Prediction of $PM_{2.5}$ level at an AQMD station : using nearby station's data

Nitish Venkatesh Septankulam Ramakrishnan, Meghana Vasanth Shettigar, Jooseok Lee

University of Colorado Boulder

Problem Space

Predicting the level of particulate matter ($PM_{2.5}$) is becoming increasingly crucial in the field of public health. For example, the United States Environmental Protection Agency (US EPA) designates particulate matter as an important source of air pollution. Accordingly, many air pollution observatory stations are set up to monitor the level of $PM_{2.5}$. However, not every station is able to measure the level of $PM_{2.5}$. For instance, only 13 of 30 stations in South Coast Air Quality Management District (AQMD) can monitor the level of $PM_{2.5}$. In this project, we aim to develop a machine learning model that can predict the level of $PM_{2.5}$ of stations that have no $PM_{2.5}$ level monitoring capability. To achieve that, we build a machine learning model that predicts the $PM_{2.5}$ level of a target station using $PM_{2.5}$ level data of nearby stations. We also utilize other commonly monitored air pollution data, such as level of NO_2 and CO, and basic meteorological data, such as wind direction and speed, to increase the performance of the model. The development direction is based on the assumption that the air quality of one station has a spatiotemporal correlation to the air quality of nearby stations.

Data

We use data from January 17, 2017 to January 16, 2022. Each row represents an hourly measurement of various air pollutants and the total number of rows is 33,520. To avoid data leakage in the test dataset, we split data into train and test dataset according to the time sequence. The train dataset is from January 17, 2017 to January 10, 2021 and the test dataset is from January 17, 2021 to January 16, 2022. We select Mira Loma station as our target station and five other stations (i.e. Upland, LakeElsinore, Temecula, Banning, and Central San Bernardino stations) as nearby stations. The selection criteria is the number of missing values in PM2.5 and proximity).

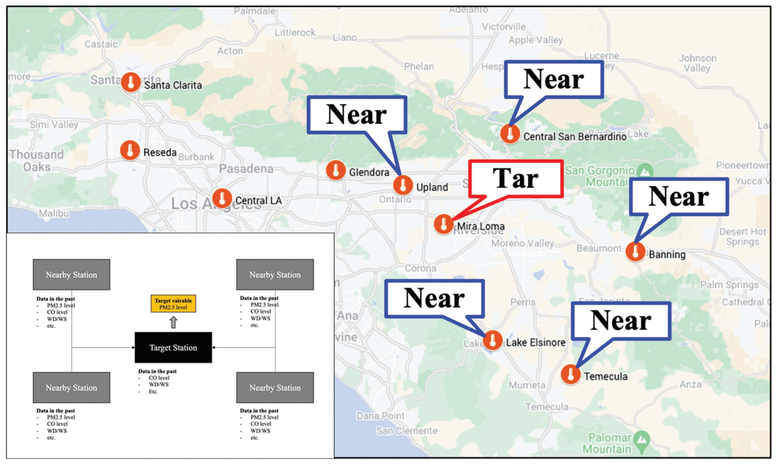

Figure 1: Illustration of Stations and Approach

Approach

We handle missing values with an interpolation technique supported by pandas library. It estimates missing values using adjacent data. In this project, we use spline interpolation to fill missing values. We also clip PM2.5 data to handle outliers. We aim to build machine learning models that predict the PM2.5 level after six steps (i.e. after six hours) using the previous seven days' data (i.e. 168 hours of past data). We developed three machine learning models (i.e. CNN, LSTM, and XGBoost) that can predict the level of $PM_{2.5}$ of stations that have no $PM_{2.5}$ level monitoring capability.

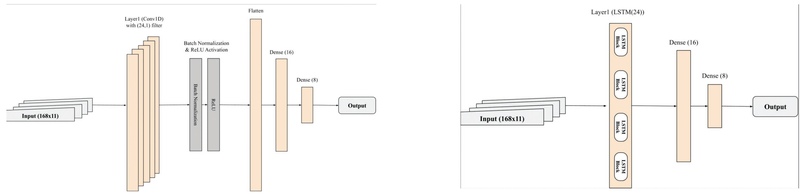

Figure 2 & 3: CNN and LSTM Architecture

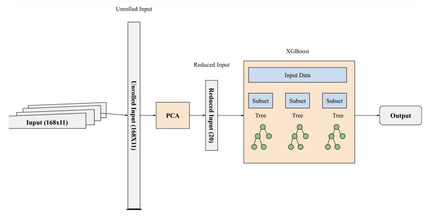

Figure 4: XGBoost Architecture

Evaluation Metrics

We use three kinds of evaluation metrics, namely mean absolute error (MAE), mean squared error (MSE), and total accuracy index (P) to compare the performance of the models. The formal definition of each metric is given below.

$$MAE = \frac{1}{n}\sum^{n}\left|y_i - \hat{y}_i\right|$$

$$MSE = \frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2$$

$$p = 1 - \frac{\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|}{\sum_{i=1}^{n} y_i}$$

No	Main model	Base model
1	CNN model *without* target station's $PM_{2.5}$ data	CNN model *with* target station's $PM_{2.5}$ data
2	LSTM model *without* target station's $PM_{2.5}$ data	LSTM model *with* target station's $PM_{2.5}$ data
3	XGBoost *without* target station's $PM_{2.5}$ data	XGBoost *with* target station's $PM_{2.5}$ data

Table 1: Models developed and corresponding base models

Results

Overall, the performance of base models, which use the target station's $PM_{2.5}$ data, show better performance in terms of three metrics, namely MSE, MAE, and Total Accuracy Index. However, the performance of main models, which do not use the target station's $PM_{2.5}$ data, show comparable performance indicating the possibility of the approach we proposed in this project, which is using nearby stations' data to predict the target station's $PM_{2.5}$ level. One limitation is that the overall performance of models is pretty low, which will be discussed in the discussion section below. Among three algorithms we used, CNN with 1D convolutional layer showed the best performance. We conducted additional error analysis to gain more insight into the problem. The error goes up as the real $PM_{2.5}$ level goes up. It seems that it is mainly due to the lack of data in higher levels of $PM_{2.5}$. As a result, the model we developed failed to accurately predict high levels of $PM_{2.5}$.

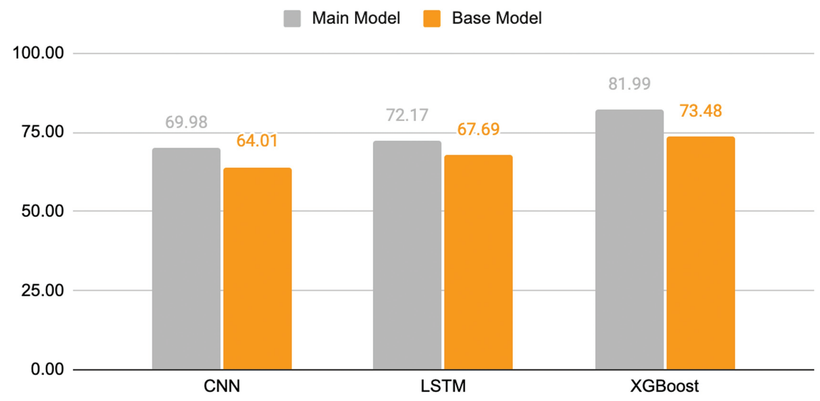

Table 2: Comparison between models (MSE)

	MSE		MAE		Total Accuracy Index	
	Main Model	Base Model	Main Model	Base Model	Main Model	Base Model
CNN	69.98	64.01	5.57	5.38	0.7085	0.7284
LSTM	72.17	67.69	5.50	5.42	0.7111	0.7187
XGBoost	81.99	73.48	6.07	5.86	0.6847	0.6320

Table 3: Performance comparison between models

The results reveal that machine learning models that do not use the target station's $PM_{2.5}$ data show comparable performance to the models that use the target station's $PM_{2.5}$ data. It showed the possibility of utilizing our approach to estimate the $PM_{2.5}$ level of stations that do not have the capability to measure it. However, there are some limitations that should be overcome. The overall performance of models we developed showed pretty low performance and R^2 of the best model we developed was 0.4552, which is quite low. To make the proposed approach practical, it is necessary to increase the R^2 of prediction models. According to the error analysis, the prediction models seem to show lower performance on high levels of $PM_{2.5}$ indicating a need to build models specialized in handling higher levels of $PM_{2.5}$.