# Research on Short-term Load Forecasting Using XGBoost Based on Similar Days

Liao Xiaoqun, Cao Nanlan, Ma Li, Kang Xiaofan
Xi'an University of Science and Technology,Xian, Shaanxi, 710054, China
18192090869@163.com

***Abstract*—In this paper, the power load data is increasing exponentially and the traditional forecasting model is fatigued and difficult to achieve high efficiency when dealing with massive data. A XGBoost load forecasting model based on similar days is proposed. This model analyzes the common laws of meteorological and daily types on the load， The XGBoost model with the second-order Taylor expansion and loss function is added to the regular term to control the complexity and over-fitting. The real charge data and temperature data in a certain area are taken as samples. The simulation results show that the XGBoost model based on similar days can predict the load in short-term load forecasting effectively.**

***Index Terms*—Short-term load forecasting ;similar days;XGBoost；**

## I. INTRODUCTION

The short-term load forecasting of power system is based on historical load data, considering the relevant factors affecting the load,reflecting the load value of a certain period of time scientifically , ensuring the safe operation of the power system effectively , and providing a basis for the power grid to formulate scheduling and power supply plans.

The traditional methods in the field of power load forecasting mainly include time series based ARMA model [1-2], Random Forest (RF) [3-4] and Gradient Boosting Regression Tree (GBRT) [5]. The prediction models of such traditional methods are easy to construct and used widely. The shortcoming is that the stability of the load sequence is required to be high, which will affect the prediction accuracy under certain conditions. In order to solve these problems, shallow machine learning algorithms represented by Support Vector Machine (SVM) [6] and Artificial Neural Network (ANN) [7] rely on excellent nonlinear fitting ability, which is widely concerned by researchers in this field. The load prediction of RF, GBRT and ARIMA methods is compared in the literature[8]. The analysis results show that ARIMA prediction accuracy is worse than that of integrated tree models such as GBRT and RF. With the increase of the number of learning samples and the increase of influencing factors, the non-parametric model optimization process is more complicated, the prediction stability is not high, and it is easy to fall into the local optimal value.

In this paper, XGBoost is introduced into the field of load forecasting. It is proposed to standardize the load factor, establish a coefficient feature map, standardize the feature to select similar days, and then use XGBoost method to predict the load data. The result validates the validity of the XGBoost model based on similar days.

## II. XGBOOST ALGORITHM PRINCIPLE

### A Gradient Boosting algorithm principle

1. Initialize the model to a constant value

$$F_0(x) = \arg\min_{r} \sum_{i=1}^{n} L(y_i, \gamma) \tag{1}$$

2. Iteratively generate M base learners

(1)Calculate the pseudo-residual

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1,2,....n \tag{2}$$

(2) based on the base learner

(3) Calculate the optimal $\gamma_m$

$$\gamma_{\mathrm{m}} = \arg\min_{\gamma} \sum L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{3}$$

(4)Update the model

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{4}$$

The main idea of the Gradient Boosting algorithm is to fit the negative gradient of the loss function in repeated iterations after optimizing the empirical loss function, and then use the linear search method to generate the optimal weak learner. The XGBoost algorithm implements the weak learner by optimizing the structured loss function, and the XGBoost algorithm does not use the linear search method, directly uses the first derivative and the second derivative of the loss function, The performance of the algorithm is improved by pre-sorting, weighted quantile, sparse matrix identification and cache recognition.

### B Principle of XGBoost algorithm

#### 1) XGBoost loss function

The XGBoost algorithm is a tree-based lifting algorithm with regularizations added to its optimized objective function.

$$\begin{aligned}\Psi(\mathrm{y}, F(X)) &= \sum_{i=1}^{N} \psi(y_i, F(x_i)) + \sum_{m=0}^{T} \Omega(f_m) \\ &= \sum_{i=1}^{N} \psi(y_i, F(x_i)) + \sum_{m=0}^{T} \gamma L_m + \frac{1}{2}\|\omega_m\|^2\end{aligned} \tag{5}$$

$\gamma$ and $\lambda$ two regularization coefficients can control the complexity and output value of the model. Suppose that at the point $P_{\mathrm{m-1}} = F_{m-1}(X)$ ,XGBoost takes the Taylor approximation of the loss function at this point directly, and then trains the weak learner $f_m$ by minimizing the approximation loss function.

978-1-7281-1307-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICITBS.2019.00167

$$\Psi_m = \sum_{i=1}^{N} \psi(y_i, F_{m-1}(x_i) + f_m(x_i) + \Omega(f_m))$$
$$\approx \sum \left[ \psi(y_i, F_{m-1}(x_i)) + g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(f_m) \tag{6}$$

*2) Determine the optimal output value of each leaf node*

Suppose that a tree is generated in the mth iteration, there are L leaf nodes $\{l_1, l_2, \dots, l_L\}$, which $I_j = \{i | q(x_i = j)\}$ is the sample index number falling on the jth leaf node, q is the result of the mth spanning tree, Mapping a sample x to a corresponding leaf node, the optimization function can be expressed as follows:

$$\overline{\Psi}_m = \sum_{i=1}^{N} \left[ g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(f_m)$$
$$= \sum_{j=1}^{L} \left[ \left( \sum_{i \in I_j} g_i \right) \omega_j + \frac{1}{2} \left( \lambda + \sum_{i \in I_j} h_i \right) \omega_j^2 \right]$$
$$= \sum_{j=1}^{L} f(\omega_j) + \gamma L \tag{7}$$

$\overline{\Psi}_m$ can be achieved by taking the minimization loss value $f(\omega_j)$ in each leaf node.

$$\overline{\Psi}_m(q) = -\frac{1}{2} \sum_{j=1}^{L} \frac{\left( \sum_{i \in I_j} g_j \right)^2}{\lambda + \sum_{i \in I_j} h_j} + \gamma L \tag{8}$$

It can be seen from the above analysis that in the loss-minimum tree q of the mth generation model $F_m(X) = F_{m-1}(X) + f_m(X)$, the optimal output value of each leaf node is determined only by the first derivative $g = (g_1, g_2, \dots, g_N)$ and the second derivative $h = (h_1, h_2, \dots, h_N)$ of the loss function $\psi$ at the point $F_{m-1}(X)$.

*3) Integration of splitting conditions and weak learners*

Assume that starting from the root node, the index set of all samples on the node is represented by I, and the index sets of the split left and right child samples are represented by $I_L$ and $I_R$, $I = I_L + I_R$ and there is a reduction of the mth tree loss function after splitting. It can be expressed as:

$$\Delta\Psi = \frac{1}{2} \left[ \frac{\left(\sum_{i \in I_L} g_i\right)^2}{\lambda + \sum_{i \in I_L} h_i} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\lambda + \sum_{i \in I_R} h_i} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\lambda + \sum_{i \in I} h_i} \right] - \gamma \tag{9}$$

From the value of equation (9) as the splitting condition, the splitting point and the splitting attribute are optimized in the splitting of each node. This gives the optimal model in the mth iteration $f_m(X)$.

T+1 weak learners are obtained after T iterations, and T+1 models are added to integrate weak learners. XGBoost uses the shrinkage method to reduce overfitting, and the model integration is expressed as follows:

$$F_m(X) = F_{m-1}(X) + \eta f_m(X), 0 < \eta \le 1 \tag{10}$$

In addition to using Shrinkage to reduce overfitting, in the XGBoost tectonic tree model, the method of randomly selecting a certain number of feature subsets in a random forest is also used in determining the optimal splitting point, and the risk of overfitting can also be suppressed. The larger the feature subset, the smaller the deviation of each weak learner, but the larger the variance. We need to use cross-validation to weigh to determine the specific ratio.

## III. METHOD OF PREDICTION

In this paper, the historical load data is preprocessed, and then the temperature data is used to calculate the correlation of similar day. According to the comparison between the date and the history to be predicted, the sample data is selected and predicted by XGBoost method to obtain the load prediction value.

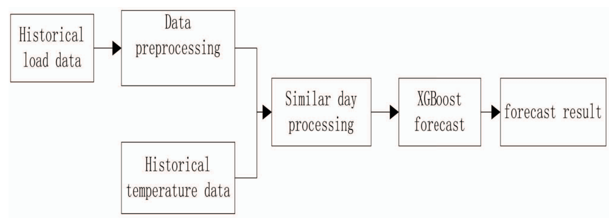


Figure 1. Process of load forecasting

### *A Data preprocessing*

1. The sample selects the historical load data of the same day type as the day to be predicted, and selects the maximum rate of change of the historical load of the adjacent points in the sample as the threshold.

2. Compare each load point of the forecast day with the previous point of the current time in order, and repair processing if bad data is detected, and the data is the basis of the next comparison.

3. The adjustment value uses a linear combination of historical load data of the same time and the same day type.

$$\begin{cases} \left| \dfrac{L(d,t) - L(d,t-1)}{L(d,t-1)} \right| > \alpha_{\max}(t,n), t \neq 0 \\ \left| \dfrac{L(d,t) - L(d-1,\infty)}{L(d-1,\infty)} \right| > \alpha_{\max}(t,n), t = 0 \\ \alpha_{\max}(t,n) = \max\left\{ \dfrac{L(d-i,t) - L(d-i,t-1)}{L(d-i,t-1)} \right\} \end{cases} \tag{11}$$

Where i=1-n,

$$L(d,t) = \lambda_1 L(d-1,t) + \lambda_2 L(d-2,t) + \cdots + \lambda_m L(d-m,t) \tag{12}$$

$$\begin{cases} \lambda_j = \beta(1-\beta)^{j-1}, \beta \in (0,1) \\ \sum_{j=1}^{m} \lambda_j = 1 \end{cases} \tag{13}$$

In equation (11), $L(d,t)$ is the load point at the time of day d; $L(d-1,\infty)$ is the last load point on day d-1; the threshold $\alpha_{\max}(t,n)$ is defined as the maximum value of load change at the time of the first n days t and t-1; i is the current date The number of days apart; $L(d-m,t)$ is the load value at time t of day d-m; the weight coefficient Am is used to indicate the effect of the load value at time t on day d-m on the load value at time t on day d; $\beta$ is the smoothing coefficient. when $\lambda_1 = \lambda_2 = \cdots \lambda_m$ , it is the average value of the historical load at the same time.

### *B Selection of similar days*

The similar day prediction is based on the approximate prediction method of cluster analysis. First, different evaluation functions are set, then the existing data is traversed, and the analysis is predicted according to the related properties of the search. Accurate selection of similar days will directly affect the accuracy of load data prediction. The following two evaluation functions for similar days are introduced.

#### *1) Time factor matching coefficient*

Day type and time distance are factors that need to be considered when selecting similar days. The expression (14) is used to analyze the similarity between the predicted day and the historical day in the time distance and the day type .

$$\delta(t) = S\beta^{t} \qquad (14)$$

When the date type of the forecast date is the same as the historical day , S takes 1; when the day type of the forecast date and the historical day are different but both are the weekends or weekdays, S takes 0.8. The similarity attenuation ratio is expressed by $\beta$, Usually take 0.90~0.98.

#### *2) Meteorological factor matching coefficient*

The forecast date and The historical day temperature data similarity can be quantified by the matching factor of the meteorological factor. First, the eigenvectors are used to vectorize meteorological factors, and then the meteorological factors are matched.

There is a significant nonlinear relationship between temperature and load, so the temperature needs to be properly adjusted so that the relationship between temperature and load is basically linear. Therefore, the load changes with temperature as shown in Figure 2.

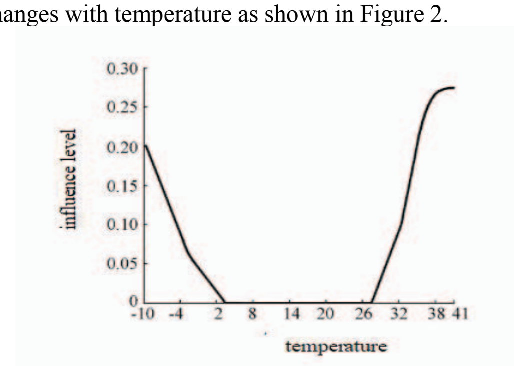


Figure 2. Effect of temperature on load

The difference between the kth feature and the day to be predicted for the day and the i-th historical day can be expressed as:

$$\Delta_i(k) = \left|X_i(k) - X_0(k)\right| \qquad (15)$$

The values normalized by the difference are expressed as:

$$\Delta_i(k) = \frac{\Delta_i(k) - \min\limits_i \Delta_i(k)}{\max\limits_i \Delta_i(k) - \min\limits_i \Delta_i(k)} \qquad (16)$$

Then its correlation coefficient can be expressed as:

$$\zeta(k) = \frac{\min\limits_i \min\limits_k \Delta_i(k) + \rho \max\limits_i \max\limits_k \Delta_i(k)}{\Delta_i(k) + \rho \max\limits_i \max\limits_k \Delta_i(k)} \qquad (17)$$

In summary, the degree of correlation between the date to be predicted and the ith historical day is:

$$r_i = \frac{1}{m}\sum_{k=1}^{m} \zeta_i(k) \qquad (18)$$

It can be known from formula (18) that the smaller the difference in meteorological factors, the larger the correlation coefficient. The value of the correlation degree is taken as the matching factor of the meteorological factor, which is determined according to the actual situation and experience of different regions. Therefore, the historical date of the evaluation function should be added to the learning sample.

### *C XGBoost prediction*

The learning samples obtained through similar days processing are predicted by the XGBoost model of the second-order Taylor expansion and the loss function added to the regular term. Compared with the traditional machine learning algorithm, XGBoost can be well extended to large data sets.

## IV. EXAMPLE ANALYSIS

### *A Experimental data sets and evaluation indicators*

The experimental data is derived from the load data of each hour of sampling from a network company of State Grid Corporation from June 2014 to June 2018 (24 load data per day), using this data to establish the model and predict the change of power load. The trend is then compared with the actual load data at that stage.

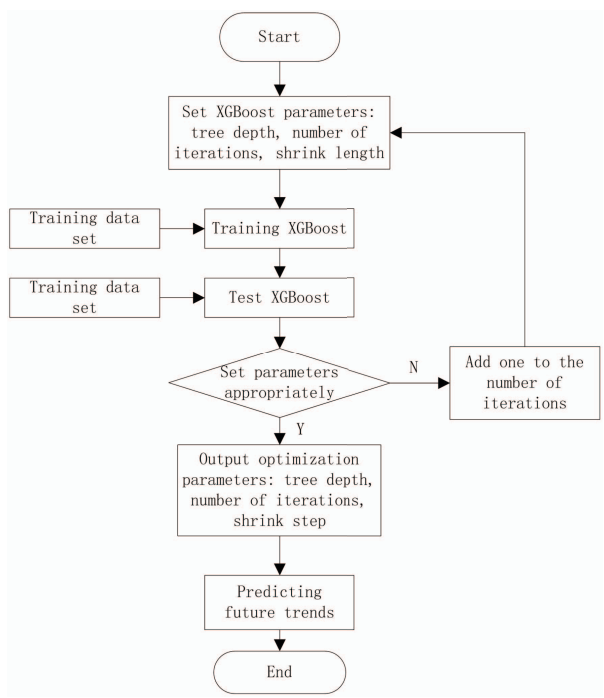


Figure 3. XGBoost load forecasting process

### B Analysis of prediction results

In order to compare the prediction effects of the XGBoost model based on similar days, this paper uses LSTM as the comparison algorithm to compare the XGBoost based on similar days with the traditional XGBoost and LSTM. Firstly, from the overall data, all three methods are used in the whole prediction period. It can better predict the ultra-short-term load of the power grid. In the area where the fluctuation of charge quantity is obvious, the XGBoost prediction result based on similar day is closer to the real load. Combined with the mean relative error (MAE) in the table, it can be seen that the relative error of XGBoost is the lowest based on the similar day; the root mean square error (RMSE) is used to analyze the dispersion of the predicted values among the three, and the RMSE numerical value is based on the similar day XGBoost prediction. The value deviation is smaller and the prediction accuracy is higher.

TABLE Ⅰ：COMPARISON OF PREDICTION RESULTS OF THREE MODELS

| model | MSE | RMSE | MAPE |
|---|---|---|---|
| XGBoost based on similary days | 4662093.64 | 0.0934 | 0.0880 |
| Traditional XGBoost | 6630414.12 | 0.1314 | 0.1201 |
| LSTM | 6835806.45 | 0.1431 | 0.1309 |

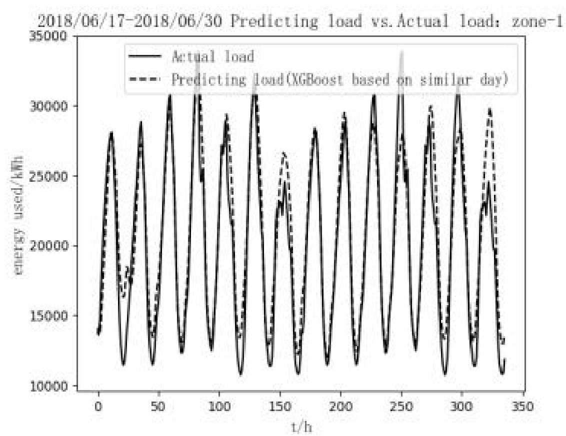


Figure 4. Based on similar days XGBoost prediction results

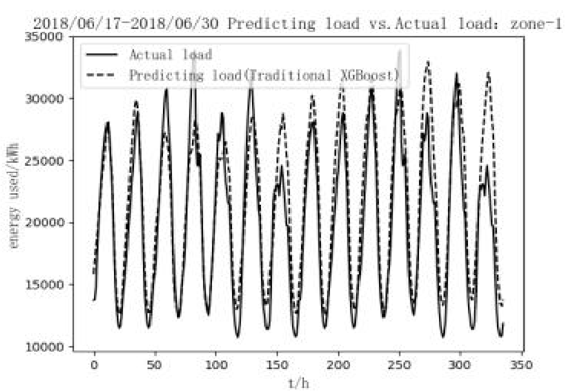


Figure 5. Traditional XGBoost prediction results

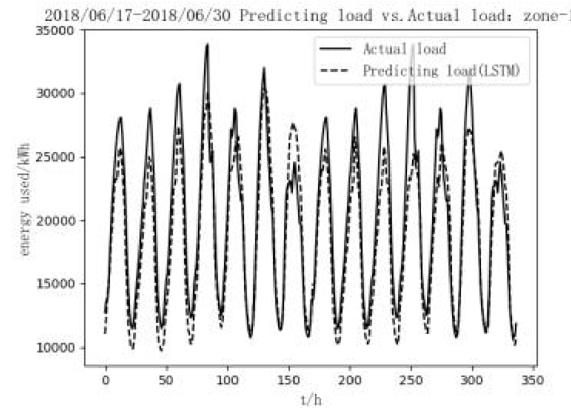


Figure 6. LSTM prediction results

Combined with Figures 4, 5 and 6, XGBoost based on similar days can be effectively fitted regardless of grid load fluctuations and fluctuation frequency, and exhibits better accuracy than traditional XGBoost and LSTM predictions.

## V. CONCLUSION

In this paper, a similar day-based XGBoost short-term load forecasting method is proposed. By classifying the main influencing factors of the standard load, the feature map is constructed reasonably to select similar days. The XGBoost model for big data and parallel learning is used for load forecasting. The RMSE of the XGBoost model is 0.093%, and the error rate is much lower than the 13.14% of the traditional XGBoost and 14.31% of the LSTM. This method is effectively validated to significantly improve the prediction accuracy.

## REFERENCES

[1]Moghram I, Rahman S. Analysis and evaluation of five short-term load forecasting techniques [J]. IEEE Transactions on power systems, 1989, 4(4): 1484-1491.

[2] Li Ni, Jiang Yuechun, Huang Shan, et al. Short-term load forecasting based on cumulative autoregressive moving average transfer function model [J]. Power Grid Technology, 2009, 33(8): 93-97.

[3] Dudek G. Short-term load forecasting using random forests[M]. Intelligent Systems' 2014. Springer International Publishing, 2015: 821-828

[4] Wu Yuyu, and Jing Han, Zhang Pei and so on. Short-term load forecasting of power system based on grey projection improved random deep forest algorithm [J]. Automation of Electric Power Systems, 2015, 39(12): 50-55.

[5] Taieb S B, Hyndman R J. A gradient boosting approach to the Kaggle load forecasting competition [J]. International Journal of Forecasting, 2014, 30 (2): 382-394.

[6] Yang Zaihe, Xiang Tieyuan, Zheng Dan. Research on short-term load forecasting of microgrid based on wavelet transform and SVM algorithm [J]. Modern Electric Power, 2014, 31(3): 74 7 9.

[7]LI Long, WEI Jing, LI Canbing, et al. Load model prediction based on artificial neural network[J]. Transactions of China Electrotechnical Society, 2014, 31(3): 225-230.

[8] Papadopoulos S, Karakatsanis I. Short-term electricity load forecasting using time series and ensemble learning methods [C]. Power and Energy Conference at Illinois (PECI),, 2015 IEEE. IEEE, 2015: 1-6.