An Ensemble Model Based on Machine Learning Methods for Short-term Power Load Forecasting

Liqiang Ren, Limin Zhang, Haipeng Wang, Lin Qi

Institute of Information Fusion, Naval Aviation University, Shandong 264001, China

Abstract. Given the significant fluctuation of errors for single forecasting model and limitation of linear combined forecasting models, A nonlinear multi-model ensemble method for short-term power load forecasting is proposed. Firstly, the power load big data is pre-processed, and multi-dimensional input feature variables are constructed and selected. On this basis, three kinds of single prediction models of random forest, support vector machine and Xgboost are modelled, and three different prediction results are obtained. Then, each individual prediction result and actual load are taken as a new training data set, and secondary learning is performed to obtain a final prediction result. The numerical experiments show that the proposed ensemble method combines the advantages of the single model, and has strong generalization ability and higher stability and accuracy, and has a high practical value.

1. Introduction

Short-term load forecasting (STLF) is an important part of power system load forecasting, and it is also one of the basic links in smart grid construction[1]. The main application of STLF is to provide load forecasting for unit combination and economic dispatch. For example, if the load demand is known in advance, the generator can be operated at the lowest possible cost, which can increase the economic efficiency of the power grid and the power plant[2]. The second application of STLF is the safety evaluation of power system. The prediction result is of great significance to the safe and stable operation of the power grid. In addition, short-term load forecasting results not only contribute to the smooth development of power management, but also contribute to the goal of energy conservation, emission reduction and environmental protection[3]. However, due to the randomness, non-linearity, and instability of the factors affecting power load changes, it is difficult to achieve high-precision predictions.

In the past decades, scholars have put forward many methods for improving the accuracy of STLF, which can be mainly divided into three categories: classical methods, artificial intelligence methods, and machine learning methods. Classical prediction methods are based on mathematical models, including time series method[4], regression analysis method[5], etc. The classical prediction model is a simple linear method. Its advantages are simple model structure and fast prediction speed. Due to the complex nonlinear characteristics of the short-term power load, the prediction accuracy of STLF by classical prediction methods is not sufficient. Artificial intelligence methods mainly include neural networks[6], extreme learning machines[7] and so on. The neural network algorithm has better self-learning ability and nonlinear fitting ability, but the network structure often needs to be determined empirically. Although the extreme learning machine can solve the problem of difficult network structure selection, its performance is unstable. Machine learning prediction models mainly include support vector machines[8], random forests[9], and gradient boosting regression trees[10]. Support vector machine has a good effect on solving practical problems such as non-linearity and high
dimensionality. The short-term load forecasting model based on support vector machine has the characteristics of high precision, strong generalization ability, and global optimum. However, the training parameters of the model need to be obtained through the optimization method, and the robustness of the outliers is not good enough, which increases the time and difficulty of training. Random forests and gradient-boosting regression trees algorithms are essentially tree-based algorithms that overcome the shortcomings of Support vector machine and are robust to outliers. However, when the load fluctuations are large, the tree-type algorithm does not have high prediction accuracy. Each prediction model has its own limitations. The prediction accuracy of each single model is different from that of the single model at the same time, and the prediction accuracy is still room for improvement. Multi-model ensemble learning completes the learning task by constructing and combining multiple learners. The generalization ability, stability, and prediction accuracy of the learning system are significantly improved by multi-model ensemble learning.

To solve the above problems, a non-linear multi-model ensemble method for short-term power load forecasting is proposed. Three optimal single prediction models including random forest, support vector machine and Xgboost were selected for modeling, and three different prediction results were obtained. Then, the prediction results of each single model are used as input variables, and the corresponding actual loads are used as target variables. The Xgboost algorithm is used for secondary learning to obtain the final combined prediction model. At the same time, in order to improve the accuracy of the model, the importance score of the input feature variable is calculated before modeling, and redundant feature variables are removed based on this score.

The rest of this paper is organized as follows. Section 2 presents the full procedure of multi-model ensemble for short-term power load forecasting, including methods for feature selection and algorithms for estimating electrical loads. Section 3 details the experimental results. The conclusion is described in Section 4.

2. Materials and Methods

2.1. Data Description and Preprocessing

This study selected the hourly electricity load data for the A region of New York State from 2001 to 2015, as well as the meteorological data for the corresponding time in the region. Through data extraction and integration, the input feature variables of the predictive model are constructed. This study considers the characteristic variables shown in Table 1. In Table 1, current load influencing factors include historical load, calendar variables, and weather variables. Due to systematic errors, "bad data" such as missing data and abnormal data are inevitable. These data seriously interfere with the training and prediction process of the model and reduce the accuracy of the prediction results. Therefore, in order to guarantee the smoothness of the load sequence, the consistency of the order of magnitude, and the good performance of the algorithm model, it is indispensable to preprocess the original data.

2.2. Missing value processing

For missing data, select the nearest m data from the missing data according to the Euclidean distance or correlation analysis, and then calculate the weighted average.

\[ l(h) = \alpha_1 l(h_1) + \alpha_2 l(h_2) + \cdots + \alpha_m l(h_m) \]  \hspace{1cm} (1)

Where \( l(h) \) is the missing data and does not exist, \([l(h_1), l(h_2), \ldots, l(h_m)]\) is the closest data vector to \( l(h) \), and \([\alpha_1, \alpha_2, \ldots, \alpha_m]\) is the weight vector determined by the Euclidean distance. That is to say, the small Euclidean distance has a high weight coefficient.
Table 1 Description of the variables of the dataset used in this study.

| Variables   | Description                      |
|-------------|----------------------------------|
| load_P       | Present load                     |
| peak_load_T  | Peak load of the day before dew   |
| L_m24       | Load of 24 hours before           |
| L_m48       | Load of 48 hours before           |
| Ldif        | Difference of L_m24 and L_m48     |
| temperature_T | Present temperature              |
| humidity_H  | Present humidity                  |
| humidity_H11 | 1 hour before temperature         |
| pressure_P  | Sealevel pressure                 |
| windspeed_P | Wind speed                       |
| pressure_P  | Sealevel pressure                 |
| month_M     | Months of the year, integer, between 1-12 |
| hour_H      | Hours of the day, integer, between 0-23 |
| day_D       | Days of the month, integer, between 1-31 |
| week_W      | Week of the year, integer, between 1-52 |
| dow_D       | Day of the week, integer, between 0-6 |
| days_D      | Days of the year, integer, between 0-365 |
| visibility_V | Present visibility               |
| visibility_V | Present visibility               |

2.3. Outlier processing

Due to the influence of random factors (such as severe temperature changes and occurrence of unexpected events), abnormal data points may sometimes appear in the load data. These abnormal points will disturb the regularity of the entire data sequence and destroy the similarity of data curves of the same date type. Thus affecting the prediction accuracy. Therefore, it is necessary to correct abnormal data.

The load curve not only has obvious seasonal features, but also presents a specific daily load pattern. In a season, the load data with the same date type are similar to each other, that is, the data in the same season has a periodicity. Therefore, it is assumed that \( I(h,k,n) \) is the load value, \( h = 1, 2, \ldots, 24 \); \( k \) represents the day of the week, \( k = 1, 2, \ldots, 7 \); \( n \) represents the week of the season, \( n = 1, 2, \ldots, N \).

Firstly, we calculate the mean \( E(h) \) and the hourly squared error \( V(h) \) of the same date type based on formulas (2) and (3).

\[
E(h) = \frac{1}{N} \sum_{n=1}^{N} l(h,k,n) \tag{2}
\]

\[
V(h) = \sigma_h^2 = \frac{1}{N} \sum_{n=1}^{N} [l(h,k,n) - E(h)]^2 \tag{3}
\]

Then, the deviation rate is defined as \( \rho(h,k,n) \).

\[
\rho(h,k,n) = \frac{|l(h,k,n) - E(h)|}{\sigma_h} \tag{4}
\]

The process of correcting outliers is as follows:

\[
\rho(h,k,n) \geq \gamma, \begin{cases} 
1 < n < N, \hat{l}(h,k,n) = l(h,k,n-1) + l(h,k,n) + l(h,k,n+1) \over 2 \\
n = 1, \hat{l}(h,k,n) = l(h,k,n+1) \\
n = N, \hat{l}(h,k,n) = l(h,k,n-1) \\
\rho(h,k,n) < \gamma, \hat{l}(h,k,n) = l(h,k,n) 
\end{cases} \tag{5}
\]

Where \( \gamma \) is the expected deviation rate. According to reference\(^{11,12}\) and several experiments, set \( \gamma = 1.25 \) in spring and autumn, \( \gamma = 1.56 \) in summer and winter.

2.4. Xgboost algorithms

Xgboost\(^{13}\) (eXtreme Gradient Boosting) is an improved algorithm based on Gradient Boosting Decision Tree (GBDT). This algorithm not only has the advantage of higher precision than the
traditional boosting algorithm, but also can process sparse data efficiently and implement distributed parallel computing flexibly. The traditional GBDT algorithm only uses the first derivative information of the loss function. Xgboost does a second-order Taylor expansion of the loss function. The first and second derivatives are used simultaneously. In addition, Xgboost incorporates regular terms in the loss function to control the complexity of the model and reduce the variance of the model, making the learned model simpler and preventing overfitting. Xgboost can improve the accuracy of predictions while maintaining a certain speed.

The XGBoost algorithm achieves an estimate of the target variable by establishing a series of decision trees and assigning each leaf node a quantized weight. It is assumed that the Xgboost model has \( k \) decision trees. The following functions are used to predict the samples.

\[
\hat{y}_i = \sum_{k=1}^{K} f_k(x_i); f_k \in \mathcal{F} \quad \quad (6)
\]

Where \( \mathcal{F} \) is the hypothesis space and \( f(x) \) is the CART regression tree.

\[
\mathcal{F} = \{ f(x) = \omega_{q(x)} | q : \mathcal{Y}^n \rightarrow T, \omega \in \mathcal{R}^T \} \quad \quad (7)
\]

Where \( q(x) \) represents the decision rule for each tree, \( T \) is the number of leaf nodes in the tree, and \( \omega \) is the sample weight (leaf score) for the leaf nodes. \( \omega_{q(x)} \) represents the predictive value of the regression tree to the sample.

Xgboost objective function as follows

\[
\mathcal{L}(\theta) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad \quad (8)
\]

where \( \sum_k \Omega(f_k) \) is a regular term, which helps to prevent overfitting of the model.

\[
\Omega(f) = \gamma T + \frac{1}{2} \lambda \| \omega \|^2, \quad \gamma \text{ is a complexity parameter, } \lambda \text{ is a fixed coefficient, } T \text{ is the number of leaf nodes, } \| \omega \|^2 \text{ is the } L_2 \text{ norm of leaf node weight, and } l \text{ is Loss function.}
\]

2.5. Feature selection

In machine learning, there are often interdependencies between feature variables, or there is no correlation between certain feature variables and target variables. These redundant features not only increase the training time of the model, but also cause the "dimensional disaster" to complicate the model, and ultimately reduce the generalization ability of the model. Feature selection can eliminate irrelevant and redundant features, reduce the number of features, shorten the running time, and improve the accuracy of the model.

The Boosting tree can output the importance score of the feature, which can measure the priority of the feature in the split. Xgboost calculates feature importance through "weight", "gain" and "coverage" \([14]\). Weights represent the number of times a feature is used for splitting; gain is the reduction in average error when using feature splitting; and coverage is the average number of samples affected by splitting. As shown in Figure 6, the feature importance score is calculated by the weight method, and features having higher importance scores are considered to have a greater influence on the power load value. The feature selection retains the top 15 features that are rated highly. Obviously, the electrical load is sensitive to temperature variables.
3. Models training and ensemble process

The complete process of short-term power load forecasting based on multi-model ensemble is illustrated as follows and presented in Figure 2.

1. Data preprocessing: correct missing and outliers, and select the feature variables for the input model.

2. Model training and ensemble: Random forests, support vector machines, and Xgboost prediction models were trained, and each individual prediction result and actual load were taken as a new training data set for secondary learning.

3. Prediction and evaluation: Use the above model to predict load values in the test set and evaluate the performance of the model.

This study adopts a nonlinear model ensemble strategy. As shown in Figure 3, the specific implementation steps of the multi-model ensemble prediction model are as follows.

Step 1: Split the training set into train_train and train_valid by the method of 5-fold cross-validation [15].

Step 2: Primary model, train five RF models with the train_train, use the grid search method to optimize the hyper-parameters of the RF model, and predict the train_valid and test data sets respectively to obtain the train_valid_pred and test_pred results. The five train_valid_preds generated by the 5-fold cross-validation are concatenated into rf_train_pred, and the five test_preds are averaged to generate rf_test_pred.

Step 3: Repeat the previous step using SVM and Xgboost to obtain svr_train_pred, svr_test_pred, xgb_train_pred, and xgb_test_pred generated by SVR and Xgboost models.

Step 4: The secondary model, The Xgboost algorithm was chosen for secondary learning to achieve
ensemble of multiple primary models. The prediction values rf_train_pred, svr_train_pred, and xgb_train_pred obtained by each single prediction model are used as input, the corresponding actual value \( y_{training} \) is used as an output training Xgboost model.

Step 5: The rf_test_pred, svr_test_pred and xgb_test_pred generated by the primary model are used as input features, Using the ensemble model based on Xgboost to predict them, the final prediction results are obtained.

![Figure.3 Multi-model ensemble model](image)

4. Results and Discussion

The proposed ensemble model is compared with the prediction results of random forest, support vector machine and Xgboost model to verify the fitting effect of the ensemble model. Then, the predicted performance of the proposed ensemble model is compared with that of other model fusion methods to illustrate the accuracy and stability of the proposed ensemble model.

4.1. Evaluation Index

Mean absolute percent error (MAPE), mean absolute error (MAE), and relative error (RE) were used as criteria for error evaluation to analyze model prediction performance.

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{Y_i - \hat{Y}_i}{Y_i} \right) \times 100\%
\]  
\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i|
\]  
\[
RE = \frac{Y_i - \hat{Y}_i}{Y_i} \times 100\%
\]

Where \( Y_i \) represents the true value of the power load, \( \hat{Y}_i \) represents the power load predicted value, and \( n \) is the predicted time point. In power load forecasting, the smaller the MAPE value, the more accurate the load forecast.

4.2. Analysis of experimental results

Experiment 1: The prediction performance of the proposed ensemble model is compared with random forest, support vector machine and Xgboost single model.

Random forest, support vector machine, Xgboost and proposed ensemble model are applied to predict the daily hourly load of A area in December 2015 17-23. The training set selects data from 2001 to 2014. The statistical results of prediction error are shown in Table 3.

From the error results in Table 2, it can be seen that the proposed ensemble model has better prediction performance than the random forest, support vector machine, and Xgboost single model.
The MAPE of the proposed ensemble model is the lowest among all models, with a prediction accuracy of 98.14%, and the random forest has the largest MAPE value. This shows that the proposed ensemble model has better prediction accuracy and greater generalization ability than the single model.

| model      | MAPE /% | MAE /MW | Max RE /% |
|------------|---------|---------|-----------|
| RF         | 2.53    | 47.00   | 11.90     |
| SVM        | 2.44    | 45.35   | 8.26      |
| Xgboost    | 2.08    | 38.15   | 7.43      |
| Ensemble model | 1.86 | 34.53   | 7.59      |

Figure 4 a and b show daily and hourly load forecast results for a day and a week for random forest, support vector machine, Xgboost single model, and proposed ensemble model, respectively. The predicted curve of the proposed ensemble model fits the actual load curve better than the prediction results of the three single models. Although the other three single models can also determine the overall trend of actual load, their forecast error is very high. Obviously, the proposed ensemble model can improve the accuracy of load forecasting.

Experiment 2: The prediction performance of different fusion methods is compared to reflect the prediction accuracy and stability of the proposed fusion model. This paper compares the nonlinear model ensemble method with the linear weighted average model fusion method proposed in paper [16]. The optimal weights of random forest, support vector machine and Xgboost obtained by grid search are 0.1, 0.3 and 0.6. The training set selects data from 2001 to 2014.

Table 3 summarizes the error results of hourly load forecasting of different combination methods in 2015. The experimental results show that the prediction accuracy of the proposed nonlinear model ensemble method is significantly higher than that of the weighted average ensemble method. Therefore, for the power load with high random and nonlinear characteristics, the nonlinear ensemble method can better fit the actual load value than the linear weighted ensemble method.

| model             | MAPE /% | MAE /MW | Max RE /% |
|-------------------|---------|---------|-----------|
| Comparison model  | 2.17    | 42.09   | 21.75     |
| Ensemble model    | 1.97    | 35.12   | 16.59     |
5. Conclusion
A nonlinear multi-model ensemble method for short-term power load forecasting is proposed. The key to multi-model ensemble method is the training of primary models and the selection of input features. The redundant feature variables are removed based on the model feature selection technology, and the missing values and outliers of the original data are preprocessed. After that, the cross-validation method was used to obtain the input feature variables of the secondary learning model, and secondary learning was performed. Based on the actual power load data, the proposed multi-algorithm multi-model ensemble method was verified and compared with the random forest, support vector machine, Xgboost, and linear weighted model ensemble methods. The comparison results show that the proposed fusion model can realize high precision prediction of complex nonlinear power load. It has the characteristics of strong generalization ability, high prediction precision and relatively stable prediction error. The proposed multi-model fusion framework is a promising alternative to STLF and has a high practical value.

Acknowledgments
This research was supported by the National Natural Science Foundation of China (Grant No.91538201); Taishan scholar project special fund project (Grant No.Ts201511020).

Author Contributions
Limin Zhang, Haipeng Wang design experiments and collect data. Liqiang Ren and Lin Qi conducted a case study and analyzed the results. Liqiang Ren wrote this paper. All authors read and approved the final manuscript.

Conflicts of Interest
The authors declare no conflict of interest.

References
[1] Li S, Goel L, Wang P. An ensemble approach for short-term load forecasting by extreme learning machine[J]. Applied Energy, 170, pp. 22-29, (2016).
[2] Hippert H S, Pedreira C E, Souza R C. Neural Networks for Short-Term Load Forecasting: A Review and Evaluation[J]. IEEE Transactions on Power Systems, 16(1), pp. 44-55, (2001).
[3] Javed F, Arshad N, Wallin F, et al. Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting[J]. Applied Energy, 96(8), pp. 150-160, (2012).
[4] Cho M Y, Hwang J C, Chen C S. Customer short term load forecasting by using ARIMA transfer function model[C]// International Conference on Energy Management and Power Delivery, 1995. Proceedings of Empd. IEEE, pp. 317-322, (1995).
[5] Dudek G. Pattern-based local linear regression models for short-term load forecasting[J]. Electric Power Systems Research, 130, pp. 139-147, (2016).
[6] Li S, Wang P, Goel L. Short-term load forecasting by wavelet transform and evolutionary extreme learning machine[J]. Electric Power Systems Research, 122, pp. 96-103, (2015).
[7] Wang Baoyi, Zhao Shuo, Zhang Shaomin. Distributed Power Load Forecasting Algorithm Based on Cloud Computing and Extreme Learning Machine[J]. Power System Technology, 38(2), pp. 526-531, (2014).
[8] Lin W M, Tu C S, Yang R F, et al. Particle swarm optimisation aided least-square support vector machine for load forecast with spikes[J]. Iet Generation Transmission & Distribution, 10(5), pp. 1145-1153, (2016).
[9] Li Yuhua, Chen Hong, Guo Kun, et al. Research on load forecasting based on random forest algorithm[J]. Computer Engineering and Applications, 52(23), pp. 236-243, (2016).
[10] Taieb S B, Hyndman R J. A gradient boosting approach to the Kaggle load forecasting competition[J]. International Journal of Forecasting, 30(2), pp. 382-394, (2014).
[11] Ming Liu. The power system load forecasting research based on wavelet and neural network theory [dissertation]. Nanjing: Nanjing University of Science and Technology, 2012.

[12] Chan Lv. Short-term load forecasting based on BP neural network [dissertation]. Hubei: Huazhong University of Science and Technology, 2007.

[13] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 785-794, (2016).

[14] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Springer[J]. Mathematical Intelligencer, 27(2), pp. 83-85, (2001).

[15] Fan Yongdong. A review of cross validation methods in model selection [D]. Shanxi University, 2013.

[16] Bai Pengfei, An Qi, Nicolaas Fransde ROOIJ, et al. Internet credit individual credit evaluation method based on multi-model integration[J]. Journal of South China Normal University (Natural Science Edition), 49(06), pp. 119-123, (2017).