Multifeature pool importance fusion based GBDT (MPIF-GBDT) for short-term electricity load prediction

Shengwei Lv 1, Gang Liu 1,* and Xue Bai 2

1 School of Automation Engineering, Shanghai University of Electrical Power, Shanghai 200090, China
2 Shanghai Brillancetech Power Equipment & Engineering Co., Ltd, Shanghai 200000, China
Email: liugang@shiep.edu.cn

Abstract. Feature selection is one of the key factors in predicting. Different feature selection algorithms have their unique preferences for elemental analysis of the data. This results in failing to determine the optimal features when a dataset goes through different feature selection algorithms to get different pools of input features, which in turn affects the prediction quality. To address this problem, the method integrates and fuses the feature importance values of two different feature selection methods. Then the input feature pools are optimized and filtered for the prediction model. Finally, the multifeature pool importance fusion based GBDT (MPIF-GBDT) is developed, which integrates the different feature selection methods and predicts the short-term power load in combination with the gradient boosting decision tree algorithm. In this paper, the tree model feature selection and the Recursive Feature Elimination (RFE) are chosen as feature selection methods. The experimental results show that MPIF-GBDT can significantly improve the accuracy of the prediction compared with the benchmark model.

1. Introduction
Building energy consumption accounts for a large proportion of the world's energy consumption and CO2 emissions. Researchers have been developing predictive simulation tools since the early 1990s. Intelligent algorithm have gradually become the focus of the research for their advantages such as fast computation speed and strong learning ability [1], and the integrated learning algorithm models attract many researchers with its good performance to conduct in-depth research. He et al.[2] used a decomposition-based quantile regression forest to predict short-term loads. Wang et al.[3] proposed a novel sparse Adaboost combined with an echo state network to predict industrial electricity consumption. The selection of a good feature input prediction engine is a very important factor to improve the performance of the model, undesirable feature input will append redundant candidates to the feature pool [4], which will not only deteriorate the efficiency of model processing, but also reduce the generalization performance and prediction accuracy of the model [5].

At present, there is still much room for research work on how to choose the optimal features. The single feature selection algorithm tends to express some aspect of the data features, and data feature extraction has a role in reducing the data dimension, but it also has a loss to the data, and often the lost information contains some of the components of the data that need to be expressed [6]. Motivated by above problems, a hybrid selection method is proposed. This method is based on the reorganization of the RFE feature importance value and the feature importance value of the tree model feature selection...
method to improve feature evaluation mechanisms by fusing the two algorithms. Combined with the gradient enhancement a MPIF-GBDT method is proposed, and then applied to the short-term electricity load prediction.

2. **The general of gradient boosting decision trees**

GBDT is an algorithm that uses decision tree as a base model to improve accuracy by optimizing gradients. GBDT estimates a prediction value with the currently trained strong learner and then computes the negative gradient value of its loss function against the strong learner, adding the corresponding loss function value of this weak learner's strong learner if the next weak learner's predicted value points in the direction of that negative gradient [7].

Currently, the gradient boosting decision tree integration algorithm is one of the most effective algorithms in supervised learning, it has a very strong robustness to outliers and can handle data with missing features very well [8]. However, simply using the GBDT model for short-term electricity load prediction will decrease the quality of the model since it neglects the influence of feature factors on the model. In order to have better feature input for the model, this paper proposes a load prediction algorithm for MPIF-GBDT based on the multifeature pool importance fusion.

3. **Multifeature pool importance fusion based GBDT (MPIF-GBDT) algorithm**

3.1. **Multiple pool feature importance algorithmic framework**

The multifeature pool importance fusion method reorganizes the importance values of features obtained by different feature selection methods to enhance quality features and remove undesirable features. MPIF-GBDT consists of two parts: the multifeature pool importance fusion and gradient boosting decision tree. The former extracts the importance values of different feature selection methods and recombines them by constructing importance vectors, importance matrices and weight assignments, and the latter sends the obtained quality features to the prediction engine to improve the prediction accuracy.

The overall process description of MPIF-GBDT is shown in Figure 1, and the process chart illustrates the specific method and process of integrating MPIF-GBDT and feature importance values.

![Figure 1. MPIF-GBDT overall process chart.](image)

3.2. **MPIF-GBDT algorithm**

The main algorithmic process of MPIF-GBDT is as follows steps:

**Step1** Calculating Feature Significance Values. The dataset is used to obtain feature importance values $I_i = \{I_{i1}, I_{i2}, I_{i3}, \ldots, I_{in}\}$ by different feature selection methods, respectively, where $i$ is the feature selection method, $S_n$ is the total number of features, $I_{ij}$ denotes the importance of the $j$-th feature after adopting the $i$-th feature selection algorithm, if the value is higher, it means that the relevance and importance of this feature is higher.

**Step2** Data alignment. It is necessary to perform data alignment based on discrepancy standardization by the importance values and importance scores respectively to avoid the inability to perform data analysis due to the different orders of magnitude of the importance values and importance scores. Specifically, by calculating the ratio of feature importance values minus their minima and their moving range, they can be mapped in a space of 0 to 1. This allows feature
importance value of different levels of magnitude to be processed and analyzed on the same domain, and finally obtain $\mathcal{T}_i \{I_{i1}, I_{i2}, \ldots, I_{in}\}$, where $\mathcal{T}_i$ is importance values of different feature selection algorithms after data alignment, $i$ is the feature selection method and $j$ is the corresponding feature.

**Step 3** Determine the number of features. To determine the number of features, this method sets the truncation coefficient $\mathcal{S}$ to give feature pool a proper range and avoid overfitting due to high dimensionality to improve generalization capabilities, which is the basis for selecting the number of features. The truncation coefficient is set according to the characteristics of the feature selection algorithm and the total number of features. Equation (1) expresses the calculation of the truncation algorithm:

$$
\mathcal{S}_{\text{num}} = \mathcal{S}_{\text{all}} \times \mathcal{S}
$$

where $\mathcal{S}_{\text{num}}$ is the number of features after feature selection for the $i$-th feature selection method. For the same dataset, the values $\mathcal{S}_{\text{num}}$ are the same.

**Step 4** Create a importance matrix. The importance values of all features are written in matrix form and computed for all feature selection algorithms. $\mathcal{I}_\omega$ is the importance value of all features computed by all feature selection algorithms as shown in Equation (2), which called the importance matrix.

$$
\mathcal{I}_\omega = \begin{bmatrix}
I_{1,1} & I_{1,2} & \cdots & I_{1,n} \\
I_{2,1} & I_{2,2} & \cdots & I_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
I_{n,1} & I_{n,2} & \cdots & I_{n,n}
\end{bmatrix}
$$

where $I_{ij}$ denotes the importance value of the $j$-th feature under the $i$-th feature selection method after feature alignment, and $n$ is the total number of feature selection methods.

**Step 5** Update the importance matrix. In order to reduce the weight of trivial features, the first $\mathcal{S}_{\text{num}}$ features with the highest importance value among the different feature selection algorithms are selected separately as a pool of different initial importance features. These features are considered high quality. The minimum values of feature importance for each of these high-quality features are defined as $I_{\text{num}}$. If the importance value of a feature computed by a feature selection algorithm is less than this feature selection algorithm's $I_{\text{num}}$, then define the importance value of this feature as 0. In this way, the weights of different features are redistributed, and the weights of low importance features are greatly reduced. And this subsection can be illustrated by Equation (3):

$$
I_{\text{update}} = \begin{bmatrix}
I_{1,1} & I_{1,2} \geq I_{\text{num}} \\
0 & I_{1,2} < I_{\text{num}}
\end{bmatrix}
$$

where $f$ is a row vector that represents the importance value of all features after feature fusion. Remove all features with zero eigenvalues of importance and define $\mathcal{I}$ as a vector of retained eigenvalues of importance, and also the candidate pool of highly important features after a feature sorting process, each corresponding to a highly important feature with k number of features.

**Step 6** Feature filter. In the pool of highly important feature candidates, the relative weights of each feature are calculated and ranked according to its feature importance value. In this paper, RFE and tree model feature selection methods are used, so $n$ takes the value of 2.

**Step 7** Initial learner. The squared error function L is chosen as the loss function, which is

$$
L(y, f(x)) = \frac{1}{2}[y - f(x)]^2
$$

Then input the training sample $U = \{(x_i, y_i) | i = 1, 2, \ldots, N\}$, and build the first CART:

$$
I_{\gamma} = \arg \min_\gamma \sum_{i=1}^N L(y_i, \gamma)
$$

where $\gamma$ is the constant that minimizes the loss function L.
**step 8** Calculate negative gradients and iterate. Compute the negative gradient for the loss function. Set the maximum number of iterations to $M$. Each round of iterations produces a weak learner, and each learner is trained on the residuals of the previous round of learners to obtain a strong learner.

To prevent overfitting, add the shrinkage parameter $v$ to regularize it and update the strong learner. In this experiment, $v$ takes 0.55.

**Step 9** Get the strong learner. Then get the strong learner.

$$f(x) = f_0(x) = f_0(x) + \sum_{m=1}^{D} \left[ \nu \sum_{l=1}^{D} \gamma_{ml} I(x \in R_{ml}) \right]$$

### 4. Experimental parameter setting and data pre-processing

This paper selects weather data, time data, and historical energy consumption data of Harvard campus buildings for nearly three years as the dataset.

Except for the feature selection algorithm, which has a particular bias for the dataset, the choice of the truncation coefficient is generally based on experience. There are 18 features in this experiment, and the truncation coefficient is taken as 0.3.

In order to have an objective evaluation of the model’s predictive performance, four model indicators are used in this paper, which are RMES, $R^2$, MAE, MAPE.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \times 100$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}$$

Here, $Y_i$ is the real value, $\hat{Y}_i$ is the predicted value, and $n$ is the number of samples used for training.

### 5. Experimental results and analysis

This paper used RFE and tree model feature selection algorithms for feature fusion and weight assignment. The original feature pool of this experiment contained only features related to meteorological factors, and the load changes are related to a variety of elements. The humanistic feature elements such as hour timestamp of the day, day timestamp of the week, day timestamp of the year, week timestamp of the year, and daytype are added to the pool to optimize the composition of the original feature pool.

![Figure 2. Feature importance value analysis comparison chart.](image-url)

Figure 2 shows the distribution of feature importance values after discretization standardization after the RFE and tree model feature selection methods. As a result, different feature selection algorithms differ in the features they consider more important due to their algorithms. Combination of different feature selection algorithms allows the integration of features that was considered to be
important by each different algorithm, thereby a pool of features of higher quality is refined compared with single feature based algorithm.

Table 1 exhibits the weights of the features after feature sorting.

| Item        | Significant feature | Insignificant feature |
|-------------|---------------------|-----------------------|
|             | occupancy           | cosHour               | daytype | day | weekend | coolingdegree | hour |
| Percentage (%) | 24.5 | 17.2 | 16.9 | 15.7 | 10.8 | 8.8 | 6.1 |

The current research results show that the integrated prediction model is generally better than the single-algorithm prediction model [9], so both the experimental group and the control group in this experiment used the integrated prediction model. The result is shown in Figure 3, where, the black line represents the actual electricity consumption, the red and blue lines represent the prediction results after MPIF-GBDT and MPIF based on extremely randomized tree regression (MPIF-ETR), and the green and yellow lines represent the prediction results after GBDT and extremely randomized tree regression algorithm (ETR). It is obvious from Figure 3 that whichever integrated prediction model is used, the prediction results of multiple pool importance fusion model are closer to the actual load consumption, which indicates that the multiple pool importance fusion can extract better features.

Figure 3. Comparison of load prediction of different models and feature selection algorithms.

To further validate the performance of the MPIF-GBDT model, this paper experimentally compares the performance of different integrated algorithm models, and the experimental results are shown in Table 2 and Table 3. It can be seen from the table that MPIF-GBDT performs better than the other models, in which the MPIF based model significantly outperforms the model without feature importance fusion in terms of prediction performance.

Table 2 gives performance evaluation for the different algorithms and models. As can be seen in Table 2, the RMSE of the integrated prediction model using MPIF is on average 56.5% lower than that without it, while $R^2$ is on average 22.7% higher. Table 2 illustrates in general that the MPIF can improve the model prediction accuracy better, while the GBDT integration model has better model performance than the ETR integration model.

Table 3 presents the percentage of cases when error falls into the specified range for different method. In the <1% percentage range, MPIF-GBDT improved by 4.9% over MPIF-ETR, while the prediction models using MPIF all reached 100% in the <25% percentage range.

Table 2. Performance evaluation for the different algorithms and models.

| Item       | R2   | MAPE | MSE  | RMSE |
|------------|------|------|------|------|
| MPIF-GBDT  | 0.9666 | 5.52 | 251.50 | 15.86 |
| MPIF-ETR   | 0.9495 | 6.38 | 380.23 | 19.50 |
| GBDT       | 0.7835 | 14.43 | 1630.39 | 40.38 |
| ETR        | 0.7786 | 13.90 | 1667.56 | 40.84 |

Table 3. Percentage of cases when error falls into range.

| Relative Error | Method       | <1%  | <2.5% | <5%  | <10% | <25% |
|----------------|--------------|------|-------|------|------|------|
| Percentage of cases when error falls into range | MPIF-GBDT | 13.6% | 30.0% | 56.2% | 85.0% | 100.0% |
|               | MPIF-ETR    | 8.7% | 31.3% | 50.0% | 73.7% | 100.0% |
|               | GBDT        | 6.2% | 16.3% | 23.7% | 41.2% | 85.0% |
|               | ETR         | 5.0% | 16.2% | 27.5% | 46.2% | 86.2% |
Figure 4 shows the scatter plots of different integrated regression models, where (a) is the MPIF-GBDT model, (b) is the GBDT model, (c) is the MPIF-ETR model and (d) is the ETR model. From the scatter plot, it can be seen that the MPIF-GBDT model has a better prediction accuracy as the prediction value is closer to the actual consumption value with a smaller fluctuation.

6. Conclusions
In this paper, a multifeature pool importance fusion scheme and a MPIF-GBDT model are proposed, and after analyzing the experimental results, the following conclusions can be drawn:

(1) Compared with the traditional feature selection algorithm, the proposed multifeature pool importance fusion method draws the advantages of different feature selection algorithms and the filtered features are of higher quality. The integrated prediction model with MPIF reduces the RMSE by 56.5% on average than that without it, which is a significant improvement.

(2) The GBDT integrated regression model has a relatively high prediction accuracy, with an average reduction of 2.05% in predicting RMSE over the extremely randomized tree regression integrated model.

(3) The MPIF-GBDT model with the lowest mean absolute percentage error, root mean square error and mean absolute error has relatively good properties, and it is the best model with the best prediction performance in this experiment.

Acknowledgments: The authors thank the editors and the anonymous reviewers for their careful works and valuable suggestions for this study. This study is supported by the National Natural Science Foundation of China (No.61673270 and No. 61203224).

References
[1] Z Wang and R S Srinivasan 2017 A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renewable and Sustainable Energy Reviews. J 75 796-808.
[2] F He, J Zhou, L Mo, K Feng, G Liu and Z He 2020 Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest Applied Energy. J 262 114396.
[3] L Wang, S Lv and Y Zeng 2018 Effective sparse adaboost method with ESN and FOA for industrial electricity consumption forecasting in China Energy. J 115 114396.
[4] U M Khaire and R Dhanalakshmi 2019 Stability of feature selection algorithm: A review. Journal of King Saud University–Computer and Information Sciences. J (available online).
[5] S Rathee and S Ratnoot. 2020 Feature selection using multi-objective CHC genetic algorithm Procedia Computer Science. J 167 1656-64.
[6] J Pérez-Rodriguez, A Haro-Garcia, J A Castillo and N Garcia-Pedrajas. 2018 A general framework for boosting feature subset selection algorithms Information Fusion. J 44 147-75.
[7] S Deng, C Wang, M Wang and Z Sun. 2019 A gradient boosting decision tree approach for insider trading identification: An empirical model evaluation of China stock market Applied Soft Computing Journal. J 83 105652.
[8] V Truong, Q Vu, H Thai and M Ha 2020 A robust method for safety evaluation of steel trusses using Gradient Tree Boosting algorithm Advances in Engineering Software. J 147 102825.

[9] F Zhai, Q Wen, Z Yang and Y Song. 2010 Hybrid forecasting model research on stock data mining Conference 4th International Conference on New Trends in Information Science and Service Science. Expert Systems with Applications (Gyeongju: South Korea) p 630.