Journal of Environment Sciences (2025), **Volume XI**, 28-33
https://doi.org/10.3126/jes.v11i1.80600

Web Application Prototype on Air Quality Index Prediction, Monitoring, and Information Dissemination System: Machine Learning and Python-Streamlit-based Application Tailored for Kathmandu Metropolitan City

Reyan Kumar Sapkota[1,*], Narayan Adhikari[1], Chiranjivee Subedi[1], Riya Jha[2], Manoj Subedi[1]
[1]Department of Civil Engineering, Institute of Engineering, Pulchowk Campus.
[2]Department of Electrical Engineering, Institute of Engineering, Pulchowk Campus.
Correspondence: 079bce137.reyan@pcampus.edu.np

Abstract

Air pollution has become a critical issue in Kathmandu Valley, significantly impacting public health and daily life. Rising vehicular emissions, industrial activities, and insufficient environmental regulations have resulted in dangerously high Air Quality Index (AQI) levels. Residents frequently experience poor air quality, yet they lack easy and organized access to location-wise early AQI information and alerts, making it challenging for citizens and local governments to make informed decisions about their health and activities. Kathmandu Metropolitan lacks an integrated real-time Air Quality Index tracker and forecast-based IT application system, further highlighting the urgency of developing such an application. This paper proposes a web application prototype (Wolmann, 2023) tailored for Kathmandu Metropolitan City to predict, monitor, and disseminate information about the air quality index (AQI). The application prototype integrates three regression-based machine learning models (Random Forest Regressor, CatBoost Regressor, and Gradient Boosting Regressor) to provide accurate AQI predictions. It also includes an intuitive user interface (UI based on Python-Streamlit) that allows the admin side of KMC to fetch accurate AQI predictions/visualizations, disseminate personalized recommendations/ email alerts for residents and the federal government, and allow station-wise mapping on Kathmandu Metropolitan City's map (using Python-Folium). Our ML Model achieved the R^2 values of 0.86, 0.87, and 0.93 for the Gradient Boosting Regressor, CatBoosting Regressor, and Random Forest Regressor, respectively.

Keywords: *CatBoost Regressor, Folium, Gradient Boosting Regressor, Streamlit, Random Forest Regressor.*

Introduction

Air pollution in Kathmandu Valley has reached alarming levels due to rising vehicular emissions, industrial activities, and insufficient environmental regulations. This study addresses the lack of an integrated real-time Air Quality Index (AQI) tracking and forecasting system in Kathmandu. The proposed web application leverages machine learning algorithms to predict AQI levels, provide real-time alerts, and disseminate information to residents and policymakers. The application also aligns with Kathmandu Metropolitan City's goals for environmental monitoring and public health awareness, contributing to SDG 11 (Sustainable Cities and Communities).

Kathmandu Metropolitan City (KMC) has recently begun the installation of Air Quality Monitoring Stations to extract the $PM_{2.5}$ (Particulate Matter of 2.5μg) concentration to assess the air quality of desired hotspots of Kathmandu Valley. Utilizing the raw data points collected by the station, appropriate ML and Neural Network-based models could be trained to develop an accurate forecasting system. Some well-known Machine Learning algorithms include Support Vector Machine (SVM), CatBoost, Gradient Boost, Logistic Regression, Decision Trees, K-Nearest Neighbors Algorithm, Random Forest Algorithm, etc. Suitable ML models need to be selected, depending on the nature of the dataset (i.e., Classification type vs Regression type). To predict the AQI using regression-based ML models, the data of independent features is

fitted to predict the target value. In this paper, our target value is our AQI, and the independent features are month, day, hour, relative humidity, precipitation, pressure, windspeed, temperature, and dew point. Upon literature review, we preliminarily concluded that AQI has a high correlation with the above parameters. For better data fitting for a large number of training data and more features, deep learning models could also be applied. Deep learning automatically performs feature extraction and modeling following data training, whereas machine learning requires data scientists or users to do it (Zhang, 2023).

For User Interface, any front-end frameworks (like React, Angular, or Vue) could be employed. Similarly, for backend handling, there are already plenty of options like Python's Flask, FastAPI, etc. For our prototype, we've used Python's Streamlit library to prepare the admin-based UI due to Streamlit being tailored particularly for such data science applications. For data visualization, Matplotlib has been used. Similarly, Python's Folium library has been used for mapping forecasted AQI of desired stations.

Materials and Methods

Data Collection and Feature Engineering

Globally, the US government installs air monitoring devices in cities of their respective embassies and consulates to monitor the particulate matter concentrations of those towns/zones. As a part of this program, the US Embassy has placed two Air Monitor Devices in Nepal: one inside the US Embassy (Maharajgunj) and another in Phora Durbar. These devices collect data about the particulate matter ($PM_{2.5}$) and Ozone concentrations and convert them to respective AQIs. These data are recorded hourly and exported in a CSV file. The final datasets for both stations (US Embassy and Phora Durbar) are found on airnow.gov and we've used these authentic datasets for training our machine-learning-based AQI prediction Model. The datasets from 2017 to the present (with around 8000 spreadsheet rows in each year's dataset) are made publicly available on airnow.gov (AirNow, 2024).

The dataset for training the machine learning models includes hourly AQI values based on $PM_{2.5}$ pollutants from the past seven years, sourced from the U.S. Embassy in Kathmandu (AirNow, 2024).

Upon literature review, we concluded that the value of the Air Quality Index primarily depends upon nine major factors: month, day, hour, relative humidity (Zender-Swiercz, 2024), precipitation (Wang, 2023), pressure, windspeed (Purnomo, 2024), temperature (Zender-Swiercz, 2024), and dew point. In the data science community, the factors on which the magnitude of our predicted value depends are called Features and our to-be-predicted quantity is called Target. In our project, the Features are: month, day, hour, relative humidity, precipitation, pressure, windspeed, temperature, dew point, and windspeed, and the Target is the Air Quality Index.

The corresponding hourly values of those historical weather parameters (month, day, hour, relative humidity, precipitation, pressure, windspeed, temperature, and dew point) were extracted using Python's Open-Meteo library.

The raw data with hourly $PM_{2.5}$-based AQI values (from 2017 to 2024) was downloaded from www.airnow.gov. Since the AQI value depends mainly on Temperature, Pressure, Relative Humidity, Dew Point, and Precipitation (Rahman, 2024), we extracted the values of these parameters from 2017 to 2024 (on an hourly basis) using Python's OpenMeteo library. The values were extracted as data frames and later converted to CSV Files. Then, missing rows and rows with negative (broken data) values from the initial dataset (downloaded from ainow.gov) were deleted. Then, the newly generated data (with the above parameters like Temperature, pressure, relative humidity, etc.) and the initial dataset (downloaded from airnow.gov (AirNow, 2024)) were merged. This merged final dataset was used to train the machine-learning models. Python's Pandas library was perfectly reliable in performing the above data engineering/pre-processing process.

Training Machine Learning Models

To predict the AQI values of those 2 stations, we've trained 3 machine learning algorithms. The

three algorithms are Gradient Boosting Regressor, CatBoost Regressor, and Random Forest Regressor. These three algorithms have been imported from Python's Scikit Learn (also called sklearn library)

First, the outliers were removed from the final merged dataset to ensure better fitting in the ML model. Then, that dataset was divided into two sections: X and Y. X is the data frame with the dependent features as major headings. The headings are Month, Day, Hour, Temperature, Relative humidity, precipitation, wind speed, Dew Point, and Pressure. Y is the data frame with the corresponding AQI value. In Y, "AQI_value" is the only heading. X and Y were split into 80% and 20%. 80% of X was chosen as X_train, and 20% was X_test. Similarly, 80% of Y was chosen as Y_train, and 20% was Y_test. The machine learning algorithm will now train (or fit the dataset on its regression lines/ planes in a process called Data fitting) on X_train and Y_train. X_test and Y_test will be utilized for checking the model's learning and prediction performance.

After splitting, the data was scaled using the StandardScalar method of Scikit Learn:

The data was fitted respectively for all three algorithms with their respective unique hyperparameters. The data was fitted with the following hyperparameters. Hyperparamters used are described in the annex.

In addition to the Standard Scaler, polynomial feature enhancement was applied to the Random Forest Regressor to increase the number of dependent features (X) and improve data fitting and prediction accuracy.

After data fitting with suitable hyperparameters, the model was used for prediction.

As per Python's OOP fundamentals, the fitted ML model is a Class. This class has a method called "predict". Using this method, we can pass our Dependent features as arguments to the model. Based on these arguments, the model class' **predict()** method returns our AQI value. After training the ML model, we converted it to a pickle file using Python's Pickle library. This readily available pickle file would later be used to create predictions.

Web Application

The web application was developed using Python's Streamlit library, providing an intuitive user

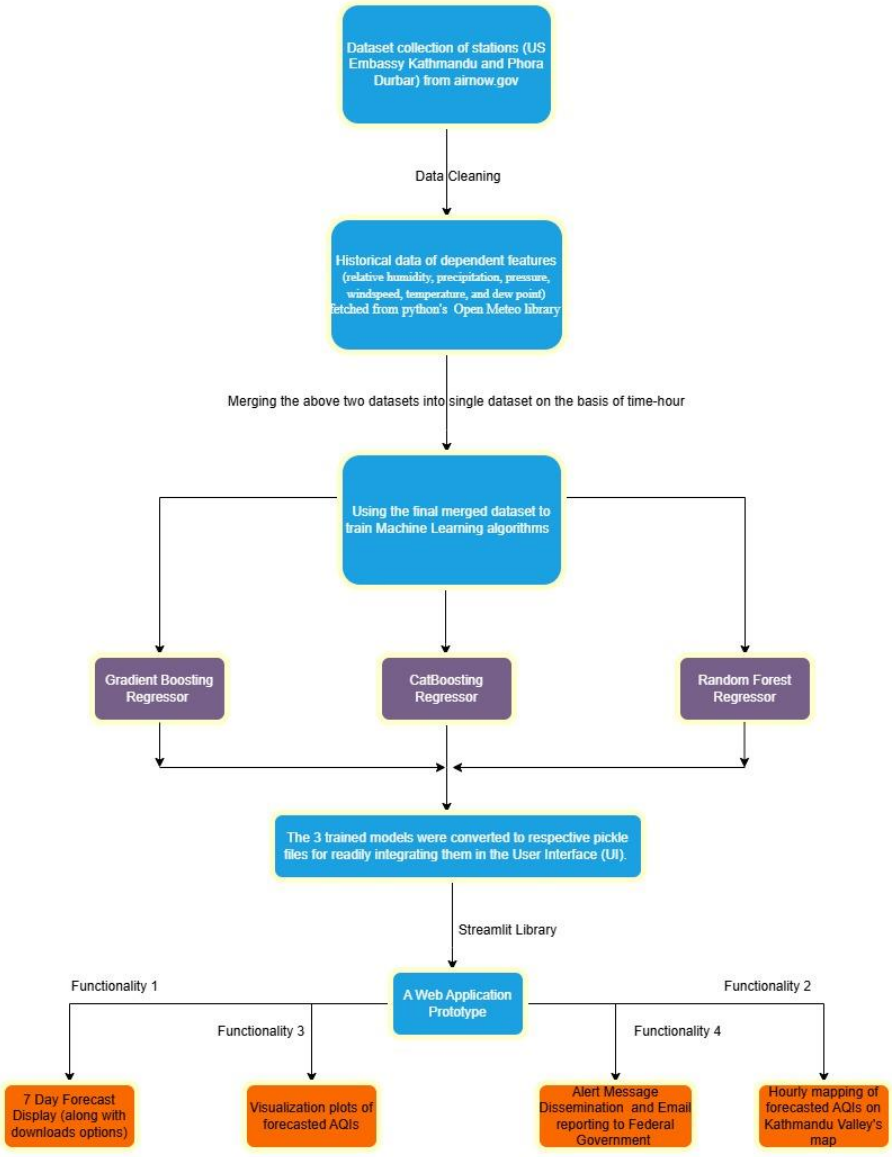

Figure 1: Flowchart depicting the entire workflow of the application.

interface for AQI predictions, visualizations, and alerts. The application also includes features such as a citizen alert system, a government reporting system, AQI trends visualization using charts, and mapping on Kathmandu Valley's map using Python's Folium library.

Results and Discussion

Model Performance

The Random Forest Regressor outperformed the other models, likely due to its ability to handle complex interactions between features and its robustness to overfitting. Polynomial feature enhancement further improved the model's accuracy by capturing non-linear relationships in the data.

The following performance metrics were tested

a. Mean Square Error, $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

Where,

$y_i = real\ i^{th}\ value$

$\hat{y}_i = predicted\ i^{th}\ value$

$n = number\ of\ data$

MAE represents the average absolute error in AQI points and is dimensionless.

b. $Coefficient\ of\ Determination,\ R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y}_i)^2}$

where,

$y_i = real\ i^{th}\ value$

$\hat{y}_i = predicted\ i^{th}\ value$

$\bar{y}_i = mean\ of\ all\ values$

c. Root Mean Square Error, $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

Where,

$y_i = real\ i^{th}\ value$

$\hat{y}_i = predicted\ i^{th}\ value$

$n = number\ of\ data$

RMSE represents the square root of the average squared error in AQI points and is also dimensionless.

Application Functionalities

We have added the following functionality to our web application. We've meticulously tried to tailor the application prototype of a web app exclusively for Kathmandu Metropolitan City's administrative purposes. Due to the availability of large volume datasets of Maharajgunj and Phora Durbar only, we have those two stations incorporated in our prototype. To achieve this goal, we've added the following functionalities/services to our prototype:

7-Day Hourly Forecast

In the UI shown below (Figure 2), each button returns the desired day's AQI prediction (done by all three algorithms) (Figure 2). Furthermore, a feature for downloading the forecast CSV has also been added (Annex).

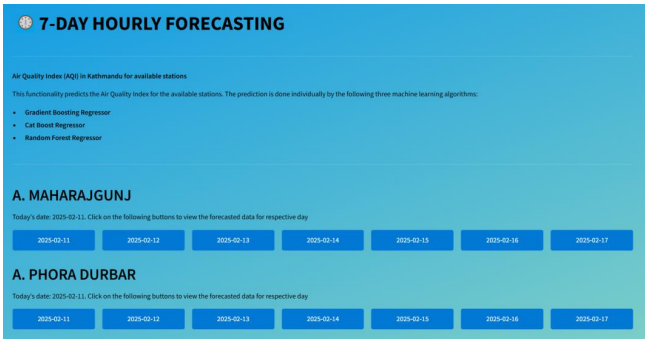

Figure 2: 7-Day Hourly Forecasting for Maharajgunj and Phora Durbar

Table 1: Model Performance Metrics

Performance Metric	Gradient Boosting Regressor	CatBoost Regressor	Random Forest Regressor
Coefficient of Determination	0.88	0.87	**0.94**
Mean Absolute Error	11	13	**9**
Root Mean Square Error	16	17	**13**

Citizen Alert System

The AQI values predicted from respective ML models were used to generate alert messages. The alerts were divided into four titles: Highest AQI and its time of occurrence (as predicted by each ML model), Health Alert, Hazard Description, and Caution (Annex). The automated message flex was designed using HTML and CSS.

Govt. Reporting System

Using a similar concept as above, a Federal Government Reporting System is also added. The required day's recommendation can be generated/ automated and emailed to the central government with the click of a button (Annex). Along with the HTML and CSS designed message, the CSV files of each ML model's 7-day (hourly) forecasted dataset are also emailed. Python's powerful email.mime library has been utilized.

Mapping of AQI of 2 stations in Kathmandu

Python's Folium library allows us to import the maps of the desired coordinate range. On this map, we've mapped the AQI values of the 2 stations. Index color has been assigned. This color (or legend) depends on the severity of the Air Quality Index value. Mapping for the respective date (Figure 3) would allow the Kathmandu Metropolitan City to have live data updates of the air quality of the given stations (Figure 4). This could help the metropolitan disaster control unit to launch mitigating actions accordingly.

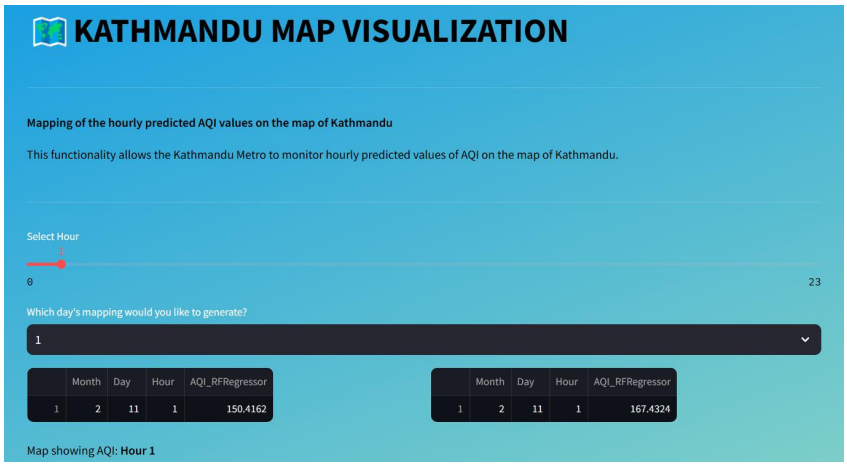

Figure 3: Mapping for the desired day and hour.

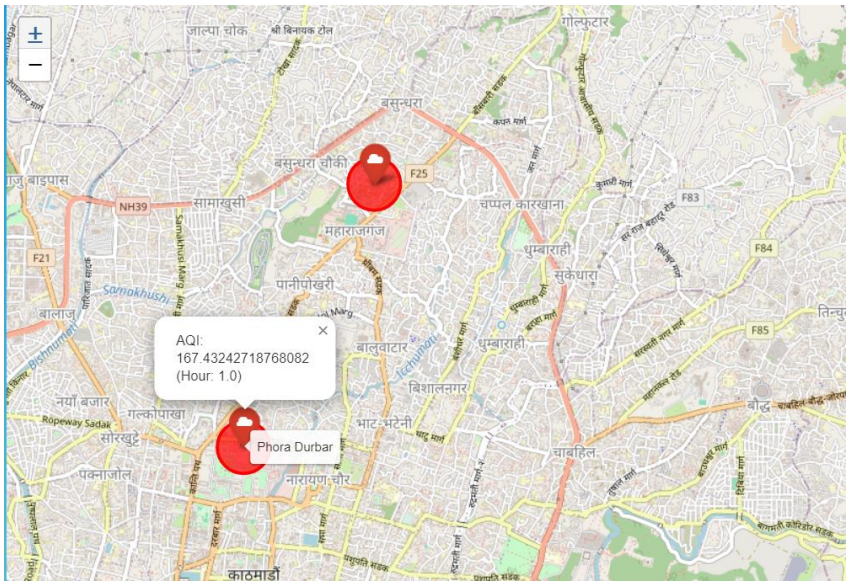

Figure 4: Mapping of AQI for 2 stations (Maharajgunj and Phora Durbar) using Python's Folium library.

Data Visualization

The hourly forecasted value of the AQI has been used to create visualization plots and charts (Figure 5).

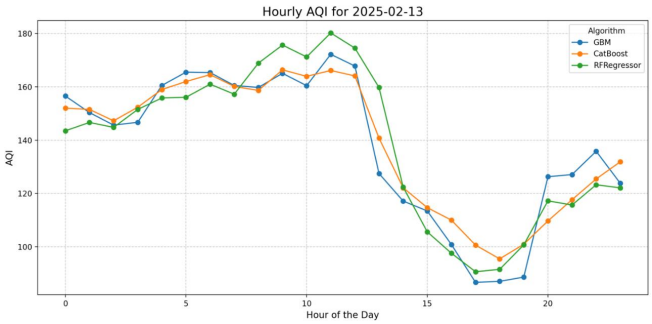

Figure 5: Generated Line Chart (by matplotlib) for a particular day

Conclusion

The proposed web application effectively addresses Kathmandu's air pollution problem by providing real-time AQI monitoring, predictions, and alerts. The system supports public health awareness and environmental monitoring, aligning with Kathmandu Metropolitan City's smart city initiatives. The Random Forest Regressor performed better over the four ML regression-based algorithms with an R^2 of 0.9385, Mean Absolute Error (MAE) of 8.88, and Root Mean Square Error (RMSE) of 13.362. The integration of these predicted values with the intuitive user interface (with the aforementioned functionalities) would serve as a valuable asset for the admin side of the Kathmandu Metropolitan City in terms of effective Air Quality Monitoring and Information dissemination. Future work might include expanding the system to cover more monitoring stations and integrating additional weather data. For larger datasets with a large number of features, Deep Learning and Neural Network alternatives could be considered.

Acknowledgements

The authors would like to acknowledge the U.S. Embassy in Kathmandu for providing the hourly dataset of AQI values (from 2017 to 2024) for Maharajgunj and Phora Durbar.

References

AirNow. (2024). *US embassies and consulates*. AirNow.gov. https://www.airnow.gov/international/us-embassies-and-consulates/#Nepal

Rahman, M. M., Wolmann, L., Zhang, Z., & Zhang, S. (2024). AirNet: Predictive machine learning model for air quality forecasting using the web interface. *Environmental Systems Research, 13*(1). https://doi.org/10.1186/s40068-024-00378-z

Purnomo, A., Andang, A., Badriah, S., Paryono, E., Sambas, A., & Umar, R. (2024). Influence of wind speed and direction on the performance of Low-Cost particulate matter sensors. *Environment and Ecology Research*, *12*(4), 446–455. https://doi.org/10.13189/eer.2024.120409

Thapa, I., & Devkota, B. (2024). Applying machine learning algorithms to estimate $PM_{2.5}$ using satellite data and meteorological data. *Journal of Engineering and Sciences, 3*(1), 74–80. https://doi.org/10.3126/jes2.v3i1.66239

Wang, R., Cui, K., Sheu, H., Wang, L., & Liu, X. (2023). Effects of precipitation on the air quality index, $PM_{2.5}$ levels and on the dry deposition of PCDD/FS in the ambient air. *Aerosol and Air Quality Research*, *23*(4), 220417. https://doi.org/10.4209/aaqr.220417

Wolmann, L., Rahman, M. M., Zhang, Z., & Zhang, S. (2023). evalPM: A framework for evaluating machine learning models for particulate matter prediction. *Environmental Monitoring and Assessment, 195*(12). https://doi.org/10.1007/s10661-023-11996-y

Zender-Świercz, E., Galiszewska, B., Telejko, M., & Starzomska, M. (2024). The effect of temperature and humidity of air on the concentration of particulate matter - $PM_{2.5}$ and PM_{10}. *Atmospheric Research*, 107733. https://doi.org/10.1016/j.atmosres.2024.107733

Zhang, H., Liu, Y., Zhang, C., & Li, N. (2025). Machine Learning Methods for Weather Forecasting: A survey. *Atmosphere*, *16*(1), 82. https://doi.org/10.3390/atmos16010082

Zhang, Z., & Zhang, S. (2023). Modeling air quality $PM_{2.5}$ forecasting using deep sparse attention-based transformer networks. *International Journal of Environmental Science and Technology*. https://doi.org/10.1007/s13762-023-04900-1