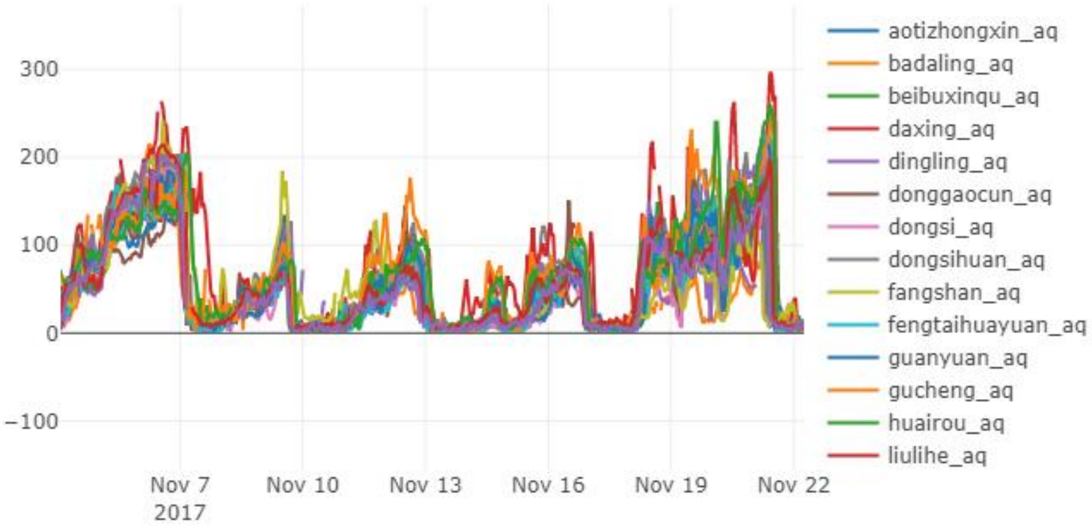
MSC-BDT5002 Project Report

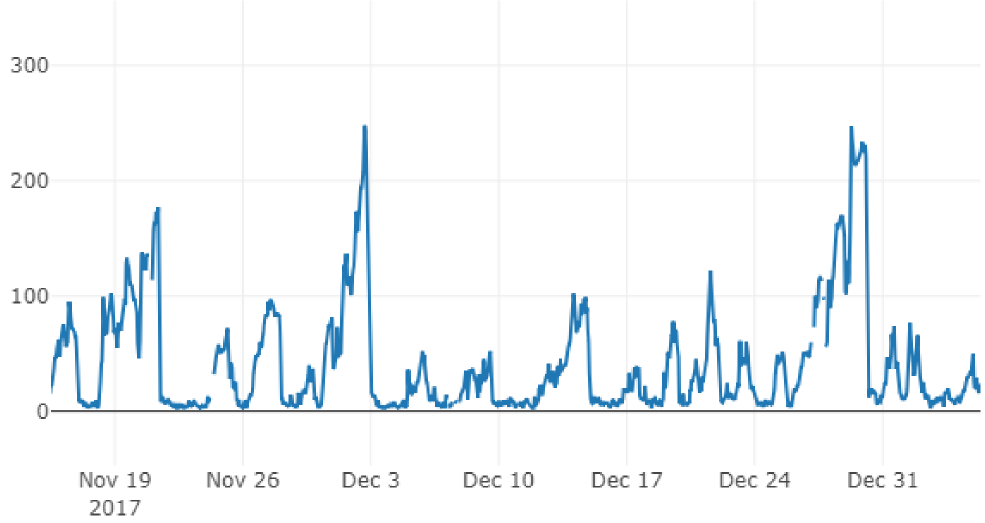
Team member:
Zheng Xiaowen: xzhengao 20548840
Wang zhicong: zwangee 20573003

I.Data Distribution

1.PM2.5

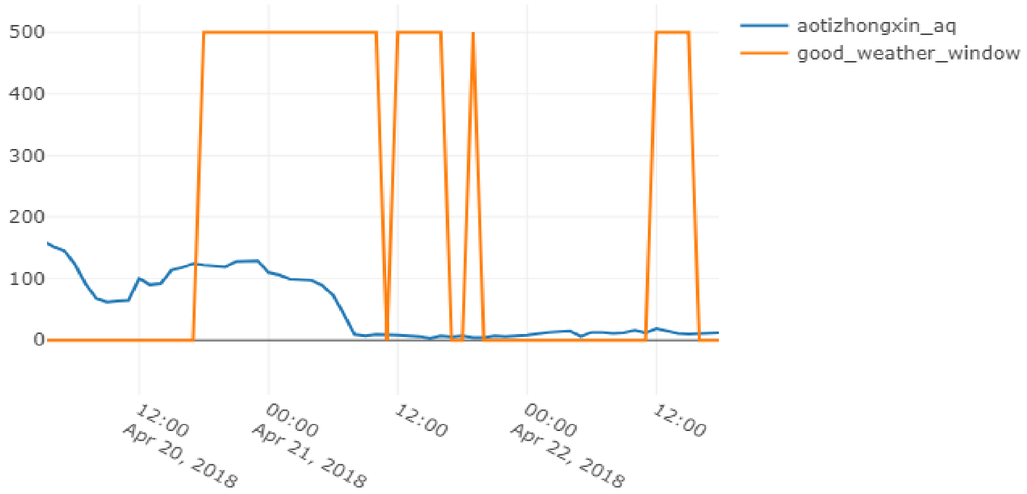


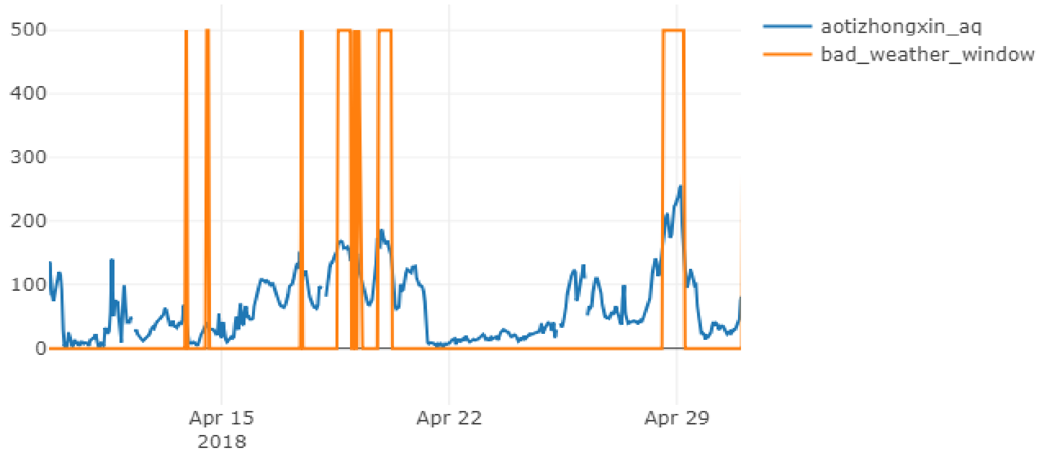
This is an overview on all of the stations of one period. We find that all the stations change with the similar trend, which will give us the inspiration that we can train all the stations together in one model.



One unique phenomena of PM25 may be the variation trency. The image shows the value of one station over the time. We may find that the pm2.5 follows such hypothesis: it will maintain the low value except something occurs. After reaching the peak or the event finishes, it will quickly go back to the normal low value. So, one important objective for predicting pm2.5 may be how to capture such sharp drop and the increase facts accurately.

weather affect on PM25

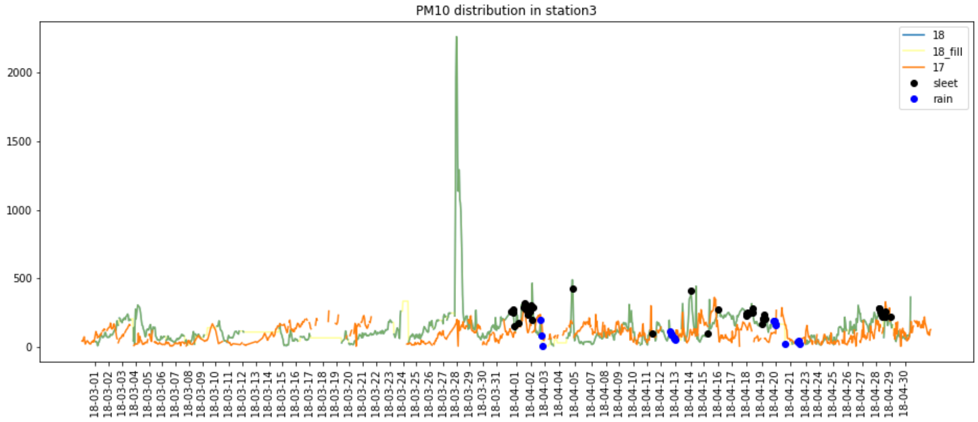


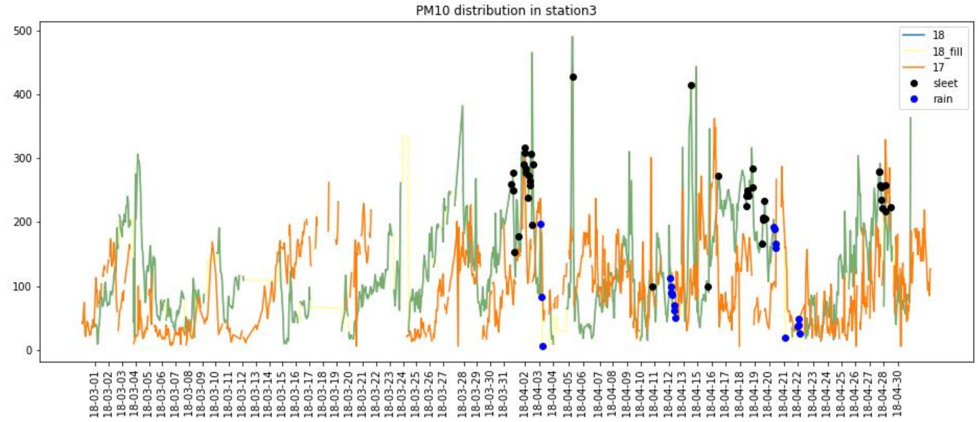


According to the above two images, PM25 is significantly affected by two kinds of weather. One is defined as good weather, like wind and rain. The other is defined as bad weather ,like haze. We find that:

- Under the good weather window, PM25 will keep low value or experience a significant drop.
- When meeting bad weather, the value of PM2.5 will have higher probability to have high value during the that period.

2.PM10

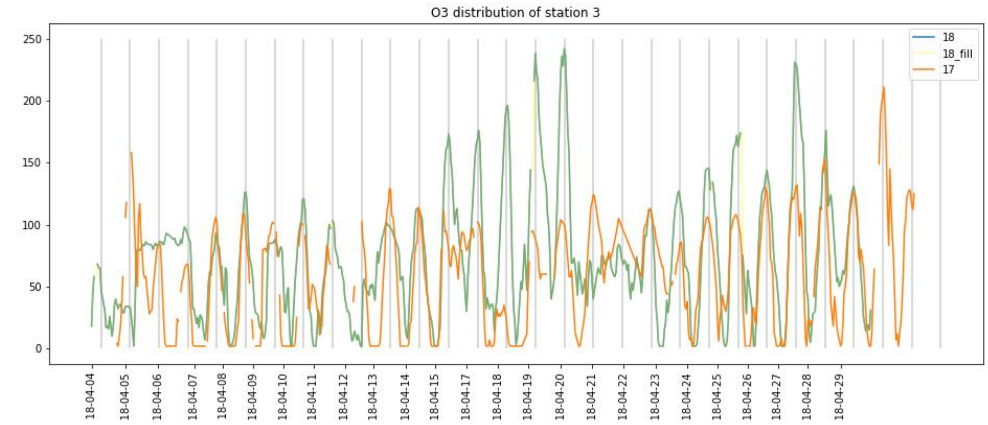




The above figures show the distribution of O3 from March to April in 2018 and 2017. The figure on the top is the data before removing points on 2018-03-28 and the figure on the bottom is the figure after removing points on 2018-03-28. The black point highlight the time when the weather is either ‘haze’,’hail’ or ‘sleet’. We group these 3 type of weather as bad weather condition. The blue point hightlight the time when it’s rainy day. We used ffill method to fill the PM10 gap in 2018. From the above distribution, we found that:

- PM10 performs abnormally on 2018-03-28. It’s better to remove the outliers.
- PM10 was high in Qingming Festival and other holidays.
- PM10 distribution is seasonal and has similar trend in 2017 and 2018.
- Some gaps of the missing value is too large to use ffill method to fill.
- PM10 is highly correlated with bad weather conditions(sleet, hail and haze) as well as rain. When it’s rainy, it’s highly possible that PM10 will start to drop and when it’s in bad weather condition, it’s highly possible that PM10 will reach its peak.

3.O3



The above figure shows the distribution of O3 from March to April in 2018 and 2017. The gray vertical line is shown every 24 hours. We ffill the missing O3 value in 2018. And the filled value is shown in yellow line.

From the above figure, we can find that

- O3 will have a peak every 24 hour. Its distribution is seasonal.
- In 2018, after ffill the misiing O3 values, the trend is still preserved.
- Distribution of O3 in 2017 and 2018 is quite similar, we can use 2017's data to train our model.

II.Data Preprocessing

1.Weather data

Weather data only misses in times,with missing rate 1.11% . we try to fill them by surrounding station because the mse of nearest stations are lower than 5% of its origin value. Unfortunately, the filling rate for this method is lower than 5%, which means if one station has no value for one time, the surrounding station have 95% probability to have no data for the same time. So, it may be meaningless to fill the data by the surrounding station due to such low filling rate. Later we will use other methods to overcome this problem, like use statistic data instead of directly use it.

2.Missing air quality values

Overview:

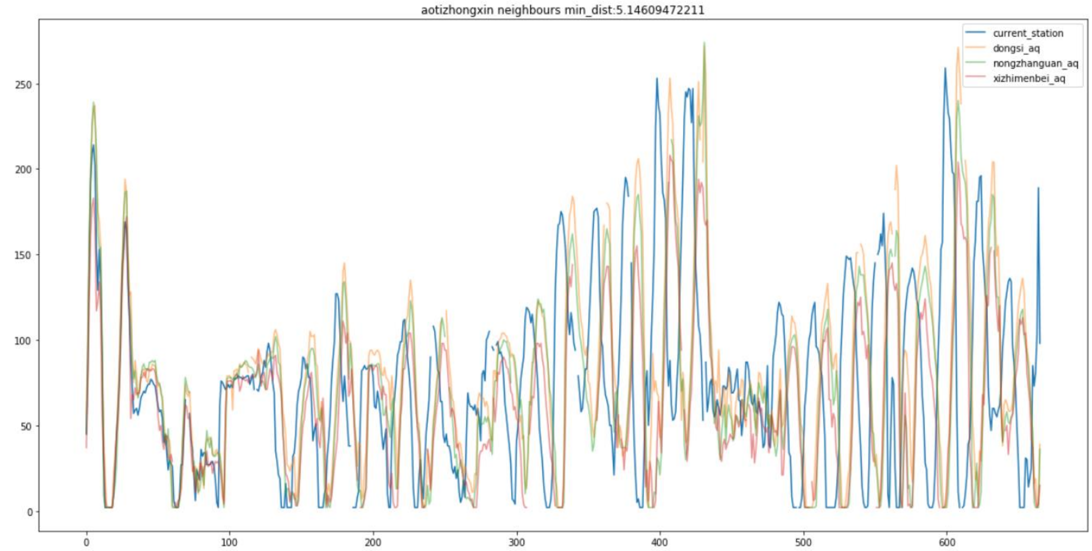
	PM25	PM10	O3
missing rate	5%	20%	5%

Possible methods:

We try several methods to fill the missing pollution index and belows are some details for each method we have tried.

2.1.Nearest aq station to fill:

After looking at the distribution of the pollution index data. We found that using air quality data from the nearby station directly could be problematic. The figure below shows the O3 distribution of aotizhongxin station and its top 3 nearest neighbors. We can see that although the trend is similar is these stations, it seems that there is a shift in between. Thus, use the pollution index values from the nearest air quality stations may suffer from lagging effect. This is not a good way to fill the missing value.



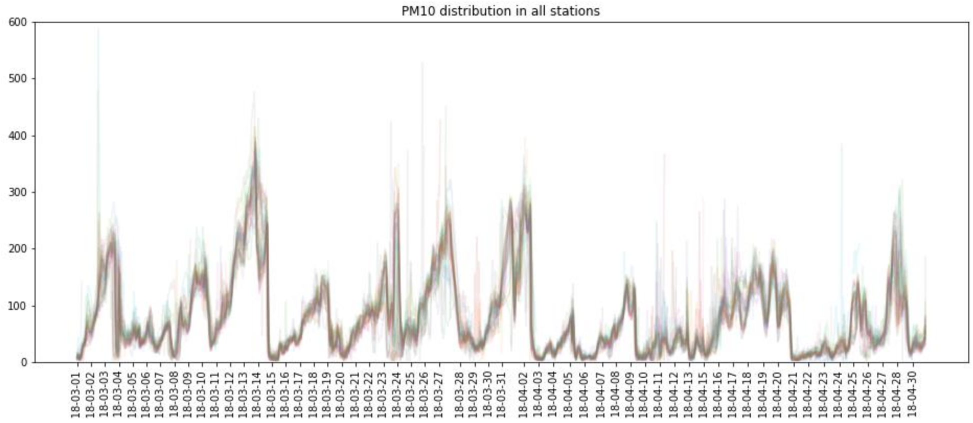
2.2.Use models to predict the gap:

The total missing rate is quite small for PM2.5 and O3, so we may use model to predict the missing value, which may have better performance than using surrounding values. Here we use xgboost as the model to predict the missing values and the validation result seems good.

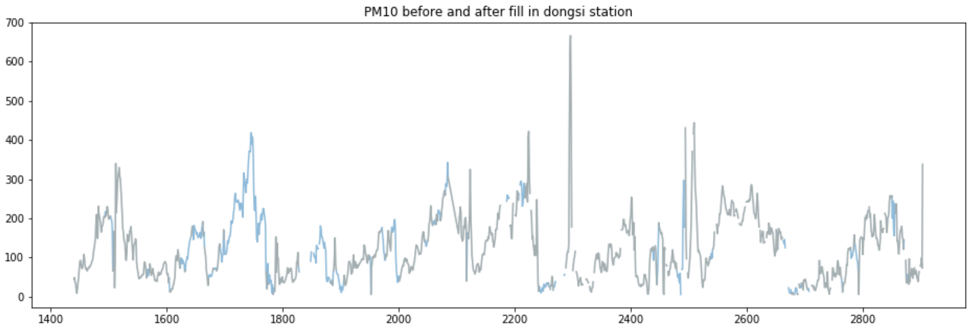
xgboost predicton	PM25	PM10	O3
self test(smape)	0.04572231877945248	0.05356661981167564	0.06384560158364744

After testing on the same setting, this method performs worse than directly fill by surrounding values. In our opinion, this may introduce addition noise to the dataset.

2.3.statistics of all the stations at the given time(final choice)



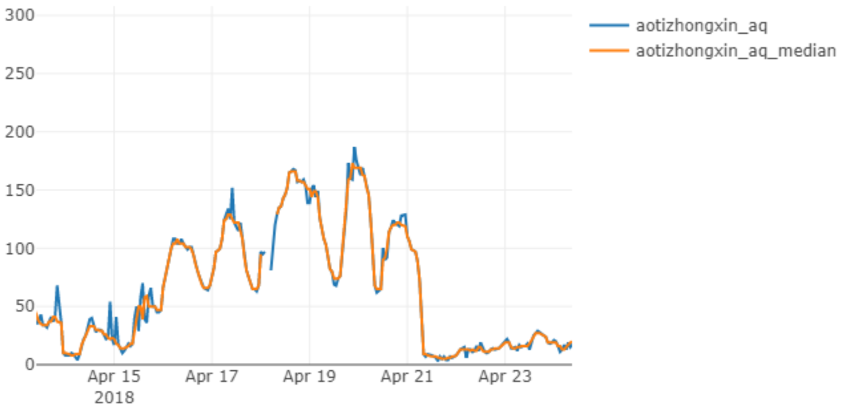
The above figure shows the distribution of PM2.5 in all stations. We found that the majority stations have similar PM2.5 and we used mean value of all the stations to fill the missing values in some stations. The image below shows the result after filling the missing values in dongsi station. There are still some values missing because the original given time index is incontinues, the mean values of all the stations are null. We generate a new time index and use ffill method to fill the gap.



3. The noise in data

The data is the real time data collected by human, which may contain noise. This noise may be mainly caused by the human and also by the environment condition when recording. Here we could not obtain very accurate information from the dataset and this may be too difficult for us to take anything into consideration. For this task, in our opinion, one basic objective is to predict the main trend for each pollution index. Too much noise may mislead the model which will give predictions far from the true values.

To solve this kind of problem, we use median filter to eliminate the extreme values in the filter window on the pollution index, which is also the label for training. This will result in a smoother curve than the origin one, which will also make our model focus on the trend of our target. The following image shows the effect of the median filter. Also, after the experiment, the result with filter is higher than the origin one.



III.Feature Enginnering

1.PM2.5

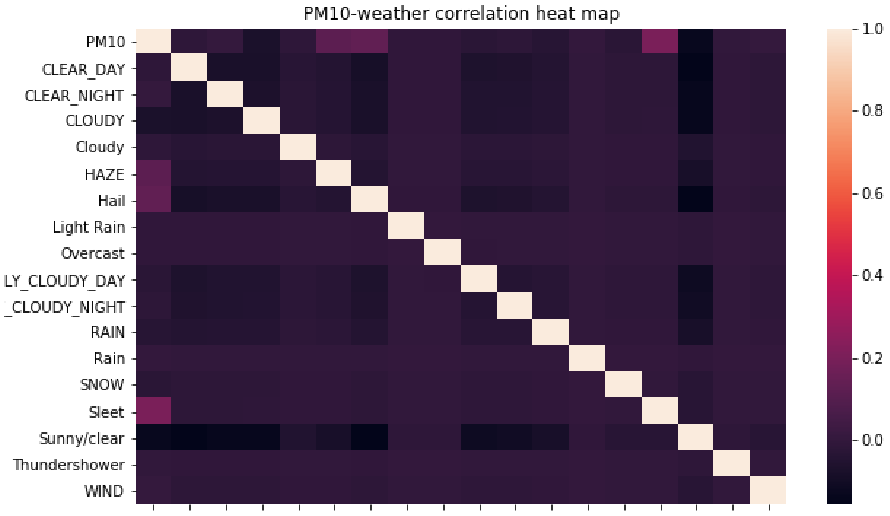
Weather: Based on previous observation on pm2.5, we classified the weather into three different classes, positive, negative and neutral weather ,instead of directly using it. We also use accumulated weather value based on this three classes, which will indicate the weather effectiveness along the duration. Unfortunately, according to the feature importance in Lightgbm, this will not have great impact on the model. One possible reason for this is that other weather features themselves have done this job.

weather feature selection: we have several basic weather features for PM2.5, but how they affect the pm2.5 need to be analyzed. We use *f_regression* in *Sklearn* to find the most correlative features for PM2.5, which are humidity and wind speed, which also have certain relation with good weather we defined before. The performance after feature selection will also improve a little.

historical values: we used statistical values of PM2.5 as features. For statistical attributes, we mainly look at var and median in a given rolling time window. We apply this to both weather attributes(wind speed & humidity) and the PM2.5 value with 6 hour as the rolling window to get the best performance.

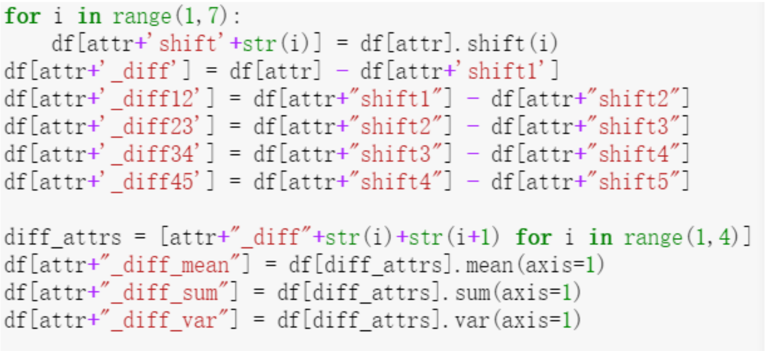
2.PM10

weather: The image below is the correlation heat map of PM10. We can find that PM10 highly relates to haze, hail and sleet, which match the distribution of PM10. We add an additional column to show whether the weather is "bad" now. If the weather is in bad condition, the value will be 1, else will be 0. We also added a column "bad_weather_6h_sum" to sum up the "bad_weather" column within 6h for a given air quality station. Apart from the weather condition, we also use weather attributes like 'pressure', 'temperature' ,'humidity' and 'wind_speed'. The statistical values of the weather attributes and current values of the weather attributes are used to build the features.



date/holiday: The distribution shows that PM10 is related to holidays. Thus we added one additional column "holiday" to show whether a given day is a holiday. We also add a column "holiday_sum" to sum up the "holiday" in rolling window of size 24. For example, if it's "2018-04-05 02:00:00" now, the holiday sum will be 2 as this is the 2nd hour of the Qingming Festival.

historical values: we used statistical values of PM10 as features. For statistical attributes, we mainly look at sum, var, mean, median in a given rolling time window. We also use PM10 at the some time in the previous days. To learn the trend, we also use the mean, sum and variance of the difference. The image belows is the screenshot of our codes that shows how we extract the statistical features of the change. More detail can be found in the **add_rolling** function in utils.py



```
for i in range(1,7):
    df[attr+’shift’+str(i)] = df[attr].shift(i)
df[attr+’_diff’] = df[attr] - df[attr+’shift1’]
df[attr+’_diff12’] = df[attr+”shift1”] - df[attr+”shift2”]
df[attr+’_diff23’] = df[attr+”shift2”] - df[attr+”shift3”]
df[attr+’_diff34’] = df[attr+”shift3”] - df[attr+”shift4”]
df[attr+’_diff45’] = df[attr+”shift4”] - df[attr+”shift5”]

diff_attrs = [attr+”_diff”+str(i)+str(i+1) for i in range(1,4)]
df[attr+”_diff_mean”] = df[diff_attrs].mean(axis=1)
df[attr+”_diff_sum”] = df[diff_attrs].sum(axis=1)
df[attr+”_diff_var”] = df[diff_attrs].var(axis=1)
```

3.O3

The holiday features and weather features used for O3 prediction is quite similar with that in PM10 and PM2.5. But due to the periodic characteristic of O3, we includes more historical values for the models to learn the trend. Note that we find O3 will have a peak every 24 hours, just like the sine and cosine function. How the O3 shift compre with the same time at the previous days can be important. Thus we first record O3 value at a given time, let’s say, 02:00:00 in 1,2,3,4,5 days before. Here for simplicity, we mark these values as v1,v2,v3,v4,v5 respectively. We used the statistical features like sum, var and mean of these values. We also included the shift, that is, v4 - v5, v3 - v4, v2 - v3, v1 - v2. We mark the, as diff4,diff3,diff2,diff1 correspondingly. We used these difference as features and also the statistical values of the difference like sum, mean and variance. To sum up, the final historical features we used are:

[rolling stat] +[shift] + [stat of shift] + [diff] + [stat of diff]

Note:

[rolling stat] is the statistical features within the rolling window;

[shift] is O3 at the same time in the previous days(v1,v2,...);

[stat of shift] is the sum,variance,mean of [shift];

[diff] is the difference between shifted values (diff1,diff2,...);

[stat of diff] is the sum, variance, mean of the [diff]. More detailed of the implementation can be found in our codes.

IV.Data used for training and validation

Since the prediction of different pollution indexes uses different features and some features are severely missing in the historical data. We trained 3 separate models to predict PM2.5, PM10, O3 independently. And the data used to train and evaluate this models are quite different.

1.PM2.5

mean & std	April	Jan to April	weekday	weekend
2017	57,48	79,88	81,90	75,83

2018	62,60	62,63	60,65	66,59

According to the statistic analysis on different time group, we find that the data in 2017 will be much different from 2018 for the corresponding time period, so we may not consider to use data long time ago to train our model.

Also as an holiday for 1/5/2018, we should also consider the difference between work days and non-work days. It seems the same for the whole year but if we look deeper into this data, we find that the data for last two weekend is much lower than before(mean 83 vs 55). We assume that 1/5/2018 will have higher probability to be similar with the recent non-work days, so we just use two weeks' data to train our model in order to conquer the misleading data from history.

2.O3

The prediction of O3 relies more on its periodic characteristic and its trend. And in the distribution of O3, we found that for March and April, the distribution is quite similar in 2017 and 2018. Thus we used data in these four months for training and the last two days of April(2018-04-29 & 2018-04-30) for validation.

3.PM10

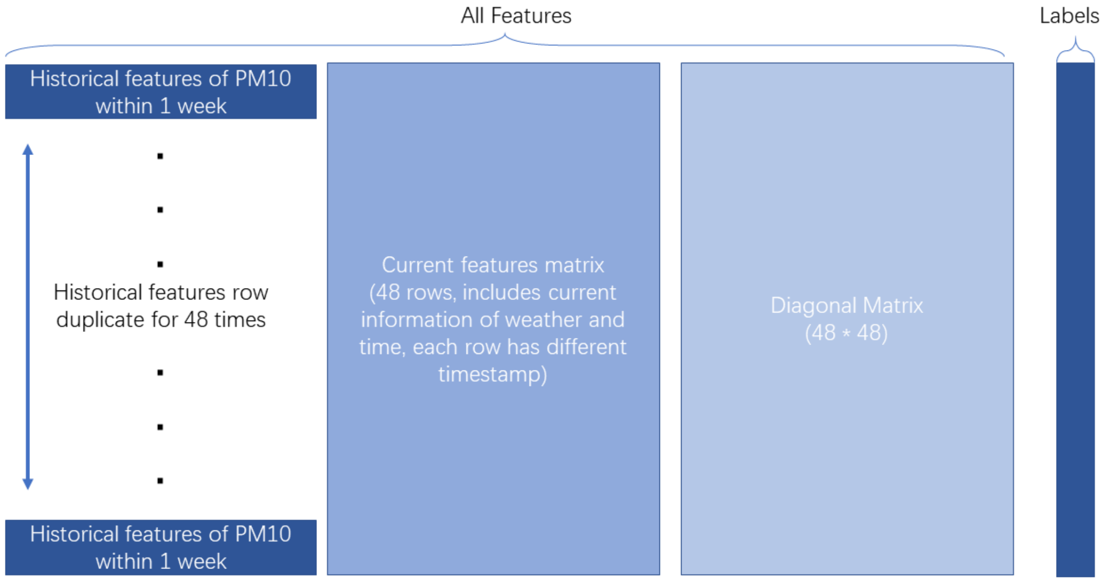
The bad weather condition features(sleet,haze,hail) turn out to be an important features for predicting PM10. However, weather columns are greatly loss in 2017 and March 2018. The lost rate of PM10 during different periods can be found in the following table. We use data in April 2018 because the weather info is quite complete and seldom loss. We use the last two days to evaluate our models and all the other days for training the models.

	2017 March & April	2018 March	2018 April
PM10 (% of null)	14.9%	28.9%	20.5%
weather (% of null)	63.1%	68.6%	1.5%

V. Our attempts for model building

1. One-time prediction method

Considering that using the prediction values as features could introduce noise and the error might keep accumulating if the previous prediction is wrong, we attempted to predict the pollution index for the coming 48 hours at once. The models we used in the one-time prediction method is just the basic machine learning models like lightgbm and gradient boosting regressor in sklearn. The way we build the training and testing set to predict the next 48 hours at once is shown in the following figures.


All Features
Labels
Historical features of PM10 within 1 week
Historical features row duplicate for 48 times
Historical features of PM10 within 1 week
Current features matrix (48 rows, includes current information of weather and time, each row has different timestamp)
Diagonal Matrix (48 * 48)

We first compressed the historical features into one row for each station. Then we generated a 48 x 48 diagonal matrix to tell the models which hour we are going to predict(the 1st, 2nd, 3rd...). The compressed historical data will be duplicated for 48 times as we will have future 48 hours to predict. That is for the consecutive 48 labels, their historical features will be exactly the same. We then concat all these features horizontally and feed them to the models.

It turns out that this kind of predicting method can not well handle the sudden drop or sudden increase. The 48 predicted labels tend to be very similar, that is, the models can not learn from the diagonal matrix. Problems could be that identical historical features can be misleading. Thus, we abandoned this methods finally.

2.Row by Row Prediction Method(Final Choice)

2.1 LSTM

The attributes for lstm is different from what we have talked about earlier in this report, so here we will briefly describe what we have done with LSTM, from data preprocess to the final result.

1.data preprocess

Time: One important restriction for RNNs are the time consistency of data. For this dataset, the time is not continuous. So we should first complete the missing time. This will also increase the total number of missing value, which may cause great problems.

Weather info: In order to use the weather information, we use the geographic information to select the weather station. For the weather station, according to the description of the weather, the grid weather make use of different technology so that this will have higher accuracy on weather information. So we just ignore the observed data. The nearer grid station will have the higher possibility to be selected. The total data matrix will contain the weather information and the prediction target. We will put all the raw attributes we have about the weather to train the model.

regularization: For RNNs, we should regularize the input data first. we just use the mean and std to map the data into z-distribution.

2.Experiments

2.1settings:

lstm parameters: It is hard to say which parameters can achieve the best result. so here we just use the fixed parameters to evaluate different settings.

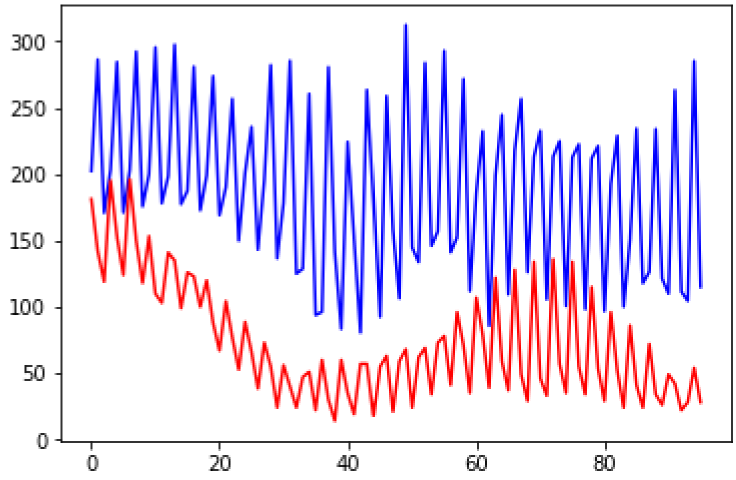
loss function: we use mse as the loss function. the only problem is that mse will give high penalty on the large difference values, however, the smape is not the same. Here we test different loss function. the result is in following. By now, the best loss function may still be mse.

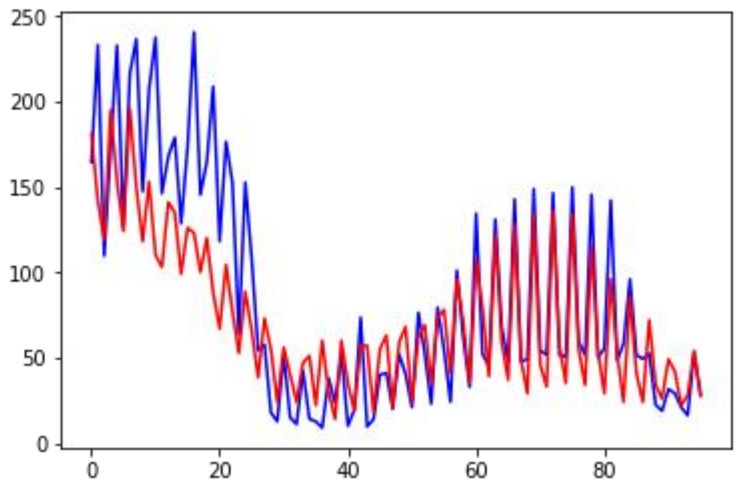
method	val loss	smape on last 48 hours
mse	562.7996	0.816452
mae	50.7740	1.113866
mape	51.3665	0.888768

2.2 Two different models

There may be two different kinds of training method. First, we will not use the targets as attributes, but this will not be reasonable because the target values have high relationship with their previous state. Second, we use the targets as attributes, but we will meet another problem. We can only use the prediction to fill in the gap for the 48 hours prediction.

The results are shown here.The red one is the ground truth and the blue one is the prediction. The upper one is method 1, the lower is method 2. for method 1, the model can learn some of the trends but if we mask the target values, the prediction will be far away from the ground truth. for method 2, if we can have the actual value for timestamp t, the prediction of timestamp t+1 will be pretty good, as shown in the image, but if we could not get the accurate value at t, the t+1 value will be much worse.





3. conclusion on LSTM model

The LSTM model may performs well on other dataset, but for this data, we may not use lstm to do the prediction. One potential problem for this is the missing values. For the other datasets, they have full data in time series, which may be one of the foundation for such network, however, for this task, we have high probability to fill in the wrong value which will mislead the model. The fact that smape of 48 hour prediction is much lower than other model is also another problem. So we will use other machine learning methods to solve this problem.

2.2 LighGBM

Since LSTM performs better than the previous one-time prediction method, we guess that even the prediction can be wrong and error can be accumulated, it's still important to include the latest information of pollution index for the model to learn the trend. Thus we decided to include the prediction result in the features. To do this, we sort the test features by time and then group the dataframe by time. We then run the features extraction and prediction for 48 times and append the predicted result to the resource pool for each iteration. In this way, we can extract features from the prediction result in the next time frame. Here we use LightGBM to predict the values. Details of the implementation can be found in **rbr_predict** function in main.py.

3. Conclusion

We use row by row prediction finally and use LightGBM as the prediction models. Compared different methods we have tried, we found that though the one-time prediction method seems to be free from accumulating prediction errors, it does not do well in learning the trend. What's more, the matrix is very sparse for the one-time prediction models(35 one hot stations columns + 4 station type columns + 48 diagonal matrix) and it consumes a lot of memory. On the other hand, row by row prediction method can do well in predicting the trend. And because LightGBM has better performance than LSTM in our validation set, we use LightGBM as our final prediction model.

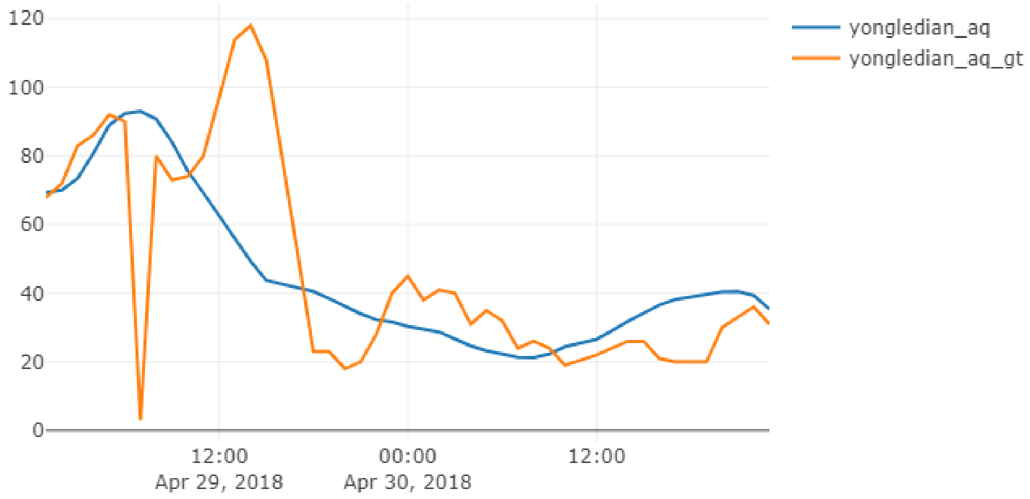
Note that all the pollution index for zhiwuyuan air quality station is null in April 2018, we use the mean of prediction as the final values for zhiwuyuan air quality station.

VI. Results & evaluation

We use the last two days(2018-04-29 ~ 2018-04-30) as the validation set.

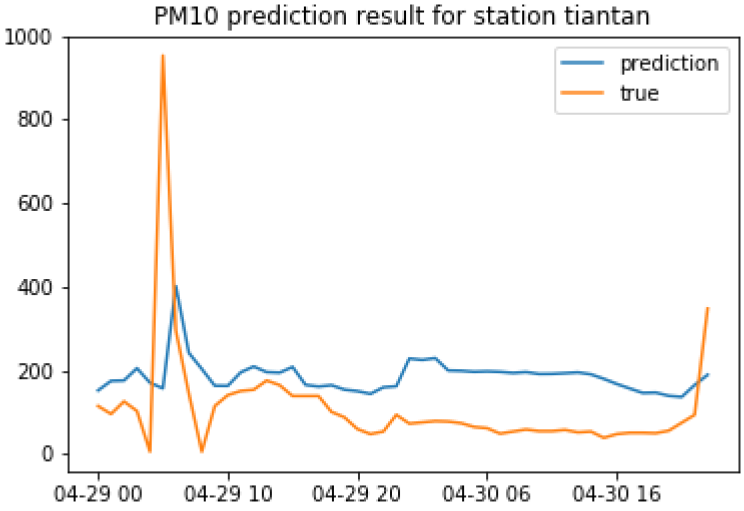
PM2.5

The total validation smape is around 0.48. The following image is the sample output for one station on the last two days. Although the prediction could not contain the sudden changes compared with the ground truth, it successfully captures the main trend of those two days, which result in a not very high smape value.



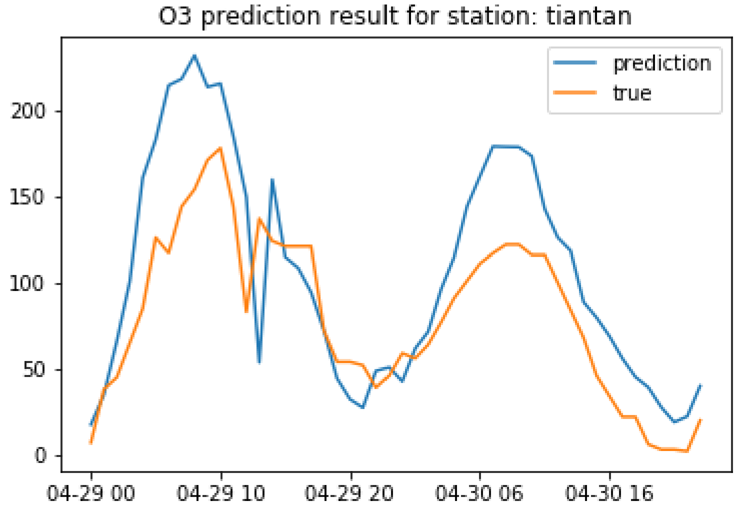
PM10

The smape for validation set is 0.61. The following graph shows the prediction result for tiantan air quality station.



O3

The overall smape score is 0.40. The following figure shows the prediction result for tiantan air quality station. Our model can roughly predict the trend. But still can not exactly predict the value.



VII. Division of labour

zheng xiaowen: PM10 and O3 analysis on final model.
wang zhicong: Experiments on LSTM. PM2.5 analysis on final model.