Report of Machine Learning Project of M2
– at EDF Lab

Junfeng CHEN Yuan ZHANG

March 9, 2018

Contents

1 Introduction

In this project, we focus on different statistical methods (for example, ...) for data prediction. We implement different methods for predicting the data of pollution (the highest 1-hour measurement of NO2 for each day).

Our goal is to predict the data of NO2 in the year 2017 from the data in the year 2015 and 2016. We use cross validation, split the data in the year 2015 and 2016 into train set and validation set, and take the data in the year 2017 as test set.

The process of data preprocessing, feature extraction, model designing and analysis of the result are shown as follows.

2 Description of the data

There is the data of NO2 pollution for the year 2015, 2016 and 2017. The features for the data are described as follows:

- Date: Date;
- Daily Max 1-hour NO_2 Concentration(ppb): the highest 1-hour measurement of NO2 in the day;
- DAILY_AQI_VALUE: daily Air Quality Index value;
- TEM: temperature (°C).
- DEW: Dew Point (°C)
- HUM: Humidity (%).
- SLP: Sea Level Press (hPa).
- VIS: Visibility (km).
- WIN: Wind (km/h).
- PRECIP: precipitation (mm).
- MUSIC: if there are music events. (categorical variable)
- Weekday: weekday corresponding to the Date. (categorical variable)

Since value of NO2 contributes to the calculation of DAILY_AQI_VALUE, so we will not use DAILY_AQI_VALUE for this prediction problem.

3 Model

3.1 Lasso Regression and Ridge Regression

3.1.1 Data Preprocessing

First, we check the data type of the features. We notice that the variable 'PRECIP' is not totally numerical, there is a special character 'T' in the columns of precipitation, it means that there is precipitation but without getting the exact values. here, we just remove the rows with the precipitation indicating 'T', just like what we do to 'NA', because there are only a dozen rows like that.

Second, we standardize the data like this:

$$x_{new} = \frac{x - \mu}{\sigma}$$

Here, x is the data of a feature, μ,σ is the mean and standard deviation of the data of the feature.

3.1.2 Feature extraction

For Lasso Regression, we do not pick features because the training process of Lasso Regression Model includes the process of feature extraction.

For Ridge Regression, we check the correlation matrix of the features and remove highly-correlated features (the threshold for the correlation is 0.9) and also refer the result of feature extraction in Lasso Regression.

3.1.3 Parameters of the Model and Results

• Lasso Regression
We simulate with the lambda ranging from 0.001082595 to 3.890646410, which is automatically picked in the function glmnet(). We consider 2 criterion for selecting parameters: root mean square error(RMSE) and mean absolute percentage error(MAPE).

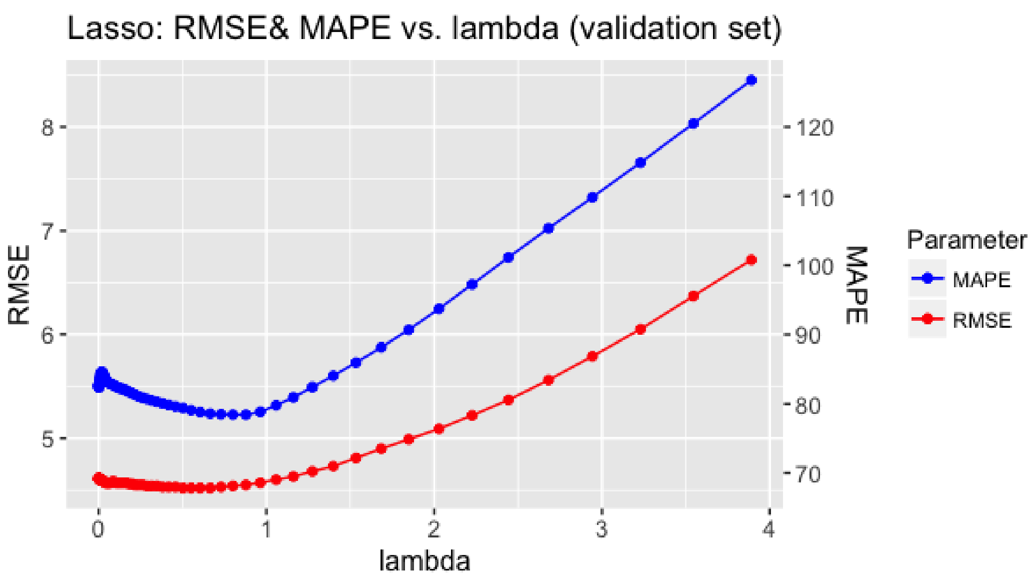
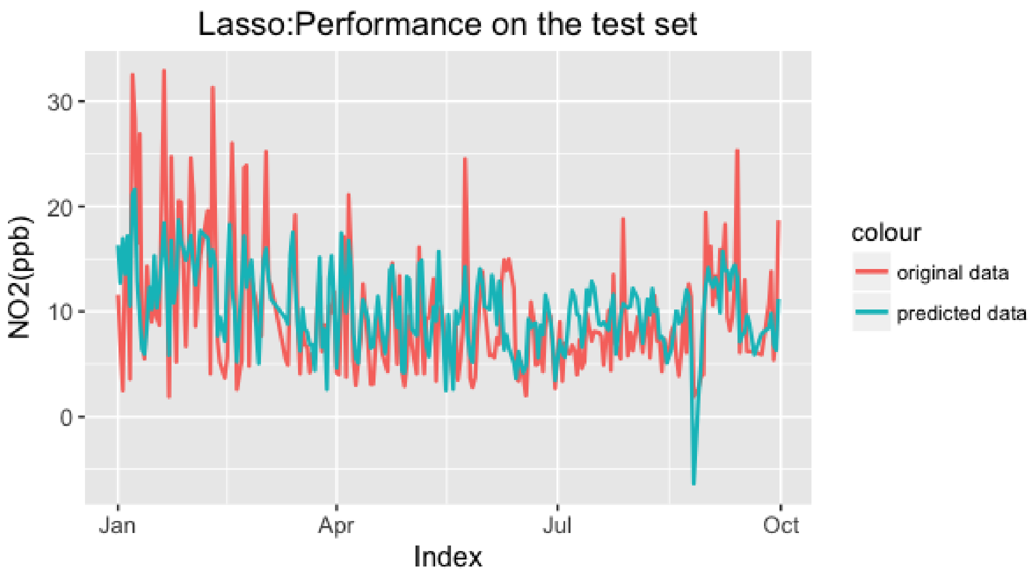


Figure 1: RMSE and MAPE of Lasso Regression on the validation set

We choose lambda=0.8781266 since it reaches the lowest mean absolute percentage error and keeps a low root mean square error. We test it on the test set and get the result as below.



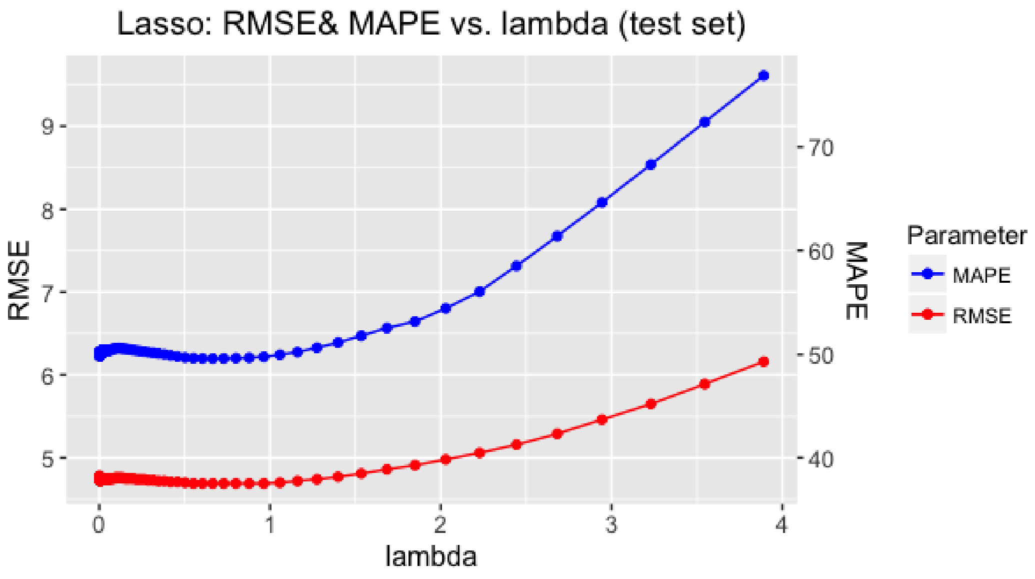


Figure 2: The performance of Lasso Regression on the test set

From the performance of Lasso Regression Model on the test set, we can see that it is not satisfying even though the predicted data has similar trend with the original data, because the lowest RMSE(4.55) and MAPE(49.58) are not small enough. Lasso Regression Model performs badly in the month January to March,June and July, in which month we can see high fluctuation.

But we also notice that the model has similar performance of RMSE on the test set compared to the validation set, but get a lower MAPE on the test set, which means there is under-fitting.

• **Ridge Regression**

Before dealing with the model, we referred the result of feature extraction in Lasso Regression Model and pick the following variables: TEM_LOW, DEW_AVG, DEW_LOW, HUM_LOW, Weekday.

We simulate with the lambda ranging from 0.3890646 to 37.1381053, which is automatically picked in the function glmnet(). We consider 2 criterion for selecting parameters: root mean square error(RMSE) and mean absolute percentage error(MAPE).

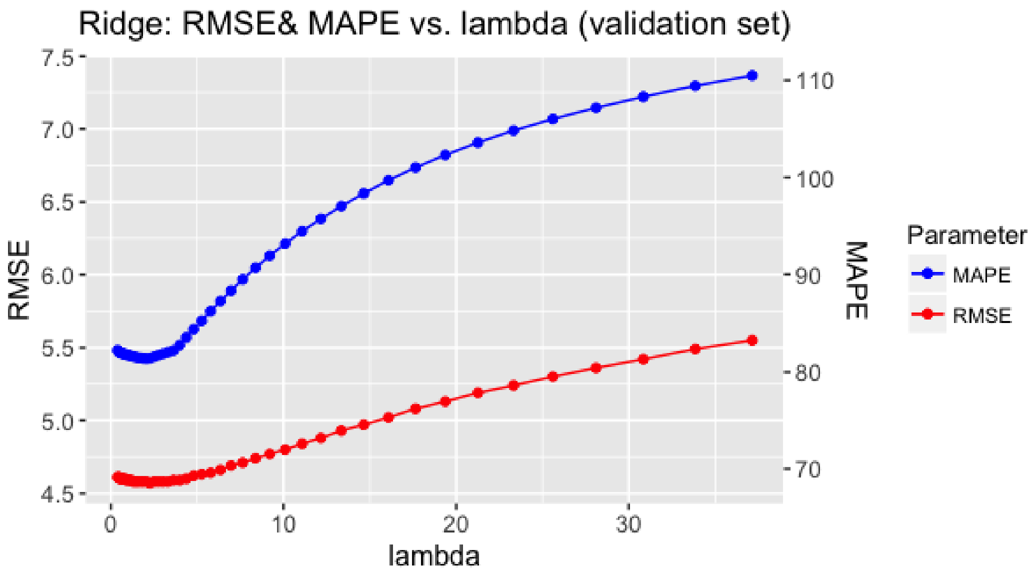
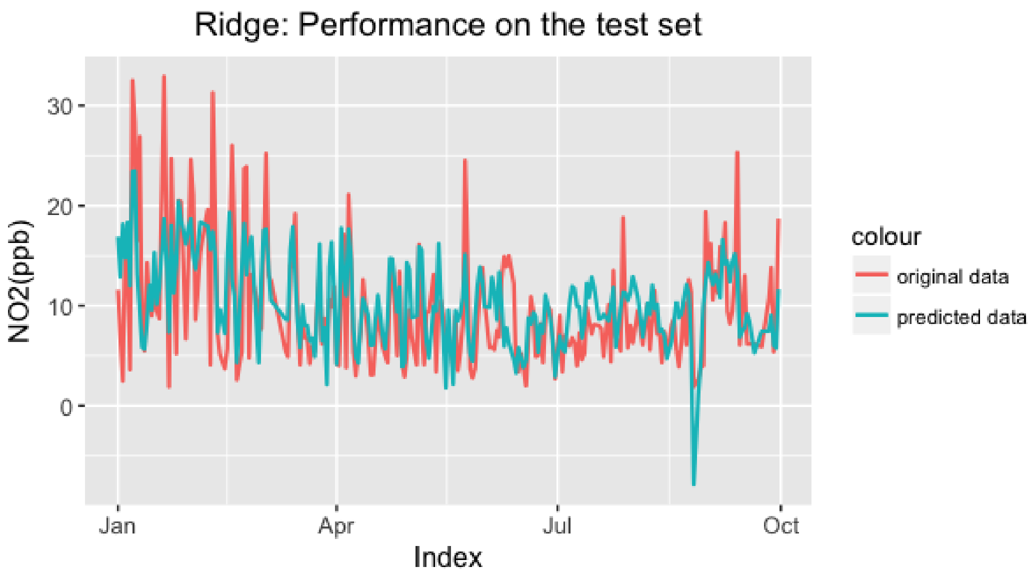


Figure 3: RMSE and MAPE of Ridge Regression on the validation set

From the figure, we observe that the lambda=2.076321 is optimal(RMSE=4.58, MAPE=81.36) on the validation set. So we choose this lambda and test it on the test set get the result as below.



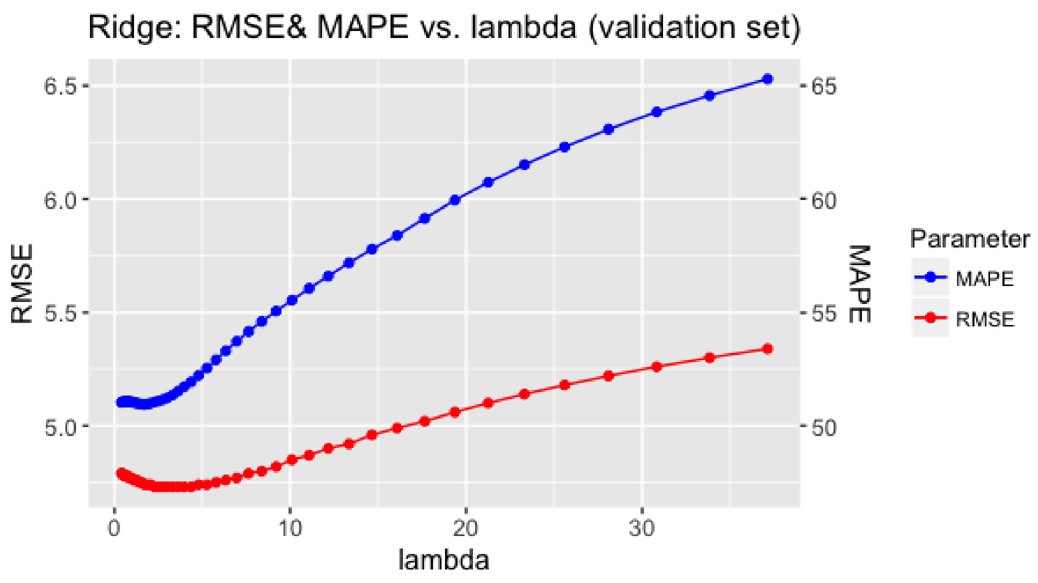


Figure 4: The performance of Ridge Regression on the test set

From the performance of Ridge Regression Model on the test set, we observe similar performance compared to Lasso Regression Model. It performs badly in the month January to March,June and July, and there is under-fitting.

3.1.4 Conclusion

From the results above, we can observe for a Lasso Regression Model and Ridge Regression Model, there is under-fitting probably because there are not enough features .

The two models are not satisfying for dealing with this prediction problem, one probable reason is that we do not get enough features , the other is that the relationship between the features we get and the value of NO2 is not linear-like. In the following chapters, we will explore more complex linear models and non-linear models.

3.2 Generalized Additive Model

3.2.1 Data Processing

We pre-process the data similarly just like in the Lasso Regression Model.

3.2.2 Feature Extraction

We extract the features according to the correlation matrix. (the threshold for correlation is 0.9).

We keep the variables TEM_HIGH, TEM_LOW, HUM_HIGH, PRECIP, VIS_HIGH, VIS_AVG, VIS_LOW, WIN_HIGH, WIN_AVG, MUSIC, Weekday.

3.2.3 Parameters of the Model and Results

The smoothing basis we choose is thin plate regression spline. We use 10-fold cross validation to select the k-the dimension of the basis used to represent the smooth term. Here, we only consider the smooth term for the variable TEM_HIGH, TEM_LOW, HUM_HIGH, PRECIP and get RMSE for different k as follows.

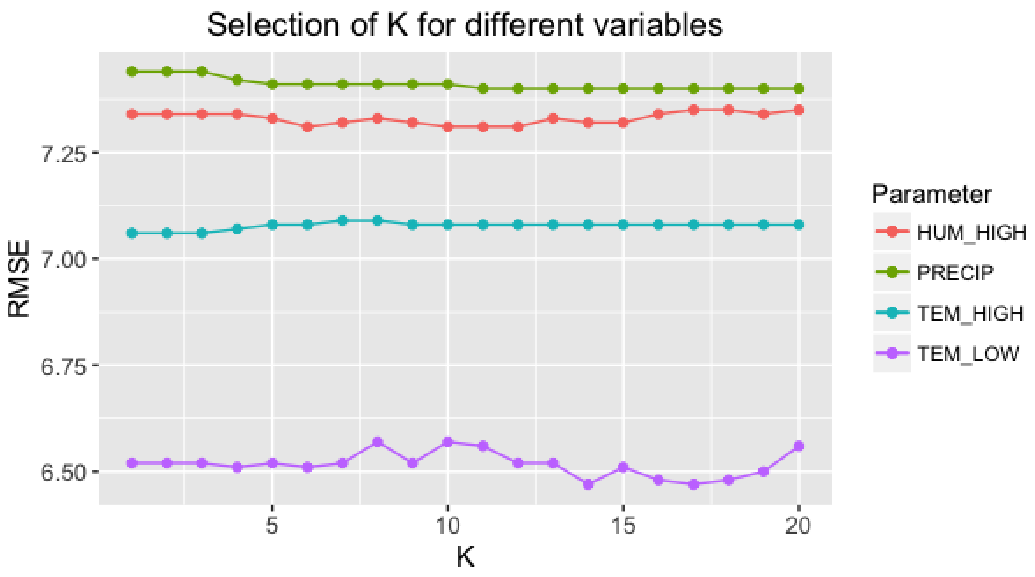


Figure 5: RMSE on the train set for different k

We can notice that for achieving lowest RMSE, we choose k=3 for TEM_HIGH, k=14 for TEM_LOW, k=6 for HUM_HIGH, k=11 for PRECIP. For other variables, we just use identity link function. The performance on the test set

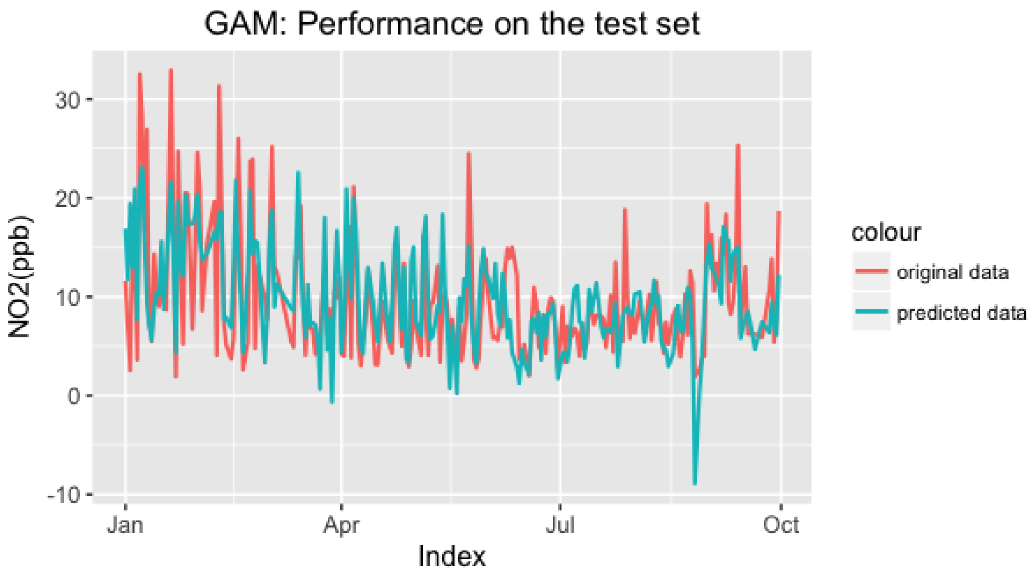


Figure 6: Performance on the test set

The RMSE and MAPE we get on the test set are 4.69, 47.72 respectively, which is better than the performance of Lasso Regression Model and Ridge Regression Model.

3.3 Decision tree and random forest

3.3.1 Data Preprocessing

This part is a bit different from other models. Tree models do not require standardized data. However the available random forest packages cannot handle missing values. In order to implement random forests, we imputed the data and turn all the columns into numeric forms, although this may not be necessary. For decision tree model, imputing is not required because rpart package deals well with missing values.

3.3.2 Decision tree

We choose rpart package to construct two decision trees, one default tree and one pruned tree. Pruning is a technique to avoid overfitting for decision tree. There are also other hyper-parameters that could be tuned, like 'xval'. Since during the course we saw that tuning the cross validation times do not boost the performance much, here we only focus on pruning, which is dominated by the complexity parameter 'cp'.
As shown in figure 7, the best complexity parameter is about 0.01. The structure of decision tree before pruning and after pruning are shown in figure 8. It can be seen that a tree without pruning is extremely complex, which may not only cause overfitting, but also occupy too much calaulation resource.

The MAPE and RMSE errors are: 60.2353 and 7.3150 for the tree not pruned, 49.6881 and 5.2154 for the pruned tree. Pruning remarkbaly improves the performance of a decision tree! The curves of original and predicted NO_2 concentration are shown in figure 9 and 10. From the curves we find that the pruned tree is more consevative, while the other tree tends to make excessive prediction.

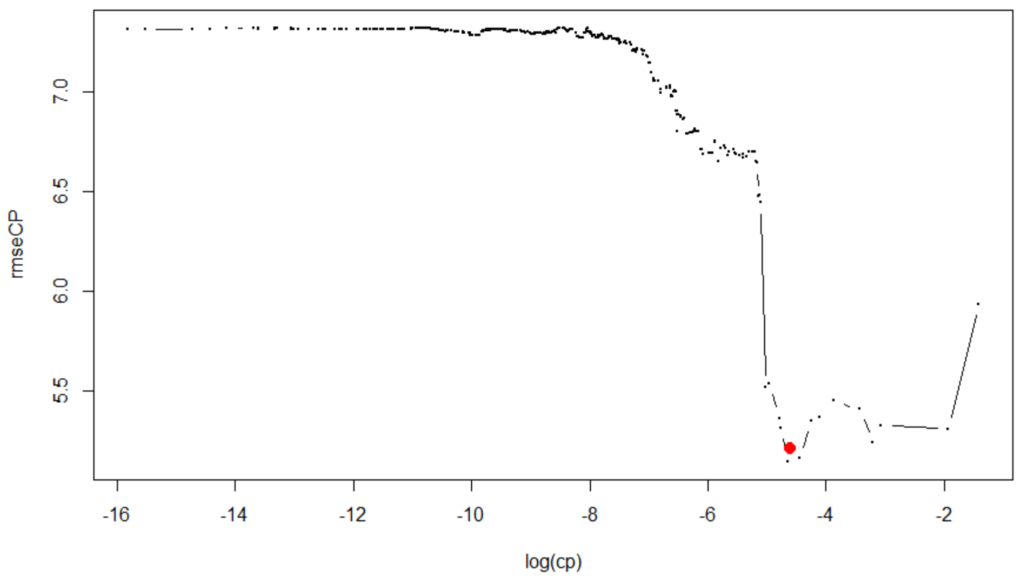


Figure 7: Influence of cp on rmse

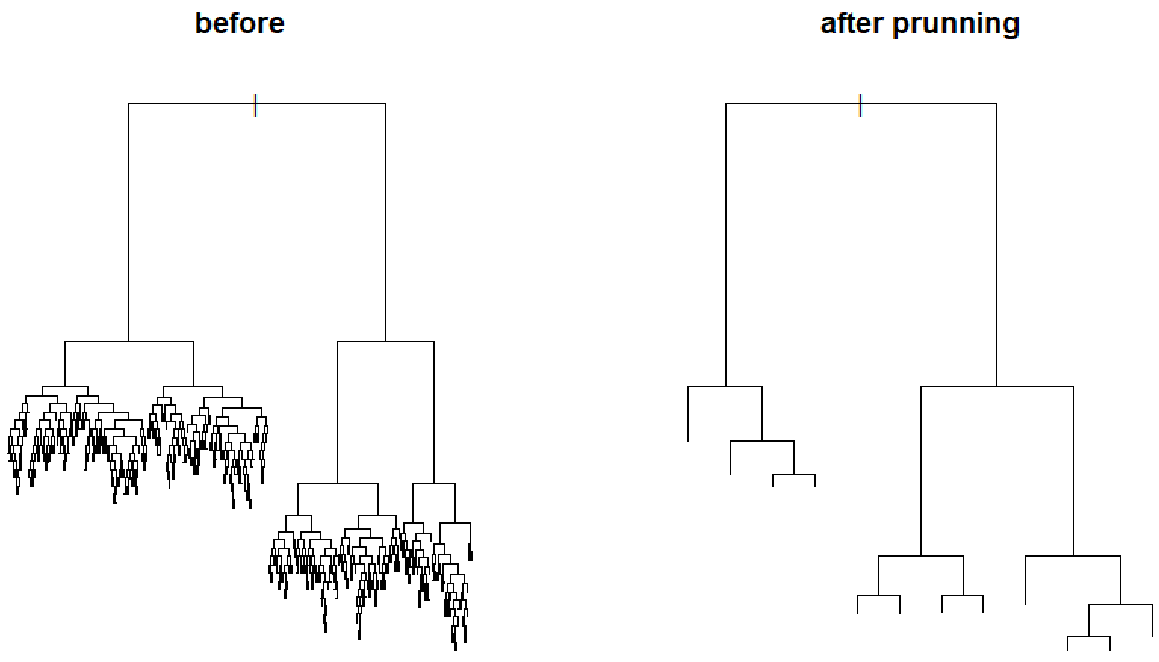


Figure 8: Influence of cp on rmse

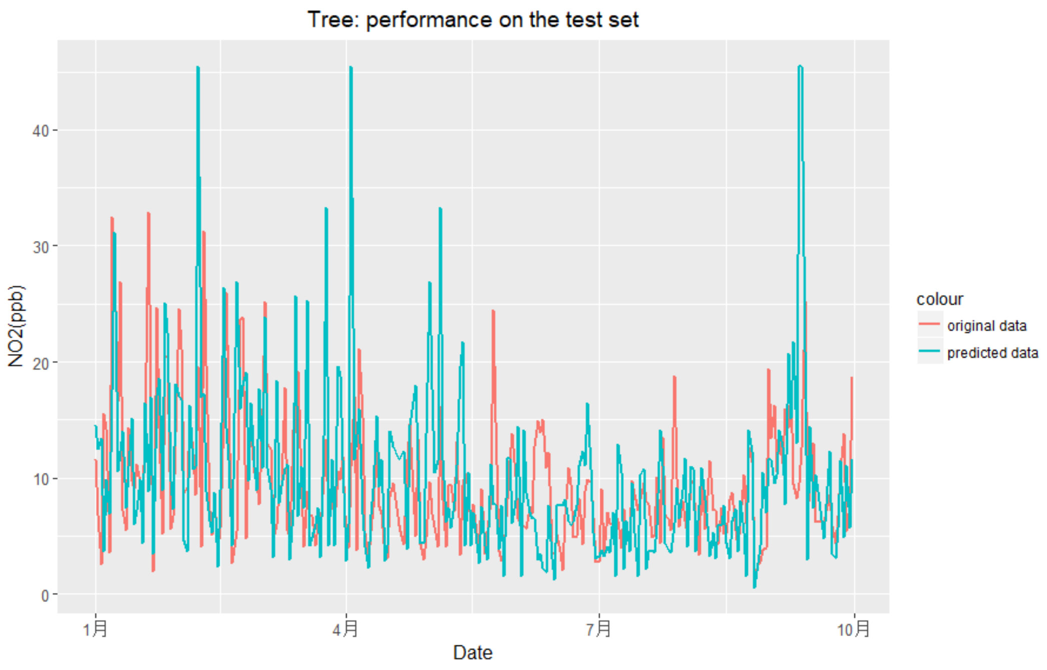


Figure 9: Decision tree: Performance on the test set

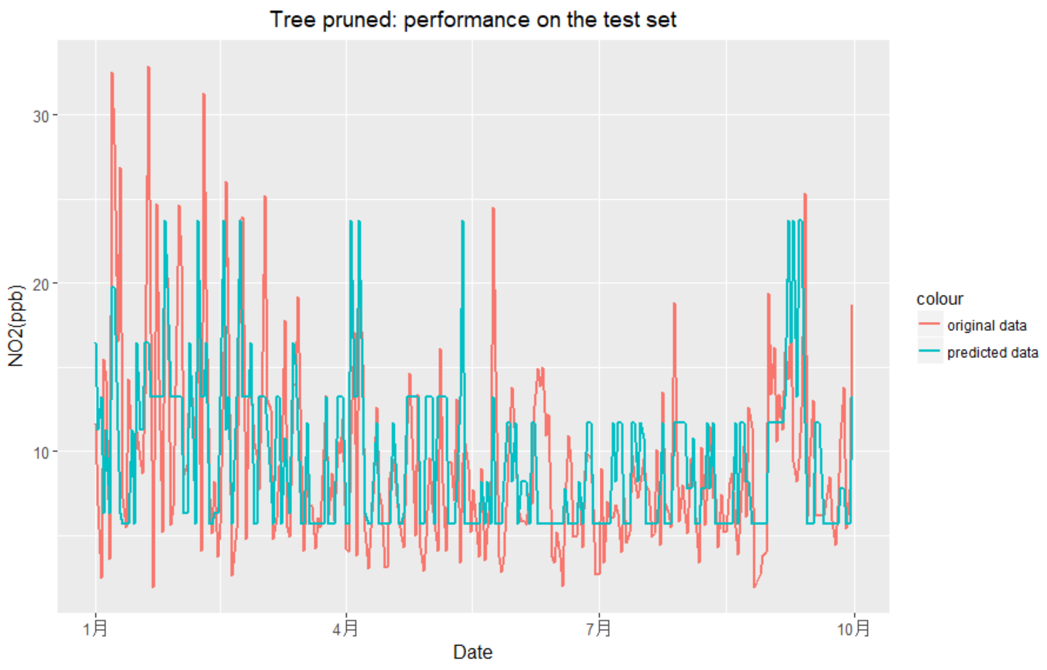


Figure 10: Decision tree: Performance on the test set

3.3.3 Random forests

Random forest is a bag of decison trees. It is able to reduce the variance, thus can hardly be overfitting. We apply package 'randomForest' and calibrate the following parameters: number of trees 'ntree', number of variables randomly sampled 'mtry', size(s) of sample to draw 'samplesize' and minimum size of terminal nodes 'nodesize'.

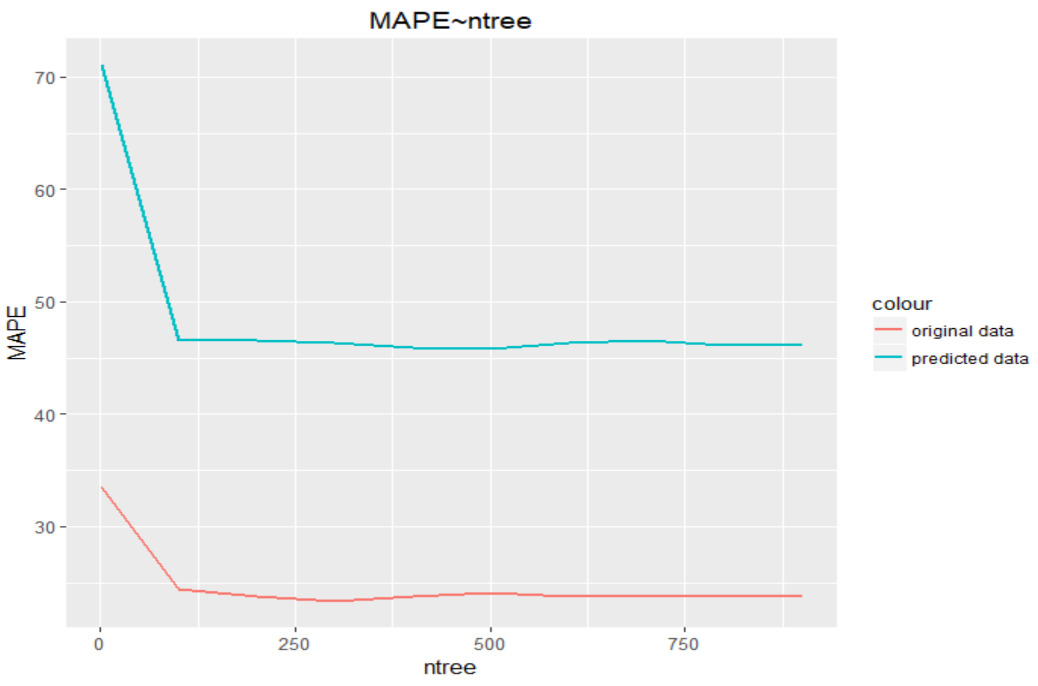


Figure 11: MAPE-ntree

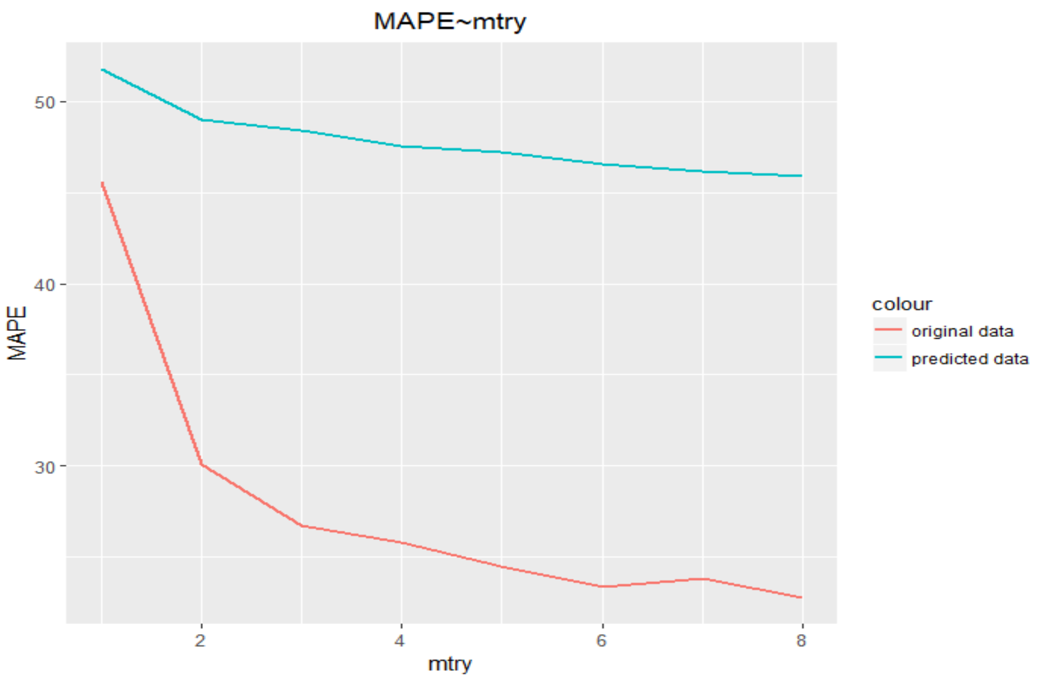


Figure 12: MAPE-mtry

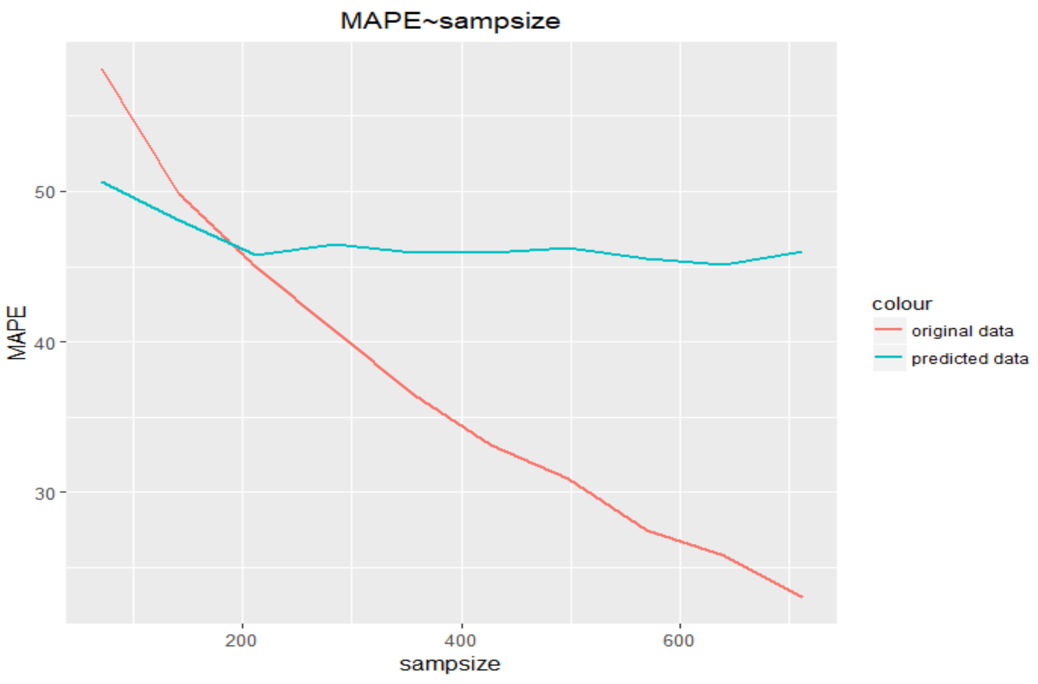


Figure 13: MAPE-sampsize

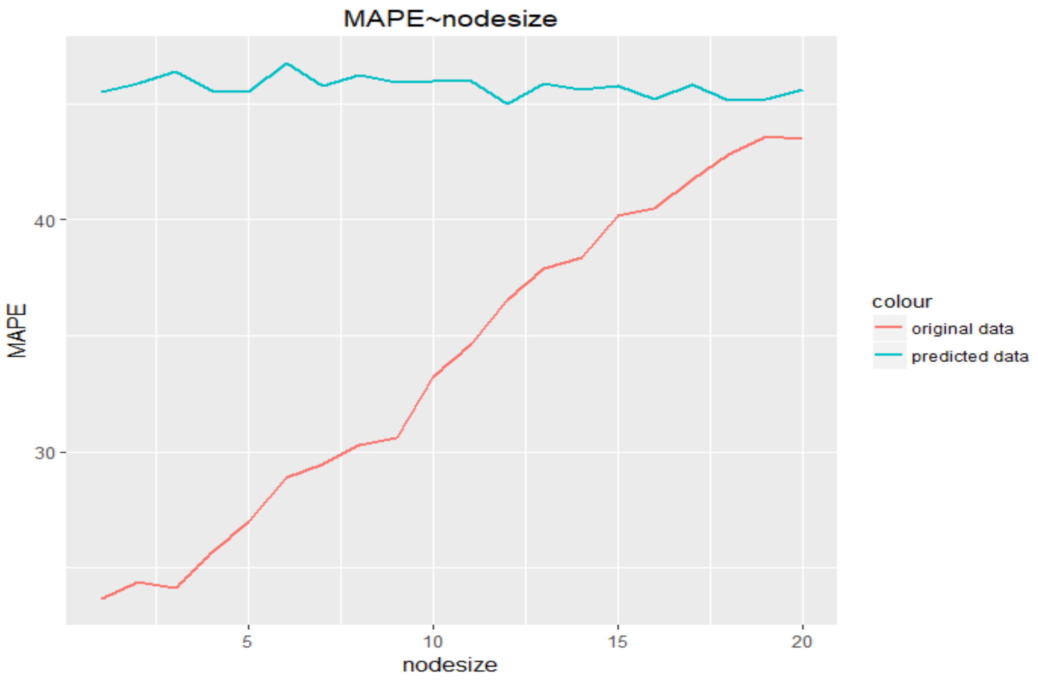


Figure 14: MAPE-nodesize

According to these figures, we decide to construct a random forest with parameters $(ntree, mtry, sampsi$
$(120, 8, 600, 1)$. Its performance is compared to a default random forest with $ntree = 500$. The MAPE and RMSE of them are 46, 0883 and 4.6078 for the default model, 34.1427 and 4.6332 for the calibrated model. Since MAPE is a better metric for average error, we can say the

calibration remarkbly improves the performance averagely. By looking carefully at the curves of concentration, we find that the two predicted curves are very similar, but at the positions with large error, the prediction of calibrated random forest is a bit more closer to the real value.

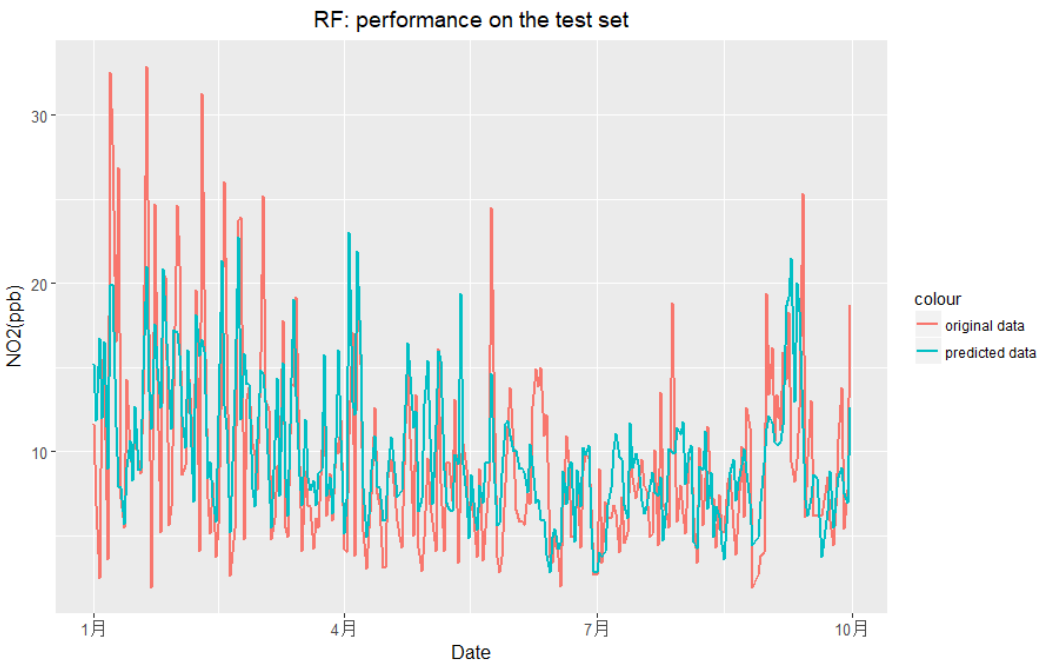


Figure 15: RF: Performance on the test set

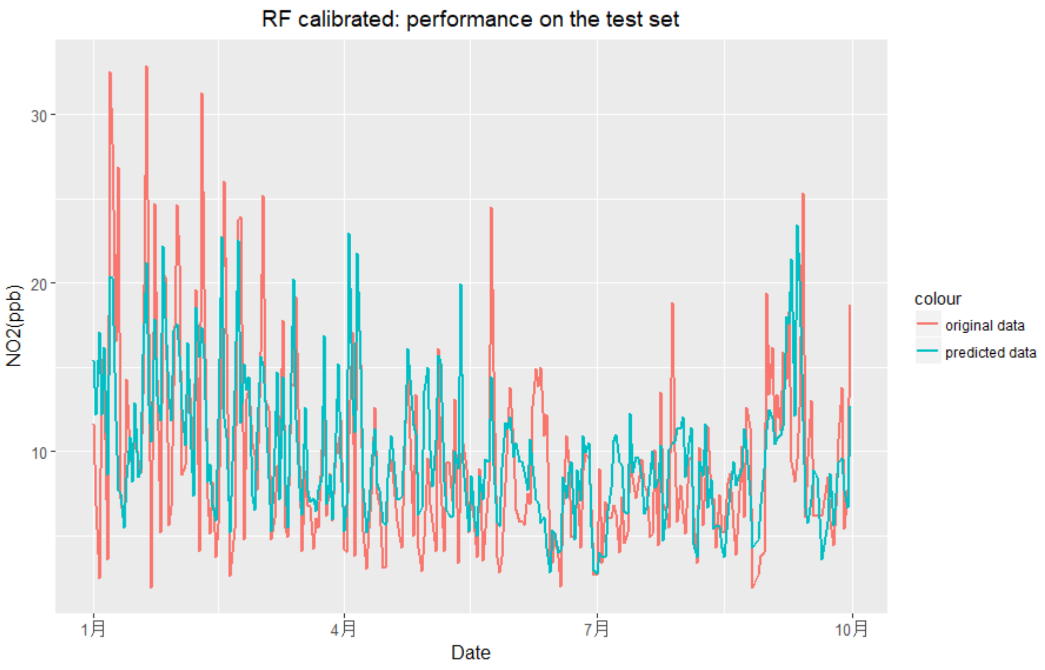


Figure 16: RF calibrated: Performance on the test set

The package randomForest allows us to get the importance of varivables, which is interesting to interpret a model. As shown in figure 17, the average wind speed is far more significant than the other variables. This corresponds to our common sense: NO_2 as a gas is easily transported along with air, thus wind is biggest contributer for the diffusion of pollution gas. This is also the reason why policy makers force coal-dependent industries installed at the downstream of monsoon far from central city. The other important features include dew point, temperature, humidity. Sea level pressure, visibility and precipitation are not so important compared to them. As a matter of fact, temperature and humidity are direct features while the other features have dependance on them. Music events do not have a strong influence as we expected.

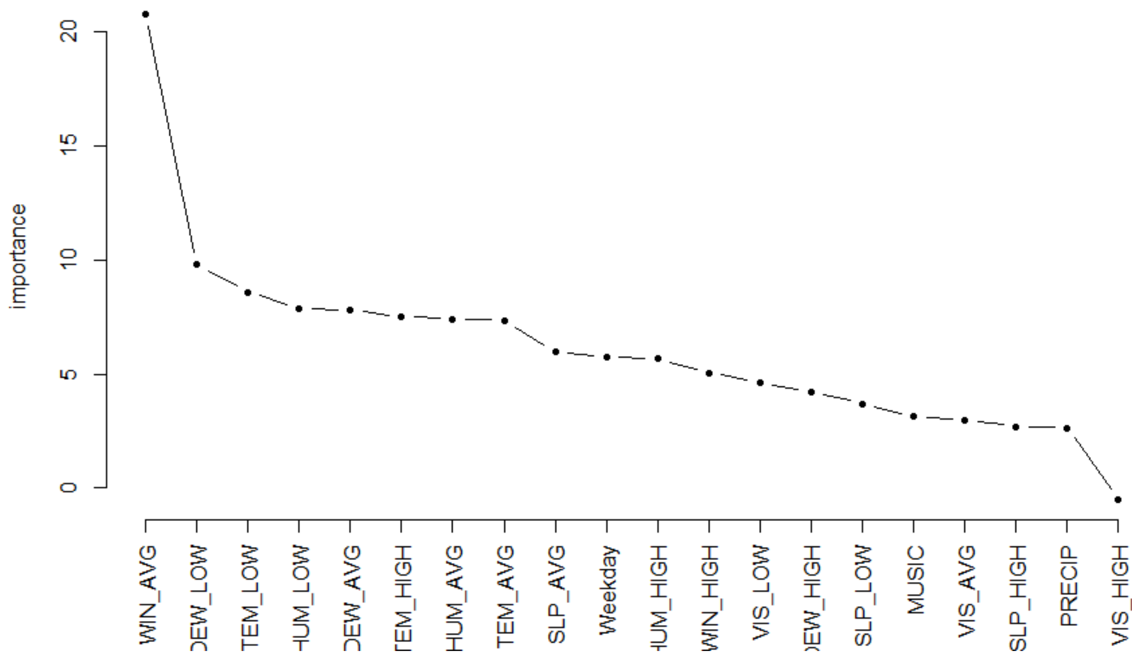


Figure 17: Importance of variables

3.3.4 Conclusion

By comparing the performance of decision tree and random forest, we see that the later is a more powerful tool, in the sense of both MAPE and RMSE. Looking at figure 10 and 16, we find that random forest learns the seasonal pattern of the data better. The decision never distinguishes the vally values. From June to September the decision tree hardly distinguish the peak values.

3.4 Time series model

3.4.1 Data Preprocessing

In order to apply effective models for the time series, we check if it is a white noise(although it does not look like any noise), and By applying Box-Ljung test to the concentration of NO_2

from 2015 to 2016, we find that the p-value$< 2.2 \times 10^{-16}$. The p-value is under 0.05, which identifies that the data is not a noise.

Then we check if the data is stationary by analysing its auto-correlation and partial auto-correlation. The data seems to be periodic yearly, thus we suppose its frequency is 360. According to the following figures, the first-order difference makes the acf and pacf both close to 0 with weak vibration. Although the second-order difference gives better result, we decide to set the auto-regressive degree to 1, to avoid losing too much information.

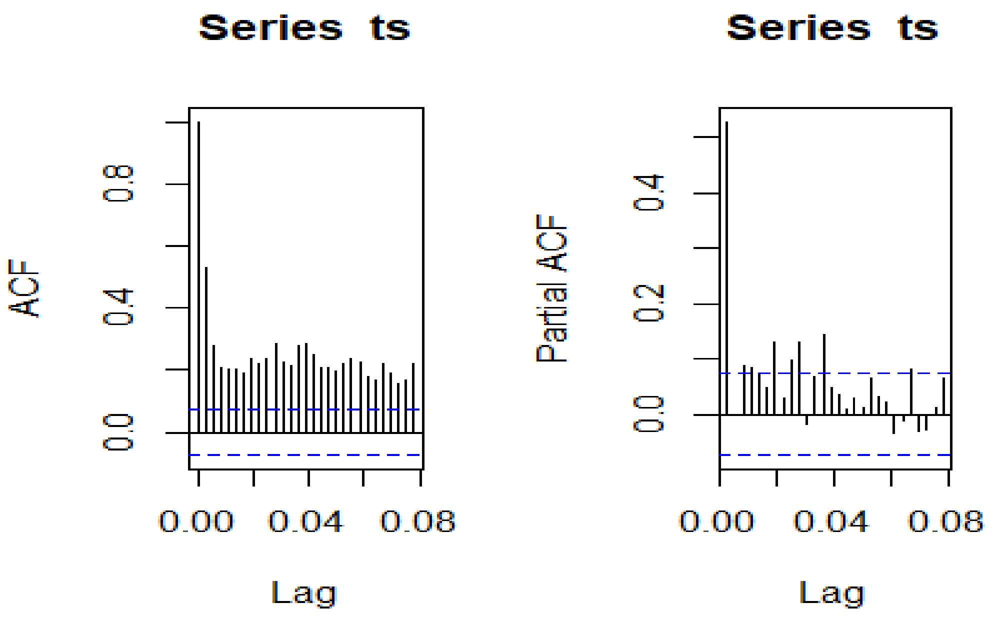


Figure 18: acf and pacf without differencing

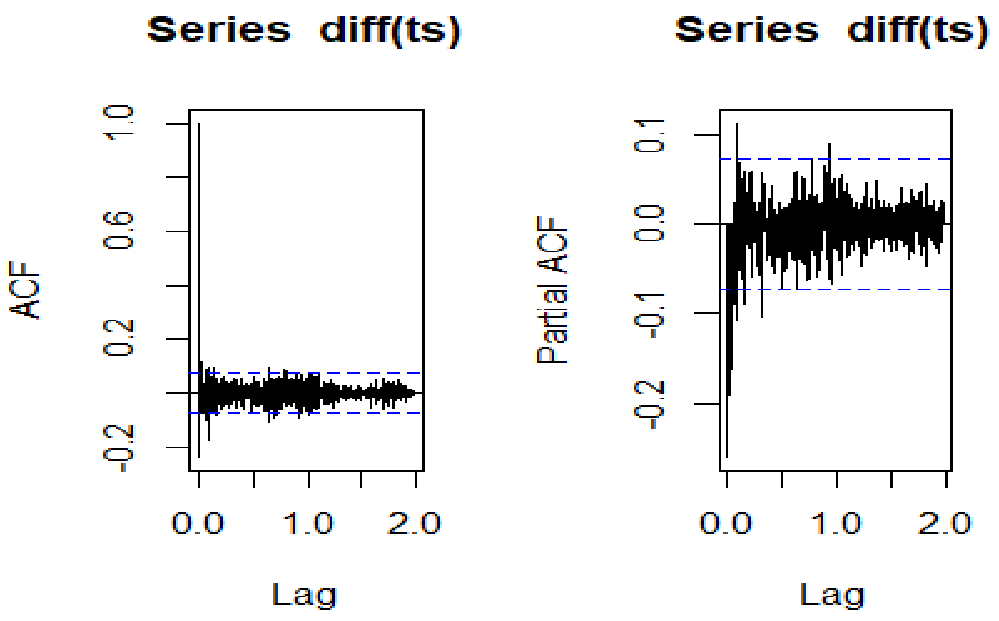


Figure 19: acf and pacf without differencing

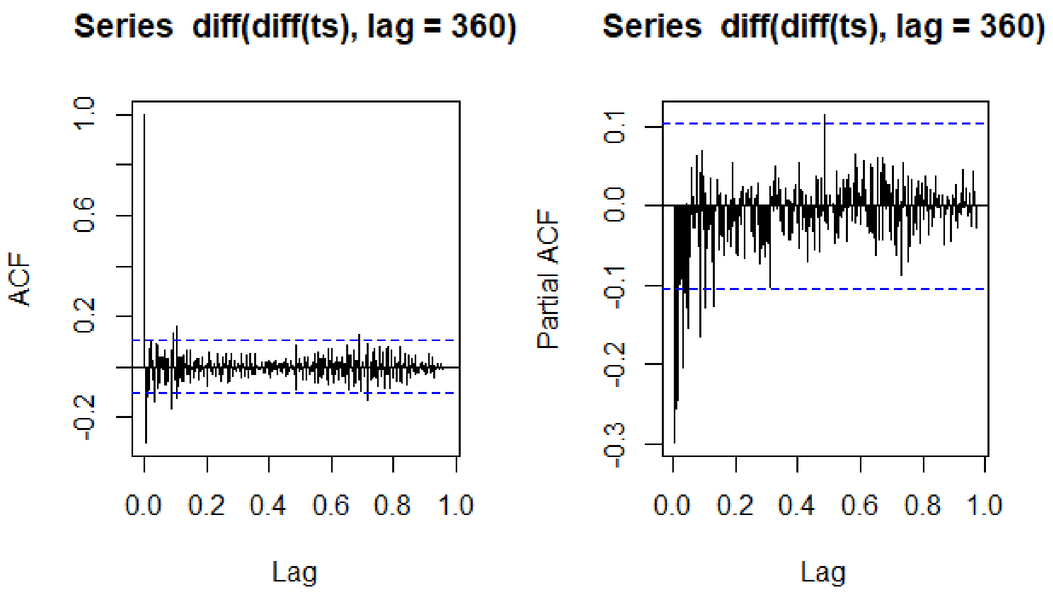


Figure 20: acf and pacf without differencing

We also standardize the NO_2 concentration as mentioned in section 3.1.1.

3.4.2 Arima

By using the auto.arima function, the AR and MA terms of the best ARIMA model are 7 and 2 respectively, but the degree of difference is recommended to be 0. We decided to follow the analysis in the section 3.5.1. So finally the ARIMA model's order parameters are set to (7,1,2).

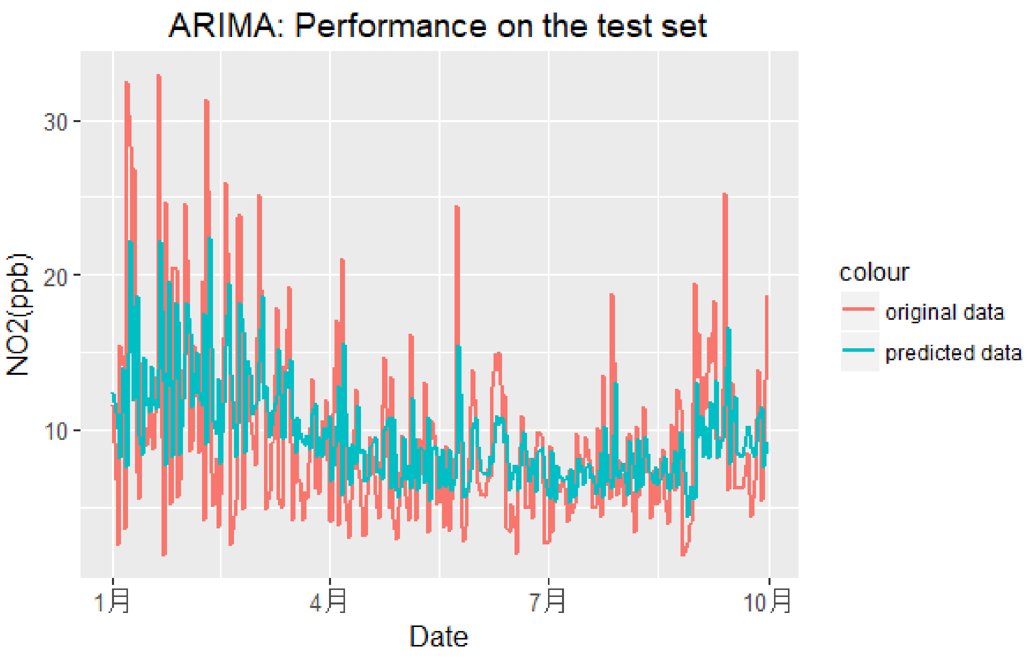


Figure 21: ARIMA: Performance on the test set

By using this ARIMA model, the MAPE and RMSE are respectively 56.5886 and 5.5798. From the curves we can see the average of predicted data is similar with that of original data. It means the model successfully learned seasonal pattern. However the model fails to match the real value of concentration, the predicted value tends to be conservative, as we see the blue curve varies much less than the red curve.

3.4.3 Combining Random forest and Arima

In this part we try to use ARIMA model to learn from the residuals of a random forest regressor. For simplicity of analysis, we use the same random forest in section 3.4.3. Its residuals are standardized before learning by used to train ARIMA.

We have a look at the acf and pacf of the residuals at first. The result shows that the residuals are stationary. We apply the auto.arima function in this case. The recommended model is (1,0,0). The final predicted concentration of NO_2 is the sum of that of random forest and that of ARIMA model. By this combination, the MAPE and RSME are reduced to 45.4778 and 4.6435 respectively. This is a big improvement, and we see the power of combining different models.

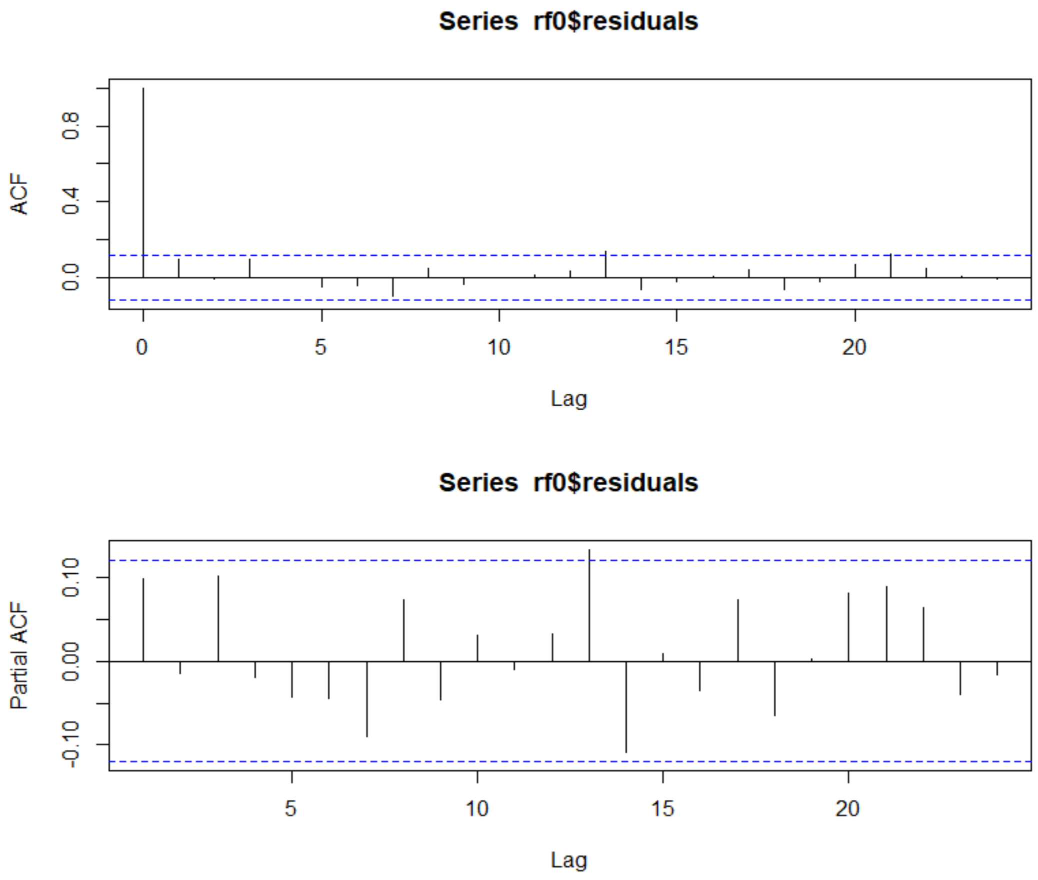


Figure 22: acf and pacf of the residuals

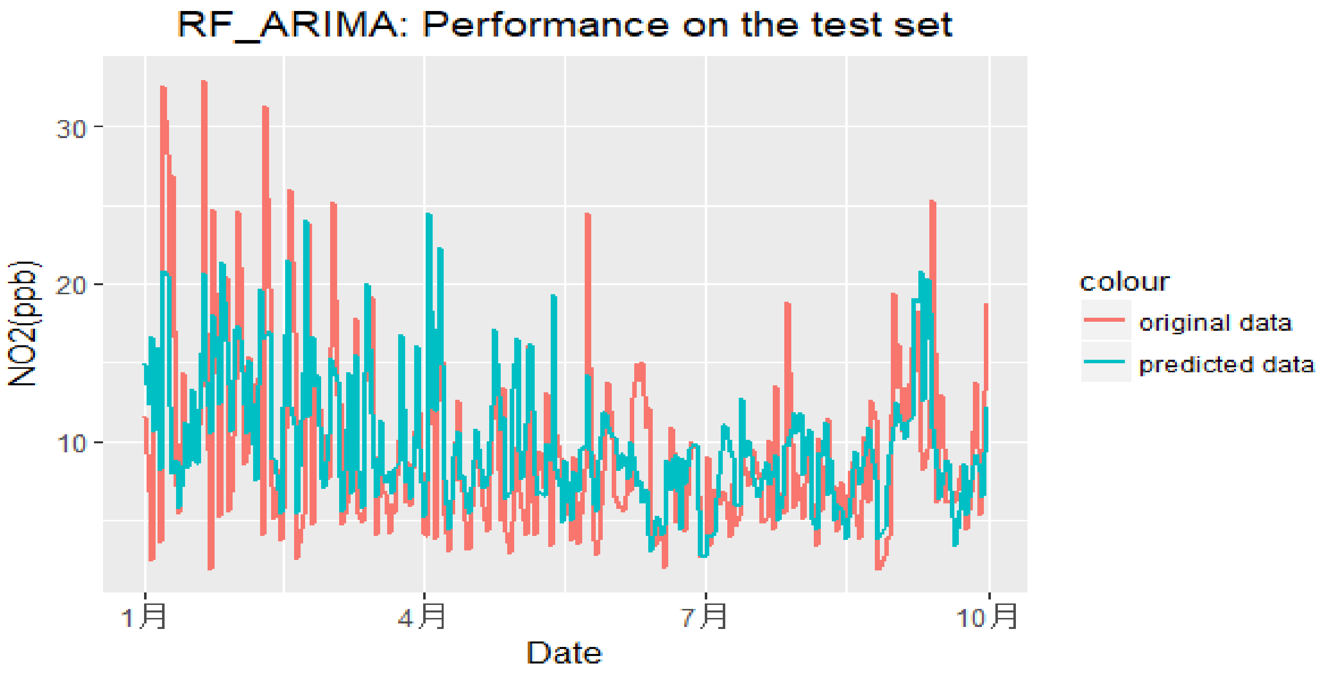


Figure 23: RF+ARIMA: Performance on the test set

3.4.4 Conclusion

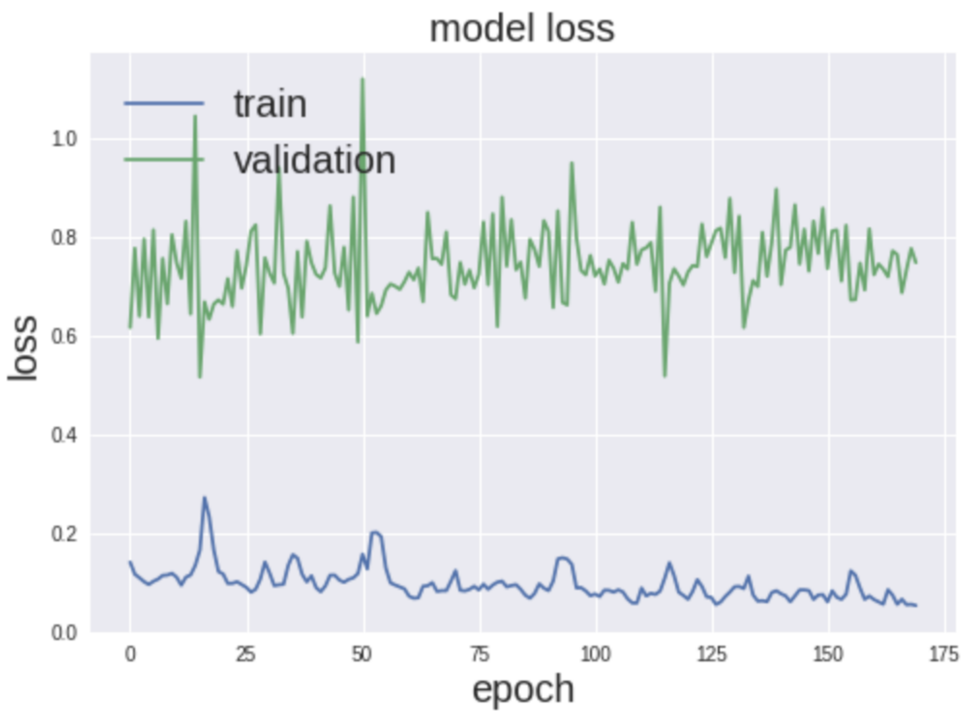
ARIMA model can better detect the seasonality patter of times series compared to tree models. Combining both models makes the performance better than a single ARIMa model. However, we find that letting ARIMA learn the residuals increases the bias, as the performance is lower than a single random forest. Since we applied the auto.arima to get the best parameters, the problem should not lie in the calibration of model, thus it could be the adaptability of ARIMA to the residuals. One possible explanation is that the residuals are too irregular, and the training set is not big enough for ARIMA. On the other hand, the combinated model is not worse than a single ARIMA. Therefore random forest is a better model in this context.

3.5 LSTM network

3.5.1 Data Processing

For LSTM Model, we just use the old data of NO2 to predict the future data of NO2. If we want to predict the data of NO2 today, the input of the model is the data of NO2 of the last 350 days, the output of the model is the data of NO2 today. We use two LSTM layers, the loss function is mean square error, the optimizer is RMSprop which is commonly used in LSTM network.

3.5.2 Results



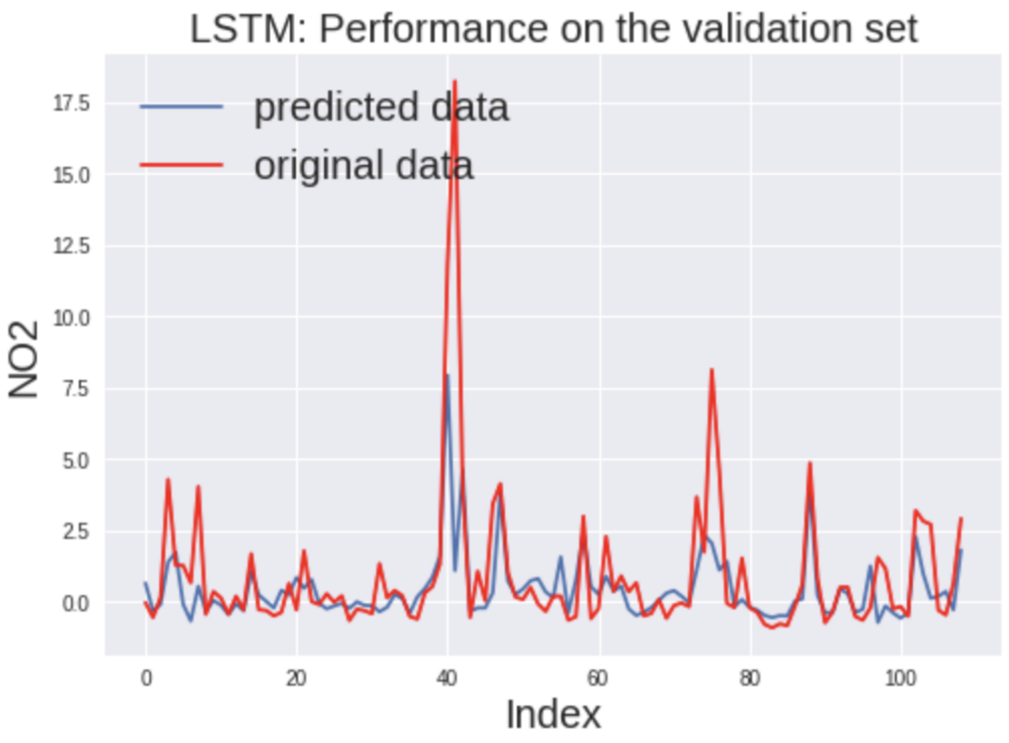


Figure 24: Performance on the validation set

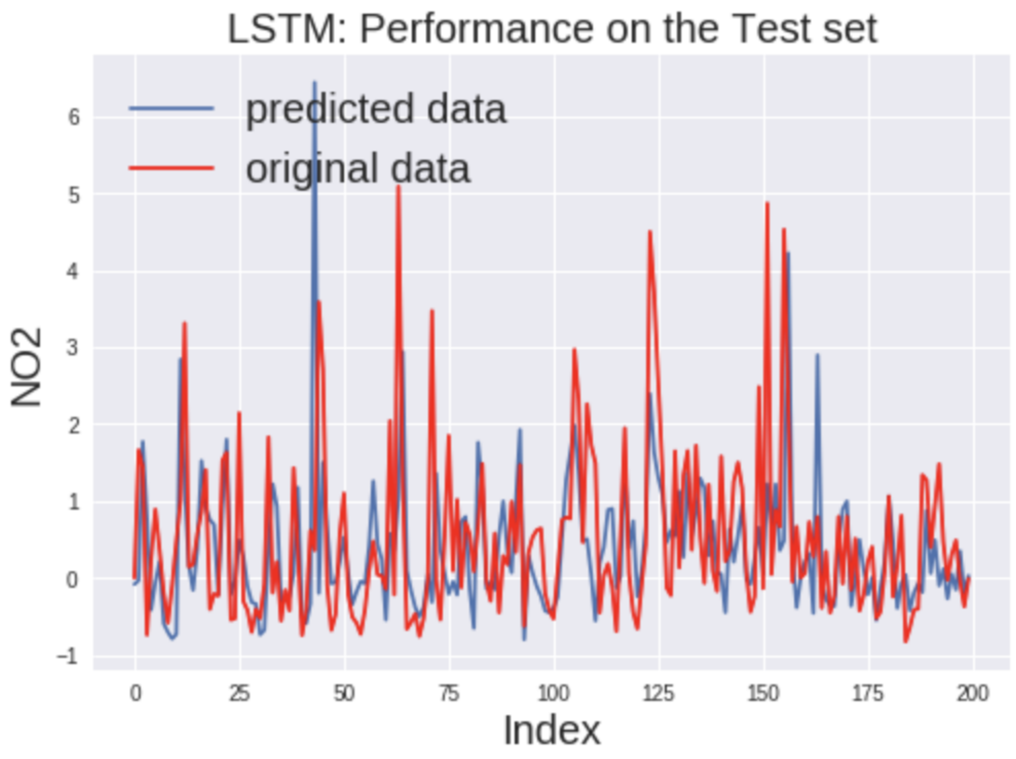


Figure 25: Performance on the test set

We can clearly see that LSTM achieves much better performance than the models above. The loss is very small compared to that of the other models and the predicted value is very similar to the original data.

4 Annex: Code

The code is uploaded to the github:
https://github.com/fern35/MasterProject