Final Report

EE4211 Data Science for Internet of Things

Class Project

Big Data Company

Group members:

Moritz Scheer | A0210531B

Zongyi Guo | A0206639Y

Yuzhe Wang | A0206675Y

Gerry Dunda | A0208770A

TABLE OF CONTENTS

2. FORECASTING 3

2.1 MOTIVATION FOR FORECASTING 3

2.2 FORECASTING USING LINEAR REGRESSION 3

2.3 FORECASTING USING SUPPORT VECTOR REGRESSION 5

3 PERFORMING LSTM AND XGBOOST MODELS IN PREDICTION 11

3.1 DATA ENGINEERING 11

3.2 MODEL DESCRIPTION 12

3.3 USING GAS CONSUMPTION TO PREDICT TEMPERATURE, HUMIDITY, VISIBILITY SEPARATELY 12

3.3.1 Temperature Prediction 12

3.3.2 Humidity Prediction 14

3.3.3 Visibility Prediction 15

3.4 USING PREVIOUS 5-DAY GAS CONSUMPTION TO PREDICT FUTURE GAS CONSUMPTION 16

3.5 USING COMBINED FEATURES TO PREDICT GAS CONSUMPTION IN FUTURE 17

3.6 SUMMARY 19

2 FORECASTING

2.1 MOTIVATION FOR FORECASTING

In this part, you will be asked to build a model to forecast the hourly readings in the future (next hour). Can you explain why you may want to forecast the gas consumption in the future? Who would find this information valuable? What can you do if you have a good forecasting model?

It might be a useful prediction to ensure the safety of the environment and cost estimation. Specifically, this prediction can be utilized to control the production management of the gas and to schedule the pipeline construction. Because the gas consumption is expected to increase as time progresses, the construction of the pipeline is necessary to meet strict demand. The gas production should not be excessive since it has direct impact on the environment in the form of the waste after its usage. This prediction can be used by the oil's company as well as the government. If the prediction has a good performance, the relevant parameter of the production, such as the date of the construction of a new pipeline, can be estimated accurately. Furthermore, the government might be able to amend the policy that regulates the price of the oil to keep the rate of the gas consumption at tolerable level.

2.2 FORECASTING USING LINEAR REGRESSION

Build a linear regression model to forecast the hourly readings in the future (next hour). Generate two plots: (i) Time series plot of the actual and predicted hourly meter readings and (ii) Scatter plot of actual vs predicted meter readings (along with the line showing how good the fit is).

The dataset splits into two parts: training set and test set. In this task, 80% of the data are considered as the training set and the rest are the test set. The test will be used to measure the performance of the linear regression in each meterid. The performance metric of the evaluator will be coefficient of determination

The following report only display three meter-ids for simplicity, 871, 4447 and 5484. Note that the blue curve of the time series plot represents the actual measurement and the green curve represents the prediction based on linear regression model. The dashed line in the scatter plot is the line showing how good is the model. The R^2 values for each meterid are also shown.

```
r2_score for id : 871 is -10.893489134558656
r2_score for id : 4447 is 0.8263534169660388
r2_score for id : 5484 is 0.7726423954537807
```

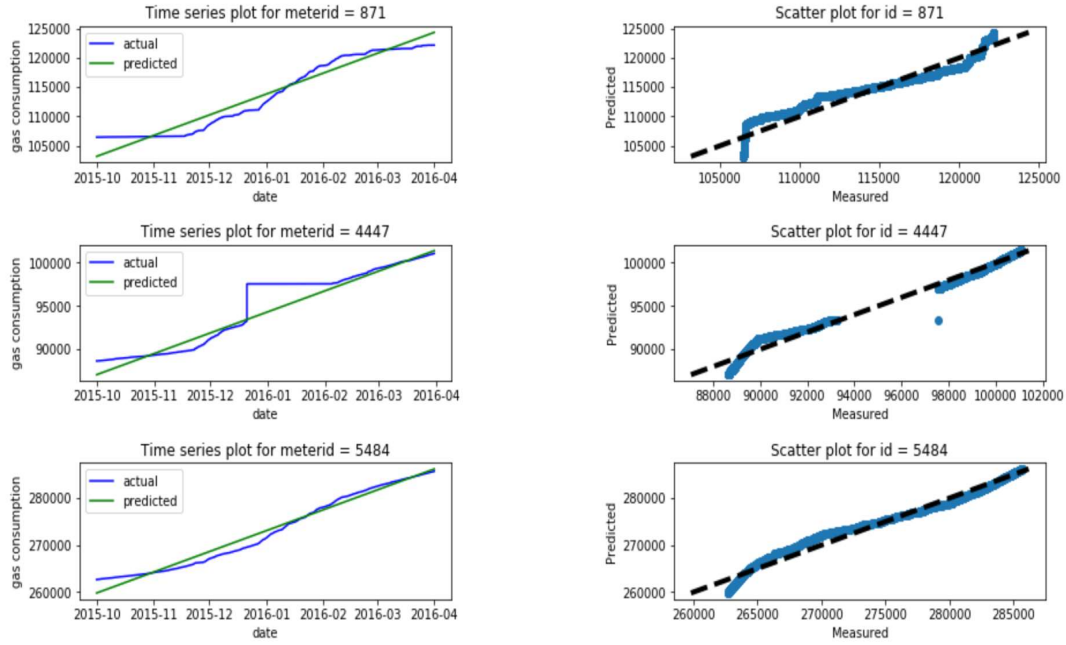

Fig. 2.1

Based on the graphs and the coefficient of determination above, there are some meter ids in which the linear regression model performs well based on the coefficient of determination. The linear regression model sometimes does not perform very well on several meter id's test sets which yield strong negative value of R^2.

To capture the performance on the whole test set of all meter ids, the average value of the coefficient of determination is calculated and its value is found -7.22.

This implies that many of meter ids in which the linear regression performs very poorly.

It is also important to consider the top 5 of the meter ids in which the model performs very well based on the coefficient of determination which are shown below.

```
1    meterid = 4447 with r^2 = 0.8263534169660388
2    meterid = 5484 with r^2 = 0.7726423954537807
3    meterid = 5810 with r^2 = 0.4958831374626059
4    meterid = 484 with r^2 = 0.4398377688988587
5    meterid = 7674 with r^2 = 0.23057363955509014
```

2.3 Forecasting using Support Vector Regression

Do the same as Question 2.2 above but use support vector regression (SVR).

Support Vector Regression (SVR) is a regression model based on the ideas of the Support Vector Machines (SVM). Hence, when initializing the model, the kernel used by the model in order to mapping data in higher dimensions and performing the kernel trick must be chosen. Furthermore, the model requires various different other hyperparameters depending on the selected kernel. Important hyperparameters are for example the penalty parameter C or gamma, who visually speaking determines the influence width of single samples. However, the latter is irrelevant for linear kernels.

Since linear kernels are the simplest ones, we started using an SVR model with a linear kernel in order to predict future readings for a specific household. We hereby used the first 70% of the samples of the specific household to train the model and the remaining 30% to test it. To ensure that the data does not get split randomly into a training and test set but in a way that the samples' time is taken into account, we wrote the function splitting().

However, using a linear kernel firstly did not work due to an extremely long runtime for the fitting process. Even after waiting for over 30 minutes, the program still did not return. We tried using the LinearSVR() model instead of the SVR() model with selecting a linear kernel. This resulted in a fast runtime and showed results for the first time. The regression result turned out to be very weird though and the program gave a convergence warning. After some, we research, we tried using feature scaling before the fitting process and this was the key to a lot of problems. Not only did the convergence warning disappear and the results of using LinearSVR() started to look good, but also other models which had an endless runtime before gave quick results after feature scaling. Apparently, not scaling the feature before the fitting process had the consequence that so many iteration for fitting the model were needed, that either the program did not return as it happened when using SVR() or the results were bad because the program stopped due to a iteration limit and gave a convergence warning as it happened using when LinearSVR().

After having fixed this problem and selected suitable hyperparameters, this are the prediction results for three randomly selected households:

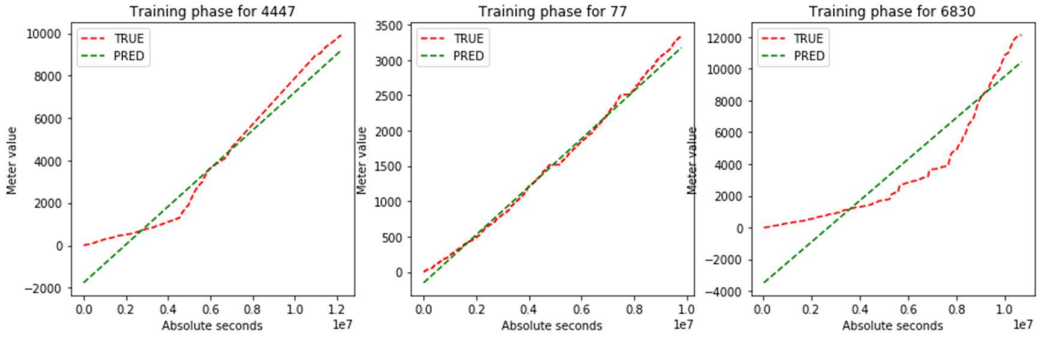

FIG. 2.2 TRAINING RESULT USING LINEAR KERNEL

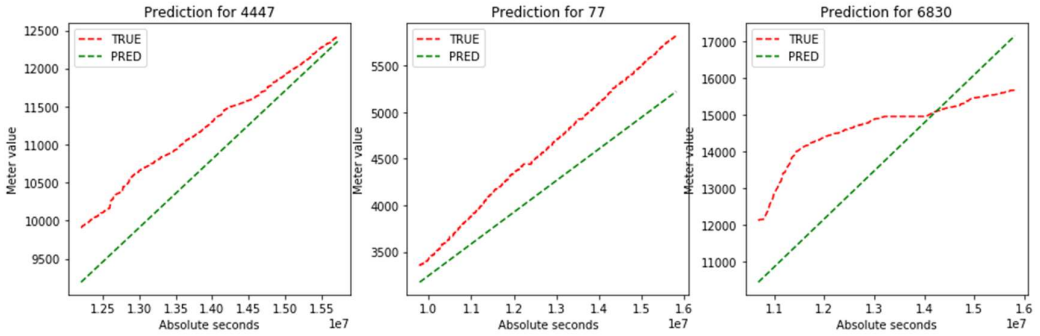

FIG. 2.3 PREDICTION RESULTS USING LINEAR KERNEL

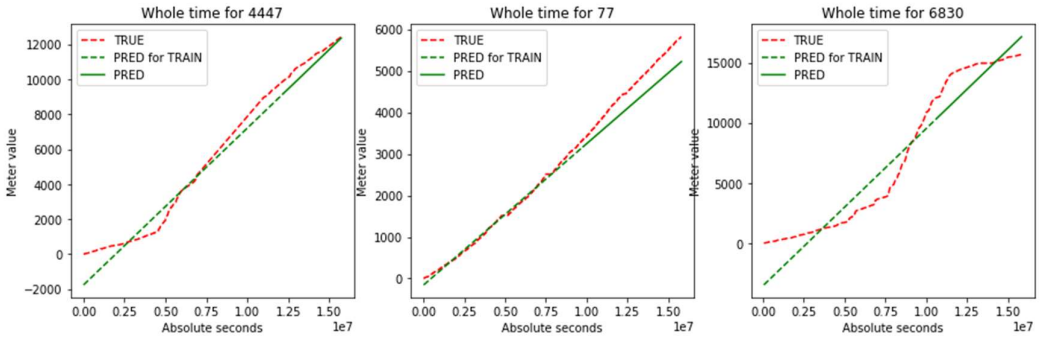

FIG. 2.4 TRAING AND PREDICTION RESULTS COMBINED USING LINEAR KERNEL

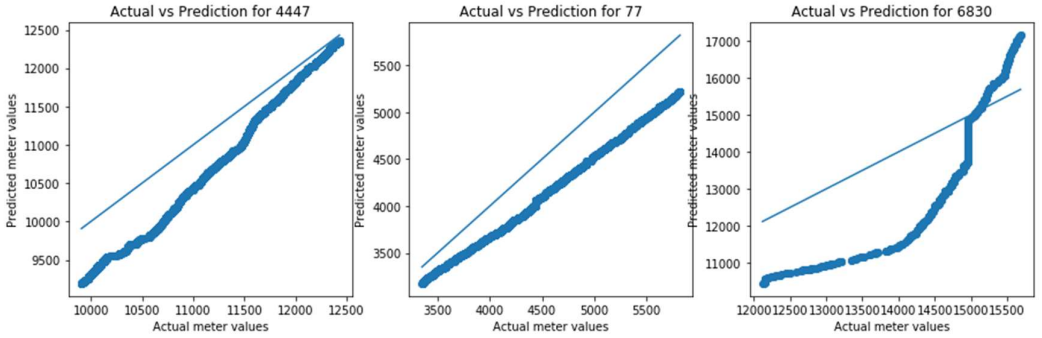

FIG. 2.5 SCATTER PLOT USING LINEAR KERNEL

```
The R^2 score of the forecasting for ID 4447 is: 0.5010734650523494
The R^2 score of the forecasting for ID 77 is: 0.6181791116768826
The R^2 score of the forecasting for ID 6830 is: -2.238023905207506
```

Next, we tried forecasting using an SVR model with a radial basis function (RBF) kernel. Interestingly, when setting the hyperparameter gamma to ‘scale’, scaling the data was not only not necessary, meaning the program neither ran endlessly nor gave a convergence warning, it did not even have any effect. The results for SVR with an RBF kernel and gamma = scale are shown below:

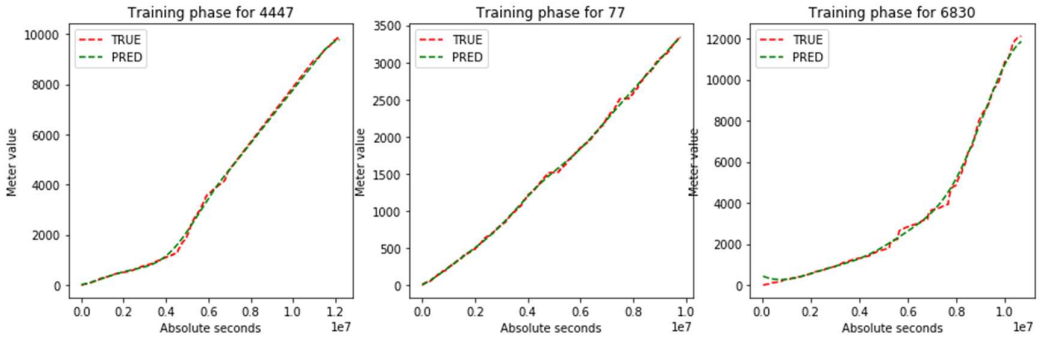

FIG. 2.6 TRAINING RESULTS USING RBF KERNEL AND GAMMA=SCALE

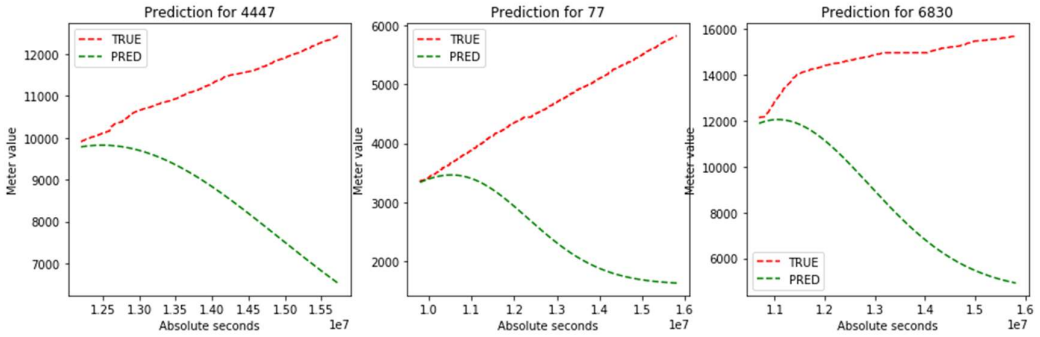

FIG. 2.7 PREDICTION RESULTS USING RBF KERNEL AND GAMMA=SCALE

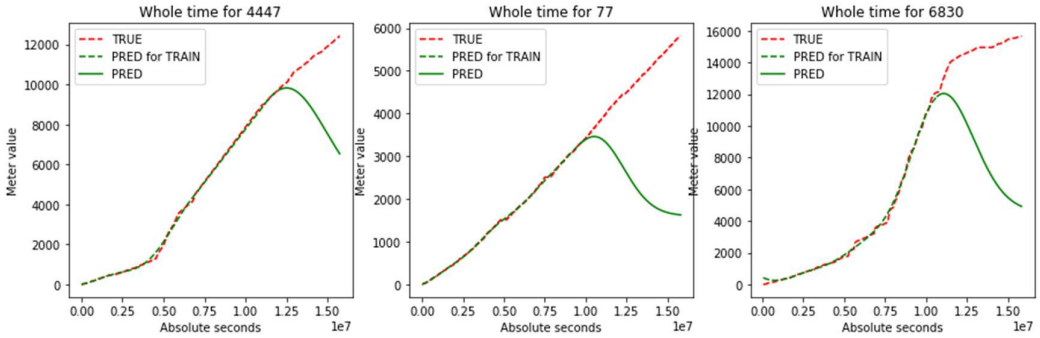

FIG. 2.8 TRAINING AND PREDICTION RESULTS COMBINED USING RBF KERNEL AND GAMMA=SCALE

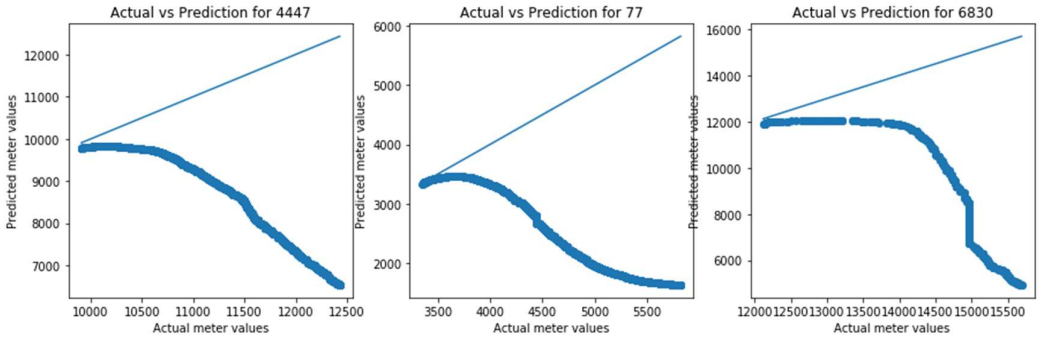

FIG. 2.9 SCATTER PLOT USING RBF KERNEL AND GAMMA=SCALE

```
The R^2 score of the forecasting for ID 4447 is: -18.423576456960316
The R^2 score of the forecasting for ID 77 is: -10.56767034677089
The R^2 score of the forecasting for ID 6830 is: -68.95299077967071
```

It can be seen, that the overall R^2-score is worse than using a linear kernel. However, the model fits way better to the training data and thereby is also more accurate shortly after the prediction ended. While a linear kernel is the preferred choice for long-term prediction, for short-term prediction, the RBF kernel with gamma=scale is superior.

The third thing we tried was using a RBF kernel again, but this time with a fixed value for gamma. Here, feature scaling in order to get useful results was required again.

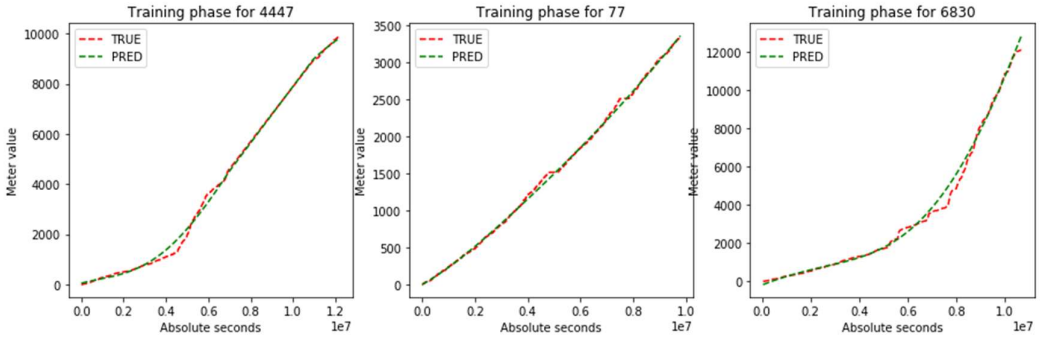

FIG. 2.10 TRAINING RESULTS USING RBF KERNEL AND FIXED GAMMA

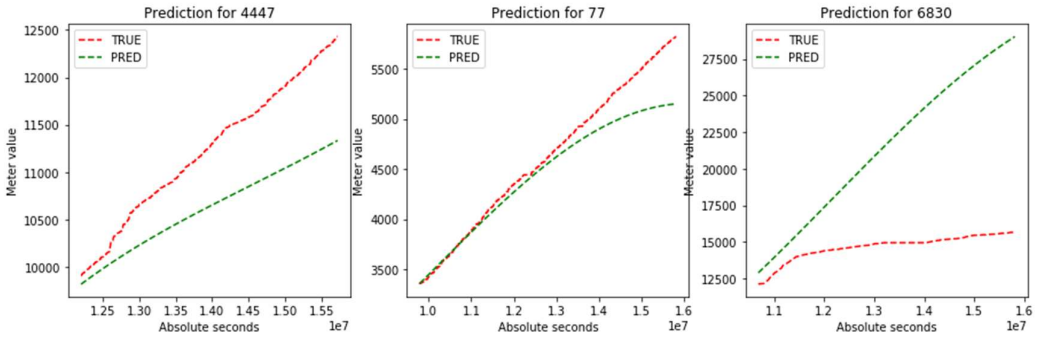

FIG. 2.11 PREDICTION RESULTS USING RBF KERNEL AND FIXED GAMMA

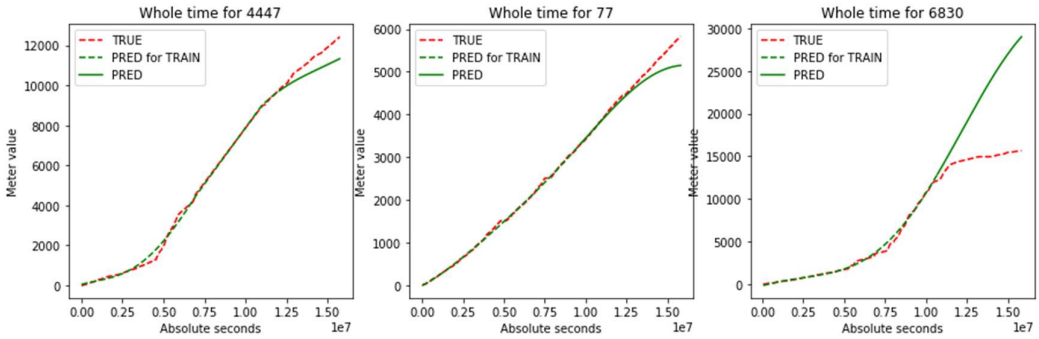

FIG. 2.12 TRAINING AND PREDICTION RESULTS COMBINED USING RBF KERNEL AND FIXED GAMMA

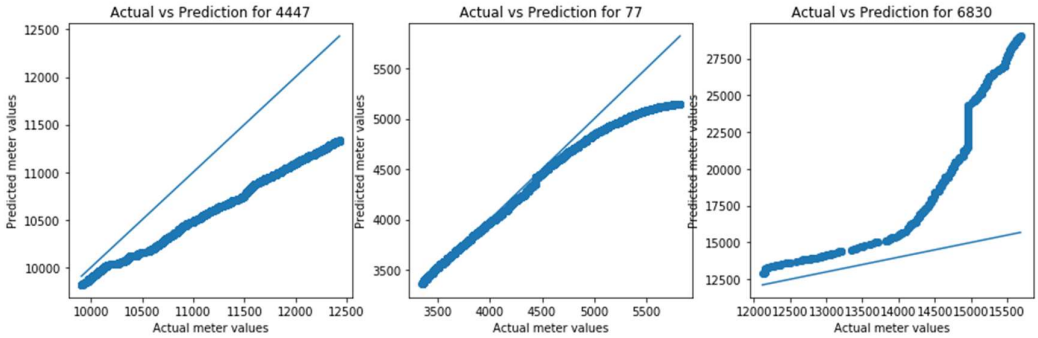

FIG. 2.13 SCATTER PLOT USING RBF KERNEL AND FIXED GAMMA

```
The R^2 score of the forecasting for ID 4447 is: 0.11239614715086786
The R^2 score of the forecasting for ID 77 is: 0.8994167357235909
The R^2 score of the forecasting for ID 6830 is: -87.79010206800524
```

Looking at the R^2-scores, choosing a fixed value for gamma instead of g amma=scale gives better results. For household 77, it with almost 0.9 eve n better than the result from the liner kernel. This model provides a very good combination of both a proper short-term and long-term prediction.

3 PERFORMING LSTM AND XGBOOST MODELS IN PREDICTION

At this point, you understand the data quite well. Propose and carry out additional analysis using the dataset given. Please be sure to justify why this additional analysis is useful and interesting.

3.1 DATA ENGINEERING

Considering the given dataset, natural gas consumption in Austin, we have to find the corresponding dataset, which can provide hourly or daily readings. Finally, we chose weather and temperature data in Austin, given the particular time period. The dataset was created by Kaggle (https://www.kaggle.com/grubenm/austin-weather), which shows not only daily temperature readings, but also other interesting features, including daily humility readings, daily visibility readings and etc. from 2013-02-21 to 2017-07-31.

Having obtained this new dataset, which are all daily readings, we have to generate daily gas consumptions of the whole Austin society. In process of generating daily gas consumptions, we found some readings of some meters are abnormally distributed, which we hadn't found in previous stage. Therefore, to simplify the question, we just ignored these 10 meters. Besides, we found that most of meter readings started from Oct.1st in 2015, and ended between Mar.29th and Mar.31st, about 180 to 182 days. Therefore, in order to keep the most of the observations, we reserved all meters, having more than 180 readings and ignored meters, which had less observations. Finally, 94 meters are remained for further analyzing and prediction. We firstly used linear interpolation to get the daily gas consumptions of each meter ID. Then, we added them up to get the gas consumption readings regarding to the whole society. The daily gas consumption in the area ranges from 2500 to 36000. From the weather dataset, we extracted 3 features out of the raw dataset, which are average daily temperature, average daily humidity, average daily visibility. Hence, we could do something interesting, trying to use gas consumption to predict these three features separately, and combine these three features to predict future gas consumption.

Having recreated the dataset, we separated it into training and test set. First 150 days readings used as the training set, and last 30 days used as test set.

3.2 Model Description

As the result of the time-series problem, the very appropriate model for dealing with this is Long Short-Term Memory (LSTM). LSTM is an improved recurrent neural network model to learn from experience to classify, process and predict time series when there are very long-time lags of unknown size between important events. Three stages are included in the architecture, forgotten, selective memory and output stage. In LSTM model, we firstly manually adjust the hyperparameters, because the whole dataset only has about 180 days, which makes it difficult for us to do cross-validation to select hyperparameters. Then, we tried to use the function from Torch to adjust Learning Rate automatically. Finally, we picked the hyperparameters with the best performance.

Besides, we also came up with another very strong and competitive model, XGBoost to handle this problem, which can make up some drawbacks of LSTM model. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. In XGBoost, we use internal libraries to adjust hyperparameters, including cross-validation and grid method. Finally, we also pick the hyperparameters with the best performance.

Then, we analyze the performance of these two models and reach our final result.

3.3 Using Gas Consumption to Predict Temperature, Humidity, Visibility Separately

3.3.1 Temperature Prediction

Note that we use first 150 days as the training set, last 30 days as the test set. The predicting time interval is one day, which means we use the gas consumption from last day to predict temperature today. The results of prediction using LSTM and XGBoost are shown below.

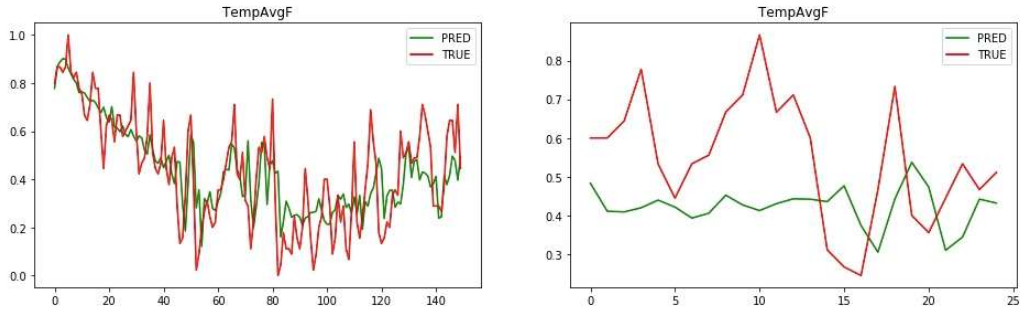

Fig.3.1 Predict Temperature (Fahrenheit) using LSTM

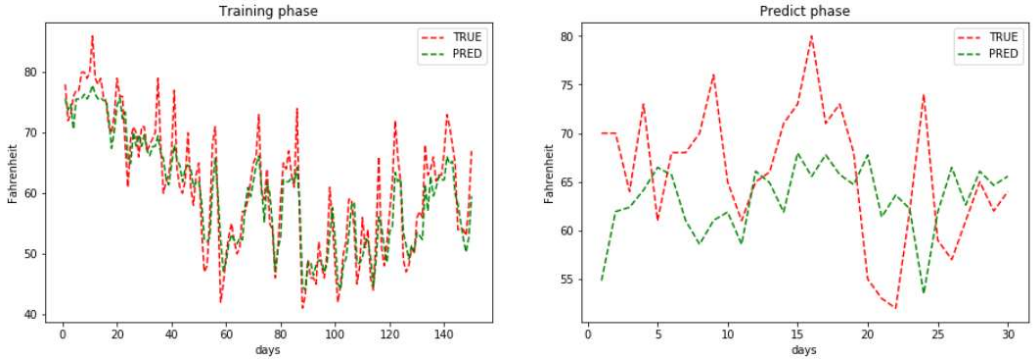

Fig.3.2 Predict Temperature (Fahrenheit) using XGBoost

MSE and MAE of performance of two models is compared in format below:

	MAE		MSE	
	Training	Test	Training	Test
LSTM	4.09	9.78	29.31	136.50
XGBoost	3.16	6.59	15.31	70.79

Comparing the performance of these two methods, XGBoost slightly beats LSTM model. When change Fahrenheit into Celsius, the error is less than 3.7 degree. The performance is quite good.

3.3.2 HUMIDITY PREDICTION

Note that we also use first 150 days as the training set, last 30 days as the test set. The predicting time interval is one day, which means we use the gas consumption from last day to predict humidity one day after. The results of prediction using LSTM and XGBoost are shown below.

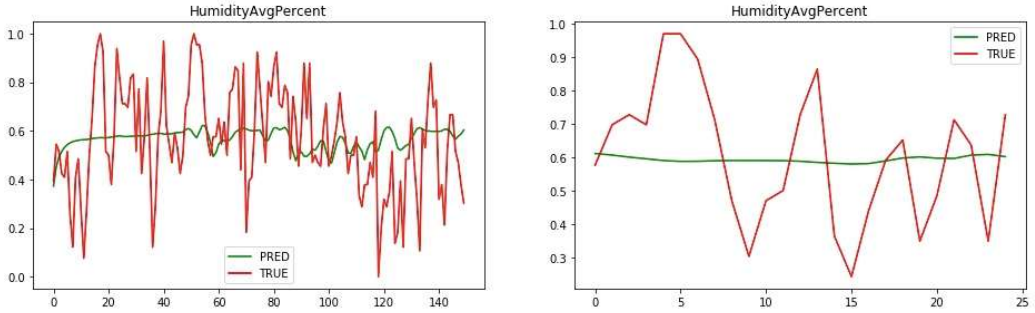

Fig.3.3 Predict Humidity (percentage) using LSTM

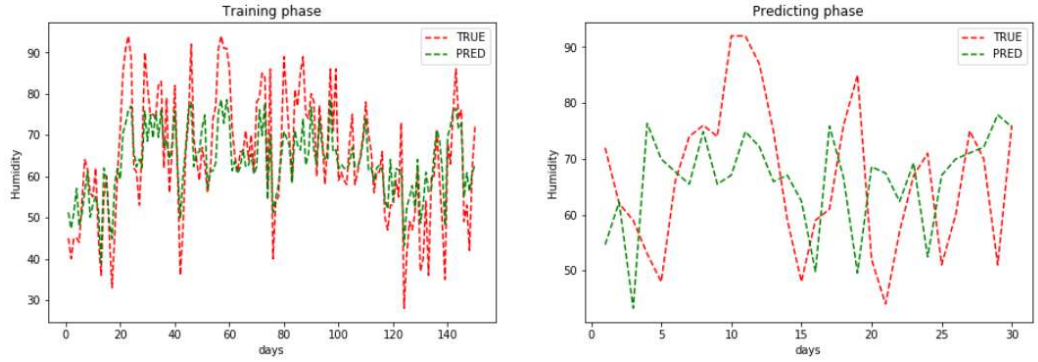

Fig.3.4 Predict Humidity (percentage) using XGBoost

MSE and MAE of performance of two models is compared in format below:

	MAE		MSE	
	Training	Test	Training	Test
LSTM	6.09	15.95	136.51	435.96
XGBoost	7.03	12.71	76.89	238.95

Comparing performance of both models, XGBoost behaves better than LSTM. LSTM is suspected overfitted to training data, because it has higher performance in training data, but worse in test data. In this case, we may choose XGBoost to predict.

3.3.3 Visibility Prediction

Note that we also use first 150 days as the training set, last 30 days as the test set. The timeline we extracted is from 2015-10-02 to 2016-03-29. The predicting time interval is one day, which means we use the gas consumption from last day to predict visibility one day after. The results of prediction using LSTM and XGBoost are shown below.

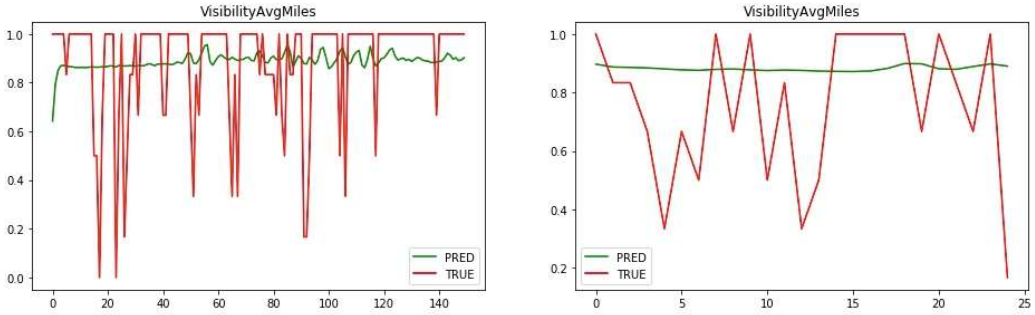

Fig.3.5 Predict Visibility using LSTM

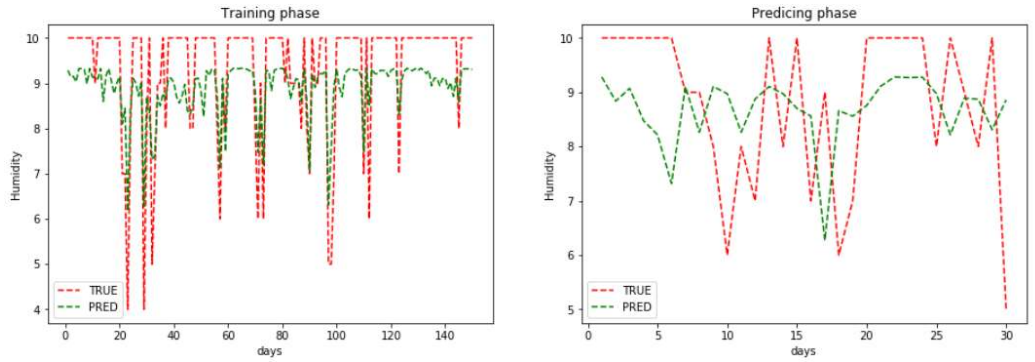

Fig.3.6 Predict Visibility using XGBoost

MSE and MAE of performance of two models is compared in format below:

	MAE		MSE	
	Training	Test	Training	Test
LSTM	0.95	1.28	1.75	2.72
XGBoost	0.85	1.37	0.91	2.63

Comparing performance of two models, they both perform very good, and LSTM is little bit better than XGBoost. Visibility from the raw dataset ranges from 4 to 10. Therefore, the prediction error, given by LSTM, is about 1.3, which can predict quite accurate.

3.4 Using Previous 5-day Gas Consumption to Predict Future Gas Consumption

Note that this time we use previous 5-day gas consumption to predict 1-day gas consumption in the near future. Besides, we also separate the dataset into 150 and 30, training and test set. As the result of using 5-day data to predict one day, the size of training set is little bit smaller, as well as test set. The results of prediction using LSTM and XGBoost are shown below.

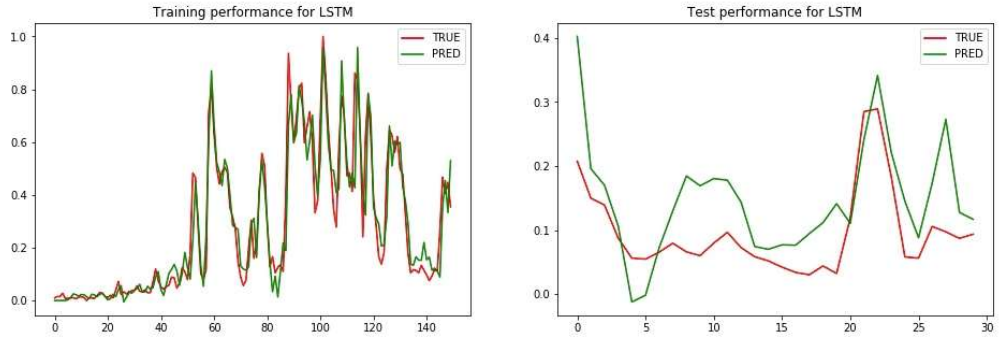

Fig.3.7 Predict Gas Consumption using LSTM

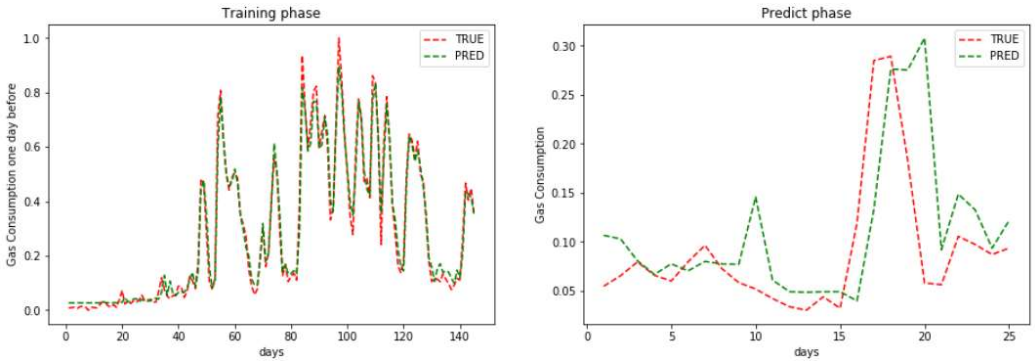

Fig.3.8 Predict Gas Consumption using XGBoost

MSE and MAE of performance of two models is compared in format below:

	MAE		MSE	
	Training	Test	Training	Test
LSTM	2453	2163	12238943	6570469
XGBoost	934	1445	1623920	5601645

From the result, we can easily reach the conclusion that in this case, XGBoost is much better than LSTM prediction. Daily gas consumption, from the raw dataset, ranges from 2400 to 36000. The error of XGBoost can reach about 1400 foot in test set, which is acceptable.

3.5 Using Combined Features to Predict Gas Consumption in Future

Now, we recreate training and test set. We use three features, including temperature, humidity, visibility to predict gas consumption in near future. The prediction time interval is one day. From our several times experiment, performance of LSTM is totally worse than XGBoost. Therefore, here we only show the performance of XGBoost. The performance is shown below.

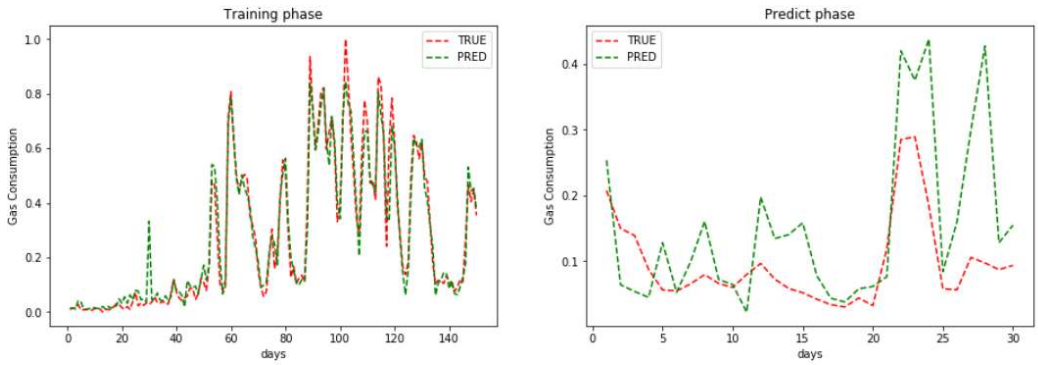

Fig.3.9 Using Combined Features to Predict Gas Consumption using XGBoost

MSE and MAE of performance of the model is compared in format below:

	MAE		MSE	
	Training	Test	Training	Test
XGBoost	1240	2539	3741734	12555687

Comparing with last experiment, which is using gas consumption to predict gas consumption in 3.4, performance of this experiment is worse than last time. Test phase MAE of last time is 1445, which is much better than 2539 now. Thus, we reach the conclusion that using gas consumption to predict gas consumption is much better than using weather and temperature features. We also believe that LSTM model is more specialized in predicting features, which are highly corelated to training features, because when predicting other features, the performance decreases dramatically.

3.6 Summary

From all experiment we have done, XGBoost model beats LSTM model for most of the time. Besides, LSTM model may have more limitations, compared with XGBoost. LSTM model behaves much better when use highly corelated features to train and predict, e.g. using gas consumption to predict gas consumption. However, XGBoost does not show this kind of limitations. Moreover, as the result of initial characteristics of XGBoost, it has several ways to prevent overfitting, like L1 and L2 regularization, limiting the depth of the trees, method of dropout and etc. Thus, XGBoost has lower possibility to overfit the training data. We may take it as another reason for why it performs better. Last but not least, we can also use the internal functions, provided by XGBoost, to automatically choose the best hyperparameters, which guarantees the best performance.