Forecasting Models for Load Time Series Prediction

Tianyu.Cui 15033963 tianyu.cui.15@ucl.ac.uk

Rui.Xu 15027712 ucabrxu@ucl.ac.uk

Yuan.Meng 14026927 yuan.meng.14@ucl.ac.uk

Abstract

In this project, we do time series prediction based on clustering, gradient boosting tree and deep learning (MLP, RNN and LSTM). The prediction performance is presented in each time series prediction method.

1 Introduction

In this project, we do time series prediction based on load time series data[5]. First, we do feature engineering to explore more useful features, and then we use clustering method to decide which time series should be predicted together in order to use the correlation information between time series. We also notice the time series have periodic behavior, so we use Fourier transformation to get useful features in frequency domain.

When we do forecasting, we use two types of learning methods. The first one is a classical boosting method called gradient boosting machine, and the second one is deep learning (MLP, RNN and LSTM). Both models have achieved reasonable results. From the result, we noticed even deep learning is extremely successful in some areas (such as computer vision and NLP), the result is still worse than gradient boosting regression tree, which is quiet a surprise. We think the reasons could be it has too many local minimums and it's very hard to tune hyper-parameters, so we didn't train the model perfectly.

2 Time Series Clustering

In this section, we want to use time series clustering method to find different time series patterns, and then we can improve the performance of prediction model by including the load information from same cluster. The reason is the time series in the same cluster can be similar and related, so if we can borrow some information from related time series, the performance can be better.

When we do clustering, we find if we choose different time series features, the result can be quiet different. It means the clusters are not stable, and the reason is the load time series have nearly the same pattern, so by using different features the result can be quiet different. But we also get some useful information by doing clustering, which is the patterns of zone 9 and zone 10 are quiet different from the other zones.

By using ACF(Auto Correlation Function), we calculate the top 40 autocorrelation coefficients for each time series, and then use these features to do clustering. We have tried spectral clustering, ward hierarchical clustering and Birch clustering method, and the number of clusters we set is 3. The result is all the zones excluding zone 9 and zone 10 were in the same cluster, and zone 9, zone 10 were in different clusters.

We use Isomap to transform the time series data from the ACF space into a two dimensional space, and do visualization. We can find that zone 9 and zone 10 are quiet different, and zone 4 is also a little bit different, but it still stay in the same cluster as the other points.

So when we predict the load of most zones(exclude zone 9 and zone 10), we use all the load data of these zones. And when we predict the load of zone 9 and zone 10, we just use their own load data.

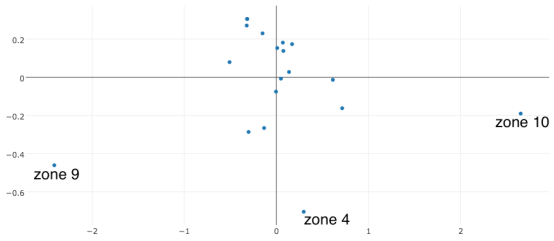


Figure 1: Time series point by Isomap

3 Feature Engineering

Besides choosing a appropriate model, representation of data also plays an important role. We trained our model with additional hand-crafted features. The temperature measurements at 11 stations were considered as features. After plotted the loads demand against the time, we expected month, day and hour to decrease the loss function. Additionally, we believe that, since the day of week and holiday effects are related to human behavior pattern, they can influence the loads demand. Therefore, we take them into consideration.

4 Calculate Time Series Periodogram

Any time series can be expressed as a combination of cosine and sine functions with different frequencies and amplitudes. We can use this fact to extract features about periodic behavior of time series. Exactly speaking, we can use periodogram to identify the cyclical behavior of time series, and the periodogram is the norm of Fourier coefficients, which is:

$$x = \sum_{j=1}^{\frac{n}{2}} [\beta_1(\frac{j}{n})\cos(2\pi\omega_j t) + \beta_2(\frac{j}{n})\sin(2\pi\omega_j t)] \tag{1}$$

$$P(\frac{n}{2}) = \beta_1(\frac{j}{n})^2 + \beta_2(\frac{j}{n})^2 \tag{2}$$

where β_1 and β_2 can be estimated by using Fourier Transform.

We first need to fix a time window size to split the time series, and in this case, we choose the window size to be 24, which means for each day, we have a set of periodogram features. When we predict ith zone, we just add the ith zone's periodogram features, and use LSTM to do prediction.

5 Gradient Boosting Regression Tree

Boosting is an ensemble method. It was designed for classification problems, and had been extended to deal with regression problems. Its basic ideal is to combine a number if weak learners into a powerful model. Boosting continuously iterates and each iteration generates a new tree which is a weak classifier. Many method were proposed for boosting to produce a new tree. Friedman presented gradient boosting machine for producing a new tree by using the ideal of gradient descent [2]. It is based on the previous trees and make a step in the direction which mostly reduces the loss function.

After generating a tree, linear models were trained on all leaves of the tree. For each leaf, a linear model is used to describe the pattern on that leaf. The algorithm produces a large number of trees or weak learners. Given a unknown

data point, all weak learners will make their prediction. Then final prediction is given by averaging all the predictions made by weak learners.

5.1 Redefining the Problem

The competition asks us to predict loads from 1st hour of 01/07/2008 to 24th hour of 07/07/2008. However the competation is closed, we can no long check our results. We decided to revise the problem. We predicted the loads from the 1st hour of 23/06/2008 to 6th hour of 30/06/2008 and evaluated our prediction using actual loads demands. Root mean squared error was served as evaluation metric.

5.2 Training Model

We implement gradient boosting machine by xgboost which is a python package written by Tianqi. Required by the competition, we used root mean squared error (RMSE) as loss function. To avoid under-fit or under fit, parameters need be carefully set.

We used all the data before 23/06/2008 as training data. However the 20 zone might be different from each other, we trained a model for each zone. Then we test our model using data from the 1st hour of 23/06/2008 to 6th hour of 30/06/2008.

5.3 Result and Discussion

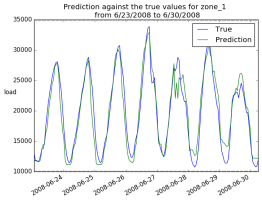
After training and testing for our model, we got an expected result.

Fig 2 3 shows predictions against actual loads demand for zone. Except zone 9 and zone 10, it seems that our model can capture the most of periodic pattern of loads. However some unexpected fluctuations which is hard to forecast and our gradient boosting model failed to capture them.

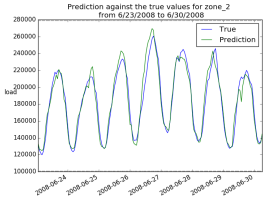
For zone 9 and zone 10, our model failed to predict. Lloyd reported in his report that there is an atypical dynamics in load 9 and a large discontinuity in zone 10 [4]. Fig 4 supports his report about the anomaly [4].

Figure 5 shows the RMSE of predictions for all zones. The loads for each zone were predicted by the best model for that zone. The actual loads for some zones are large than other zones, so are the RMSEs.

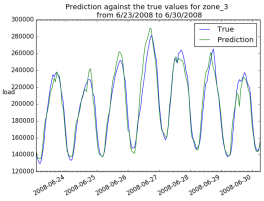
Gradient boosting machine gives an satisfied result.



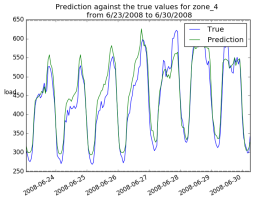
(a) zone 1



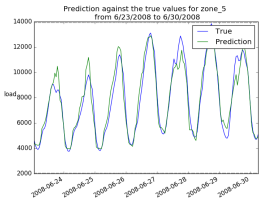
(b) zone 2



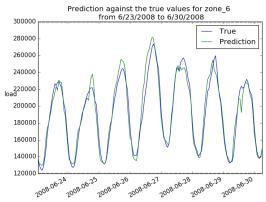
(c) zone 3



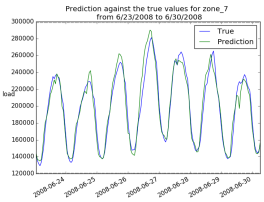
(d) zone 4



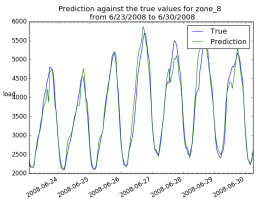
(e) zone 5



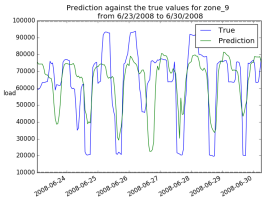
(f) zone 6



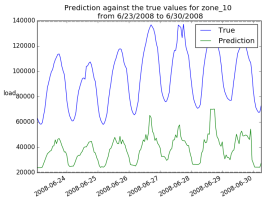
(g) zone 7



(h) zone 8

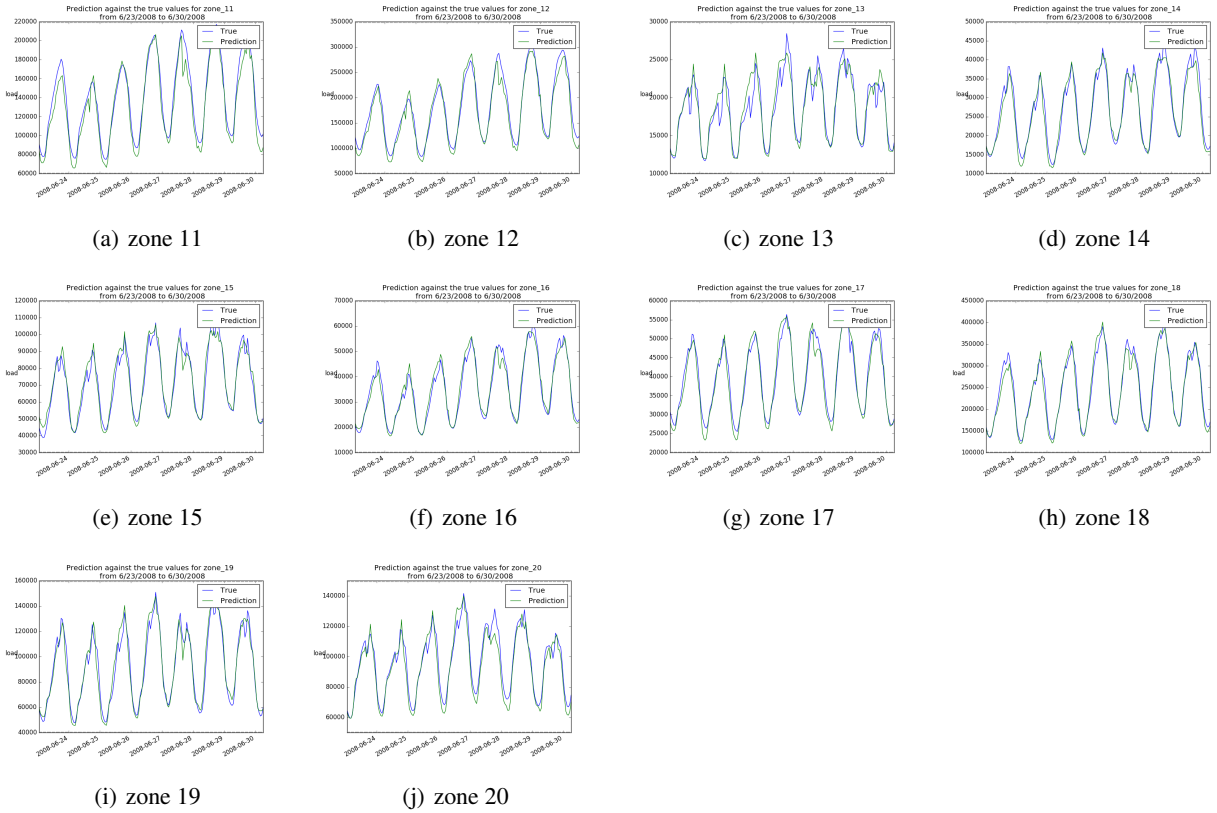


(i) zone 9



(j) zone 10

Figure 2: prediction over 20 zones



(a) zone 11 (b) zone 12 (c) zone 13 (d) zone 14

(e) zone 15 (f) zone 16 (g) zone 17 (h) zone 18

(i) zone 19 (j) zone 20

Figure 3: prediction over 20 zones

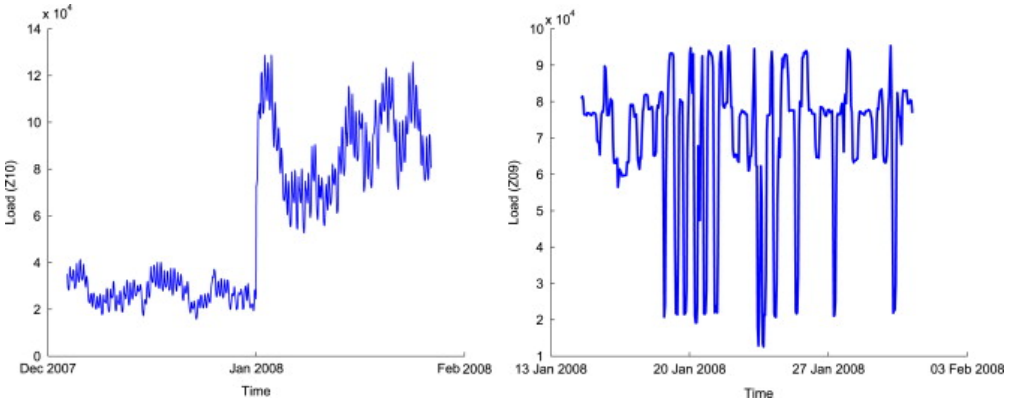


Figure 4: Left: Raw data at zone 10, showing a large discontinuity. Right: Raw data at zone 9, showing atypical load dynamics.

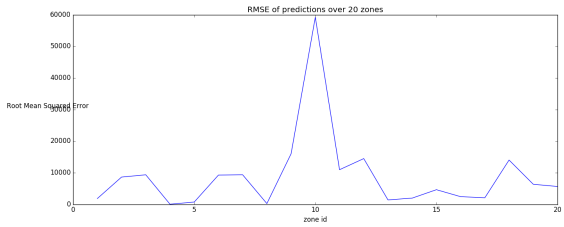


Figure 5: RMSE over 20 zones

6 Deep Neural Network

Neural network is a principled way to do the time series prediction. It can simulate any shapes function via multiple layers, which represents the hidden transitions between time steps. At the same time, some neural network can give the simulated function temporal properties, which is essential to time series prediction.

In this project, we have tested 3 approach of neural network: the Multilayer perceptron, the simple RNN and the Long Short Term Memory. We analysis the dataset to be divided into 1580 days of training set and 6 days of test set.

The feature will be affected by the clustering. After the clustering process, we find that the load in zone 9 and zone 10 is the outlier, which has the feature representation in time series different from other zones. Thus we collect the dataset without zone 9 and zone 10 and stack as part of the feature in each hour. The zone 9 and zone 10 will be predicted separately. These dataset with high correlations will be stack in vector space and seen as features in the prediction with the temperature feature data from 11 stations, date information feature and the Fourier Transform data feature. Which can be expressed as:

$$\mathbf{x} = [x_1, x_2, ..., x_n, date, month, temperature, f1, ..fm] \; n \neq 9, 10 \tag{3}$$

Where $x_1, x_2, ..., x_n$ is the highly correlated load, and the dataset with low correlations is not considered in the feature. And also, before the training and prediction, all the data need to be normalized individually by $normalized = \frac{data - mean}{standard\ deviation}$

6.1 Multilayer Perceptron

In the first approach, we apply the multilayer perceptron on the time series prediction. The MLP with 3 layers has the structure shown as follows:

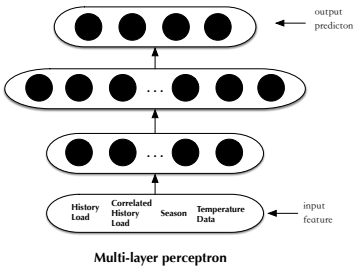


Figure 6: The Multilayer Perceptron

The strategy is that we find the electricity load time series at the same day in with 1 week interval has a similar shape and periodic properties in zones except zone 9 and zone 10. Thus, we develop a multilayer perceptron

which is has similar property of Auto-Encoder. The input of the multilayer perceptron is the load data in 20 zones without zone 9 and zone 10 and the temperature data in 11 stations. The date feature will not be considered in the input because the input and output has 1 week interval, the date feature will mislead the prediction. After the training, the multilayer perceptron will learn a transition between the load data and the load data 1 week after.

The prediction of the multilayer perceptron is shown below:

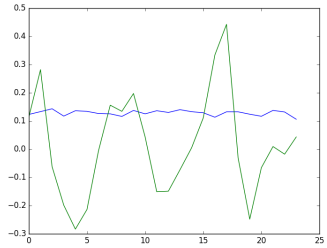


Figure 7: The Multilayer Perceptron Prediction

According to the graph, the prediction of the multilayer perceptron is not good. There is a translation between the prediction and the true label. And the shape of the amplitude is also changed. Thus the transition cannot be learned only on the multilayer perceptron without temporal properties. Thus, the Recurrent Neural Network is applied.

6.2 RNN

In the second approach, we applied the Simple Recurrent Neural Network for the prediction. The structure of the simple RNN is shown as figure below:

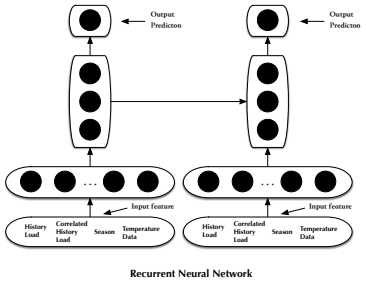


Figure 8: The Simple RNN

The strategy is that we see the electricity load as a time series, the load at the time step $t + 1$ is has transitions to time step 1 to t in latent space. Our target is to learn a transformations between the time step t and time step 1 to t to the output prediction.

During the training process, we divide the training data into 158 batches. Each batch has 10 days, which is 240 hours. Each time step input into the network is one hour. Thus each batch has 240 input time steps. The longer input batch to the RNN will provide better temporal representation for the prediction. The input and output of the neural network has one time step interval. The input data is the electricity load in 20 zones except zone 9 and zone 10, the temperature data in 11 stations and the date, week day feature. The output is the electricity load in $t + 1$.

The prediction test is shown below:

The load prediction after 50 time steps becomes intractable. The prediction saturate gradually to a level. The main

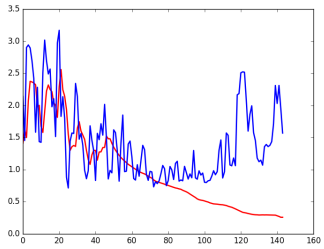


Figure 9: The Simple RNN Prediction

cause of the problem is that the gradient can be written as:

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \tag{4}$$

$$= \prod_{i=k+1}^{t} W^T diag[\phi'(h_{I-1})] \tag{5}$$

$$||\frac{\partial h_t}{\partial h_k}|| = (\gamma_\theta \gamma_\phi)^{t-k} \tag{6}$$

If the history is long, the gradient may explode as $(\gamma_\theta \gamma_\phi)^{t-k}$ is large. Thus the RNN cannot remember long-term memory. This time series has 240 time steps predictions, Simple RNN is not suitable for this task. Thus we tried the LSTM.

6.3 LSTM

In the third approach, we are using the LSTM. The LSTM is a gated unit of RNN, which will enforce the memory length of the RNN. The structure of the LSTM is shown in the figure below:

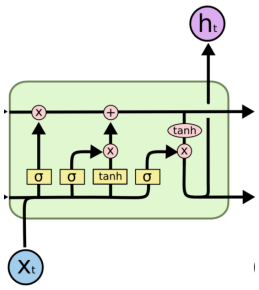


Figure 10: The LSTM

In this approach, we are stacking 2 layers of LSTM. The LSTM has a forget gate and a memory cell. It will decide which should be remembered, which should be forget. Some of the non-neccareay features will not be remembered into the memory. Thus, the LSTM can remember longer term of time series. In this approach, the input sequence is the 1 to t time steps, the output sequence is the 2 to $t+1$.

In the training process, it is similar to the RNN, the data is divided into 158 batches, and each of the batches has 10 days. And also, the features are the same as the RNN approach. The training input sequence and the training label sequence has 1 step interval. Then, in the testing process, we are using the features in time step t to generate the

electrical load on $t + 1$.

The electrical load prediction of 20 zones is shown on figure below, the blue line is true time series, the red line is the predicted time series:

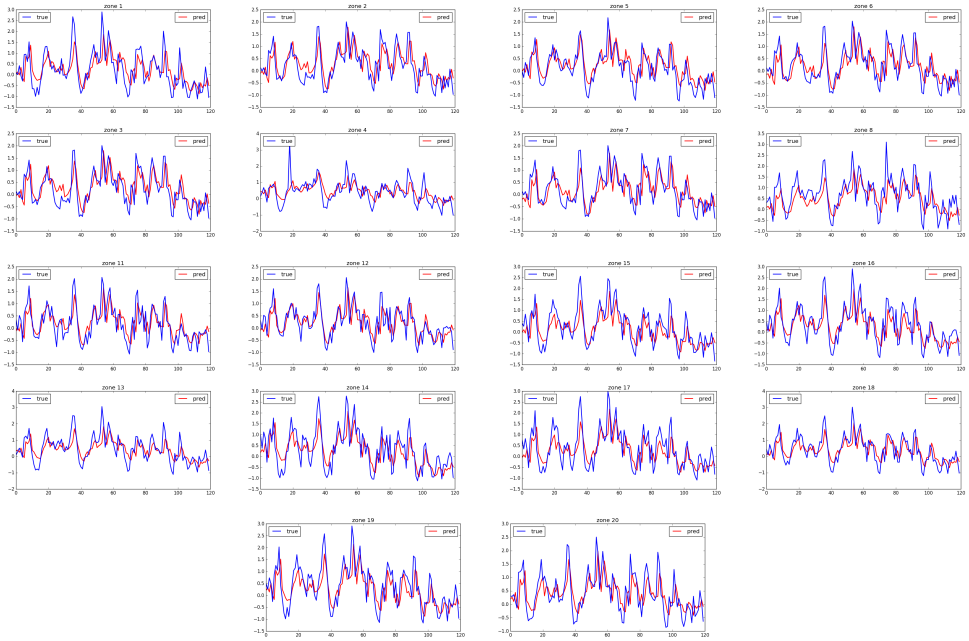


Figure 11: The LSTM Prediction without Zone9 and Zone10

And also, because the zone 9 and zone 10 is the outlier, which is difficult to predict with features belong to other zones. We predict the zone 9 and zone 10 separately. The prediction of zone 9 and zone 10 can be shown in the figure:

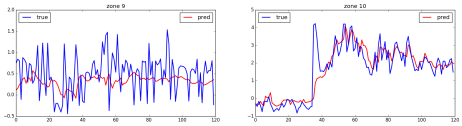


Figure 12: The LSTM Prediction of Zone9 and Zone10

As a result, the performance(root mean square error) can be shown in the table.

Table 1: The root mean square error

MLP	28176372
RNN	104728
LSTM	19639

As a result, the LSTM has the best prediction performance in the neural network time series prediction.

7 Conclusion

To conclude, we do time series prediction based on load time series data[5]. The clustering method is used to decide which time series should be predicted together in order to use the correlation information between time series. We also notice the time series have periodic behavior, so we use Fourier transformation to get useful features in frequency domain. At the same time, we use two types of learning methods for prediction: gradient boosting machine, and deep learning (MLP, RNN and LSTM). Both models have achieved reasonable results. From the result, we noticed even deep learning is extremely successful in some areas (such as computer vision and NLP), the gradient boosting regression tree is better than the deep learning in this scenario, which is quiet a surprise. We think the reasons could be it has too many local minimums and it's very hard to tune hyper-parameters, so we didn't train the model perfectly.

8 Appendix

The github site for our code is `https://github.com/tycui/IRDM2016-TimeSeriesForecasting`

References

[1] Enzo Busseti, Ian Osband, Scott Wong *Deep Learning for Time Series Modeling* 2012.

[2] Friedman J H. *Greedy function approximation: a gradient boosting machine[J]*Annals of statistics, 2001: 1189-1232.

[3] Hong T, Pinson P, Fan S. *Global energy forecasting competition 2012[J].* International Journal of Forecasting, 2014, 30(2): 357-363.

[4] James Robert Lloyd. *GEFCom2012 hierarchical load forecasting: Gradient boosting machines and Gaussian processes* 2013.07.02

[5] https://onlinecourses.science.psu.edu/stat510/node/71

[6] https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting

[7] http://colah.github.io/posts/2015-08-Understanding-LSTMs/