5002 Project: Air_Quality_Prediction

GROUP_NUM:62
NAME: Guo, Mingjun
ITSC: 20527755
E-MAIL: mguoaf@connect.ust.hk

Abstract. This report is about performing air quality prediction algorithm. The algorithm is to predict concentration levels of several pollutants (PM2.5 | PM10| O3) over the coming 24*2 hours (two day) for 35 stations in Beijing, China. The prediction algorithm is mainly composed of 4 parts: data preprocessing, feature engineering, modeling and results. Finally, the predicting results show a good match with the original data and shows the good performance of the predicting method.

Keywords: air quality prediction

1. Introduction

Over the past years, air pollution has become progressively more severe in many large cities, such as Beijing. In 2011, in the Los Angeles Times cited Dane Westerdahl, an air quality expert from Cornell University, describing the air quality of Beijing as ’downwind from a forest fire’. Among different air pollutants, air particles, or Particulate Matters (PM), are one of the deadliest forms. Particles with a diameter of 2.5 μm or less (called PM2.5) can penetrate deeply into human lungs and enter blood vessels, causing DNA mutations, cancer, central neural system damage, and premature death. Existing biomedical research demonstrates that, once inhaled, PM2.5 can hardly be self-cleaned by the human immune system. Therefore, accurately monitoring and predicting the concentration of PM2.5 and other air particles have become increasingly crucial. With precise predictions of air pollution levels, the public and governments can respond with appropriate decisions, such as closing schools and discouraging outdoor activities, to greatly mitigate the harmful consequences of air pollution.

In this project, we are requested to predict concentration levels of several pollutants over the coming 24*2 hours (two days) for 35 stations in Beijing, China. And we are provided air quality data and meteorological(weather) data from January 2017 to April 30 (including) 2018, and We need to predict the pollution level of PM2.5, PM10, O3 between May1 to May 2, 2018 (once an hour, 48 times for one station in total).

The reminder of the project is organized as follows:

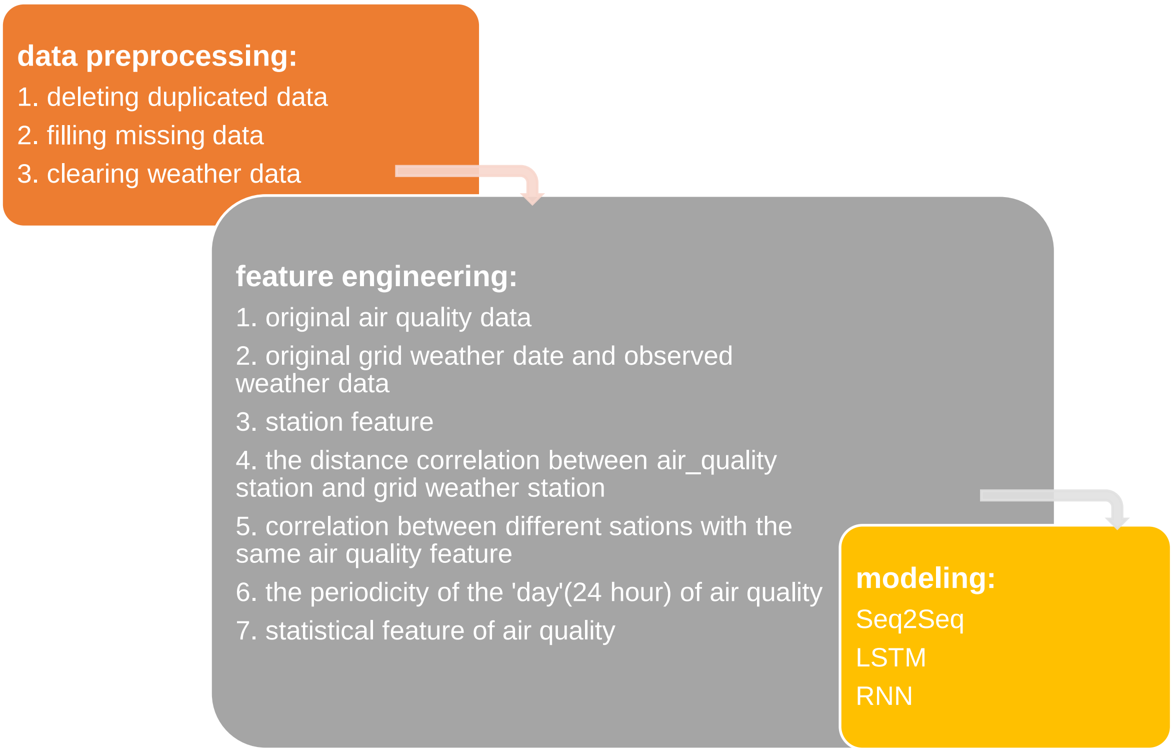
Section2 (approach):

- Data preprocessing

- Feature engineering
- Modeling

Section3(experiments):

- Predicting results
- Results visualization.



data preprocessing:
1. deleting duplicated data
2. filling missing data
3. clearing weather data
feature engineering:
1. original air quality data
2. original grid weather date and observed weather data
3. station feature
4. the distance correlation between air_quality station and grid weather station
5. correlation between different sations with the same air quality feature
6. the periodicity of the 'day'(24 hour) of air quality
7. statistical feature of air quality
modeling:
Seq2Seq
LSTM
RNN

Figure.1. the architecture of my method

2. Approach

In this section, we demonstrate the architecture of my method, which is shown in Figure 1. It shows that method mainly consists of three parts, namely data preprocessing, feature engineering and modeling. The details of each part are demonstrated in the following sections.

2.1 data preprocessing

Before I started to build the model and run the algorithm, I preprocess the data (both air quality data and meteorological data). And I mainly take the following steps:

- Deleting the duplicated data.
- Filling missing data.

(1) for air quality data:

a) deleting the duplicated data:

air quality dataset begins at 2017-01-01 14:00:00 and ends at 2018-04-30

23:00:00, and totally have 10726 tuples. We define ‘duplicated data’ by the tuples with same station at same time point, and there is no duplicated data.

b) filling missing data:

3 missing classification:

(1) lack of integrity: at some ‘time’ point, all features of all stations are missing;

(2) lack of examples: at some ‘time’ point, all data of some stations are missing;

(3) partial lack of examples: at some point in ‘time’, part of the data of some stations are missing;

solution:

for (1): drop_hours: continuous missing time no more than 5 hours a day, give up using the data, otherwise keep_hours: use the nearest time data interpolation method to fill;

for (2) (3): use the data of the nearest station to interpolate.

experiments:

for (1): from 2017-01-01 14:00:00 to 2018-04-30 23:00:00, there are 11626 hours, but dataset only contains 10769 hours, which show there are 857 missing hours for all stations (621 drop hours + 236 keep hours), so we interpolate 236 hours by the nearest time (before missing and after missing) to fill, and put the data of 621 hours by np.nan;

for (2\3): first we find the nearest station for every station, and then use the data at same time of nearest station to fill the missing value for the station.

After fill the missing value, we get the 11626 hours air quality dataset.

You can see the details on the file **aq_data_preprocess.ipynb.**

(2) for meteorological data (we choose grid weather data):

a) deleting the duplicated data:

grid weather dataset begins at 2017-01-01 00:00:00 and ends at 2018-04-30 23:00:00, and totally have 11640 tuples. We define ‘duplicated data’ by the tuples with same station in same time point, and there is no duplicated data.

b) filling missing data:

3 missing classification:

(1) lack of integrity: at some point in ‘time’, all features of all stations are missing;

(2) lack of examples: at some point in ‘time’, all data of some stations are missing;

(3) partial lack of examples: at some point in ‘time’, part of the data of some stations are missing;

solution:

for (1): drop_hours: continuous missing time no more than 5 hours a day, give up using the data, otherwise keep_hours: use the nearest time data interpolation method to fill;

for (2) (3): use the data of the nearest station to interpolate.

experiments:
for (1): from 2017-01-01 00:00:00 to 2018-04-30 23:00:00, there are 11640 hours, but dataset only contains 11518 hours, which show there are 112 missing hours for every grid station, so If the continuous missing time is <= 5 hours, fill the missing value by the nearest time (before missing and after missing), and If it exceeds 5 hours, replace by np.nan.
for (2\3): there is no missing value for every grid station.

You can see the details on the file **weather_data_preprocess.ipynb.**

2.2. feature engineering

we extract the features in this part, and the detailed data exploration and feature engineering steps listed as follows:

- original air quality features: we use the data of the first 5 days to predict the next two days.
- statistical feature of air quality: we statistic the max, min, mean, std, median, range on different locations and for different feature (PM2.5|PM10|O3|NO2 etc.).
- weather features: For every air quality station, we add the temperature, pressure, humidity, wind_direction, wind_speed of the nearest grid station as the features for every air-quality-station.
- The periodicity of the 'day' (24 hours) of air quality: we statistic the range and the trend for every station and every air quality feature.
  (1) air quality data exploration:
    - the correlation of the same feature between different air quality station

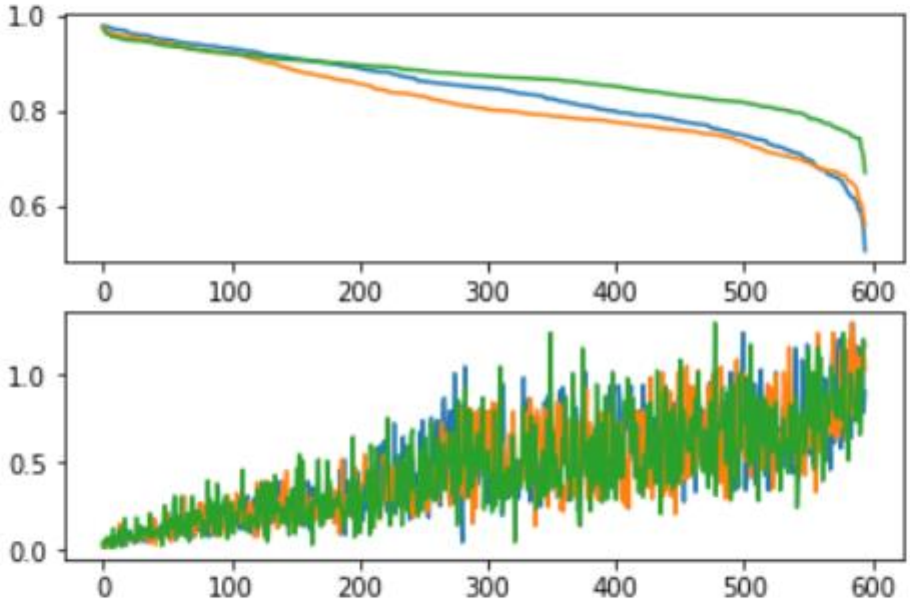


Figure.2. the correlation between distance and feature correlation.

We can see the negative correlation between "distance" and "correlation": for 'PM2.5': r= -0.7776390076845455; for 'PM10': r= -0.8233083334288264; for 'O3': r= -0.7665545841346856; and r coefficient range (0.5-1).

- The periodicity of the ‘day’ (24 hours) of air quality stations for different feature

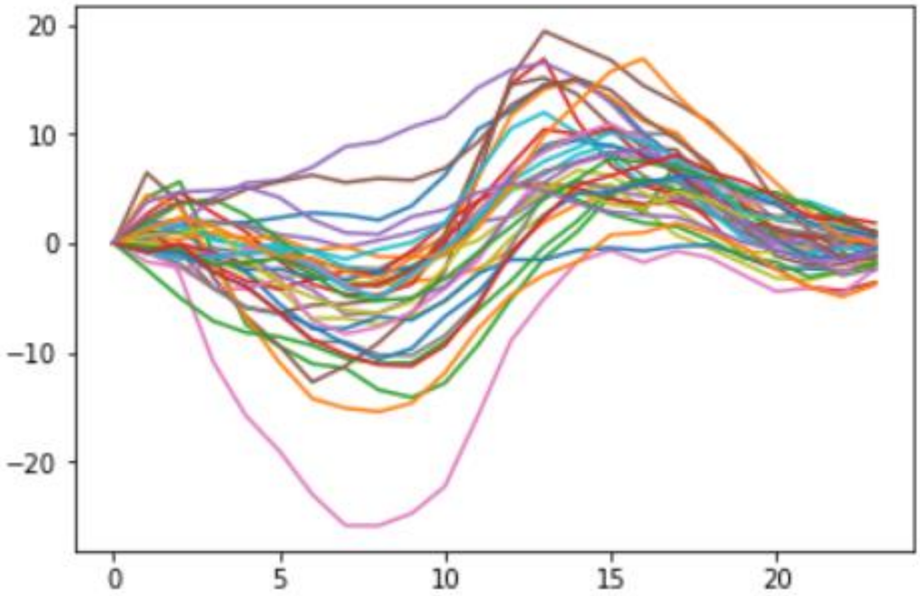


Figure.3. PM2.5

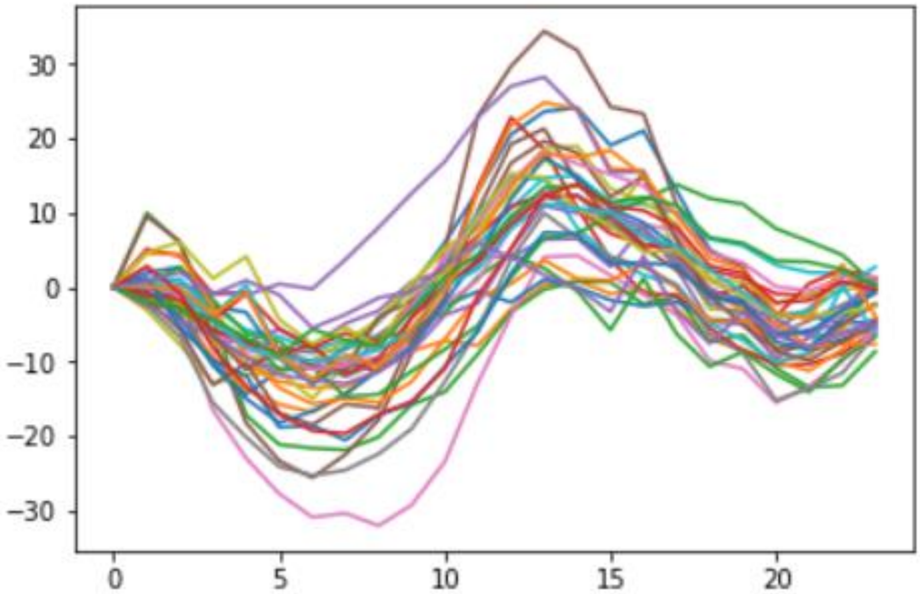


Figure.4. PM10

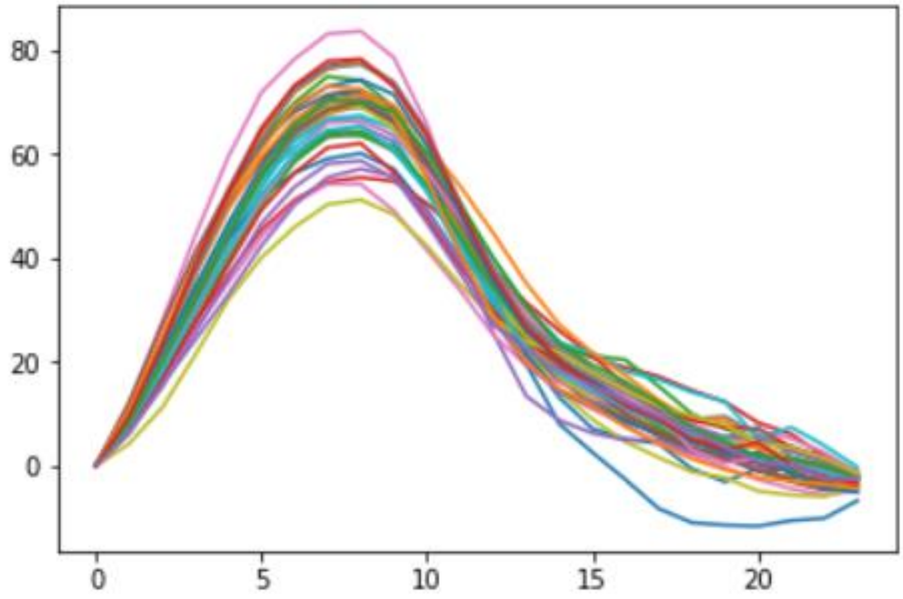


Figure.5. O3

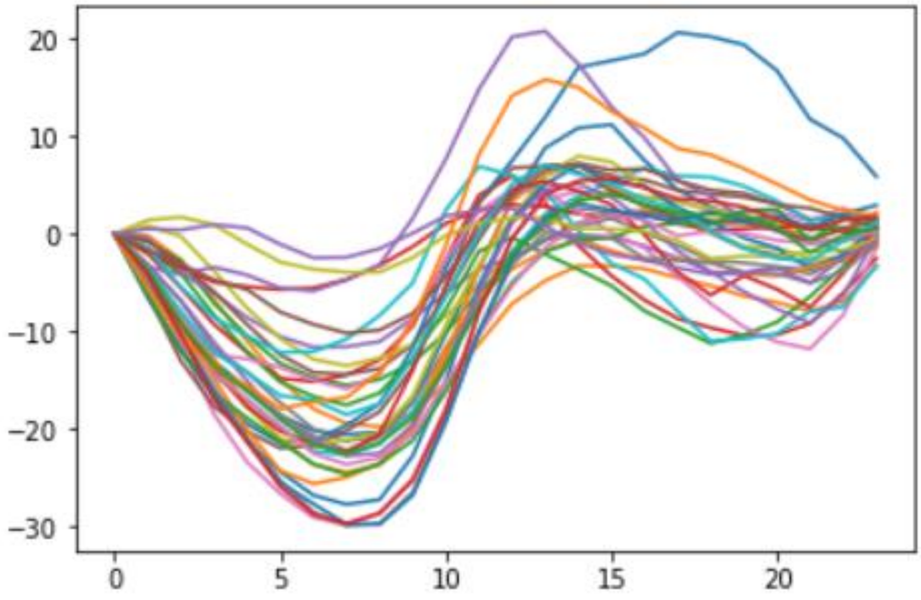


Figure.6. NO2

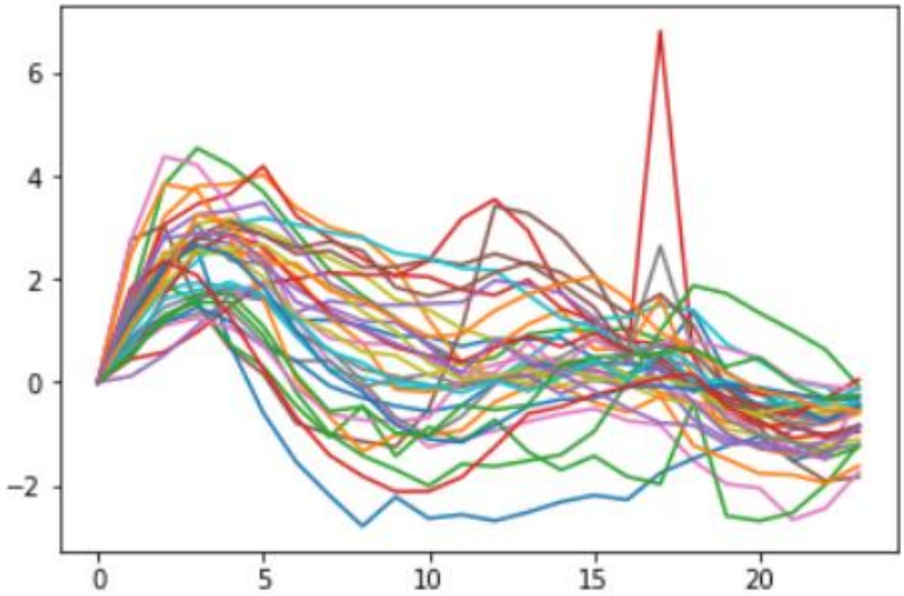


Figure.7. SO2

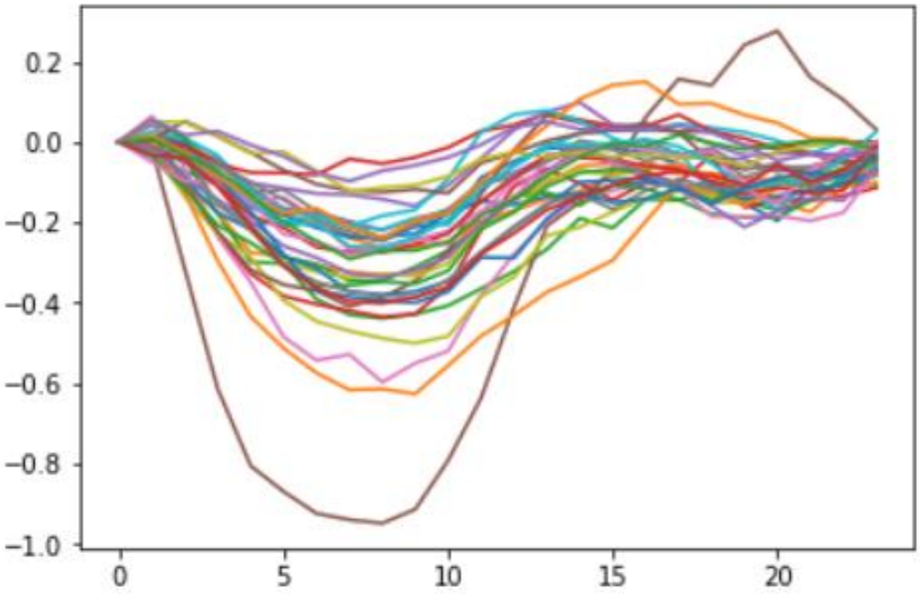


Figure.8. CO

In summary, (1) Different features show different trends within one day: single feature modeling or multiple feature modeling can be considered; (2) strong correlation between three features between different locations (r coefficient range (0.5-1)), so consider single station modeling or multiple stations modeling.

You can see the details on the file **aq_data_exploration.ipynb.**

(2) weather data exploration:

- there are 35 air quality stations, 18 observed stations and 651 grid weather stations in Beijing.

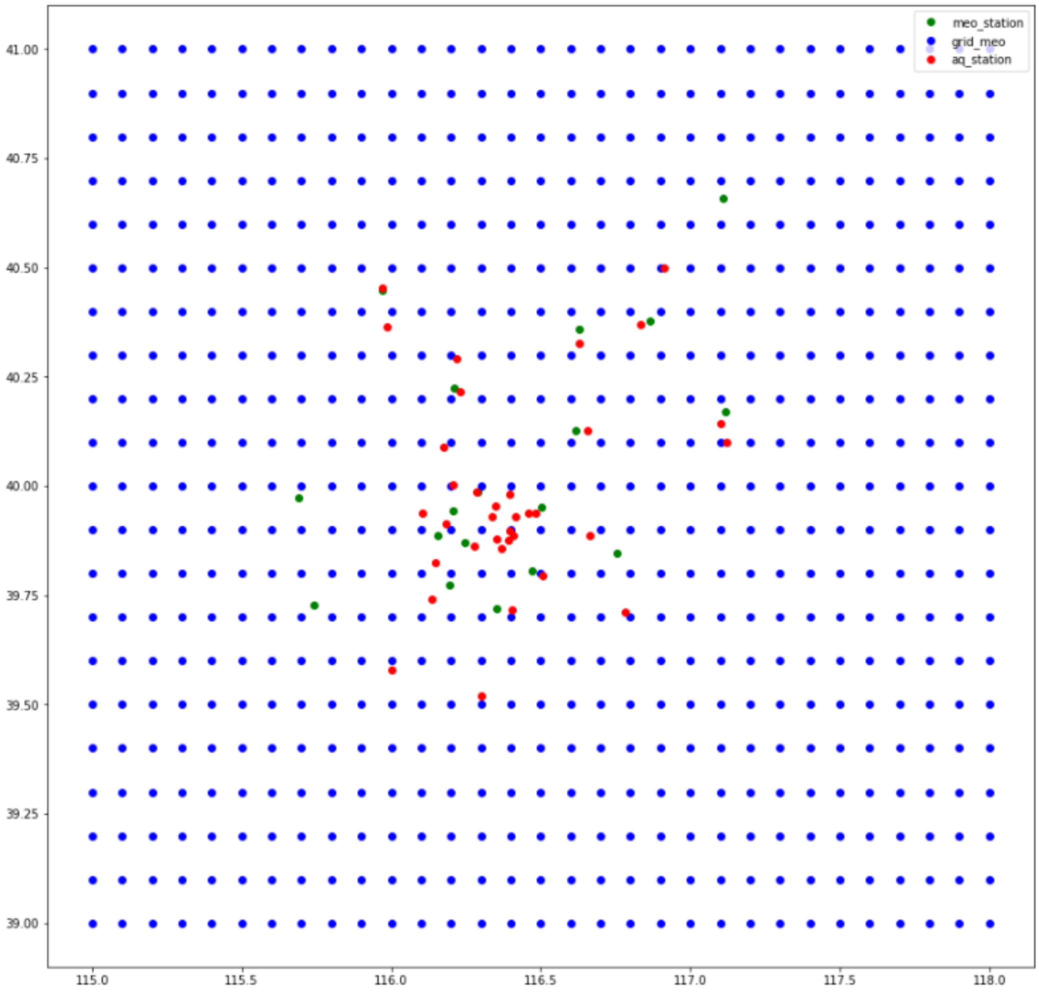


Figure.9. station distribution

We can see that for every air quality station, there is one grid weather station near it, so we can make full use of the weather feature of grid weather station.

You can see the details on the file **weather_data_exploration.ipynb.**

2.3. modeling

Seq2seq model is a machine learning model that use decoder and encoder to learn serialized feature pattern from data. Seq2seq model is applied to a lot of machine learning applications, especially NLP applications like Machine translation. In this project, seq2seq is applied to generate time series forecast of air quality. To be more specific, we adopt Stacked LSTM Architecture, the reason is: RNNs are inherently deep in time, since their hidden state is a function of all previous hidden states, and RNNs could also benefit from depth in space by stacking multiple recurrent hidden layers on top of each other, just as feedforward layers are stacked in conventional deep networks.

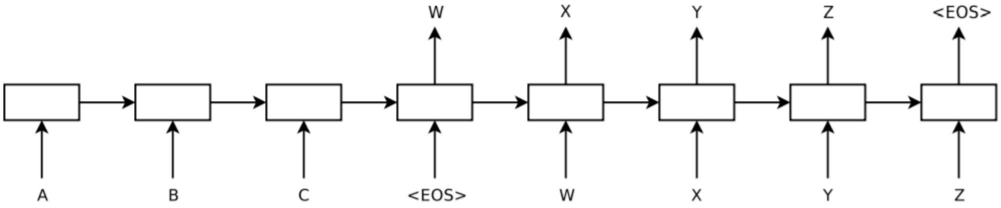


Figure.10. the basic graph of seq2seq model

We use first 5 days air quality features and weather features to predict next two days air quality features (PM2.5|PM10|O3) together. And in the training time, our last parameters are set like:

- Hiddendim=512
- num_stacked_layers=3
- learning_rate=0.0001
- regularization: L1
- GRADIENT_CLIPPING=2.5
- Iteration=100

To be more specific, we use the norm data for training and validation. And in the last, when we get the prediction result from norm data, we transform the prediction result by using the mean and std of original data to get our final prediction.

You can see the details on the file **main.py, train_seq2seq.py, seq2seq_model.py.**

3. Experiments:

We train our model by using 2500 time series (every is composed of 5 days features as input and 2 days features as output) chosen from 2017-01-02 00:00:00 to 2018-03-30 23:00:00, and use the 21 time series chosen from 2018-04-01 00:00:00 to 2018-04-30 00:00:00 as the validation sets to choose the best model with the least SMAPE, and then use the last five days for 2018-04 to predict the air quality for May 1 and May 2, 2018. And the results show the matching trend with original data.

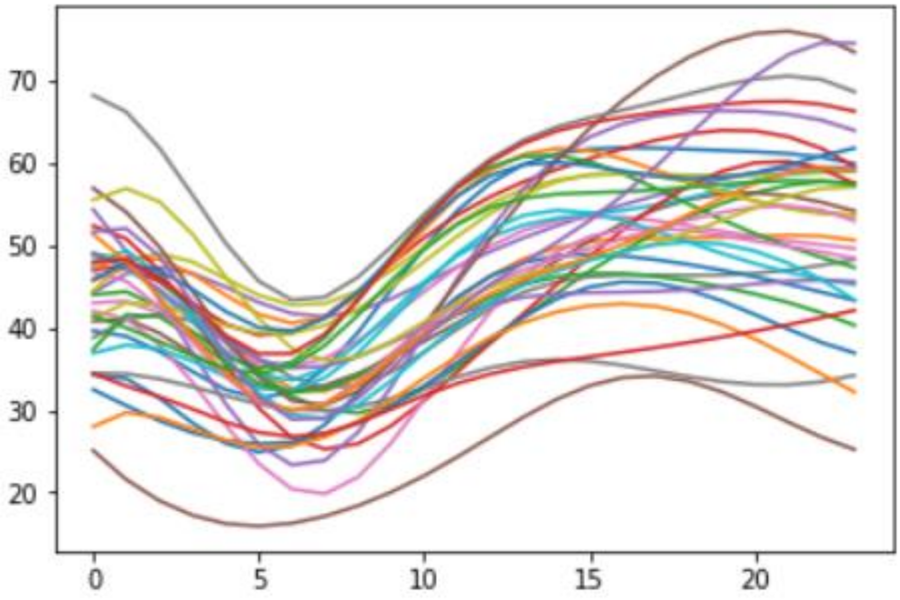


Figure.11. output PM2.5

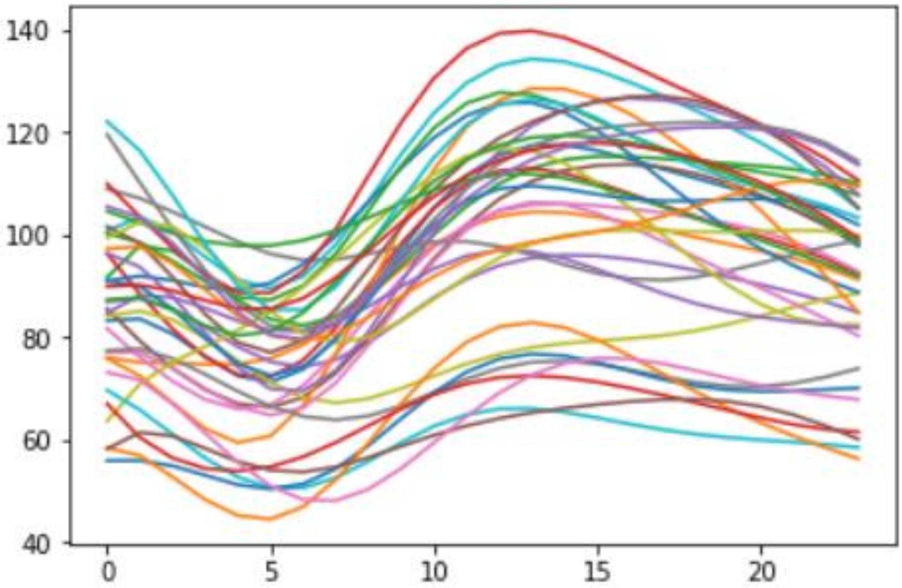


Figure.12. output PM10

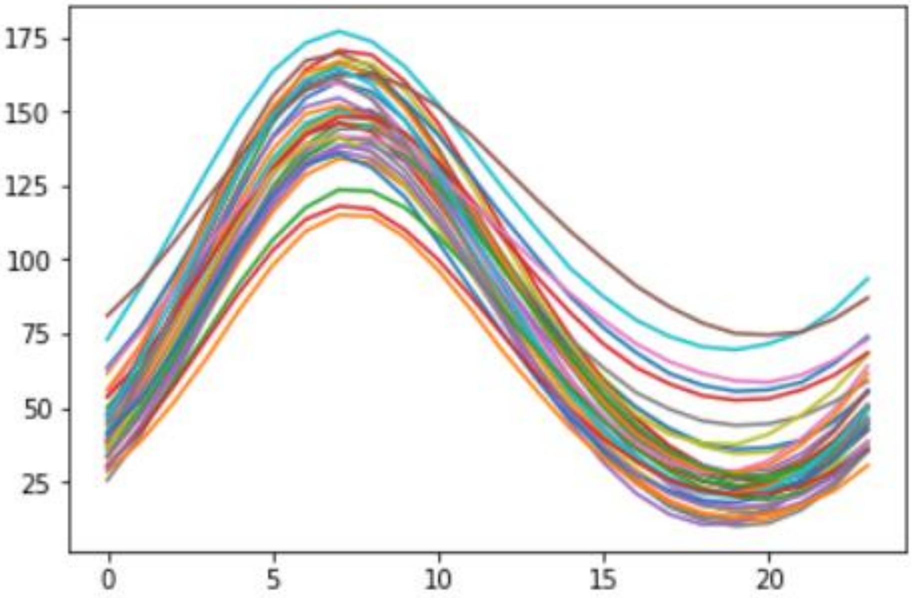


Figure.13. output O3

You can see the details on the file **output_visiualization.ipynb**.

4. Conclusions

In this report, we present my method to predict air quality for 35 stations in Beijing. In summary, in data preprocessing part, we fill the missing value according to the distance correlation of different station and time correlation for same station; in feature engineering part, we find the nearest grid weather station for every air quality station and then make fully use of the weather feature; in modeling part, we adopt seq2seq RNN model to generate time series forecast of air quality. Noting that I will extract more useful features like Holiday and week feature and adopt ensemble methods to predict better.

Thanks for your time.

References:

[1] http://articles.latimes.com/2011/oct/29/world/ la-fg-china-air-quality-20111030.

[2] Becker, S., Fenton, M.J., Soukup, J.M.: Involvement of microbial components and toll-like receptors 2 and 4 in cytokine responses to air pollution particles. American journal of respiratory cell and molecular biology 27(5), 611–618 (2002).

[3] Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS 2014).