Proceedings of 14th IOE Graduate Conference
Peer Reviewed
Year: 2023 Month: December Volume: 14
ISSN: 2350-8914 (Online), 2350-8906 (Print)

Comparative Analysis of Time Series Forecasting Models for Predicting PM2.5 level in Kathmandu : SARIMA, Prophet and XGBoost

Isha Khanal [a], Om Prakash Dhakal [b], Manoj Kumar Guragain [c], Pukar Karki [d], Bhagat Pandey [e]

[a, b, c, d] *Department of Electronics & Computer Engineering, Purwanchal Campus, IOE, Tribhuvan University, Nepal*

[a] ishakhanal@ioepc.edu.np, [b] ODhakal1@gmail.com, [c] mkguragai@gmail.com, [d] pukar@ioepc.edu.np, [e] pandeybhagat369@gmail.com

Abstract

This study conducts an in-depth comparative analysis of time series forecasting models with a specific focus on predicting PM2.5 levels in Kathmandu Valley, Nepal. PM2.5, fine particulate matter with a diameter of 2.5 micrometers or less, is chosen as the central parameter of interest due to its critical role in air quality assessment and its significant impact on public health. Kathmandu Valley, a rapidly urbanizing region, faces severe air quality challenges, making it an ideal study area. PM2.5 is particularly concerning because it can penetrate deep into the respiratory system, leading to various adverse health effects. As such, accurate forecasting of PM2.5 levels is crucial for air quality management and public health initiatives in the region. This research involves the comprehensive collection and preprocessing of historical air quality data and relevant meteorological variables. Three robust time series forecasting models : SARIMA,Prophet and XGBoost are meticulously developed, fine-tuned, and rigorously evaluated. The objective is to identify the most effective model for forecasting PM2.5 concentrations in Kathmandu Valley. The study not only seeks to determine the best-performing model but also explores the potential implications of accurate PM2.5 predictions. These implications extend to informing local air quality management strategies, facilitating early warning systems, and ultimately contributing to better environmental and public health outcomes in Kathmandu Valley. The findings of this research hold significant value for regions facing similar air quality challenges worldwide and underscore the importance of predictive modeling in addressing critical environmental issues.

Keywords

Time series forecasting, PM 2.5, SARIMA, Prophet, XGBoost

1. Introduction

Air quality plays a vital role in sustaining human life as it directly impacts on our well-being. Monitoring and understanding air quality are crucial for safeguarding our health. Unfortunately, air pollution has become a significant global issue, leading to numerous physiological disorders and even respiratory fatalities. According to scientific evidence, air pollution poses the single greatest environmental risk. The rapid industrialization and population growth have contributed to escalating levels of toxic gas emissions, resulting in a decline in air quality. Hazardous substances contaminate the air, posing severe health risks to individuals.

To assess and communicate air pollution levels, the Air Quality Index (AQI) employs a numerical scale. This index considers twelve parameters or air pollutants, including nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), particulate matter with a diameter of 10 microns or less (PM10), particulate matter with a diameter of 2.5 microns or less (PM2.5), ammonia (NH3), and benzene. In most of the applications, the six pollutants PM10, PM2.5, SO2, NO2, CO, and O3 are used to calculate the air quality index (AQI). However, the choice of specific pollutants depends on the intended purpose and several factors like data availability, measurement methods, and monitoring frequency. A higher AQI value indicates a greater level of air contamination, which can pose significant health risks.

PM2.5 pollution is a complex issue with far-reaching consequences. These tiny particles, often originating from combustion processes and industrial emissions, can penetrate deep into the respiratory system, leading to a range of health problems, including respiratory diseases, cardiovascular issues, and even premature mortality. In the context of Kathmandu Valley, where rapid urbanization and vehicular emissions are prevalent, the adverse effects of PM2.5 pollution are of profound concern. The valley's unique topography, seasonal weather patterns, and pollution sources make it a challenging but crucial area for air quality research and intervention.

The primary purpose of this research is to develop a robust framework for the prediction of PM2.5 levels in Kathmandu Valley. Accurate forecasting of PM2.5 concentrations can serve as a vital tool for air quality management and mitigation efforts.The findings of this research can be leveraged for developing predictive systems for other cities as well in the future and it will also help in the formulation and implementation of effective air quality management strategies, such as targeted emission reduction measures, urban planning, and public awareness campaigns. Moreover, the predictive models can be integrated into real-time air quality monitoring systems, providing timely and accurate information to the general public, policymakers, and relevant authorities.

2. Background and rationale

Nepal's air quality ranked at the bottom among 180 countries in terms of Environmental Performance Index (EPI) in 2020

[1]. Kathmandu city, in particular, is recognized as one of the most polluted cities in Asia [2]. Nepal is experiencing rapid urbanization, with approximately 6.2 million Nepalese residing in urban areas as of 2020 [3]. This trend is expected to continue, and it is projected that the urban population could reach 60 million by 2040 [4]. With a population density of 13,225 individuals per square kilometer, Kathmandu is the largest urban agglomeration in Nepal, accounting for 20 percent of the urban population within an area of 50.67 square kilometers [5]. The surge in vehicle ownership is also evident in Nepal, with the number of registered vehicles reaching approximately 90,000 in the year 2015/16, as reported by the Department of Transport Management [6]. Vehicle emissions, particularly from diesel-powered vehicles, are a significant concern as they are considered highly toxic pollutants and carcinogens. Apart from vehicles, the air pollution in Kathmandu is worsened by factors such as unregulated road excavation for the ongoing Melamchi water project, brick kilns, unplanned road expansion, improper disposal of construction materials along busy roadsides, and the frequent use of old, inefficient automobiles that contribute to pollution [7]. To examine the seasonal variations in air pollution, Karki et al. conducted a study in Kathmandu Valley in 2015 [8]. Their findings revealed that $NO2, CO$, and PM2.5 concentrations were highest during the winter and spring seasons. Winter had the maximum levels of these pollutants, while autumn had the minimum levels. Given the adverse consequences of air pollution, there is an urgent need to develop effective strategies for monitoring, understanding, and predicting air quality levels in Nepal. The Air Quality Index (AQI) serves as a standardized metric to communicate air quality information to the public and policymakers. However, accurately forecasting AQI levels in Nepal remains a challenge due to the complex and dynamic nature of air pollution. The rationale behind this thesis topic lies in the potential of predictive modeling to address these challenges. By analyzing extensive data sets from existing air quality monitoring stations, it becomes possible to develop models that can accurately forecast PM 2.5 levels which plays significant role in determining AQI. These models can provide valuable insights into the factors influencing air pollution, including pollutant emissions, weather conditions, and geographical features.

3. Literature Review

Different algorithms have been found to be effective for different datasets and pollutants. Some of the most popular algorithms include SVM, M5P, ANN, gradient boost,XGBoost, AR, SVR, LSTM, ARIMA, SARIMA,prophet etc. The choice of algorithm depends on a number of factors, including the size and quality of the dataset, the type of pollutant being predicted, and the desired accuracy of the prediction. Deep learning methods have also been shown to be effective for predicting air pollution. However, these methods can be more computationally expensive than traditional machine learning algorithms. The results of these studies suggest that machine learning can be a valuable tool for predicting air pollution. However, more research is needed to determine the best algorithm for a particular application. Some of the specific examples of studies that have used machine learning to predict air pollution are discussed here.

JK Sethi and Mittal used ARIMA and VR models to analyze the prediction of AQI in univariate and multivariate models respectively [9]. They found out that the ARIMA model best predicted AQI based on their RMSE and MAE score. Ruchita has Integrated Gated Recurrent Unit with Long Short-Term Memory to create a hybrid model which proved to improve RMSE score when compared with other machine learning models [10]. In one of the research projects done in Zhale, Lebanon by A. Atoui et.al, Exponential Smoothing, TBATs and SARIMA models have been implemented. They have concluded that SARIMA is most accurate model to predict AQI for Zhale [11].

In the context of Nepal as well, some studies have been conducted in the past. One of the earliest studies was conducted by Ghimire et al in 2012, who used a multiple linear regression (MLR) model to predict PM2.5 levels in Kathmandu [12]. The study found that the MLR model was able to predict PM2.5 levels with a high degree of accuracy. Another study, conducted by Adhikari, used a support vector machine (SVM) model to predict PM2.5 levels in Kathmandu [13]. The study found that the SVM model was able to predict PM2.5 levels with a slightly higher degree of accuracy than the MLR model. More recently, Liu and Chen used a hybrid method to predict PM2.5 levels in Kathmandu [14]. The hybrid method consisted of a binary grey wolf optimization-based feature reduction, discrete wavelet packet transform-based decomposition, extreme learning machine and adaptive boosting-based prediction model.The hybrid method consisted of a binary grey wolf optimization-based feature reduction, discrete wavelet packet transform-based decomposition, extreme learning machine and adaptive boosting-based prediction model. The study found that the hybrid method was able to predict PM2.5 levels with the highest degree of accuracy of any of the studies reviewed.

4. Data Collection and Pre-processing

4.1 Data Collection

Department of Environment has made AQMS data publicly available in its official website. The data from the year 2016-2021 has been published under its Publications. Hence, data set has been collected from the official government website. The historical data of major two pollutants, PM 2.5 and ozone are present in the excel format. Only PM 2.5 has been considered in this project it has more significant impact then the ozone value in air quality.

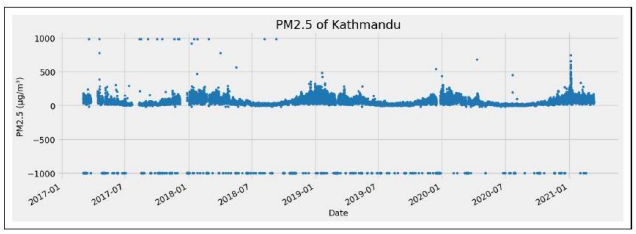

Figure 1: Graphical representation of PM 2.5 value over the range of 2016-2021 before pre-processing

4.2 Data Pre-processing

The dataset used for PM2.5 level forecasting underwent thorough preprocessing to ensure the quality and reliability of the data.

The preprocessing steps included addressing missing values and mitigating the impact of outliers, which are essential for creating accurate forecasting models.

Handling Missing Values In the dataset, two types of missing values are prevalent: NaN values and -999.0 values. NaN values account for 1.64 percentage of the dataset, while -999.0 values constitute 7.54 percentage. Collectively, these missing values represent 9.06 percentage of the dataset. This absence of data significantly impacts the analysis, particularly within a specific interval of the time series, resulting in less than optimal outcomes. The SARIMA model, unfortunately, lacks inherent capabilities to manage missing values autonomously. To address this, a back-filling technique was employed to fill these gaps in the data. However, the model's performance was sub-optimal even after applying this method. On the contrary, models like XGBoost and Prophet possess the ability to handle missing values without external intervention. While attempts were made to apply back filling to these models as well, the analysis reveals that omitting the back-filling step yielded better results. This suggests that these models can effectively handle missing data without compromising their forecasting accuracy.

Outlier Treatment The dataset underwent a thorough outlier detection process, involving meticulous graphical visualization for PM 2.5 values. This meticulous manual examination aimed to identify potential outliers, which accounted for approximately 0.12 percentage of the dataset. During this analysis, outliers in the higher range were addressed by replacing them with the 75th percentile value, while negative outliers (those falling below the lower range) were similarly transformed using the 25 th percentile value. This rigorous approach ensured that outliers, irrespective of their position in the distribution, were appropriately managed.

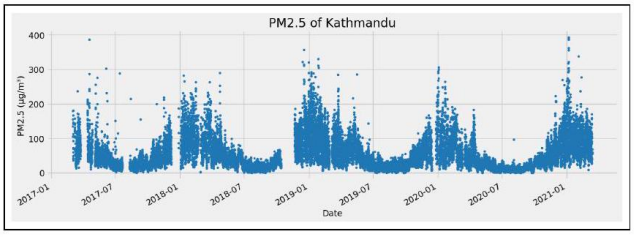

Figure 2: Graphical representation of PM 2.5 value over the range of 2016-2021 after pre-processing

The pre-processed data is then split into train and test data. The data over the range of year 2017-2019 has been used as train data while the data over the range of year 2020-2021 has been used as test data which has been later used to validate the training model.

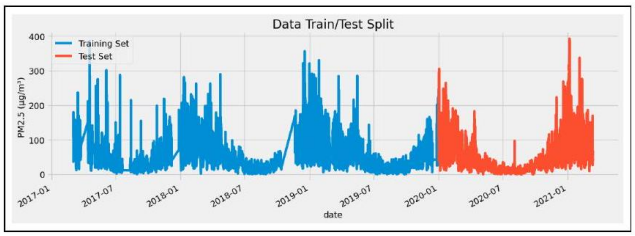

Figure 3: Graphical representation of pre-processed dataset after splitting

5. Methodology

5.1 Training Models

There are a variety of statistical, machine learning and deep learning models that can be used for time series forecasting, such as ARIMA, SARIMA, Prophet, XGBoost, LSTM, Temporal Fusion Transformer, etc. The choice of algorithm will depend on the characteristics of the data and the desired accuracy of the model.SARIMA,Prophet and XGBoost in particular has been evaluated in this study.

5.1.1 SARIMA

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. Autoregressive Integrated Moving Average is a statistical time series forecasting model that combines autoregressive, differencing, and moving average components to analyze and predict data points based on their historical patterns and relationships.SARIMA includes additional components to capture the seasonal patterns in the data. The components of SARIMA are as follows:

Autoregression(AR) The autoregressive component in SARIMA is similar to that in ARIMA and represents the correlation between the current observation and its past observations. The "p" in SARIMA (p, d, q)(P, D, Q)s represents the order of autoregression, which is the number of past observations used to predict the current value, considering both the seasonal and non-seasonal patterns. Mathematically, an AR(p) model can be represented as:

$$y(t) = c + \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + \varepsilon(t) \quad (1)$$

where $y(t)$ is the value at time t, ϕ_1 to ϕ_p are the autoregressive coefficients, and $\varepsilon(t)$ is the white noise or error term.

Differencing (I) The differencing component in SARIMA is also similar to ARIMA and is used to remove trends and make the data stationary. The "d" in SARIMA (p, d, q) (P, D, Q)s represents the order of differencing, which accounts for the non-seasonal differences. Mathematically, differencing can be represented as:

$$y'(t) = y(t) - y(t-1) \quad (2)$$

where $y'(t)$ is the differenced series.

Moving Average (MA) The moving average component in SARIMA is similar to ARIMA and represents the correlation between the current observation and the residual errors from past observations. The "q" in SARIMA (p, d, q) (P, D, Q)s represents the order of the moving average, which considers both the seasonal and non-seasonal patterns. Mathematically, an MA(q) model can be represented as:

$$y(t) = c + \varepsilon(t) + \theta_1\varepsilon(t-1) + \theta_2\varepsilon(t-2) + \ldots + \theta_q\varepsilon(t-q) \quad (3)$$

where $\varepsilon(t)$ is the white noise or error term, θ_1 to θ_p are the moving average coefficients, and c is a constant term.

Seasonal Autoregression (SAR) The seasonal autoregressive component captures the correlation between the current observation and past observations at the same seasonality period. The "P" in SARIMA (p, d, q) (P, D, Q)s represents the seasonal order of autoregression.

Seasonal Differencing (SI) The seasonal differencing component is used to remove the seasonal trends and make the data stationary at the seasonal period. The "D" in SARIMA (p, d, q)(P, D, Q) s represents the seasonal order of differencing

Seasonal Moving Average (SMA) The seasonal moving average component captures the correlation between the current observation and residual errors from past observations at the same seasonal period. The " Q " in SARIMA (p, d, q)(P, D, Q)s represents the seasonal order of the moving average.

SARIMA is particularly useful for air quality index forecasting because it takes into account these seasonal variations. By incorporating seasonal components (SAR, SI, and SMA) into the model, SARIMA can capture the recurring patterns and produce more accurate forecasts. This helps in predicting air quality levels for specific time periods with higher precision, making it a valuable tool for environmental monitoring agencies, city planners, and public health authorities.

5.1.2 Prophet

Prophet is a forecasting model developed by Facebook's Core Data Science team. It is designed to handle time series data and is particularly well-suited for forecasting with strong seasonal patterns, holidays, and multiple seasonalities. Here's an explanation of the key features and workings of the Prophet model:

Additive Model Prophet uses an additive model that decomposes a time series into three main components: trend, seasonality, and holidays. This decomposition helps capture the underlying patterns in the data.

Trend Component The trend component represents the overall direction or trajectory of the time series data. Prophet allows for both linear and non-linear trend modeling, making it flexible in capturing different trend shapes.

Seasonality Component Prophet accommodates multiple types of seasonalities, such as daily, weekly, and yearly patterns. It uses Fourier series to model seasonality, enabling the model to capture complex seasonal variations.

Holiday Effects Prophet allows you to include holiday effects in your time series analysis. You can specify custom holiday dates and their impacts on the data. This is especially useful for modeling the effects of holidays and special events on the time series.

Handling Missing Data Prophet is robust in handling missing data and outliers, making it suitable for real-world datasets that may contain gaps or anomalies.

Automatic Change point Detection The model automatically detects change points in the data, indicating where the time series behavior shifts. This helps in identifying important turning points.

Uncertainty Estimation Prophet provides uncertainty intervals (prediction intervals) for its forecasts, allowing you to assess the level of confidence in the predictions.

Scalability It is scale-able and can handle large datasets efficiently, making it practical for a wide range of applications.

Customization Prophet offers various parameters for customization, such as the ability to set prior scales, control seasonality components, and fine-tune the forecasting process.

Ease of Use Prophet is designed to be user-friendly and accessible to users with limited expertise in time series forecasting. It provides a straightforward interface for data preparation and model configuration.

Prophet has gained popularity for its ability to produce accurate forecasts with minimal effort, making it a valuable tool for businesses and researchers across various domains, including retail, finance, and demand forecasting. It is particularly well-suited for data sets with strong seasonal patterns and holidays, where traditional time series models may require more complex modeling techniques.

5.1.3 XGBoost

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. XGBoost is known for its speed, performance, and ability to handle complex relationships in the data.Key characteristics and features of XGBoost include:

Gradient Boosting XGBoost is based on the concept of gradient boosting, which builds an ensemble model by combining the predictions of multiple weak learners, typically decision trees. It iteratively adds decision trees, and each subsequent tree corrects the errors made by the previous ones, leading to a stronger and more accurate final model.

Regularization XGBoost incorporates regularization techniques to prevent overfitting and enhance model generalization. It includes L1 (Lasso) and L2 (Ridge) regularization terms in the objective function, controlling the complexity of the individual decision trees and the overall model.

Tree Pruning XGBoost uses a process called "tree pruning" to reduce the depth of decision trees, which helps in reducing model complexity and avoiding overfitting. Tree pruning is performed based on the importance of features and their contributions to the model's performance.

Handling Missing Values XGBoost has built-in support for handling missing values in the data. It can automatically learn how to treat missing data during the training process, making it convenient to work with datasets that have incomplete information.

Cross-validation XGBoost supports k-fold cross-validation, which helps in estimating the model's performance and selecting optimal hyperparameters. This aids in finding the best trade-off between model complexity and performance.

Feature Importance XGBoost provides a measure of feature importance, allowing users to understand which features have the most significant impact on the model's predictions. This feature is valuable for feature selection and data analysis.

Parallel Processing XGBoost is designed to take advantage of parallel processing capabilities, making it computationally efficient and capable of handling large datasets quickly.

Flexibility XGBoost can be used for various types of machine learning tasks, including regression, classification, ranking, and user-defined objectives. It is widely used in competitions like Kaggle due to its versatility and ability to deliver high accuracy.

Multiple Language Support XGBoost is implemented in multiple programming languages, including Python, R, Java, and Julia, making it accessible to a wide range of data scientists and developers.

XGBoost uses a combination of decision trees and gradient boosting to optimize a loss function and improve the accuracy of the predictions. XGBoost can handle complex and nonlinear relationships in the data, as well as incorporate external features, such as meteorological data, traffic data, or emission data, that may affect the air quality. XGBoost has been shown to outperform other machine learning methods, such as linear regression, random forest, or neural networks, in some studies of AQI prediction

5.2 Model Evaluation

Once the model has been trained, it is important to evaluate its performance. To analyze the performance of a machine learning model we need some metrics. These metrics are statistical criteria that can be used to measure and monitor the performance of a model. As this study deals with prediction, RMSE has been considered as the performance metrics.

5.2.1 Root mean square error (RMSE)

RMSE is the square root of the average of the squared difference between the target value and the value predicted by the model. It is the square root of mean square error (MSE). The implementation is very much similar to MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{4}$$

Where:

RMSE : Root Mean Squared Error

y_i : The observed (actual) value for data point i

$\hat{y}_i$: The predicted value for data point i

n : The total number of data points

5.2.2 Mean Absolute Error (MAE)

MAE is the arithmetic average of the difference between the ground truth and the predicted values. It can also be defined as measure of errors between paired observations expressing the same phenomenon. It tells us how far the predictions differed from the actual result. Mathematical representation for MAE is given below.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{5}$$

$n =$ number of data points or observations

$Y_i =$ actual values (ground truth) for the i-th data point

$\hat{Y}_i =$ predicted values for the i-th data point

6. Results and Discussion

The three different models namely SARIMA, Prophet and XGBoost has been experimented over the training data set. MAE and RMSE of each models has been calculated to evaluate the performance of the models.

6.1 SARIMA

Specific parameters, order = $(0,1,3)$, and seasonal order = $(0,1,1,12)$ are utilized to set up the SARIMA model. Subsequently, the model's performance is assessed using information criteria: AIC (Akaike Information Criterion): 288.57 BIC (Bayesian Information Criterion): 298.03 HQIC (Hannan-Quinn Information Criterion): 292.16 Lower values across these criteria indicate a superior model fit, considering the trade-off between model complexity and goodness of fit. Similarly, in the experiment, the calculated value of RMSE is 16.991 for trained SARIMA model with mean absolute error being 54.839. The figure below shows the validation graph plot for SARIMA model :

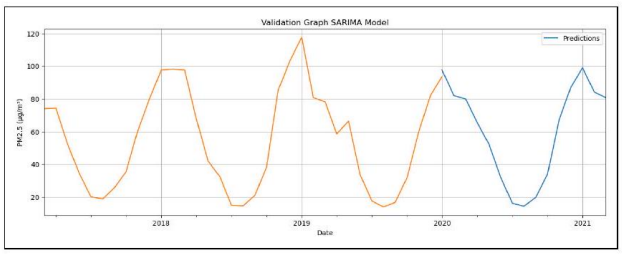

Figure 4: Validation graph for SARIMA model

The validation graph indicates that the predicted values closely follow the same curve direction as the actual values.

6.2 Prophet

The Prophet model is trained utilizing the following optimal parameters:

Ip (Log Probability): Currently stands at 39798.8, indicating the log posterior density.

k (Trend Growth Rate): Presently set to 0.341135 , representing the growth rate within the trend component.
m (Initial Trend Level): Currently estimated at 0.12248, indicating the baseline or starting level of the trend.

delta (Seasonal Effects): An array of coefficients capturing various seasonal variations in the data.

sigma obs (Observation Noise): Currently measured at 0.06778 , representing the standard deviation of observation noise.

beta (Additional Regressors): Array reflecting the impact of external factors or regressors on predictions.

trend (Modeled Trend): An array outlining the predicted trend over time.

During the training process, the Prophet model showcases promising performance metrics:

RMSE (Root Mean Squared Error): Currently calculated as 39.5599, indicating the root of the mean squared differences between predicted and actual values. MAE (Mean Absolute Error): Currently stands at 28.262, signifying the mean of absolute differences between predicted and actual values. The RMSE value of 39.5599 and MAE value of 28.262 demonstrate the Prophet model's accuracy in predicting the target variable during the ongoing training phase. Lower values in these metrics reflect higher accuracy in predictions, emphasizing the model's capability to capture underlying patterns within the dataset.

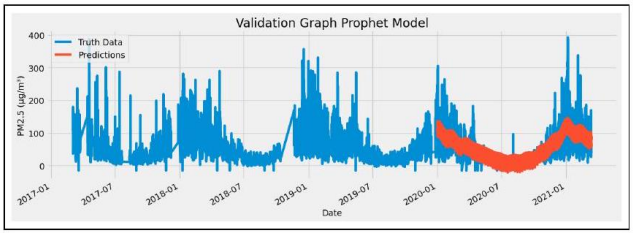

Figure 5: Validation graph for Prophet model

The validation graph indicates that the predicted values closely aligns with the actual values as they follow the same curve direction but increased value of RMSE suggests that there is some discrepancies between the predicted and actual values .

6.3 XGBoost

When hyper-tuning an XGBoost model, several methods optimize its performance. Grid search exhaustively tests predefined hyperparameter values, while random search randomly samples from specified ranges. Utilizing cross-validation techniques, like k-fold, allows for robust evaluation across different parameter settings using validation or test sets. Defining a parameter space with ranges for hyperparameters, such as learning rate or max depth, facilitates systematic exploration. Through iterative optimization, diverse hyperparameter combinations are tested to identify the set that maximizes model performance. Additionally, Bayesian optimization methods leverage past evaluations to guide the search efficiently, reducing the number of iterations needed for finding optimal hyperparameters. These approaches systematically explore and evaluate hyperparameter configurations to enhance the XGBoost model's predictive accuracy and generalization capability on a given dataset. For hyper tuning the XGBoost model, we opted for grid search as our primary method for hyperparameter tuning. This systematic technique exhaustively explores a predefined grid of hyperparameter values, diligently evaluating each combination's impact on the model's performance. Our grid search procedure yielded the best parameters 'learning rate': 0.1 , 'max depth': 3 , 'n estimators': 100, optimizing the XGBoost model's configuration to enhance predictive accuracy and generalization on the provided dataset. Additionally, the tabulated hyperparameters used in the grid search are as follows:

Hyperparameter	Obtained Value
learning_rate	0.1
max_depth	3
n_estimators	100

Table 1: Tabulated Hyperparameters for Grid Search

Moreover, after training the XGBoost model with these optimized hyperparameters, the model's performance metrics were evaluated. The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were found to be 30.36 and 4.65, respectively. These values represent the model's accuracy in predicting the target variable, where lower values indicate better performance in capturing the underlying patterns within the dataset.

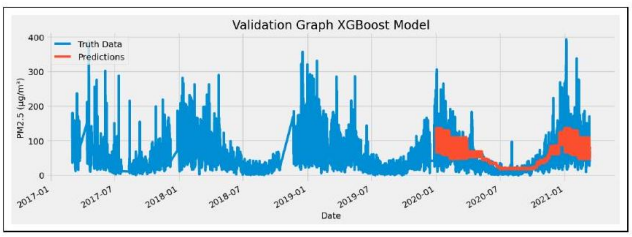

Figure 6: Validation graph for XGBoost model

The validation graph for the trained XGBoost model typically shows a close alignment between the predicted values and the actual data points, with relatively small and random prediction errors.

The SARIMA model has the lowest RMSE (16.991), indicating the smallest average error in predicting PM2.5 levels. If minimizing prediction errors is a top priority, SARIMA performs the best in this aspect. The XGBoost model has the lowest MAE (4.65), meaning it, on average, makes smaller absolute errors in predictions compared to the other models. This indicates that XGBoost tends to have a more accurate prediction in terms of magnitude. The Prophet model has the highest RMSE and an intermediate MAE, which suggests it has a higher average prediction error compared to SARIMA but a lower error compared to SARIMA in terms of MAE. It falls between the other two models in both RMSE and MAE.

7. Conclusion

The main aim of this research is to predict PM 2.5 value to determine the AQI which can be used later for implementing effective regulations to reduce AQI. For this purpose it's essential to prioritize accurate magnitude predictions, as the AQI calculation relies on specific concentration thresholds for pollutants. Therefore, minimizing the Mean Absolute Error (MAE) becomes crucial in this context, as MAE measures the accuracy of predictions in terms of magnitude. Lower MAE indicates better performance in this scenario.Considering the specific application to predict AQI, the XGBoost model stands out as the best-suited option. The XGBoost model has the lowest MAE (4.65), indicating that, on average, it makes smaller absolute errors in predicting PM2.5 values. This aligns well with the goal of accurately determining AQI values, which depend on the accurate magnitude of pollutant concentrations. Xgboost model has also been used by S.Bhatta and his team [15] to predict pm 2.5 of Kathmandu with similar data and their MAE score has been found to be 15.82 as per their study. Improvement in MAE score is observed in this experiment. While SARIMA has a lower RMSE, its relatively high MAE suggests that it might not be as well-suited for accurate magnitude predictions as XGBoost. Prophet, while performing better than SARIMA in terms of MAE, still falls behind XGBoost.Therefore, the XGBoost model is the recommended choice, as it provides the best balance between RMSE and MAE and is likely to provide more accurate magnitude predictions of PM2.5 values, which are essential for calculating AQI and implementing effective air quality regulations.

Future Enhancements

For future enhancements in the experiment aimed at predicting PM2.5 values for AQI determination using the XGBoost model, several key improvements can be considered. These include the incorporation of additional relevant features through feature engineering, fine-tuning model hyperparameters, and exploring ensemble techniques to leverage model strengths. Time series decomposition methods, advanced machine learning models, and robust cross-validation strategies should be explored to enhance prediction accuracy. Moreover, the development of uncertainty estimation methods, support for real-time monitoring, and the consideration of spatial dependencies can provide more comprehensive and actionable insights. Model interpretability, data quality improvement, and the inclusion of external environmental factors should also be prioritized. Effective communication tools and collaboration with stakeholders will further enhance the model's utility in informing regulatory decisions and promoting better air quality management

Acknowledgments

The completion of this research paper owes much to the invaluable support and guidance of numerous individuals. Foremost, my deepest gratitude goes to my supervisor, Mr. Om Prakash Dhakal, whose mentorship and expert insights significantly shaped the research process. His wisdom and encouragement were indispensable throughout this endeavor. I extend sincere appreciation to Mr. Manoj Kumar Guragain, the Head of the Department, for fostering an environment conducive to research and learning. Special thanks to Mr. Pukar Karki for his dedicated mentorship, valuable feedback, and assistance in navigating challenges along the way. My heartfelt thanks also go to Mr. Bishnu Chaudhary, the Program Coordinator, for facilitating this academic journey and providing valuable support. Lastly, I am immensely grateful to my husband, Mr. Bhagat Pandey, for his unwavering support, encouragement, and invaluable assistance throughout the project. His dedication was a pillar of strength.

References

[1] Yale University. Environmental performance index 2018. *Yale University, New Haven, Conn, USA*, 2018.

[2] K Parajuly. Clean up the air in kathmandu. *Nature*, 533(7603):321–321, 2016.

[3] Nepal population (live). https://www.worldometers.info/world-population/nepal-population/.

[4] World Health Organization (WHO). *Nepal Urban Health Profile*. 2017.

[5] Kathmandu metropolitan city. https://www.kathmandu.gov.np/ne/node/94, 2020.

[6] Department of Transport Management (DoTM). Vehicle registered in bagmati zone in fiscal year 072-73, 2017.

[7] B Saud and G Paudel. The threat of ambient air pollution in kathmandu, nepal. *Journal of Environmental and Public Health*, pages 1–7, 2018.

[8] K. B. Karki, P. Dhakal, and S. L. Shrestha et al. *Situation Analysis of Ambient Air Pollution and Respiratory Effects in Kathmandu Valley*. Kathmandu, Nepal, 2015.

[9] JK Sethi and M Mittal. Analysis of air quality using univariate and multivariate time series models. In *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 823–827, 2020.

[10] R Patil. Prediction of air quality index data using machine learning and deep learning. Master's thesis, National College of Ireland, 2021.

[11] A Atoui, K Slim, SA Andaloussi, R Moilleron, and Z Khraibani. Time series analysis and forecasting of the air quality index of atmospheric air pollutants in zahlé, lebanon. *Atmospheric and Climate Sciences*, 12(4):167, 2022.

[12] N Ghimire, S Shrestha, B Khanal, and Y Aryal. Prediction of pm 2.5 concentration in kathmandu using multiple linear regression model. *Journal of Environmental Management*, 108:112–117, 2012.

[13] A Adhikari, S Adhikari, P Shrestha, and S Pokhrel. Prediction of pm 2.5 concentration in kathmandu using support vector machine. *Journal of Environmental Management*, 191:84–91, 2017.

[14] Y Liu and Y Chen. Predictive modeling and analysis of air quality - visualizing before and during covid-19 scenarios. *PLOS ONE*, 15(1):e0228667, 2020.

[15] S Bhatta and Y Yang. Reconstructing pm2.5 data record for the kathmandu valley using a machine learning model. *Atmosphere*, 14:1073, 2023.