	0.10	0.18	14,990
Unhealthy for Sensitive Groups	0.00	0.00	0.00	103
Unhealthy	0.36	0.91	0.52	11,923
Very Unhealthy	0.09	0.08	0.08	987
Hazardous	0.29	0.20	0.23	14,065
<b>Accuracy</b>	0.36 (on 42,069 instances)			
<b>Macro Avg</b>	0.43	0.38	0.34	42,069
<b>Weighted Avg</b>	0.51	0.36	0.29	42,069

Table 3: Classification Report

Station	Pollutant	MSE	MAPE	RMSE
site_1431	co	476.2848511	0.818838239	21.82395172
site_1431	nh3	468.022644	1.322046399	21.63383102
site_1431	no	533.4741211	0.300118834	23.09705925
site_1431	no2	387.1081848	0.347151875	19.67506599
site_1431	nox	188.6734467	1.895005822	13.73584557
site_1431	ozone	1946.36731	0.219041005	44.11765289
site_1431	pm10	797.6629028	0.819417179	28.24292755
site_1431	pm25	27.60520935	1.101059794	5.25406599
site_1431	so2	11659.03906	3.528970242	107.9770279
site_1431	avg	1831.58197	1.150183265	42.79698553
site_105	co	3.786857367	7.49151E+13	1.94598496
site_105	no	513.8780518	1.333410144	22.66887856
site_105	no2	197.2561188	1.970907211	14.04478931
site_105	nox	505.6847229	2.303E+15	22.48743439
site_105	ozone	66.20107269	0.664623857	8.136404037
site_105	pm10	1093.631348	0.26621747	33.07009888
site_105	pm25	857.1174316	0.360151201	29.27656746
site_105	avg	462.5079433	3.39702E+14	21.50599784

Table 4: Pollutant Data Analysis

## Alternate Transformer

### 1 Introduction

We have included this primarily as a way to showcase the difference hyperparameter tuning can make.

**Differences from previous approach:** This one primarily focuses on data preprocessing, utilizing forward and backward filling for imputation and applying a MinMaxScaler for feature scaling while excluding timestamp columns from processing. In contrast, the old one presents a comprehensive machine learning pipeline that includes mean imputation, standard scaling, feature engineering with lagged variables, model training, evaluation, and detailed AQI calculations.

Most importantly, however this new model is much more complex. The model has an 8 times larger dimension, twice the number of heads, 8 times the encoder layers, and 4 times the number of feedforward layers. This model was already trained for a much longer time, on the entire dataset, and on a more powerful GPU. The data processing stays largely the same.

## 2 Training Results



Figure 17: Training vs Validation Loss. We are using epoch 12 for all further evaluation



### 3 Evaluation Results

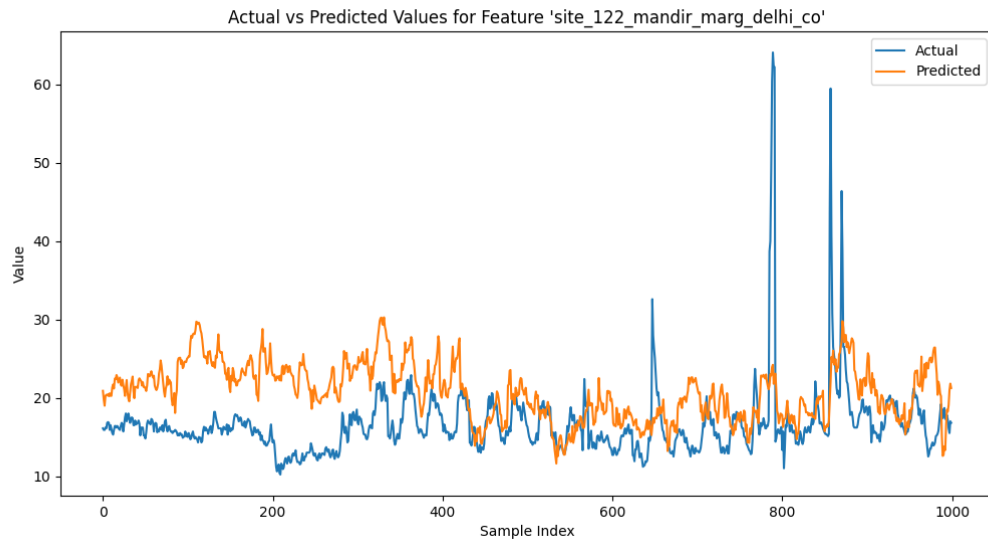


Figure 18: 1

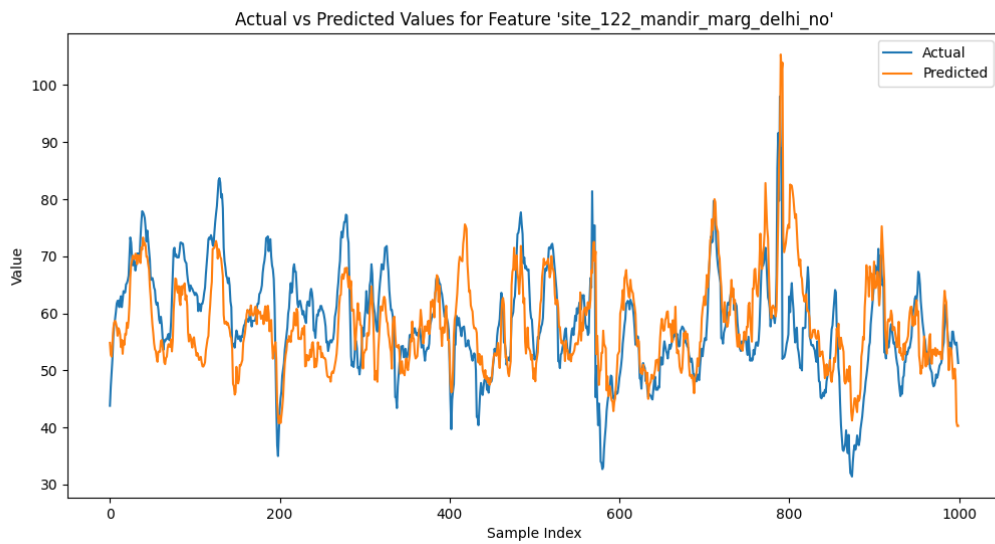


Figure 19: 2

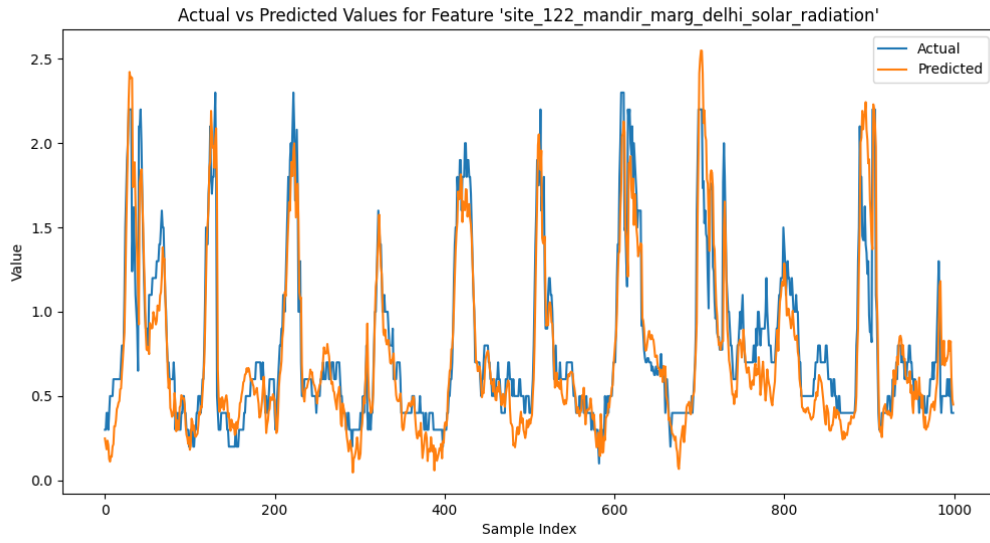


Figure 20: 3

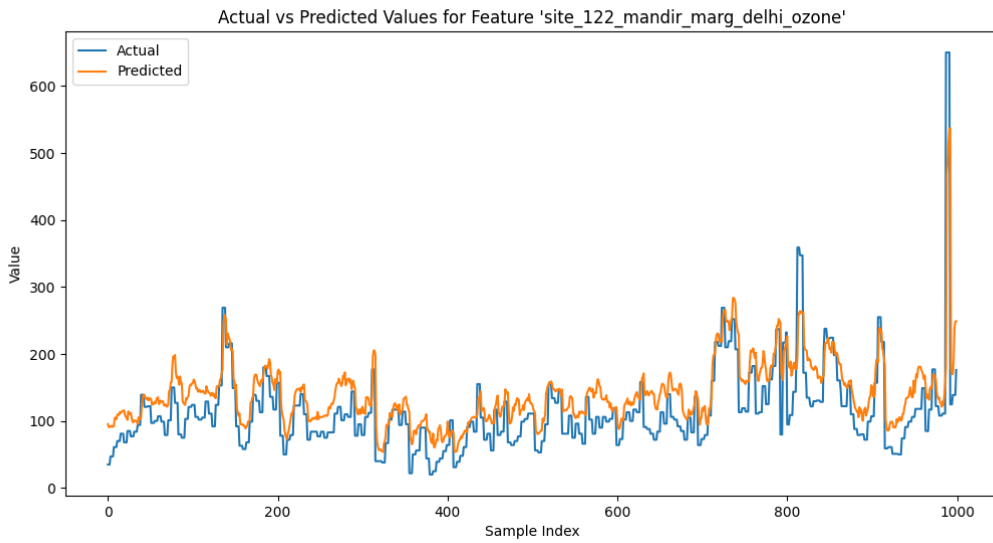


Figure 21: 4

## LSTM

We are testing the ability of LSTM's to excel at capturing long-term dependencies in sequential data. Unlike traditional models, LSTMs can retain and learn patterns over extended periods, making them ideal for time series data where past observations significantly influence future outcomes. Again, we have trained this on the full dataset, for around 4 hours.

# Model Overview

In this project, we developed a Long Short-Term Memory (LSTM) model using PyTorch to handle sequential data effectively. The model architecture and data preprocessing steps are outlined below.

## Model Architecture

### Parameters:

- **num\_features:** Number of input features per timestep.
- **hidden\_size:** Number of features in the LSTM's hidden state.
- **num\_layers:** Number of stacked LSTM layers.
- **output\_size:** Dimension of the model's output.

### Layers:

- **LSTM Layer:** Captures temporal dependencies in the input sequence.
- **Fully Connected (FC) Layer:** Transforms the LSTM output to the desired output size.

**Initialization** involves setting up the LSTM and FC layers using PyTorch's `nn.LSTM` and `nn.Linear` modules.

## Data Flow

1. **Input:** Sequence data with shape `(batch_size, seq_length, num_features)`.
2. **LSTM Processing:** The input passes through the LSTM layer, producing hidden states for each timestep.
3. **Sequence Summarization:** The hidden state from the last timestep is extracted to represent the entire sequence.
4. **Output Generation:** The summarized representation is fed into the FC layer to produce the final output with shape `(batch_size, output_size)`.

## Training Strategy

- **Loss Function:** Mean Squared Error (MSE) for regression tasks or Cross-Entropy Loss for classification.
- **Optimizer:** Adam optimizer is utilized for its adaptive learning rate capabilities.
- **Backpropagation Through Time (BPTT):** This technique is employed to compute gradients across the unrolled network for training.

## Data Preprocessing

- **Scaling:** Features are normalized using `MinMaxScaler` to fit within the range  $[0, 1]$ .
- **Handling Missing Values:** Columns with all missing values are removed. Remaining missing entries are filled using forward and backward filling methods.
- **Verification:** A sample plot compares original and scaled feature values to ensure proper scaling.
- **Saving Processed Data:** The cleaned and scaled data is saved for subsequent model training.

## Conclusion

The integration of the LSTM model with robust data preprocessing techniques ensures effective training and reliable predictions. This setup is well-suited for tasks involving sequential data, leveraging the strengths of LSTM in capturing temporal patterns.

## 1 Results

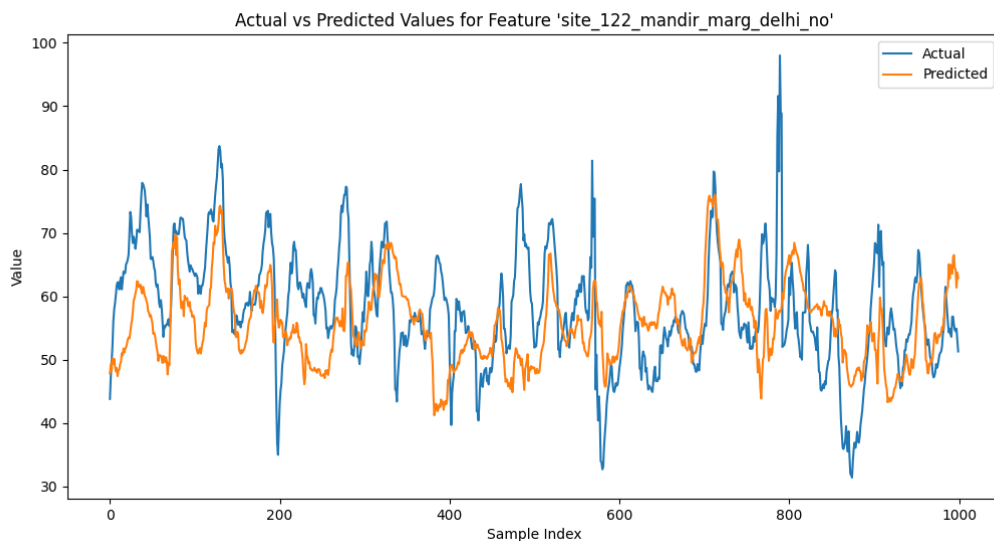


Figure 22: 1

## Limitations

1. Our model only works for data from Delhi. It will not generalize to AQI prediction for all of India.
2. Our data had 40% values `NaN`, which reduced our accuracy.

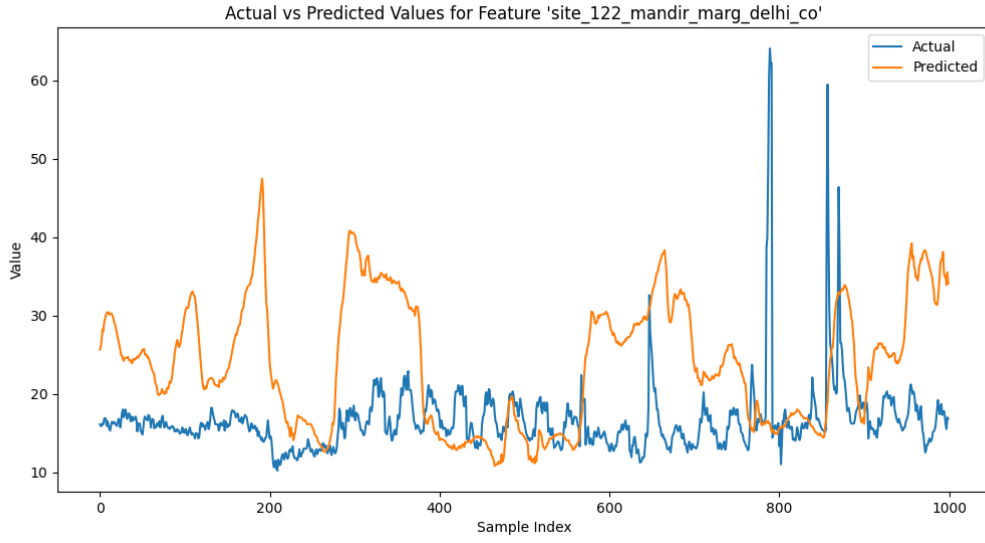


Figure 23: 2

Table 5: Pollutant Statistics for Stations

Station	Pollutant	MSE	MAPE	RMSE
site_1420	co	579.196	0.4272	24.0665
site_1420	nh3	473.704	1.1684	21.7647
site_1420	no	423.844	0.8168	20.5875
site_1420	no2	542.849	0.3168	23.2991
site_1420	nox	1192.172	3.6870	34.5278
site_1420	ozone	3925.371	0.7698	62.6528
site_1420	pm10	3338.504	0.7276	57.7798
site_1420	pm25	240.632	3.1522	15.5123
site_1420	so2	4229.040	1.0416	65.0311
site_1420	avg	1660.590	1.3453	40.7503
site_1432	co	386.570	0.8341	19.6614
site_1432	nh3	1449.205	2.0503	38.0684
site_1432	no	707.982	2.5677	26.6079
site_1432	no2	1020.669	0.5903	31.9479
site_1432	nox	352.389	2.4910	18.7720
site_1432	ozone	4150.792	0.2367	64.4266
site_1432	pm10	1677.495	0.3764	40.9572
site_1432	pm25	86.152	3.0566	9.2818
site_1432	so2	4153.913	1.6184	64.4509
site_1432	avg	1553.908	1.5357	39.4196

3. We would have been able to create a model to better approximate the complex dataset if we had more computational power.

## Next Steps

1. **Incorporate Traffic Data:** Use real-time and historical traffic data to capture vehicular emission patterns, which are a significant contributor to air pollution.
2. **Utilize Satellite Imagery:** Analyze satellite data to track large-scale pollution sources and monitor changes in AQI over time.