# Beijing Multi-Site Air-Quality Index: Time-Series Analysis

Krutin Rathod
*Department of Computer Science*
*Stevens Institute of Technology*
Hoboken, United States
krathod3@stevens.edu

Nihar Patel
*Mathematical Sciences*
*Stevens Institute of Technology*
Hoboken, United States
npatel17@stevens.edu

Sanika Mhadgut
*Department of Computer Science*
*Stevens Institute of Technology*
Hoboken, United States
smhadgut@stevens.edu

***Abstract*—This project focuses on forecasting air quality variables using machine learning algorithms applied to the 'Beijing Multi-Site Air Quality' dataset obtained from the UC Irvine Machine Learning Repository. The dataset comprises hourly atmospheric measurements from 12 cities, including pollutants such as PM2.5, PM10, SO2, NO2, CO, and O3, along with meteorological variables. Our goal is to develop accurate predictive models to forecast these variables, aiding in air quality management and public health initiatives. To achieve this, we employ various machine learning algorithms, including SARIMA, Prophet, VAR, and Multi-Layer Perceptron (MLP) neural networks. The SARIMA model captures both seasonal and non-seasonal variations in the data, while Prophet excels in handling strong seasonal effects. VAR models enable us to analyze dynamic interdependencies among multiple variables. Additionally, MLP neural networks offer flexibility in capturing complex temporal patterns. Through extensive experimentation and evaluation, we demonstrate the effectiveness of these models in forecasting air quality variables. Our contributions include identifying optimal model configurations, assessing predictive accuracy, and providing insights into the underlying patterns and dynamics of air pollution. These findings offer valuable guidance for policymakers and environmental agencies in mitigating air pollution and improving public health.**

## I. Introduction

Air pollution is a major health and environmental concern for urban centers globally. Beijing, a densely populated metropolis in China, has a documented history of air quality issues. In 2013, Bejing was suffocated with a poisonous haze for four days. PM2.5 particulate levels reached their highest point and caused havoc in the world. In our project, we focus on analyzing and forecasting air pollution levels in Beijing, China, using the 'Beijing Multi-Site Air Quality' dataset. By analyzing these time-dependent features, we aim to develop a model capable of forecasting PM2.5 concentrations and potentially other pollutants of interest.

Our methodology involves a multi-step process. First, we will clean and prepare the data by addressing missing values, handling categorical variables, and potentially transforming features to improve model performance. Second, we will conduct an exploratory data analysis, visualizing and analyzing the relationships between pollutants and meteorological variables to understand their impact on air quality. This will include investigating correlations, seasonality, and potential trends within the data. Third, we will evaluate different machine learning models. This may include established time series forecasting techniques like ARIMA and Prophet, as well as potential multivariate models suitable for capturing the complex interactions between air pollutants and meteorological factors. Finally, we will assess the chosen model's performance using relevant metrics and potentially refine the model architecture or training process to achieve optimal prediction accuracy. Exploratory data analysis reveals correlations between pollutants and meteorological variables, as illustrated in Figures 2 and 3. Furthermore, we conduct auto-correlation and partial auto-correlation analysis to assess temporal patterns and identify suitable parameters for time series modeling, as depicted in Figure 4.

By developing a robust air quality prediction model, this project can contribute valuable insights for environmental monitoring strategies and potentially inform pollution mitigation efforts in Beijing.

## II. Related Work

We have looked at two important studies about the analysis of the Beijing Air Quality dataset. Firstly, Zhang et al. (2019) studied how to predict PM2.5 pollution levels in Beijing using both Autoregressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) models. Their study showed that while ARIMA is simple and easy to understand, LSTM performed better at capturing the complex patterns and long-term relationships in the data. However, using LSTM required more computer power and larger amounts of training data.[1] Secondly, Wang et al. (2020) examined the patterns of air pollution in Beijing over time and space using statistical methods and Geographic Information System (GIS) techniques.[2] Their analysis stressed the importance of considering differences in air quality across different areas, which can help in identifying areas with the highest pollution levels and planning targeted actions to improve air quality. These studies provide useful insights into understanding and addressing air pollution problems using data from the Beijing Air Quality dataset. Building upon these findings, this project aims to explore the potential of combining time series and spatial analysis techniques. By incorporating both temporal trends and spatial variability, we can potentially develop a more robust and informative model for air quality prediction in Beijing. This approach could offer valuable insights for environmental monitoring and targeted pollution mitigation strategies.

## III. Our Solution

This section elaborates on your solution to the problem.

### A. Description of Dataset

For our project, we utilized the 'Beijing Multi-Site Air Quality' dataset from the UC Irvine Machine Learning Repository. This dataset comprises 12 individual CSV files containing time-dependent atmospheric measurements, including target variables such as $PM_{2.5}$, $PM_{10}$, $SO_2$, $NO_2$, $CO$, and $O_3$, as well as independent variables like temperature, pressure, dew point temperature, rainfall, wind direction, wind speed, and station name, along with corresponding timestamps.

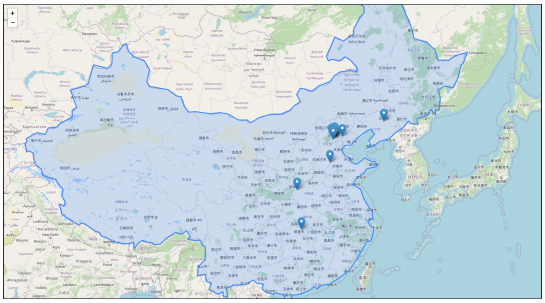

Fig. 1: Map of China highlighting city

Each file in the dataset covers a detailed time span from March $1^{st}$, 2013, to February $28^{th}$, 2017. These files contain hourly measurements of air pollutants gathered from 12 cities, meticulously collected by the Beijing Municipal Environmental Monitoring Center. This extensive dataset enables in-depth analysis of pollution trends across different urban areas. The geographical distribution of monitoring stations is visually represented in Figure 1, offering valuable insights into regional air quality variations.

| | year | month | day | hour | PM2.5 | PM10 | SO2 | NO2 | CO | O3 | TEMP | PRES | DEWP | RAIN | wd | WSPM | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2013 | 3 | 1 | 0 | 6.0 | 6.0 | 4.0 | 8.0 | 300.0 | 81.0 | -0.5 | 1024.5 | -21.4 | 0.0 | NNW | 5.7 | Tiantan |
| **1** | 2013 | 3 | 1 | 1 | 6.0 | 29.0 | 5.0 | 9.0 | 300.0 | 80.0 | -0.7 | 1025.1 | -22.1 | 0.0 | NW | 3.9 | Tiantan |
| **2** | 2013 | 3 | 1 | 2 | 6.0 | 6.0 | 4.0 | 12.0 | 300.0 | 75.0 | -1.2 | 1025.3 | -24.6 | 0.0 | NNW | 5.3 | Tiantan |
| **3** | 2013 | 3 | 1 | 3 | 6.0 | 6.0 | 4.0 | 12.0 | 300.0 | 74.0 | -1.4 | 1026.2 | -25.5 | 0.0 | N | 4.9 | Tiantan |
| **4** | 2013 | 3 | 1 | 4 | 5.0 | 5.0 | 7.0 | 15.0 | 400.0 | 70.0 | -1.9 | 1027.1 | -24.5 | 0.0 | NNW | 3.2 | Tiantan |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **35059** | 2017 | 2 | 28 | 19 | 20.0 | 48.0 | 2.0 | NaN | 500.0 | NaN | 12.5 | 1013.5 | -16.2 | 0.0 | NW | 2.4 | Tiantan |
| **35060** | 2017 | 2 | 28 | 20 | 11.0 | 34.0 | 3.0 | 36.0 | 500.0 | NaN | 11.6 | 1013.6 | -15.1 | 0.0 | WNW | 0.9 | Tiantan |
| **35061** | 2017 | 2 | 28 | 21 | 18.0 | 32.0 | 4.0 | 48.0 | 500.0 | 48.0 | 10.8 | 1014.2 | -13.3 | 0.0 | NW | 1.1 | Tiantan |
| **35062** | 2017 | 2 | 28 | 22 | 15.0 | 42.0 | 5.0 | 52.0 | 600.0 | 44.0 | 10.5 | 1014.4 | -12.9 | 0.0 | NNW | 1.2 | Tiantan |
| **35063** | 2017 | 2 | 28 | 23 | 15.0 | 50.0 | 5.0 | 68.0 | 700.0 | 21.0 | 8.6 | 1014.1 | -15.9 | 0.0 | NNE | 1.3 | Tiantan |

Fig. 2: Data Preview of the file

Figure 2 displays a data preview extracted from one of the files, each containing 35,000 instances of records collected at hourly intervals. The dataset encompasses essential features such as $PM_{2.5}$, $PM_{10}$, $SO_2$, $NO_2$, $CO$, $O_3$, $TEMP$, $PRES$, $DEWP$, $RAIN$, and $WSPM$. $PM_{2.5}$ represents the density measure of Particulate Matter with a diameter of 2.5 micrometers or smaller, while $PM_{10}$ denotes the density measure of matter with a size of 10 micrometers or less. Additionally, $SO_2$, $NO_2$, $CO$, and $O_3$ indicate the concentration levels of Sulphur dioxide, Nitrogen dioxide, Carbon monoxide, and Ozone respectively. These variables constitute the focus of our analysis, with their temporal variability illustrated in Figure 3.


Fig. 3: Target variable Plot

Regarding independent variables, we include $TEMP$, $PRES$, $DEWP$, $RAIN$, and $WSPM$, representing Temperature, Pressure, Dew Point Temperature, Rainfall, and Wind Speed respectively. Additionally, the variable $wd$ signifies Wind Direction.

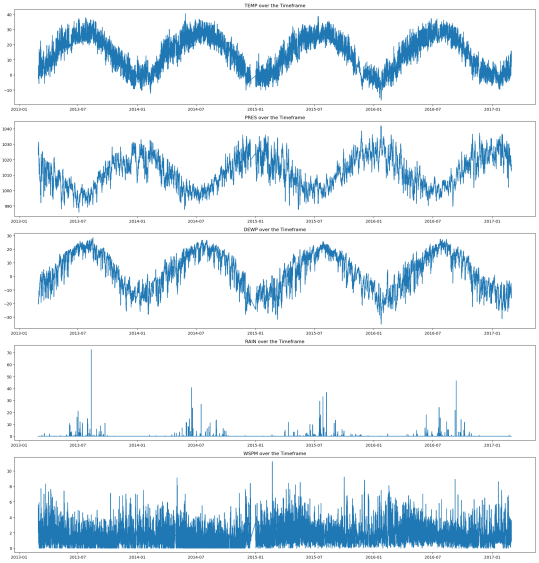


Fig. 4: Independent variable Plot

All target and independent variables are time-dependent, depicted in Figure 3 and Figure 4. To streamline our workflow, we've consolidated string columns like $year$, $month$, $day$, and $hour$ into a unified $datetime64$ column named $timestamp$.

In our dataset, we encountered missing values denoted as $NaN$. To address this, we employed a forward-fill interpo-

lation method, replacing missing values with the most recent preceding valid numerical value until the next valid numerical value was encountered.

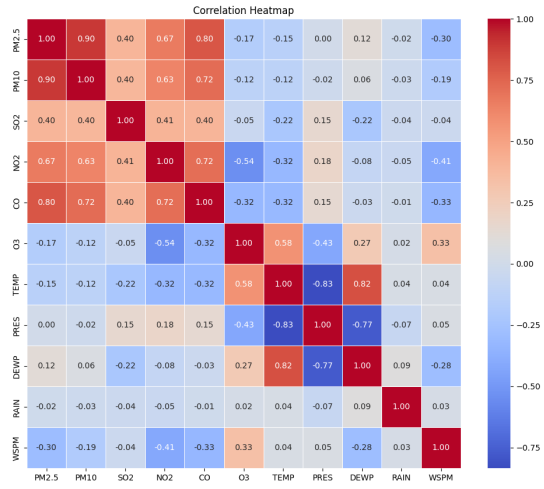


Fig. 5: Correlation Heat Map of all the Features

Figure 7 presents a Heat Map illustrating the correlation between each feature in the dataset. Notably, the pollutants (target variables) exhibit strong correlations. Additionally, we observe intriguing relationships among the independent variables: $Temperature$ and $Pressure$ demonstrate a notable negative correlation, whereas $Temperature$ and $DewPointTemperature$ exhibit a significant positive correlation.

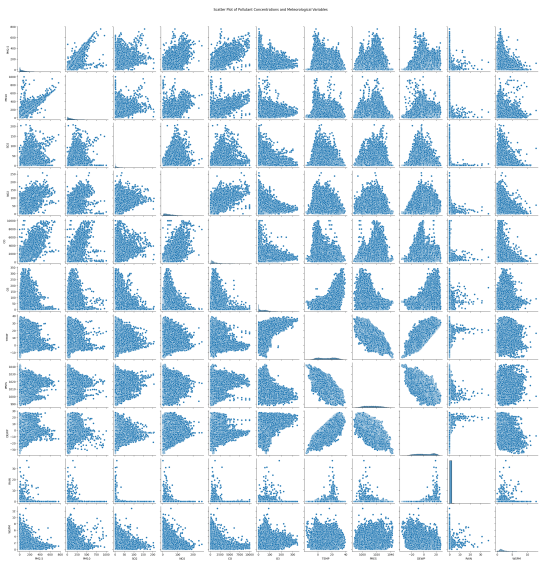

Fig. 6: Pair-Plot of Pollutant Concentrations and Meteorological Variables

Figure 6 showcases the pairwise relationships between pollutant concentrations and meteorological variables. Pollutant concentrations, including PM2.5, PM10, SO2, NO2, CO, and O3, exhibit varying degrees of correlation, suggesting potential sources of pollution. Meteorological variables such as TEMP (Temperature), PRES (Pressure), DEWP (Dew Point Temperature), RAIN (Rainfall), and WSPM (Wind Speed) also demonstrate interrelationships. For instance, temperature and dew point temperature might exhibit a linear relationship, while wind speed could fluctuate with changes in pressure or temperature. Notably, outliers in the scatter plots may indicate anomalous observations or extreme values, possibly arising from measurement errors, unusual weather events, or specific pollution sources.

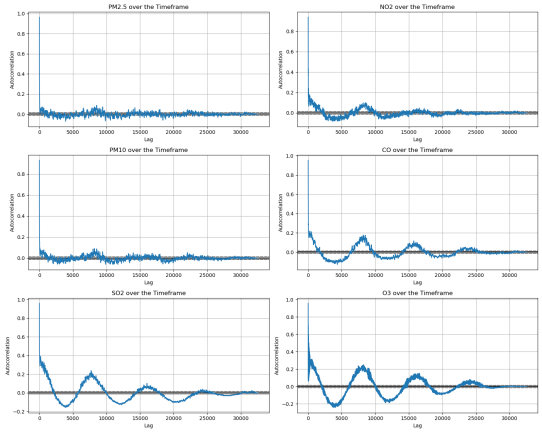


Fig. 7: Auto-correlation of all the Target Features

Auto-correlation refers to the degree of similarity between a given time series and a lagged version of itself over successive time intervals [3].

| | count | mean | std | min | 25% | 50% | 75% | max | median | 1st quartile | 3rd quartile |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **PM2.5** | 31100.0 | 79.245241 | 81.492876 | 2.0000 | 19.0 | 54.0 | 111.0 | 762.0 | 54.0 | 19.0 | 111.0 |
| **PM10** | 31100.0 | 99.144428 | 89.973374 | 2.0000 | 31.0 | 77.0 | 139.0 | 999.0 | 77.0 | 31.0 | 139.0 |
| **SO2** | 31100.0 | 13.981728 | 19.958511 | 0.2856 | 2.0 | 5.0 | 17.0 | 207.0 | 5.0 | 2.0 | 17.0 |
| **NO2** | 31100.0 | 44.407718 | 30.965834 | 2.0000 | 20.0 | 38.0 | 63.0 | 258.0 | 38.0 | 20.0 | 63.0 |
| **CO** | 31100.0 | 1202.186076 | 1161.331577 | 100.0000 | 400.0 | 900.0 | 1500.0 | 10000.0 | 900.0 | 400.0 | 1500.0 |
| **O3** | 31100.0 | 55.270282 | 54.191037 | 0.2142 | 11.0 | 44.0 | 77.0 | 340.0 | 44.0 | 11.0 | 77.0 |
| **TEMP** | 31100.0 | 12.942435 | 11.493197 | -16.8000 | 2.6 | 13.5 | 22.9 | 39.0 | 13.5 | 2.6 | 22.9 |
| **PRES** | 31100.0 | 1013.481529 | 10.158541 | 988.0000 | 1005.1 | 1013.2 | 1021.5 | 1042.8 | 1013.2 | 1005.1 | 1021.5 |
| **DEWP** | 31100.0 | 1.755331 | 13.724698 | -36.0000 | -9.5 | 1.8 | 14.4 | 27.5 | 1.8 | -9.5 | 14.4 |
| **RAIN** | 31100.0 | 0.055051 | 0.709264 | 0.0000 | 0.0 | 0.0 | 0.0 | 37.3 | 0.0 | 0.0 | 0.0 |
| **WSPM** | 31100.0 | 1.840608 | 1.292341 | 0.0000 | 1.0 | 1.5 | 2.3 | 12.8 | 1.5 | 1.0 | 2.3 |

Fig. 8: Summary Statistics

Figure 8 presents summary statistics of the dataset, encompassing various pollutants and meteorological parameters. The data provides valuable insights into environmental conditions, with PM2.5 and PM10 concentrations exhibiting wide variability, suggesting fluctuations in particulate matter levels. Additionally, pollutants such as SO2, NO2, and CO demonstrate diverse concentration ranges, indicating potential emission sources and their impact on air quality.

Meteorological variables, including temperature, pressure, dew point, and wind speed, also exhibit variability, reflecting the dynamic nature of weather conditions. Temperature ranges from -16.8°C to 39.0°C, with pressure averaging around 1013.48 hPa. Dew point values show considerable variation, indicating fluctuations in atmospheric moisture levels.

The dataset includes rainfall data, with the majority of observations showing zero rainfall, suggesting predominantly dry weather conditions during the recorded period. Wind

speed averages around 1.84 m/s, with a slightly right-skewed distribution.

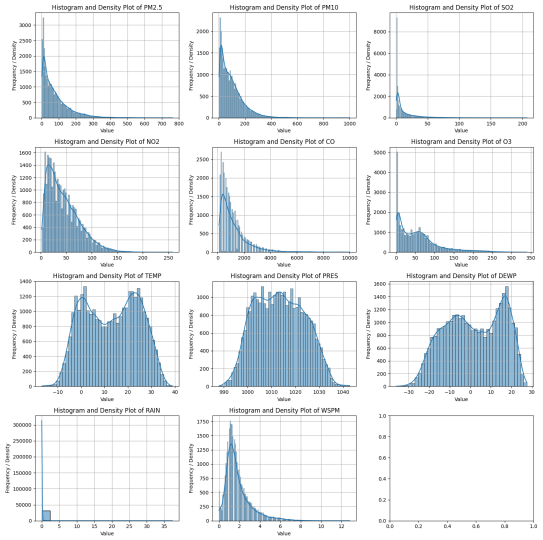


Fig. 9: Density Histogram

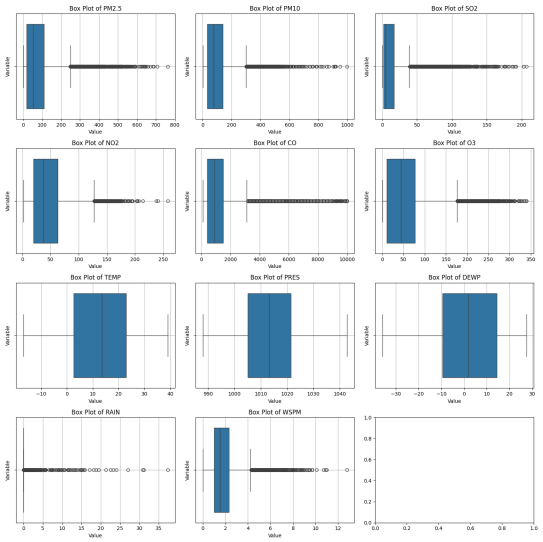


Fig. 10: Boxplot

Figure 9 is the combined plot of historgram and density for each variable. Figure 10 is the Box PLot for each variable. The variables representing pollutant concentrations, including PM2.5, PM10, SO2, NO2, CO, and O3, generally exhibit right-skewed distributions. This indicates that most observations have lower values, with a few higher outliers. The density plots show the probability density function, with peaks indicating the most common values. In contrast, meteorological variables such as TEMP (Temperature), PRES (Pressure), DEWP (Dew Point Temperature), RAIN (Rainfall), and WSPM (Wind Speed) display different patterns. TEMP shows a relatively normal distribution, with temperatures centered around a mean value. PRES appears to have a somewhat normal distribution but may have some outliers. DEWP is similar to temperature, with a slight skew towards higher dew points. RAIN likely consists mostly of zero values (no rainfall), with occasional higher values. WSPM is skewed to the right, indicating that lower wind speeds are more common, with a few instances of higher wind speeds.

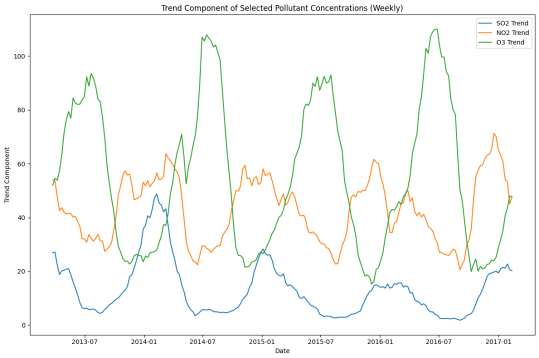


Fig. 11: Weekly Trend Comparison of 'SO2', 'NO2', 'O3' gases

Figure 11 is the combined plot of Weekly Trend Comparison of 'SO2', 'NO2', 'O3' gases. We can observe that trend of SO2 and NO2 is inversely proportional to O3. SO2 and NO2 are primary pollutants emitted from various sources such as industrial processes, vehicles, and power plants. These pollutants can undergo chemical reactions in the atmosphere, leading to the formation of secondary pollutants such as O3. The trend component of NO2 exhibits a slightly increasing trend over time. This suggests a gradual rise in NO2 concentrations over the observed period, indicating a potential long-term upward trend in NO2 pollution levels. However, SO2 shows a relatively long term decreasing trend. O3 has stable pattern with minor fluctuations. Regarding other variables, both PM2.5 and PM10 exhibit similar trends, these fluctuations could be attributed to changes in emissions, atmospheric conditions, and meteorological factors influencing particle dispersion. The trend of CO demonstrates relatively smoother variations compared to other gases. The trend of temperature, pressure and DEWP exhibits a seasonal pattern with periodic fluctuations corresponding to changes in weather conditions throughout the year. The trend of rainfall shows distinct patterns of wet and dry periods, reflecting the seasonal distribution of precipitation over time. Variations in wind speed reflect changes in the speed and direction of air masses, influencing pollutant dispersion and atmospheric mixing.

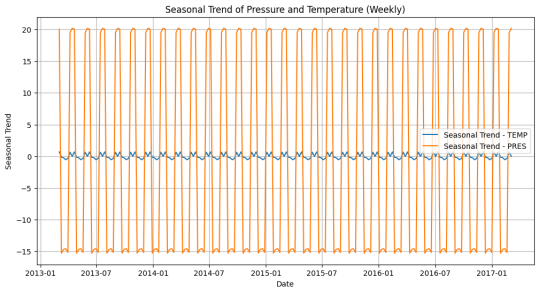


Fig. 12: Seasonal Analysis

Figure 12 is the Seasonality plot of 'TEMP' and 'PRES'.

The seasonal trend of temperature shows typical seasonal variations, with higher temperatures during summer months and lower temperatures during winter months, reflecting the seasonal changes in weather. The seasonal trend of pressure also demonstrates periodic fluctuations but with less variability compared to temperature. Pressure tends to follow a more stable pattern across seasons, with relatively minor fluctuations compared to temperature. Changes in pressure can be influenced by weather systems such as high and low-pressure systems, but these variations may not be as directly tied to seasonal changes as temperature.

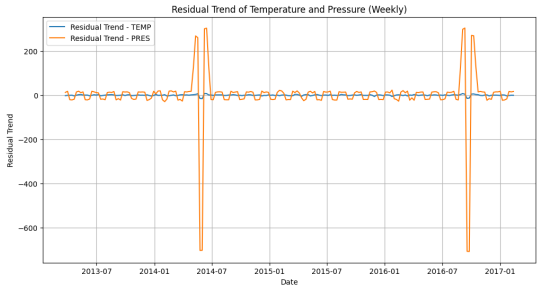


Fig. 13: Residual Analysis

Figure 13 is the Residual plot of 'TEMP' and 'PRES'. The residual trend shows random fluctuations around zero, indicating that the seasonal and trend components adequately explain the variation in temperature. The residual trend exhibits sporadic deviations from zero, indicating occasional anomalies in pressure data that are not captured by the seasonal and trend components.

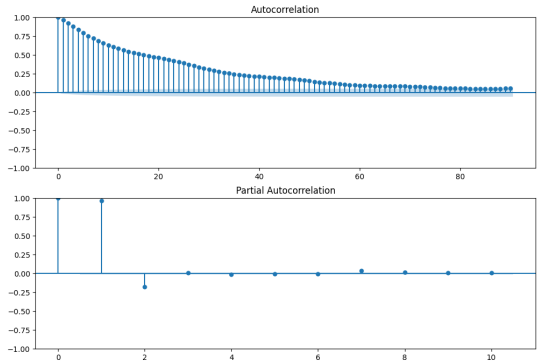


Fig. 14: Autocorrelation and Partial Autocorrelation

Figure 14 is the combined plot of Auto-correlation and Partial Auto-correlation function. By this plot we can identify the patterns and seasonality of the time series data. It will also be useful in determining the Auto regressive (AR) and moving average (MA) terms in ARIMA modeling. Initially we performed the Adfuller test, to identify the stationarity of the data. The statistical p-value for the PM2.5 column pollutant came in the order of $10^{-20}$. Thus, the data is found out to be stationary. Further as the data is stationary, we will plot the auto-correlation function and partial auto-correlation function as seen above. "p" equals AR model lags, "d" equals differencing, "q" equals MA lags. The inference for the ARIMA model will be, p = 3 as it shuts off PACF at 2 & q = 0 as there is no sudden shut off or exponential decrease in the partial auto correlation graph.

### B. Machine Learning Algorithms

*1) Arimax & Sarimax:* ARIMA (Auto-Regressive Integrated Moving Average) is machine learning model used to forecast the time series dataset. It is based on the autoregressive (AR), differencing (I), and moving average (MA) components of the data. SARIMA (Seasonal ARIMA) is built on top of the ARIMA model as it also incorporates the seasonal components, allowing for the modeling of time series data with seasonal patterns. Both are popular and important models in the field of time series analysis. They both capture the complex underlying pattern of the dataset. The ARIMA model is more focused towards the non-seasonal variations, whereas SARIMA model is used for both the seasonal and non seasonal variations. By leveraging these models, analysts can gain insights into temporal trends and make informed forecasts for various applications.

*2) Prophet:* Prophet is a forecasting model developed by Facebook's Core Data Science team. It is designed to handle time series data with strong seasonal effects and multiple seasonality components. Prophet incorporates a decomposable time series model with three main components: trend, seasonality, and holiday effects (if applicable). The model is flexible, robust, and easy to use, making it suitable for forecasting tasks across various domains. It automatically detects changepoints in the time series data and can handle missing data and outliers effectively.

*3) Vector autoregression (VAR):* VAR is a multivariate time series model that represents each variable in the system as a linear function of its lagged values and lagged values of other variables in the system. It is widely used for modeling and forecasting the joint behavior of multiple time series variables that influence each other. VAR models are capable of capturing dynamic interdependencies and feedback mechanisms among the variables in the system. They are particularly useful for analyzing economic and financial time series data, as well as in other domains where variables are mutually time-dependent. Estimation and inference in VAR models typically involve techniques such as least squares estimation, maximum likelihood estimation, and impulse response analysis.

*4) Multi-Layer Perceptron (Neural Network):* In the realm of machine learning models applied to time series problems, the Multi-layer Perceptron (MLP) regressor stands out as a versatile approach. In time-series analysis, MLPRegressor can be a powerful tool for predicting future values based on historical data. Time-series data often exhibits complex patterns and dependencies that traditional statistical methods may struggle to capture. MLPRegressor, with its ability to model nonlinear relationships and capture intricate temporal dependencies, can offer significant advantages in such scenarios. But training and hyper-tuning models like this can be a bit challenging task and these networks are prone to common optimization problems like over-fitting and vanishing gradient.

### C. Implementation Details

*1) Arimax & Sarimax:* The SARIMA model is used to understand the seasonal and non seasonal components of the time

series dataset. It provides us with a framework for modeling time series variability. Non-seasonal parameters, denoted by p, d, and q, capture the series' inherent dynamics, distinguishing between autoregressive and moving average effects. On the other hand, seasonal parameters, symbolized by P, D, Q, and m, extend this modeling approach to account for cyclic patterns inherent in the data, such as those occurring quarterly, monthly, weekly, or daily.

We identified an optimal SARIMA model based on the performance metrics of Akaike Information Criterion (AIC). We have used autoArima() function from the pyramid package to find the best configuration for our SARIMA model. By iteratively exploring parameter combinations, the function identifies the set that minimizes the chosen criterion, ensuring a well-fitted model to the training dataset. Despite this rigorous search, observations from our model summary underscore the significance of certain lagged variables in the forecast, particularly those linked to seasonal terms. Furthermore, the model's performance evaluation, measured through the root mean square error (RMSE), indicates its robustness in predicting PM10 concentrations, with deviations falling within one standard deviation of the actual data distribution.

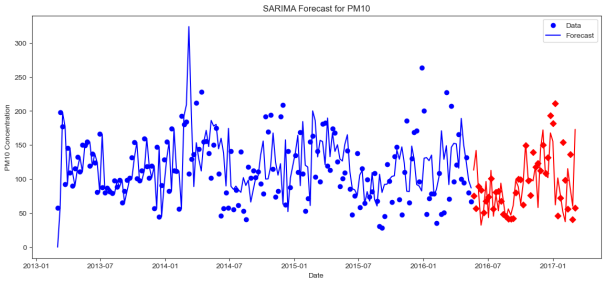


Fig. 15: SARIMA forecasting for $PM_{2.5}$

While the SARIMA model demonstrates promising performance in capturing time series dynamics, visual inspection of its predictions against actual observations reveals some limitations. Instances of overestimation and underestimation, particularly around certain time periods, highlight potential areas for model refinement. These discrepancies underscore the need for continued evaluation and refinement of the model's performance, ensuring its reliability and accuracy in forecasting air pollution levels. Additionally, thorough examination of model residuals and their distribution provides valuable insights into areas where the model may benefit from further calibration, guiding future iterations and enhancements to improve its predictive capabilities.

| Pollutant | MAE | MSE | RMSE |
|---|---|---|---|
| PM2.5 | 32.68 | 1902.81 | 43.62 |
| PM10 | 32.80 | 2095.06 | 45.77 |
| SO2 | 4.91 | 54.72 | 7.40 |
| NO2 | 14.18 | 371.24 | 19.27 |
| CO | 425.50 | 434612.54 | 659.25 |
| O3 | 14.29 | 301.20 | 17.36 |

TABLE I: Metrics table for SARIMAX model (weekly basis)

*2) Prophet:* Implementing the Prophet model involves several steps. First, the time series data needs to be prepared, including cleaning and formatting. Prophet can handle missing data and outliers effectively, but it's still essential to preprocess the data to ensure accuracy.

Next, the Prophet model is instantiated and configured. This includes specifying any additional parameters, such as seasonality settings and holiday effects, if applicable. Prophet is designed to handle time series data with strong seasonal effects and multiple seasonality components, making it particularly suitable for forecasting tasks in environments with complex seasonal patterns.

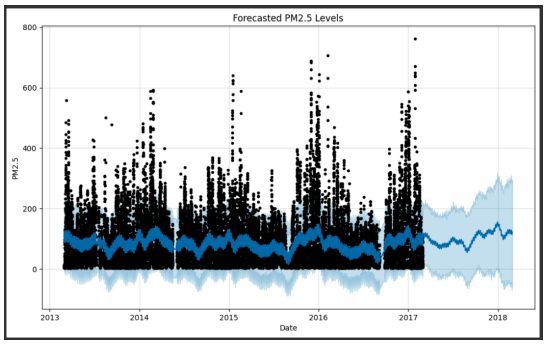


Fig. 16: Prophet $PM_{2.5}$

Once the model is configured, it's trained using historical time series data. Prophet automatically detects changepoints in the data, identifying points where significant changes in trend occur. This feature helps capture changes in the underlying patterns of the time series, improving the accuracy of the forecasts.

After training, the model can be used to make predictions for future time periods. Prophet provides intuitive methods for visualizing the forecasted data, including confidence intervals to assess the uncertainty of the predictions. Figure 16 is prediction of 2.5 using Prophet Model.

Overall, implementing the Prophet model involves preparing the data, configuring the model, training it on historical data, and using it to make forecasts for future time periods. Prophet's flexibility, robustness, and ease of use make it a valuable tool for forecasting tasks across various domains, particularly those with strong seasonal effects and multiple seasonality components.

In this analysis, we used the Prophet forecasting model to predict air quality variables including PM2.5, CO, temperature (TEMP), pressure (PRES), dew point temperature (DEWP), rainfall (RAIN), wind direction (WD), and wind speed (WSPM) on a weekly basis.

The data was preprocessed by converting the 'wd' column (wind direction) to numeric using label encoding and then aggregated at a weekly level. After fitting individual Prophet models for each variable, we made predictions for the next 52 weeks. The plots above show the observed values (blue) and forecasted values (red) for each variable over time. The shaded area represents the uncertainty range of the forecast. Based on the forecasts, we can observe the trend and seasonality patterns

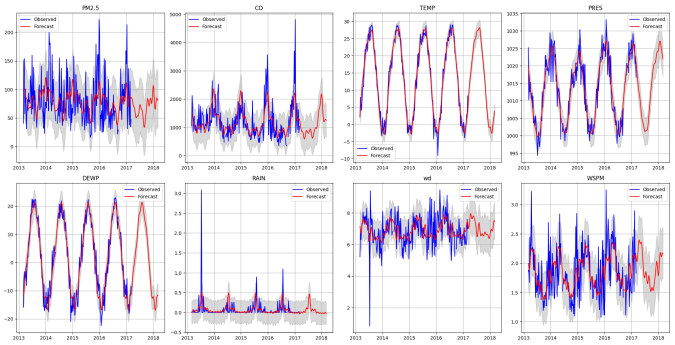


Fig. 17: Prophet on Weekly Data

for each variable. These forecasts can be used to anticipate changes in air quality parameters and make informed decisions regarding environmental management and public health.

Figure 17 is prediction of all the variables using Prophet Model.

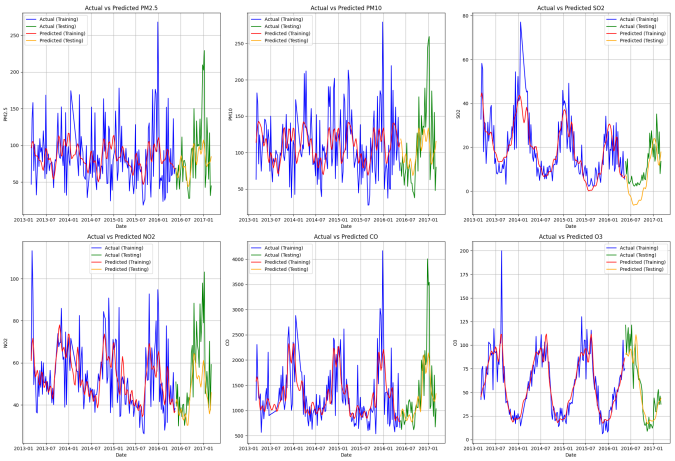


Fig. 18: Prophet on Independent Variables

Figure 18 provides a visual comparison between the actual values and the model's predicted values for each air quality variable (PM2.5, PM10, SO2, NO2, CO, and O3). Here are some inferences that can be drawn from the plot: The blue line represents the actual values of the air quality variable over time, while the red line represents the predicted values on the training data. The green line represents the actual values on the testing data, and the orange line represents the predicted values on the testing data. Ideally, the predicted values should closely follow the actual values. A large deviation between the actual and predicted values indicates lower prediction accuracy. The extent to which the red and orange lines overlap with the blue and green lines indicates how well the model fits the data. A high degree of overlap suggests a good fit, while significant deviations indicate potential shortcomings in the model. The plot allows you to observe any underlying trends and seasonal patterns in the data. The model attempts to capture these patterns to make predictions.The model's ability to capture and predict these outliers can be evaluated based on the proximity of the predicted values to the actual values.Variations in prediction accuracy and model fit across variables may indicate differences in the predictability of different pollutants or potential areas for model improvement.

| Variable | Train MAE | Train MSE | Train RMSE |
|---|---|---|---|
| PM2.5 | 32.00 | 1723.21 | 41.51 |
| PM10 | 36.07 | 2099.62 | 45.82 |
| SO2 | 7.49 | 91.28 | 9.55 |
| NO2 | 10.34 | 199.14 | 14.11 |
| CO | 369.47 | 273480 | 522.95 |
| O3 | 15.26 | 447.18 | 21.15 |

TABLE II: Training Metrics for Prophet Model (weekly)

The model trained on the weekly aggregated air quality data using the Prophet time series forecasting algorithm achieved the performance metrics show in figure . In this case, lower MAE, MSE, RMSE values for certain variables on the testing set compared to the training set suggest that the model may be slightly overfitting to the training data. However, the differences are relatively small, indicating reasonably good generalization. Overall, while the model demonstrates reasonably good performance in predicting air quality variables, particularly for variables like SO2 and O3, there may be some room for improvement, especially in terms of reducing overfitting and enhancing generalization to unseen data. Regular model evaluation, tuning, and potentially incorporating additional features or refining the model architecture could help address these issues. Additionally, considering other evaluation metrics and exploring different modeling approaches might provide further insights into improving predictive performance.

| Variable | Test MAE | Test MSE | Test RMSE |
|---|---|---|---|
| PM2.5 | 28.97 | 1634.19 | 40.43 |
| PM10 | 34.56 | 2205.06 | 46.96 |
| SO2 | 7.40 | 65.32 | 8.08 |
| NO2 | 11.81 | 271.55 | 16.48 |
| CO | 360.37 | 296395 | 544.42 |
| O3 | 10.96 | 207.03 | 14.39 |

TABLE III: Testing Metrics for Prophet Model (weakly)

*3) Multi Layer Perception:* As we have seen, the Multilayer Perceptron (MLP) can be very useful in forecasting time-series data. To implement MLP, we utilized the MLPRegressor model from scikit-learn. Our study illustrates its application in predicting future values based on historical time series data. The methodology begins with partitioning the dataset into training and testing subsets, ensuring adequate data for model training and testing. The selection of pertinent features, such as temperature, pressure, dew point, rainfall, and wind speed, as independent variables, and the identification of dependent target variables, including PM2.5, PM10, SO2, NO2, CO, and O3, sets the stage for model training. After trying and testing methods, we achieved good accuracy by applying a hidden layer of four layers, having 550, 500, 200, 150, and 50 perceptrons respectively in the hidden layers. As for our input and output layers, they will be independent variables and dependent variables respectively.

Through iterative training, the MLPRegressor model learns the intricate patterns and relationships within the data, op-

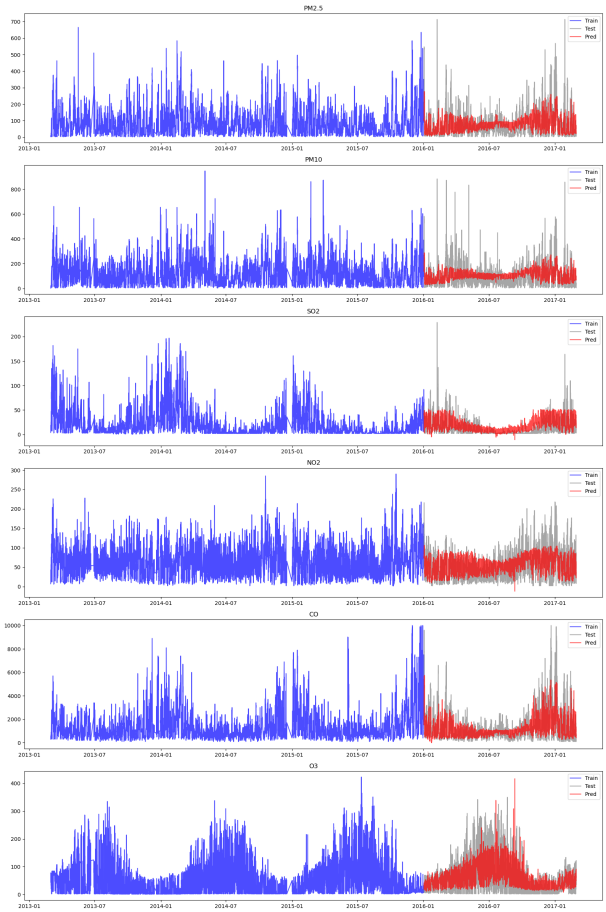


Fig. 19: Prediction using Multi-Layer Perceptron

timizing its parameters to minimize prediction errors. Post-training, the model is deployed to forecast target variables for the test data, with predictions subsequently evaluated against actual values. Evaluation metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) provide quantitative insights into the model's performance across different variables. Analysis of results underscores varying levels of efficacy, with the model demonstrating robust performance for certain variables like SO2, NO2, and O3, while encountering challenges in accurately predicting others such as PM2.5, PM10, and CO, this problem is caused by the large input values for those columns. This study underscores the significance of the MLPRegressor model in time-series analysis, offering a flexible and powerful approach to capturing temporal dependencies and making informed predictions. Further refinement and optimization may enhance the model's performance, particularly for complex or challenging variables, paving the way for broader applications in real-world scenarios.

Table IV displays the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) values obtained from training and testing the models on hourly data. The ability of MLP models to abstract and learn from such noisy data showcases their robustness and learning capacity. In terms of capturing trends and forecasting on an hourly basis, MLP models outperform other alternatives.

| | MAE | MSE | RMSE |
|---|---|---|---|
| **PM2.5** | 45.32 | 4527.56 | 67.29 |
| **PM10** | 57.61 | 6380.81 | 79.88 |
| **SO2** | 11.99 | 255.31 | 15.98 |
| **NO2** | 22.14 | 785.45 | 28.03 |
| **CO** | 596.51 | 850028.95 | 921.97 |
| **O3** | 26.27 | 1284.68 | 35.84 |

TABLE IV: Metrics table for MLP model (hourly)

*4) Vector Auto Regressor (VAR):* VAR models are particularly suitable for capturing the dynamic interdependencies among multiple time series variables, making them well-suited for complex systems where variables influence each other over time.

Initially, our analysis began with data preprocessing to ensure the integrity and quality of our dataset. We meticulously removed missing values from the DataFrame, ensuring that our subsequent analysis would be based on complete and reliable data. Additionally, we created a copy of the DataFrame named data to facilitate further manipulation and analysis, ensuring the integrity of the original dataset.

Subsequently, we partitioned our data into distinct training and testing sets, a critical step in evaluating the performance and generalization ability of our model. We allocated approximately 70% of the data for training purposes, enabling the model to learn from past patterns and relationships within the data. The remaining 30% was reserved for testing, allowing us to assess the model's predictive accuracy on unseen data.

Next, we initiated the VAR model with the training data, laying the foundation for our forecasting endeavors. Leveraging the select_order() method, we embarked on the task of determining the optimal lag order for our model. By setting a maximum lag of 11, we systematically evaluated different lag orders to identify the most suitable one for our dataset. The selected lag order was meticulously summarized and displayed, providing valuable insights into the temporal dynamics of our data.

Following initialization, we proceeded to fit the VAR model to the training data, leveraging the selected lag order to capture the temporal dependencies and relationships among the variables. The fitted model was subjected to rigorous evaluation, with a detailed summary generated to assess its performance and statistical significance. This comprehensive evaluation provided valuable insights into the model's ability to capture the underlying patterns and dynamics within the data.

With the model successfully fitted, we embarked on forecasting future values based on the testing data, leveraging the predictive capabilities of the VAR model. The forecasted values were meticulously stored in a DataFrame named predictions, facilitating a detailed comparison with the actual test data. Additionally, we computed and printed the mean values of both the test data and forecasted values, enabling a quantitative assessment of the model's predictive accuracy.

In addition to forecasting future values, we conducted an extended forecasting step to predict values for the next 100

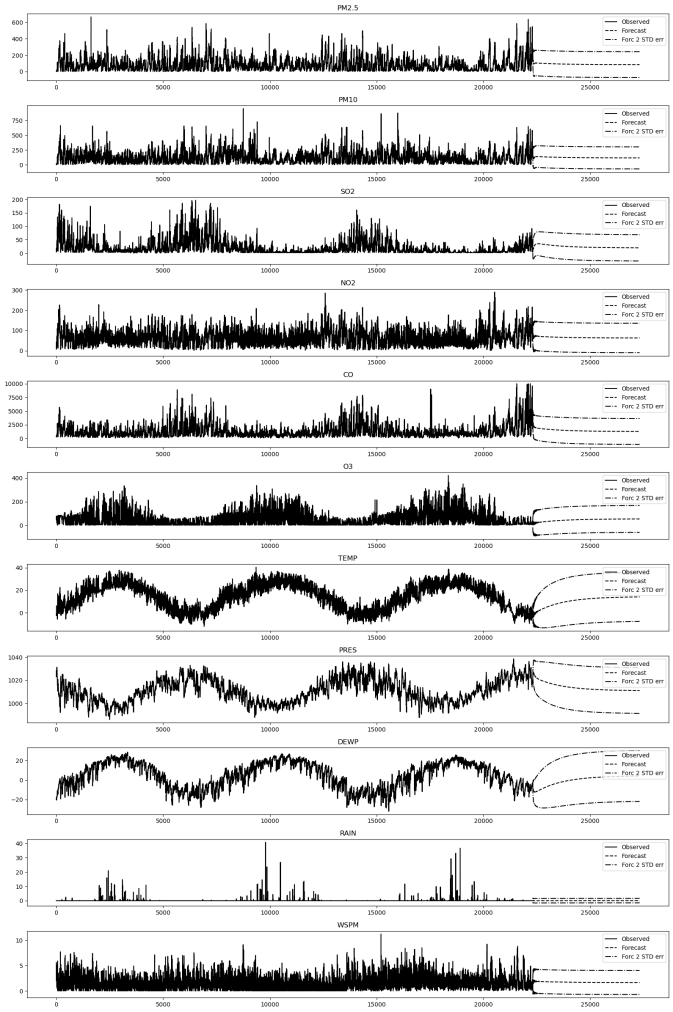


Fig. 20: Prediction using Vector Auto Regressor

time steps. The forecasted values were stored in a DataFrame named df_forecast, indexed with datetime values starting from '2015-01-01' and an hourly frequency. This extended forecasting step provided valuable insights into the long-term trends and patterns within the data, facilitating informed decision-making and strategic planning.

Finally, to aid in the interpretation and visualization of the forecasted values, we leveraged the plot_forecast() method to generate insightful visualizations. By adjusting the figure size for clarity, we ensured that the plots effectively conveyed the forecasted trends and patterns, enabling stakeholders to make informed decisions based on the model's predictions.

## IV. Comparison

We have employed three primary machine learning models in our analysis, namely Prophet, Multi-Layer Perceptron Neural Network, and SARIMAX model. To assess the performance and accuracy of these models, we utilized metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Error (MSE) on our dataset. Below, we provide a comparative analysis of the insights gained and the performance achieved by each model.

The SARIMA model demonstrates proficiency in capturing both seasonality and inherent dynamics within time series data. Notably, it exhibits strong performance in predicting PM10 concentrations, as indicated by its RMSE value of 45.77, suggesting accurate trend capture. However, the model showcases limitations in perfectly capturing all fluctuations, occasionally resulting in instances of over and underestimation.

Prophet emerges as well-suited for datasets with pronounced seasonality, requiring minimal data preprocessing. It delivers promising results for variables such as SO2 (RMSE: 7.40) and O3 (RMSE: 17.36), underscoring its efficacy in capturing seasonal patterns.

The MLP neural network model demonstrates robust performance, particularly for variables like SO2 (RMSE: 15.98) and NO2 (RMSE: 28.03). However, it encounters challenges with variables like PM2.5 (RMSE: 67.29), possibly due to the large input values necessitating further model architecture optimization. While MLP models exhibit proficiency in handling noisy data, careful feature selection and hyper parameter tuning are imperative for optimal outcomes.

In summary, Prophet and SARIMA excel in capturing seasonal patterns, while MLP can achieve high accuracy for specific variables with appropriate configuration. Both SARIMA and Prophet may benefit from further tuning to mitigate overfitting to the training data.

## V. Future Directions

For Future Directions, there are several potential avenues for further enhancing the performance and applicability of our models. Firstly, there is an opportunity to integrate additional external factors, such as meteorological data and traffic patterns, into our predictive models to better understand their influence on air quality. Prophet model and other Uni-variate models can be used in the pipeline to incorporate categorical columns, providing a more comprehensive analysis. Secondly, exploring advanced modeling techniques, including machine learning algorithms (like Random Forest, Boosting algorithms) and deep learning architectures (like neural networks, long short term memory), beyond traditional time series analysis could improve forecasting accuracy. Vector Auto Regressor can be improved and hyper-tuned to forecast more accurately, and to mitigate variance. Additionally, incorporating spatial analysis and geospatial modeling techniques can help identify spatial patterns and hotspots of air pollution across different regions of Beijing & China. RNN and LSTM can be used as models to learn and take advantage of deep learning models and their capacity to learn complex patterns. We can also impute more data to tackle the issue of missing values and investigate further into feature engineering techniques to better understand the relationships between pollutants and meteorological variables. These future directions aim to contribute to the advancement of air quality prediction and management strategies, ultimately benefiting public health and environmental sustainability.

## VI. Conclusion

In this project, we conducted an in-depth analysis of air quality data from the 'Beijing Multi-Site Air Quality' dataset, focusing on forecasting key pollutants such as PM2.5,

PM10, SO2, NO2, CO, and O3. Our exploration encompassed various machine learning algorithms and time series analysis techniques, aiming to accurately predict air pollution levels. Through rigorous experimentation, we evaluated the performance of models including SARIMA, Prophet, Vector Auto Regression (VAR), and Multi-Layer Perceptron (MLP) neural networks. Each model demonstrated strengths in capturing different aspects of temporal dynamics and spatial variations in air quality parameters. SARIMA excelled in capturing both seasonal and non-seasonal variations, while Prophet effectively handled strong seasonal effects. MLP neural networks showcased flexibility in capturing nonlinear relationships and temporal dependencies. However, all models encountered challenges, particularly in accurately predicting variables with complex dynamics such as PM2.5 and PM10. Moving forward, avenues for improvement include leveraging ensemble methods, incorporating exogenous variables, and continuous monitoring and evaluation of model performance. Despite these challenges, our analysis represents a significant step towards enhancing our understanding of air pollution trends and improving our ability to forecast future air quality conditions, ultimately contributing to efforts to mitigate the adverse effects of air pollution on public health and the environment.

## REFERENCES


[1] Y. Tian, Y. Jiang, Q. Liu, D. Xu, S. Zhao, L. He, H. Liu, and H. Xu, "Temporal and spatial trends in air quality in beijing," *Landscape and Urban Planning*, vol. 185, p. 35–43, May 2019.

[2] L. Zhao, Z. Li, and L. Qu, "Forecasting of beijing pm2.5 with a hybrid arima model based on integrated aic and improved gs fixed-order methods and seasonal decomposition," *Heliyon*, vol. 8, no. 12, Dec 2022.

[3] T. Smith, "Autocorrelation: What it is, how it works, tests." [Online]. Available: https://www.investopedia.com/terms/a/autocorrelation.asp#:~:text=Autocorrelation%20represents%20the%20degree%20of,value%20and%20its%20past%20values.