

A learning framework based on weighted knowledge transfer for holiday load forecasting

Pan ZENG¹, Chang SHENG¹, Min JIN¹



Abstract Since the variation pattern of load during holidays is different than that of non-holidays, forecasting holiday load is a challenging task. With a focus on this problem, we propose a learning framework based on weighted knowledge transfer for daily peak load forecasting during holidays. First, we select source cities which can provide extra hidden knowledge to improve the forecast accuracy of the load of the target city. Then, all the instances which are from source cities and the target city will be weighted and trained by the improved weighted transfer learning algorithm which is based on the TrAdaBoost algorithm and can decrease negative transfer. We evaluate our method with the classical support vector machine method and a method based on knowledge transfer on a real data set, which includes eleven cities from Guangdong province to illustrate the performance of the method. To solve the problem of limited historical holiday load data, we transfer the data from nearby cities based on the fact that nearby cities in Guangdong province have a similar economic development level and similar load variation pattern. The results of comparative experiments show that the forecasting framework proposed by

this paper outperforms these methods in terms of mean absolute percent error and mean absolute scaled error.

Keywords Load forecasting, Holiday effect, Sparse data, Weighted transfer learning

1 Introduction

Short-term power load forecasting (STLF) is one of the key technologies to achieve the observability and controllability of a strong smart grid. The ever-increasing supply of distributed power, the use of renewable energy, and the establishment of new grid systems have raised an urgent need for accurate and effective load forecasting. During the past few decades, several STLF methods, including traditional and artificial intelligence methods, have been proposed [1, 2]. Traditional techniques, including linear regression, auto regressive moving average (ARMA) [3], grey model [4], Kalman filter-based methods, etc., use a mathematical model to map the features to the load consumption so as to make predictions. However, the relationship between fluctuation of load and exogenous factors is complex and nonlinear, which makes it extremely difficult for us to build a precise model. In recent years, artificial intelligence methods, such as fuzzy regression models [5], ensemble forecasting method [6], artificial neural network (ANN) [7, 8] and support vector machines (SVMs) [9, 10], have been widely used to forecast electricity loads. ANN is based on multilayered perceptrons and has a good performance on time series forecasting. Several different types of ANNs, including radial basis function (RBF) neural networks [11], back propagation (BP) neural network and fuzzy neural networks [12] are also used in load forecasting. However, the computational

CrossCheck date: 28 May 2018

Received: 20 October 2017/Accepted: 28 May 2018/Published online: 4 September 2018

© The Author(s) 2018

✉ Min JIN
jinmin@hnu.edu.cn

Pan ZENG
beckin@126.com

Chang SHENG
545154622@qq.com

¹ College of Computer Science and Electronic Engineering, Hunan University, Changsha, China



speed of ANN is not fast enough, and its solution is sometimes easily trapped into a local optimum. The SVM, which was proposed by Vapnik, is a novel powerful machine learning method based on statistical learning theory. It can effectively avoid the overfitting problem in ANN. Since the SVM problem is a convex quadratic programming problem, it can always obtain a unique and globally optimal solution. SVM was originally designed for classification problems and has achieved good performances. With the introduction of Vapnik's insensitive loss function, a modified SVM, which is called support vector regression (SVR) was proposed. Recently, SVR has been applied to various applications, such as battery remaining useful life estimation [13], natural gas demand prediction [14], etc., and has shown excellent performance. SVR has also been widely used in load forecasting.

Lin [15], the champion of the EUNITE Competition 2001, has confirmed the superiority of the SVR. However, the forecasting model proposed by Lin does not perform well for holidays. This is primarily because holiday trends are different from those of non-holidays and the amount of data on holidays is sparse. Ehab E. Elattar [16] presented an locally weighted support vector machine regression (LWSVR) model to predict load based on a time series. Hu [17] proposed a model selection method for SVR in short-term load forecasting. Che [18] proposed a kernel function selection method for SVR in the short-term load forecasting problem. Hong [19] developed a method that used the 24 solar terms calendar to categorize days of a year for load forecasting instead of the Gregorian calendar. However, these methods failed to pay attention to the load forecasting for holidays and the performance for holidays is yet to be improved. The loads of holidays are usually less predictable than those of non-holidays. On the one hand, the load variation pattern of holidays is quite different from that of non-holidays, because of the great change in human activity. On the other hand, the historical load data of holidays is always limited. This is called the problem of sparse data. Therefore, in order to increase the load forecasting accuracy of holidays, both the problems mentioned above should be well considered and solved. In the Global Energy Forecasting Competition (GEFCom2012) [20], some competitors had an emphasis on holiday load forecasting. Reference [21] treated 6 special holidays as weekend days: Memorial Day, Labour Day, Thanksgiving Day and the day after, New Year's Day, Independence Day and Christmas Day and improved the forecasting accuracy. However, since other public holidays are not treated as weekend days and the dates of holidays are different in other areas, this method should be modified when used in such areas. Reference [22] modeled the holiday effect by using a factor variable, which set the day before holiday as a non-zero value and holiday as a zero value. While the variable on other days was set as another different value. This

method divided days into three categories: holidays, the days before holidays and other days. This improved the load forecasting accuracy. Reference [23] analyzed the difficulties in holiday load forecasting and proposed a series of approaches to deal with public holidays and weekends, and further drew a conclusion that the most promising approach is to add public holiday dummies to the model but set the weekday dummies at the holidays to zero. With this approach the load forecasting accuracy during holidays is greatly increased. These works are significant for research on holiday load forecasting and are using different approaches to solve the first problem of holiday load forecasting. However, they failed to consider the second problem, namely, the problem of sparse data. Reference [24] reveals that the availability of historical holiday load data has an impact on the choice of holiday modeling technique, and that limited historical data leads to the limitation of the choice of the holiday modeling methods. That is to say, in order to improve the forecasting accuracy on holidays, just marking the holidays by using different variables is not sufficient. More attention should be paid to solve the problem of sparse data.

Transfer learning could solve the problem of sparse data by using the knowledge learned from some source tasks and finally improving the learning process of the target task. Transfer learning has been successfully used in many fields, such as natural language processing, face recognition, load forecasting etc., and even supports cross-domain knowledge transfer. In load forecasting, the target task is to forecast the load of the target city, and source tasks are that of source cities, i.e. cities that can provide hidden knowledge to improve the forecast accuracy of the load of the target city. Zhang [25] proposed a method based on transfer learning for load forecasting. With the use of extra knowledge transferred from other cities, the overall forecast accuracy has been improved. However, transferring all load data from source cities results in two issues. First, a large amount of noisy data could be imported, which has a negative impact on the forecast result. Second, since the influence of the data from the target city and source cities is different, we can't treat these two kinds of data equally.

In this paper, a learning framework based on weighted knowledge transfer for holiday load forecasting is proposed. First, a new holiday feature is introduced to indicate the turning point of holiday loads, thus solving the first problem of holiday load forecasting. Moreover, by using weighted transferring historical holiday data from source cities to the target city, the sparse holiday data sets can be appropriately enriched and the load forecasting accuracy of holidays can be improved without influencing the load forecasting performance of non-holidays. Lastly, a forecasting framework named HWT-SVR (weighted transfer learning of holidays), which can give different weights to holiday load data from different sources, is proposed.

Different weights reflect the degree of relevant influence of the different training samples from the target city and source cities on the holiday load data from the target city and embody the essential attributes of different load data. In this way, we can get a more accurate forecasting model. Furthermore, with the use of the improved TrAdaBoost algorithm we proposed, we solve the negative transfer phenomenon that occurs in certain cities. Figure 1 shows the flowchart of the proposed forecasting framework.

The rest of the paper is organized as follows. Section 2 analyzes data preprocessing in detail, as well as feature selection and data set selection. Section 3 proposes the method based on weighted knowledge transfer of holidays. Section 4 describes the whole procedure of the proposed forecasting framework in this paper. Section 5 illustrates the forecast performance of this framework with some case studies. Section 6 concludes the paper.

2 Data processing and selection

This section primarily covers four aspects: motivation of the study, load data preprocessing, feature selection and data set selection. After these steps, the model can more accurately capture the load variation.

2.1 Study motivation

The variation of load on holidays is dramatically different than on non-holidays. Taking China as an example, the overall electric load during the Spring Festival

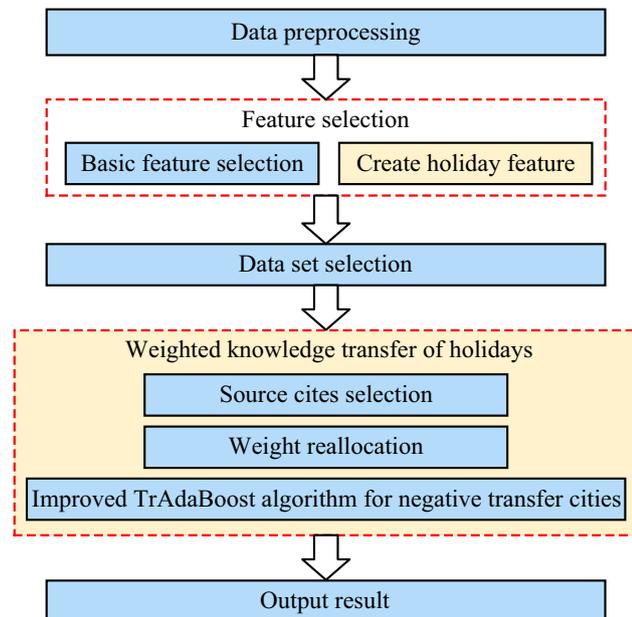


Fig. 1 Flowchart of the proposed forecasting framework

represents the valley of the year because of the shutdown of large-scale productive activities. Therefore, the accuracy of load forecasting on holidays is always lower than on non-holidays, and it's more difficult to predict holiday load especially when there are fewer historical load data on holidays than on non-holidays. Moreover, the traditional holidays in China are celebrated by lunar calendar, so they are not fixed to date, and this makes it more challenging to predict the load on holidays. Improving the forecasting accuracy on holidays will greatly improve overall forecasting accuracy. In China, there are four important and official holidays: New Year's Day, Spring Festival, May Day and National Day, as shown in Fig. 2. We classified these four holidays into four types. The Spring Festival is celebrated according to lunar calendar, which makes it different from the other three holidays. The New Year's Day holiday only lasts three days and we classified it into another type. The May Day holiday is in summer and lasts seven days, and the National Day is in autumn. Apparently, the load during holidays is lower than on non-holidays.

2.2 Data preprocessing

Noise and missing data can affect the accuracy of a forecast. In this work, Pauta criteria [26] have been used to analyze the load data. Pauta criteria are used to detect exceptional data. Assume x_i is a series of data, a data point satisfying (1) will be treated as exceptional data.

$$|x_i - \bar{x}| > 3\sigma \tag{1}$$

where \bar{x} represents the mean value of x_i and σ represents the standard error.

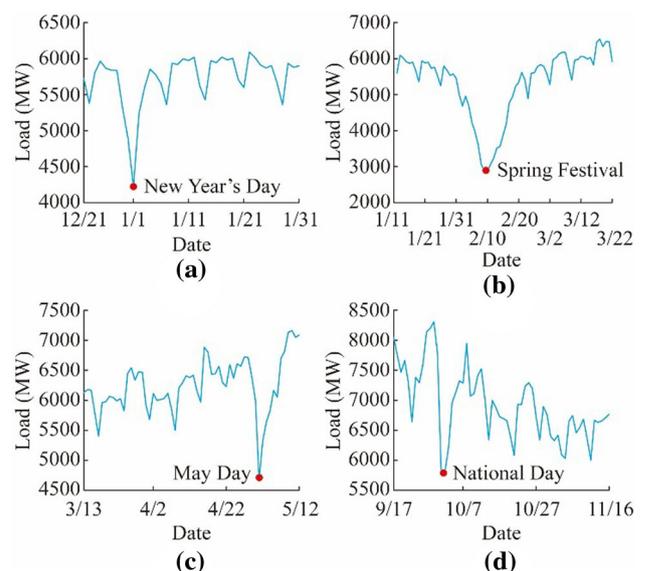


Fig. 2 Holiday load data



This study used the method of linear interpolation [4] to correct exceptional data points. For example, Fig. 3 shows the raw data of daily load consumption. It can be seen that there are some exceptional data and missing data. The actual daily peak load of one city after being processed is plotted in Fig. 4.

After dealing with exceptional data, all load values will be scaled by (2), where y' is the normalized load, y is the actual load, u is the average of the actual load, and σ is the standard deviation of the actual load.

$$y' = (y - u) / \sigma \quad (2)$$

2.3 Feature selection

Input variable selection is an extremely important step in load forecasting, which directly affects forecasting performance [27, 28]. In this paper, we select features primarily according to the analysis of load variation trends.

2.3.1 Basic feature selection

To make use of the periodicity of the load trend, the past load values up to 7 days are selected as a part of the basic

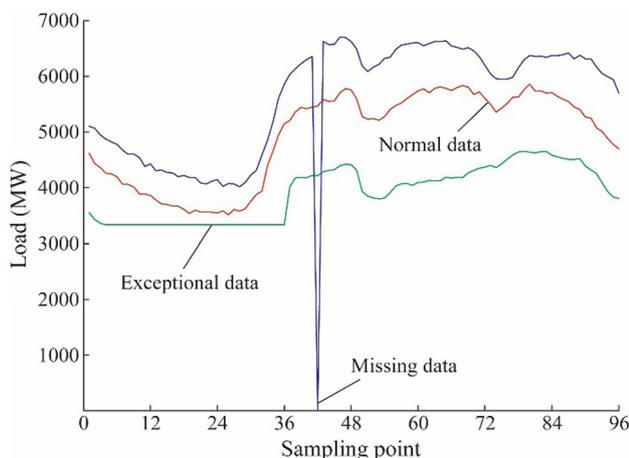


Fig. 3 Daily load data before processing

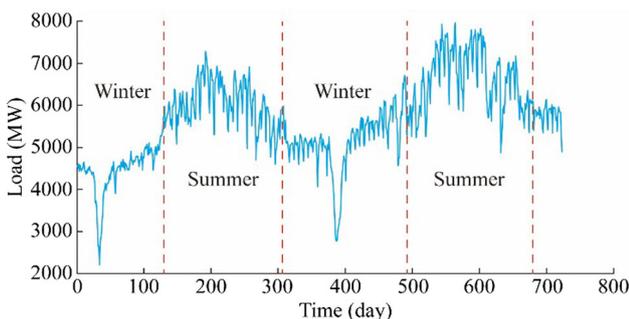


Fig. 4 Peak load of one city after being processed

features. In order to let the algorithm capture the regulation of the load trend more accurately, we use seven Boolean variables as the weekday and holiday indicator. For example, we set the second Boolean variable to true for Tuesday and all first six variables to false for Sunday.

2.3.2 Create holiday feature

The load value of a holiday is difficult to forecast accurately, not only because the data of holiday is too sparse compared with other days, but also the variation of the trends of a holiday is large. Figure 2 shows all four types of holiday. It is obvious that there is a turning point in each holiday.

Since the pattern of the trend of load on holiday is fixed relatively during each year, the turning point of each holiday is fixed too. To make the algorithm perform better at these turning points, we add another Boolean variable which is used to indicate the turning point of the holiday.

Finally, the total number of features is 15 in our work, as described in (3).

$$(W_1, W_2, \dots, W_6, H_1, H_2, L_1, L_2, \dots, L_7) \quad (3)$$

2.4 Data set selection

The summer load is significantly higher than that in winter, as shown in Fig. 4. To reduce the interference of different categories of data, the load can be forecasted with the data set of the forecasting day that corresponds to the season. Many methods exist to select the data set. The k-means clustering algorithm is the most frequently used in machine learning. The algorithm can automatically cluster a data set based on the distance of the data point to the objective function. The peak daily load is first clustered into two categories by the algorithm using the Euclidean distance function, and then, each month will be grouped into the corresponding season based on the number of days. The clustering results are shown in Fig. 5, where class 1 is winter and class 2 is summer. It is clear that January to April and November to December represent clusters for winter, whereas the other months are the summer clusters. To ensure the clustering result is closer to the situation of forecasting, we can take the load data within the last two years before the predicted day.

3 Weighted knowledge transfer of holidays

This section describes in detail the source cities' selection, weights' distribution based on HWT-SVR and the improved TrAdaBoost algorithm to solve the negative transfer problem.

In transfer learning, the input and output data observations of the source tasks and target task are connected by a hidden variable that means there is a latent or uncertain

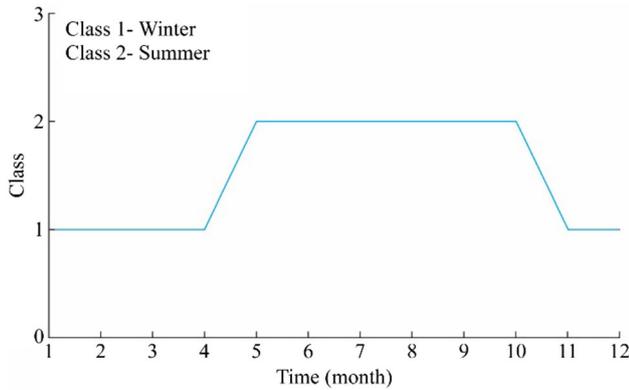


Fig. 5 Result of the k-means algorithm

relation between input and output variables, as shown in Fig. 6. Therefore, additional knowledge and information can be transferred from the source tasks to the target task, which improves forecasting performance. In load forecasting, the amount of holiday data is sparse, and the trends of the holidays and non-holidays are completely different; thus, it is difficult for the model to accurately forecast holidays. To solve this problem, the holiday data of relevant source cities are transferred to the target city.

However, there may be potential interference regarding the transfer of holiday data to the target city, as shown by the dashed lines of Fig. 6. The negative transfer may come from data of unrelated cities and related cities, where the distribution of certain load points is different from that of the target city. Therefore, selecting appropriate source cities and data of source cities are two key factors.

3.1 Source city selection

Source city selection plays a vital role in transfer learning. If the selection is inappropriate, it may lead to negative transfer.

For a set of candidate cities $T = \{T_k | k = 1, 2, \dots, m\}$, if the i^{th} city T_i is the target city, the goal of the source city selection step is to select source cities from the remaining

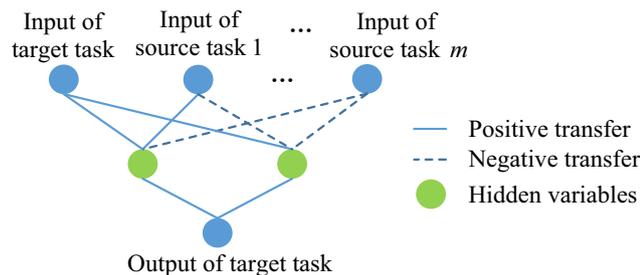


Fig. 6 Target task, source task and negative transfer in transfer learning problems

$m - 1$ cities. Thus, the number of possible combinations is 2^{m-1} . Exhaustive search on such a large number of candidate combinations is impossible. In order to overcome this problem, source cities are selected in the following way.

First, the similarity between target city and each candidate source city, i.e. the similarity between the load trend of the target city and the load trend of each candidate source city needs to be established. We use these similarities to represent source city selection priority. For a set of candidate source cities $\{T_1, T_2, \dots, T_m\}$, let $\{(x_t^{(j)}, y_t^{(j)}), t = 1, 2, \dots, n\}$ be the original training data set of city T_j , where $x_t^{(j)}$ is the attribute, i.e. the feature part of the instance of j^{th} candidate source city on time point t ; $y_t^{(j)}$ is the label part of the instance. Thus, we can use the similarity between $y^{(i)}$ and $y^{(j)}$ to represent the similarity between city T_i and T_j , here $y^{(i)}$ can be represented as (4).

$$y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \dots, y_n^{(i)}) \tag{4}$$

The incidence degree between each point is defined in (5), where $|y_0^k - y_i^k|$ is the absolute difference between the source city load y_i^k and the target city load y_0^k , $\min_i \min_k |y_0^k - y_i^k|$ is the two-level minimum difference, which means the minimum difference among all points ($k = 1, 2, \dots, m$) and all of the source cities y_i ($t = 1, 2, \dots, n$), $\max_i \max_k |y_0^k - y_i^k|$ is the two-level maximum difference, which has a similar meaning to that of the two-level minimum difference, and ρ is the resolution coefficient, whose value ranges from 0 to 1 and is normally taken as 0.5.

$$\xi_i(k) = \frac{\min_i \min_k |y_0^k - y_i^k| + \rho \max_i \max_k |y_0^k - y_i^k|}{|y_0^k - y_i^k| + \rho \max_i \max_k |y_0^k - y_i^k|} \tag{5}$$

Integrating the incidence degree of each point, the incidence degree between two vectors is obtained as follows:

$$r_i = \frac{1}{N} \sum_{k=1}^N \xi_i(k) \tag{6}$$

Second, we need to determine the number of source cities. Too many source cities not only result in negative transfer, which reduces accuracy, but also increase the computational time. Too few cities will reduce the forecasting accuracy because there is not enough supplementary information. To take into account the forecast performance and computational efficiency, an optimization algorithm is required to determine the number of source cities to be used, which can be described by objective function (7), where m and n are the number of source cities to be selected and the number of candidate source cities, respectively, and $MAPE(m, n)$ is the mean absolute percentage error of HWT-SVR based on the m



source cities; λ is a tuning parameter that controls the trade-off between the number of source cities and $MAPE(m, n)$, where the first item minimizes the number of source cities selected to reduce the computational overhead. The second item requires that the number of possible source cities is as large as possible to optimize forecasting performance. The objective function is minimized by these two items such that the most appropriate number of source cities is determined. A previous paper [29] specifically discusses the principle of the optimization algorithm which is used to determine the number of source cities to be transferred.

$$\min_m f(m) = \lambda \frac{m}{n} + (1 - \lambda) \cdot MAPE(m, n) \quad (7)$$

In our research, the case study is based on 11 cities from Guangdong province. Guangzhou is the provincial capital of Guangdong province. Guangzhou and Shenzhen are the most two economically developed cities in Guangdong province. Since the electric load is affected by economic development, the load variation pattern of Guangzhou and Shenzhen are similar. For Dongguan and Foshan, in terms of economic development, they are several years behind Guangzhou and Shenzhen. Geographically, these 4 cities are not far from each other. Therefore, the load variation pattern of Dongguan and Foshan are similar to the load variation pattern of Guangzhou and Shenzhen a few years ago. As for Zhaoqing and Meizhou, their economies are not as developed as Guangzhou and Shenzhen, and their load variation pattern tends to be similar. Therefore, in practice, we could transfer knowledge learnt from source cities to target city.

3.2 Weights reallocation

Weighted support vector regression (WSVR) [16] is an improved support vector regression algorithm, which not only has the advantages of SVR such as the ability to reach the global optimal solution and good performance on a small amount of data, but also has the ability to weight each training instance. Thus, it can give more attention to the instances that have higher weight. The parameters of WSVR can be obtained by solving a quadratic programming problem with linear equality and inequality constraints.

Our aim is to forecast the target city, and the distribution of data is different between target city and each source city. To decrease negative transfer, we need to ensure the weight of instances of the target city is higher than the weight of instances of source cities. The weight of each instance of the target city, that is, the reference city, will be set to 1, and the weight of each instance of source cities is computed using the Pearson correlation coefficient as described in (8), where X and Y are the feature part of the instance of source city and target city, respectively; f is the dimension of the feature; and

\bar{X} and \bar{Y} are the average of each feature vector, respectively. The value ranges from 0 to 1; the larger the value, the more similar the features are and vice versa.

$$w_i = \frac{\sum_{k=1}^f (X_k - \bar{X})(Y_k - \bar{Y})}{\sqrt{\sum_{k=1}^f (X_k - \bar{X})^2} \sqrt{\sum_{k=1}^f (Y_k - \bar{Y})^2}} \quad (8)$$

3.3 Dealing with negative transfer

While useful information is transferred from source cities to target city, some irrelevant information will also be transferred to the target city, i.e. the negative transfer. This is not only because of the difference of trend, but also because of the difference of the load data distribution. TrAdaBoost [30] is an algorithm based on AdaBoost that can be used to reduce the influence of negative transfer. It will decrease the weights of those instances whose distributions are different from the data distribution of the target city. The algorithm process is given in Algorithm 1.

Algorithm 1 TrAdaBoost algorithm

Input: the two labeled data sets T_d , T_s and the unlabeled data set S , a base learning algorithm **Learner**, and the maximum number of iterations N . Initialize the initial weight vector, $w^1 = (w_1^1, w_2^1, \dots, w_{n+m}^1)$

For $t = 1, 2, \dots, N$ **do**

1. Set $t = 1, 2, \dots, N$
2. Call **Learner**, providing it with the combined training set T with the distribution p^t over T and the unlabeled data set S .
3. Calculate the error rate of ε_t on T_s

$$\varepsilon_t = \frac{\sum_{i=n+1}^{n+m} w_i^t |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}$$

4. Set

$$\beta = 1 / (1 + \sqrt{2 \ln(n/N)})$$

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t)$$
5. Update the new weight vector

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{h_t(x_i) - c(x_i)} & x_i \in \text{source task} \\ w_i^t \beta^{-|h_t(x_i) - c(x_i)|} & x_i \in \text{target task} \end{cases}$$

end for

Output: the weights of training data.

TrAdaBoost algorithm is designed for classification problems, whereas load forecasting is a regression problem. In practical application, the error rate in the 2nd step of TrAdaBoost should be less than or equal to 0.5 to ensure that it will gradually decrease as the number of iterations

increases. For classification problems, the result of $h_t(x_i)$ must be either 0 (negative sample) or 1 (positive sample). Thus, the aforementioned condition is easy to satisfy. In order to allow this algorithm to fit the regression problem, we change the equation in the 2nd step of the algorithm to (9), where $P_t(x_i)$ is the predicted load value; $r(x_i)$ is the real load value; and δ is an adjusted parameter that is less than 1. The actual meaning is that if the percentage predicted error of the sample is less than δ , it is considered to be correct; otherwise, it is an error. By setting the threshold, the algorithm can effectively reduce negative transfer. The value should be obtained by experiment.

$$\varepsilon_t = \frac{\sum_{i=n+1}^{n+m} w_i^t q(x_i)}{\sum_{i=n+1}^{n+m} w_i^t} \tag{9}$$

$$q(x_i) = \begin{cases} 0 & \text{if } \frac{|P_t(x_i) - r(x_i)|}{r(x_i)} < \delta \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

4 Proposed forecasting framework

The specific steps of the forecasting framework based on HWT-SVR are described as follows.

- Step 1:* Correct abnormal data and use (2) to scale the corrected data.
- Step 2:* Extract the label part of instances of data and construct vectors using (4) for all cities.
- Step 3:* Use (6) to calculate similarities between the target city and candidate source cities and rank the candidate source cities according to the similarities we figured out.
- Step 4:* Use optimization function (7) to determine the number of source cities selected; then, select the appropriate cities as the transfer cities according to the sorting order from Step 3.
- Step 5:* Create datasets according to (3), remove all non-holiday instances in source cities, and calculate the weights of each training instance according to (8).
- Step 6:* Use the k -means algorithm to partition the data set and train the HWT-SVR model with the weights of Step 5 and the corresponding data set.
- Step 7:* Predict the load of the target city using the HWT-SVR model in Step 6.
- Step 8:* Readjust the weights of the data set of the negative transfer cities with the improved TrAdaBoost algorithm; then, train the HWT-SVM model, and predict the load.

5 Prediction case

For the case study, input variables and forecasting model were presented in the above section. To illustrate the feasibility of the proposed method, we made three case studies with the daily peak load data of 11 cities from Guangdong province from 2005 to 2007. The data of 2005 and 2006 are set to be the training set and the data of 2007 are set to be the test set. For each month, the month before the predicting month in the same season is set to be the validation set. For example, when predicting the load of January of 2007, since January is in winter as we analyzed before, winter month from 2005 to 2006 are set to be the training set except for December 2006, which is the validation set. The value of λ and δ are respectively 0.1 and 0.25, obtained by experiment.

5.1 Some common mistakes

To assess the performance of the forecasting model, two accuracy measures, including MAPE and MASE (mean absolute scaled error), were used in this study. MAPE is a measure of the accuracy of the forecasting model, where the smaller the value is, the more accurate the forecasting is. The MASE is a scaled error that is scaled by a naive forecast value, which is less than one if the forecast is better than the naive method, and the smaller the MASE value is, the better the forecasting performance is. The definitions of these metrics are shown as follows:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{L_i - \hat{L}_i}{L_i} \right| \times 100 \tag{11}$$

$$MASE = \frac{\sum_{i=1}^n |L_i - \hat{L}_i|}{\frac{n}{n-1} \sum_{i=2}^n |L_i - L_{i-1}|} \tag{12}$$

where L_i and \hat{L}_i are the actual and forecasted load demand value of the i^{th} sample, respectively; n is the forecast horizon.

5.2 Case 1

In this test case, we forecast the load of Guangzhou City. We compared our proposed method, i.e. HWT-SVR with three other methods. The first method, which is denoted as SVR, is proposed by Lin and helped his team win the EUNITE Competition 2001. The second method is T-SVR which is based on the method of transfer learning proposed by Zhang. The third method, denoted as HF-SVR, is based on the SVR model with the use of the holiday feature we proposed in Sect. 2. The forecasting results of these methods for



Guangzhou City in January are shown in Fig. 7. We can see that the holiday feature we proposed improves the forecasting accuracy of the holidays. Table 1 also presents the MAPE and MASE. It is clear that the HF-SVR model does not fully capture the holiday characteristics. Taking into account the fact that the holiday data are too sparse, we transfer the holiday data of source cities. The HWT-SVR result is represented by the red curve. We can see that the HWT-SVR model performs better than the other models; also, the MAPE and MASE are better than those of the others. On the other hand, from the T-SVR result, which is represented by the yellow curve, it is not difficult to see that the performance of the forecasting result for non-holidays decreased, caused by transferring the non-holiday data. Figures 8, 9 and 10 and Table 2 show the results of load forecasting in February, May and October, corresponding to Spring Festival, May Day and National Day. As shown in Table 2, our method gains the best performance in May and October, and the accuracy in February is slightly lower than that of SVR. Figure 11 shows the actual and predicted load and the percentage error of SVR, T-SVR and HWT-SVR. We can see from the figure that the percentage errors of SVR and T-SVR are not stable and fluctuate dramatically during holidays. The percentages of HWT-SVR are relatively steady and the fluctuation around holidays is not as obvious as that of SVR and T-SVR (Fig. 12).

5.3 Case 2

In this test case, we compared HWT-SVR with three other methods mentioned in the previous test case on other cities. The result is shown in Table 3. From the MAPE of each city, it can be seen that the proposed forecasting model exhibited an improved forecasting performance for most cities. However, the MAPE was more improved by the HF-SVR method for Meizhou City and less improved

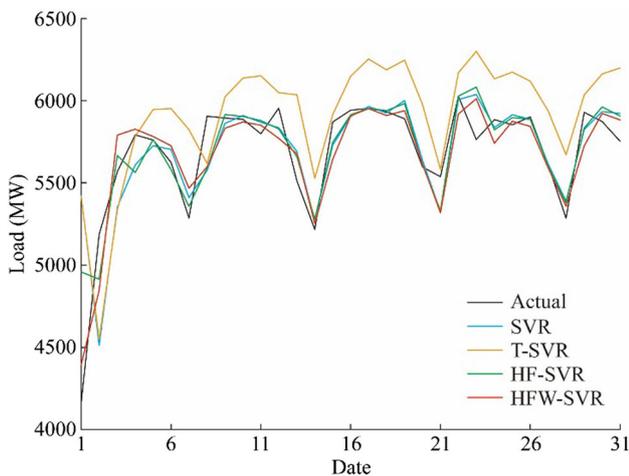


Fig. 7 Forecasted and actual peak daily load in January

Table 1 MAPE and MASE of January

| Method | MAPE (%) | MASE |
|---------|-------------|-------------|
| SVR | 3.50 | 0.74 |
| T-SVR | 3.19 | 0.67 |
| HF-SVR | 3.11 | 0.66 |
| HWT-SVR | 2.14 | 0.46 |

Note: Bold indicates the best performance of different methods

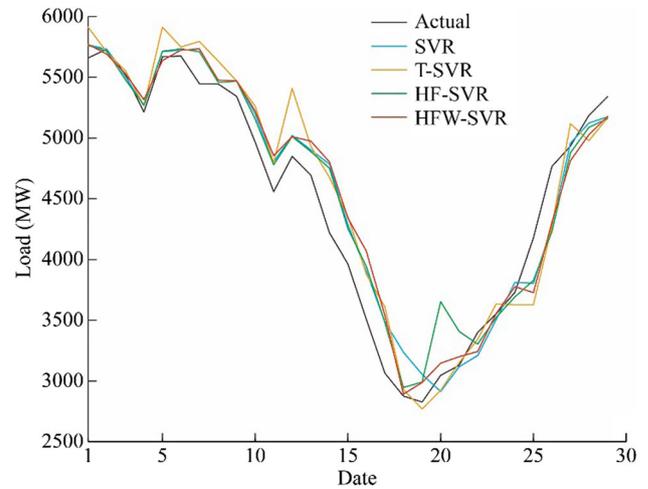


Fig. 8 Forecasted and actual peak daily load in February

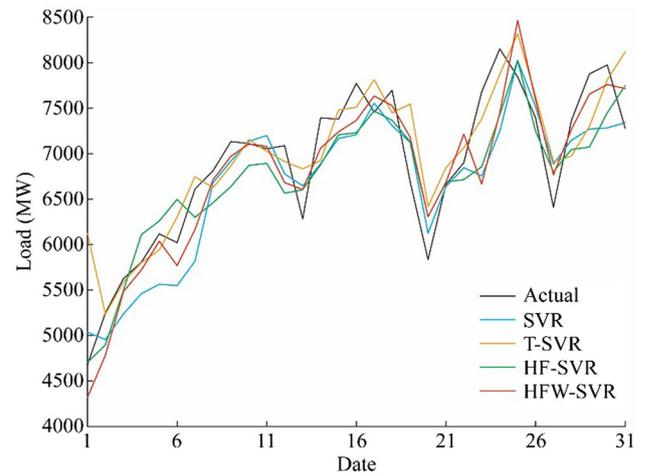


Fig. 9 Forecasted and actual peak daily load in May

for the HWT-SVR model, which was even lower than that of the SVR model. The reason is that negative transfer occurred for the city. In the next case, we will present the result of the method of the improved negative transfer. The numbers of source cities for each target city are also listed in the last column of Table 3 after applying the optimization algorithms. The number of transferred source cities in

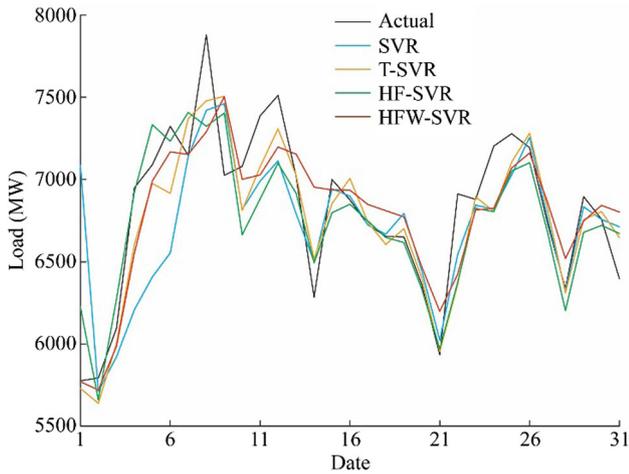


Fig. 10 Forecasted and actual peak daily load in October

Table 2 MAPE and MASE of October

| Month | Method | MAPE (%) | MASE |
|----------|---------|-------------|-------------|
| February | SVR | 4.71 | 0.75 |
| | T-SVR | 5.20 | 0.90 |
| | HF-SVR | 4.94 | 0.78 |
| | HWT-SVR | 4.72 | 0.76 |
| May | SVR | 5.26 | 0.78 |
| | T-SVR | 5.09 | 0.72 |
| | HF-SVR | 5.03 | 0.70 |
| | HWT-SVR | 4.36 | 0.65 |
| October | SVR | 3.48 | 0.73 |
| | T-SVR | 3.00 | 0.63 |
| | HF-SVR | 3.15 | 0.66 |
| | HWT-SVR | 2.33 | 0.49 |

Note: Bold indicates the best performance of different methods

each target city is not greater than 5; if an excessive number of source cities is selected, it will cause negative transfer. In addition, regarding a special case, the number of source cities transferred is only one for Foshan City. When additional cities were transferred, the MAPE and MASE decreased. The load distribution of the city is significantly different from that of the others. Therefore, we can conclude that if the load distribution is clearly different, it is a negative result of transfer learning. However, this situation is rare, and thus, we can apply the method.

5.4 Case 3

In this test case, we tested THWT-SVR, which combines HWT-SVR and the proposed improved TrAdaBoost algorithm on Meizhou city to evaluate its ability to deal

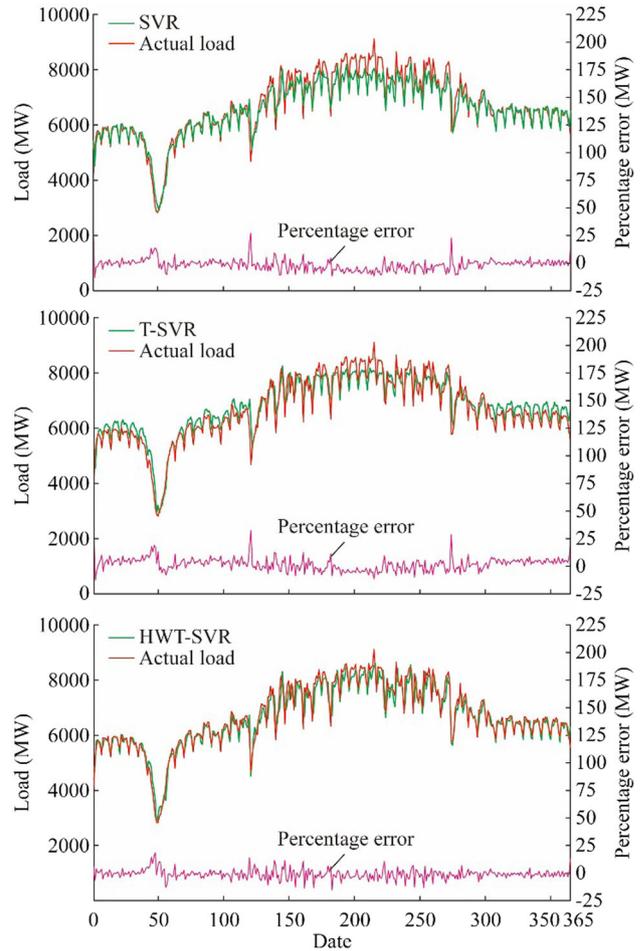


Fig. 11 Result of 2007

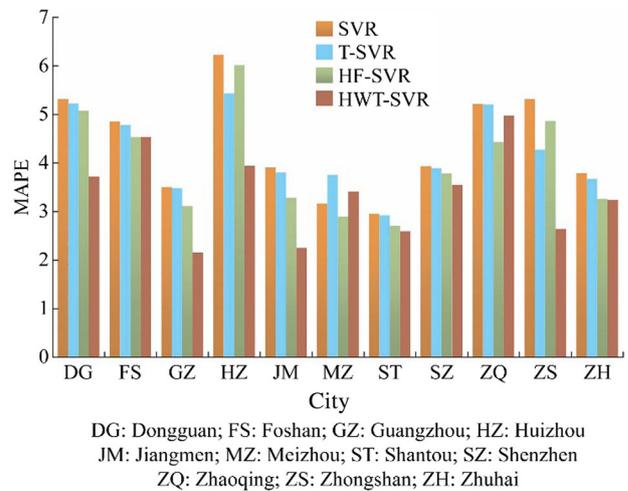


Fig. 12 MAPE results of January of each task city with the comparison methods

with negative transfer. The results are shown in Table 4. As observed from the MAPE and MASE of Table 4, the



Table 3 MAPE and MASE of October

| Target city | SVR | T-SVR | HF-SVR | HET-SVR | NSC |
|-------------|------|-------|--------|---------|-----|
| Dongguan | 5.31 | 5.22 | 5.07 | 3.72 | 3 |
| Foshan | 4.85 | 4.78 | 4.53 | 4.53 | 3 |
| Guangzhou | 3.50 | 3.48 | 3.11 | 2.14 | 5 |
| Huizhou | 6.22 | 5.43 | 6.01 | 3.94 | 4 |
| Jiangmen | 3.90 | 3.80 | 3.28 | 2.25 | 2 |
| Meizhou | 3.16 | 3.75 | 2.89 | 3.41 | 2 |
| Shantou | 2.95 | 2.92 | 2.70 | 2.59 | 1 |
| Shenzhen | 3.93 | 3.89 | 3.78 | 3.55 | 5 |
| Zhaoqing | 5.21 | 5.20 | 4.43 | 4.97 | 4 |
| Zhongshan | 5.31 | 4.27 | 4.86 | 2.64 | 4 |
| Zhuhai | 3.79 | 3.67 | 3.26 | 3.24 | 3 |

Table 4 MAPE and MASE with the improved TrAdaBoost algorithm

| Method | MAPE (%) | MASE |
|----------|-------------|-------------|
| SVR | 3.16 | 0.82 |
| HWT-SVR | 3.41 | 0.88 |
| THWT-SVR | 2.88 | 0.74 |

Note: Bold indicates the best performance of different methods

algorithm can reasonably allocate the weights of the data set to reduce negative transfer.

6 Conclusion

This paper proposes an STL approach based on weighted transfer learning of holidays. First, this approach abstracts a new holiday feature to indicate holidays. Secondly, the holiday data of source cities are transferred to the target city to appropriately enrich sparse holiday data and further aid in forecasting. According to the reliability of different data sources, the training data of the target city and source cities are allocated different weights to enhance forecasting performance. In addition to overcome negative transfer issues of a target city, the improved TrAdaBoost algorithm adjusts the initial weights based on the Pearson correlation coefficient to solve the problem.

We compared our method with SVR and T-SVR on more than a dozen of cities to illustrate its feasibility. The case studies show that the performance metrics were improved by the method proposed in this paper. From the comparative experimental analysis, the improvement is due to the new holiday feature, transfer learning of the holidays and the improved TrAdaBoost algorithm. The proposed method is suitable for a power system that has the load data of source cities, particularly for those where the accuracy

of holiday forecasting is low. In future work, we will consider more features to better increase the performance of forecasting, such as weather data, economic condition, climate, etc.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61773157), and in part by the National Scientific and Technological Achievement Transformation Project of China (No. 201255).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- [1] Hong T (2010) Short term electric load forecasting. <https://search.proquest.com/docview/852985857>. Accessed Jan 2010
- [2] Hong T, Fan S (2016) Probabilistic electric load forecasting: a tutorial review. *Int J Forecast* 32(3):914–938
- [3] Li YY, Han D, Yan Z (2018) Long-term system load forecasting based on data-driven linear clustering method. *J Mod Power Syst Clean Energy* 6(2):306–316
- [4] Jin M, Zhou X, Zhang ZM et al (2012) Short-term power load forecasting using grey correlation contest modeling. *Expert Syst Appl* 39(1):773–779
- [5] Song KB, Baek YS, Hong DH et al (2005) Short-term load forecasting for the holidays using fuzzy linear regression method. *IEEE Trans Power Syst* 20(1):96–101
- [6] Zhou M, Jin M (2017) Holographic ensemble forecasting method for short-term power load. *IEEE Trans Smart Grid*. <https://doi.org/10.1109/TSG.2017.2743015>
- [7] Rana M, Koprinska I (2016) Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* 182:118–132
- [8] Khotanzad A, Afkhami-Rohani R, Maratukulam D (1998) ANNSTLF-artificial neural network short-term load forecaster-generation three. *IEEE Trans Power Syst* 13(4):1413–1422
- [9] Ceperic E, Ceperic V, Baric A (2013) A strategy for short-term load forecasting by support vector regression machines. *IEEE Trans Power Syst* 28(4):4356–4364
- [10] Zeng P, Jin M (2018) Peak load forecasting based on multi-source data and day-to-day topological network. *IET Gener Transm Dis* 12(6):1374–1381
- [11] Ko CN, Lee CM (2013) Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter. *Energy* 49:413–422
- [12] Lou CW, Bong MC (2015) A novel random fuzzy neural networks for tackling uncertainties of electric load forecasting. *Int J Electr Power* 73:34–44
- [13] Patil MA, Tagade P, Hariharan KS et al (2015) A novel multistage support vector machine based approach for Li ion battery remaining useful life estimation. *Appl Energy* 159:285–297
- [14] Zhu L, Li MS, Wu QH et al (2015) Short-term natural gas demand prediction based on support vector regression with false neighbours filtered. *Energy* 80:428–436

- [15] Chen BJ, Chang MW, Lin CJ (2004) Load forecasting using support vector machines: a study on EUNITE competition 2001. *IEEE Trans Power Syst* 19(4):1821–1830
- [16] Elattar EE, Goulermas J, Wu QH (2010) Electric load forecasting based on locally weighted support vector regression. *IEEE Trans Syst Man Cybern C* 40(4):438–447
- [17] Hu ZY, Bao YK, Xiong T (2014) Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression. *Appl Soft Comput* 25:15–25
- [18] Che JX, Wang JZ (2014) Short-term load forecasting using a kernel-based support vector regression combination model. *Appl Energy* 132:602–609
- [19] Xie JR, Hong T (2018) Load forecasting using 24 solar terms. *J Mod Power Syst Clean Energy* 6(2):208–214
- [20] Hong T, Pinson P, Fan S (2014) Global energy forecasting competition 2012. *Int J Forecast* 30(2):357–363
- [21] Charlton N, Singleton C (2014) A refined parametric model for short term load forecasting. *Int J Forecast* 30(2):364–368
- [22] Ben Taieb S, Hyndman RJ (2014) A gradient boosting approach to the Kaggle load forecasting competition. *Int J Forecast* 30(2):382–394
- [23] Ziel F (2018) Modeling public holidays in load forecasting: a German case study. *J Mod Power Syst Clean Energy* 6(2):191–207
- [24] Xie J, Chien A (2016) Holiday demand forecasting in the electric utility industry. In: *Proceedings of the 2016 SAS global forum, Las Vegas, USA, 18–21 April 2016*, 13 pp
- [25] Zhang YL, Luo GM (2015) Short term power load prediction with knowledge transfer. *Inf Syst* 53:161–169
- [26] Zhao H, Min F, Zhu W (2013) Test-cost-sensitive attribute reduction of data with normal distribution measurement errors. *Math Probl Eng* 3:4928–4942
- [27] Hu ZY, Bao YK, Xiong T et al (2015) Hybrid filter-wrapper feature selection for short-term load forecasting. *Eng Appl Artif Intell* 40:17–27
- [28] Koprinska I, Rana M, Agelidis VG (2015) Correlation and instance based feature selection for electricity load forecasting. *Knowl Based Syst* 82:29–40
- [29] Che JX, Wang JZ, Tang YJ (2012) Optimal training subset in a support vector regression electric load forecasting model. *Appl Soft Comput* 12(5):1523–1531
- [30] Dai W, Yang Q, Xue GR et al (2007) Boosting for transfer learning. In: *Proceedings of the 24th international conference on machine learning, Corvallis, USA, 20–24 June 2007*, pp 193–200

Pan ZENG received the B.E. degree in building electricity and intelligence from Qingdao University of Technology, Qingdao, China, in 2012, and the M.S. degree in software engineering from Hunan University, Changsha, in 2016. He is currently working toward the Ph.D. degree in the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include machine learning, artificial intelligence, load forecasting, and their engineering applications.

Chang SHENG received the M.S. degree in software engineering from Hunan University, Changsha, China, in 2016. His research interests include machine learning, load forecasting, and their engineering applications.

Min JIN received the B.E. degree in automation and the M.S. degree in control theory and engineering from Central South University of Technology, Changsha, China, in 1995 and 1997, respectively, and the Ph.D. degree in control theory and engineering from Central South University, Changsha, in 2000. From August 2000 to August 2004, she was a Senior Engineer with Dongfang Electronics Information Industry Company, Yantai, China. From December 2010 to January 2012, she was a Visiting Scholar with the Department of Electrical and Computer Science, Georgia Institute of Technology, Atlanta, GA, USA. Since November 2004, she has been with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where she is currently a professor. Her research interests include artificial intelligence, big data and industry 4.0, load forecasting, and their engineering applications.

