Bagging Based Multi-Source Learning and Transfer Regression for Electricity Load Forecasting

Zhaorui Meng

Abstract—**Accurate electricity load forecasting is critical to power system operation. Prediction error of load forecasting can be greatly reduced by utilizing knowledge transferred from other related tasks. To further improve the effectiveness of transfer, knowledge can be transferred from multiple sources to increase the chance of finding samples closely related to the target. In this work, a multi-source instances transfer algorithm based on domain-to-domain similarity and sample to domain similarity is developed and a bagging-based re-sampling transfer regression framework is constructed. Experimental evaluation on a real-world dataset shows that forecasting performance can be significantly improved by transferring useful data from more sources. Negative transfer is avoided effectively.**

Index Terms—Load forecasting, bagging, multi-source, transfer learning

I. Introduction

Electricity load forecasting plays an important role in ensuring power system planning, reliability and economic operation [1]. As the rapid development of highly uncertain energy sources, such as solar and wind energy generation, accurate forecast can be a crucial issue in the management of electricity companies. Effective power load forecasting is also the basis of real-time electricity price, which requires the high efficiency of forecasting algorithm [2].

Based on the importance of power load forecasting, a lot of effective forecasting theories and techniques have been published by some scholars. In the early researches, classic statistic models, such as the AR (auto-regressive) [3], the ARMAX (auto- regressive moving average with external input) model [4]and the SS (state-space) model with Kalman filter [5] are used in the short-term power load forecast. In the past decade, techniques inspired by machine learning and artificial intelligence research such as Deep Belief Network (DBN) [6], Fuzzy-Neural Networks [7] and Support Vector Machine (SVM) [8] have also been applied to load forecasting.

Manuscript received June 3, 2021; revised March 25,2022.

Zhaorui Meng is a senior engineer in School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, Fujian,361024, China (Corresponding author; Email: mengzhaorui@ xmut.edu.cn).

Most of research works about load forecasting focus on predicting based on data from a single location. Transfer learning is a hot research topic in recent years. By transfer learning [9], the performance of the target task can be improved by using knowledge transferred from source tasks. Transfer learning has been applied into variety applications, such as: image classification [10], NLP (natural language processing) [11], collaborative filtering problems [12] and brain-computer interfaces [13].

In general, the ability to transfer knowledge from a source to a target depends on the relationship between source domain and target domain. The stronger the relationship is, the more knowledge can be transferred from source to target domain. On the other hand, if the relationship is poor, it may lead to the worse performance of target task, which is known as negative transfer [14]. To avoid this effect, one would have to answer the question “what to transfer”. One strategy is to import knowledge from multiple sources, instead of one. By this way, the chance to transfer useful. knowledge closely related to the target domain increases dramatically. The method proposed in this paper transfer instances to target domain from multiple sources by a multi-similarity approach.

In more detail, the main contributions of our work are as follows: to better utilize knowledge from multiple sources, we propose a multi-similarity method between source domain and target domain to avoid negative transfer. The relation between source domain and target is measured from two aspects: one is source domain to target domain, and the other one is sample of source domain to target domain. By multi-similarity measurement, the relationship between source domain and target domain can be explored effectively. A good foundation for further transfer learning is built. We propose a bagging-based transfer regression algorithm through resampling the combination of selected source data and target data. By experiment results on a real-world electricity dataset, comparing with AdaBoostRegressor, our proposed algorithm can improve the prediction performance 11% at most while choosing zone 17 as target location and both zone 7 and zone 8 as source locations. And negative transfer can be avoided effectively, compared with the other two transfer regression approaches. More specifically, while transferring data from two sources, Multi-Source TrBagging brought negative transfer in 5 out of 9 cases. And Two stage TrAdaBoost.R2 brought negative transfer in 7 out of 9 cases.

.

Negative transfer is avoided completely by the proposed algorithm.

The rest of this paper is organized as follows. Section II reviews methods related to our work. Section III describes the details of our proposed algorithm, multi-similarity measurements and resampling-based transfer regression algorithm. Section IV analyses the experiment results on a real-world dataset.

II. Background

Originally Dai et al. [15] proposed a boosting based transfer learning approach TrAdaBoost for classification problem. Later Pardoe and Stone [16] extended the approach to the cases of regression. Two transfer regression approaches are proposed: boosted transfer stacking and two-stage TrAdaBoost.R2. In the above related works, the examples reweighting method are used for boosting. However, after comparing the results of 10 ensemble algorithms with 4 learners on 15 datasets [17], it is believed that ensemble by resampling generally performs better than that by reweighting. A weighted-resampling-based transfer learning algorithm is proposed in work [18]. It shows outstanding performance compared with TrAdaBoost, an algorithm boosting by reweighting.

In transfer learning, negative transfer happens when the performance of learning in the target domain is reduced by the source domain data and task. To void the negative transfer, some research work has been published on this work. In [14], Rosenstein et al. showed that the performance of the target might be hurt by transfer learning if two tasks are too dissimilar. David et al. [19] proposed an approach analyzing the relatedness among tasks by task clustering techniques, which maybe a guidance on how to avoid negative transfer automatically. Knowledge transferred from multi-sources is one of the approaches reducing negative transfer. Yao et al. [20] adopted a multi-source boosting approach for classification problem, which reduce the negative transfer greatly as the number of sources increases. Huang et al. [21] proposed another multi-source boosting transfer learning approach, which shows better performance than single source transfer learning. Zhang et al. [22] proposed an instance transfer learning method based on multisource dynamic TrAdaBoost. The experiment result shows that the negative transfer is avoided well by transferring knowledge from multi-source.

In electricity load forecasting, it's very often that there is no enough data to build a reliable forecasting model. To overcome the problem, there are some research works applying transfer learning approach on load forecasting. Research from Zhang et al. [23] and Fiot et al. [24] proposed multi-task learning for load forecasting, which forecasting several locations simultaneously. The difference between the proposed method with multi-task learning is that we only care about target task, which is more natural in real world application. Zhang et al. [25] proposed a transfer learning approached based on Gaussian process. To reduce negative transfer, source task is selected based on Gaussian process. Wu et al. [26] introduced a boosting based multiple kernel transfer regression for electricity load forecasting. Different with above two methods, our approach transfer data from multi-sources instead of one source so that negative transfer can be avoided.

III. Instance Based Transfer Learning with Multi-Source Selection

The transfer learning method we proposed consists of two main stages. In the first stage, source data that can help learning target task are selected from multi-source. In this research, we use multi-similarity to measure distance between source tasks and target task. Multi-similarity includes domain-to-domain similarity and sample to domain similarity. Only samples selected by multi-similarity criteria can be transferred into target task. In the second stage, samples selected from source data are combined with target data. A set of bootstrapping samples from combined dataset are chosen to train a learner. Learners performed well on the target training data are selected as the final learned ensemble for the target task.

Formally, in transfer learning from multiple sources problem, there are M source domains and one target domain. Denote the s-th source domain$D^s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, x_i^s and y_i^s are the feature space and label of the i-th sample. N_s is the sample number of the s-th source. Target domain is denoted as $D^T = \{(x_i^T, y_i^T)\}_{i=1}^{N_T}$. N_T is the sample number of the target domain.

3.1. Multi-source data selection by multi-similarity

At first, we use Maximum Mean Discrepancy (MMD) to measure the similarity between target domain and source domain. According to [27], MMD is used to detect difference between two different distributions. MMD is easily to be computed in Reproducing Kernel Hilbert Space (RKHS).

Denote the similarity between the s-th source domain with target domain as DD_s. Based on definition of MMD, DD_s is

$$DD_s = \mathrm{MMD}(D^s, D^T) = \left\| \frac{1}{N_T}\sum_{i=1}^{N_T} \phi(x_i^T) - \frac{1}{N_s}\sum_{j=1}^{N_s} \phi(x_i^s) \right\|_H \quad (1)$$

where Φ is known as feature space mapping from original space to RKHS.

Denote average sample distance SD_{s_i} as average distance between the i-th sample from s-th source with N_k neighboring samples of target domain, which are the result of one of clustering algorithms.

$$SD_{s_i} = \frac{1}{N_k}\sum_{k=i}^{N_k} d(x_i^T, x_k^s) \quad (2)$$

The smaller $SD_{s_i}$is, the more similar the sample from source domain is with target domain.

Finally, the similarity weight W_{s_i} between the i-th sample from the s-th source domain with target domain is:

$$W_{s_i} = \alpha DD_s + \beta SD_{s_i} \quad (3)$$

where α and β are parameters to control effect of domain-to-domain similarity and sample to domain similarity respectively. Empirically, both α and β are set to 0.5 in this paper.

Algorithm 1. Source Data Selection

Input: Target training data D^T and all M source data D^s, α, β, n

Output: the set R_S of selected source instances

Step 1. for i =1,2,…,M do
 compute DD_s according to Eq. (1)
 end

Step 2. normalize DD_s to 1
Step 3. for i = 1,2,…,M do
for each instance (x_i^s, y_i^s)in D^s do
compute SD_{s_i} according to Eq. (2)
end
normalize SD_{s_i} to 1
compute sample weight W_{s_i} according to Eq. (3)
end
Step 4. select top n instances from all M source data D^s based on sample weight W_{s_i}.

By algorithm 1, similarities between source tasks and target task are measured by both domain-to-domain similarity and sample to domain similarity. More helpful data can be passed to the second stage of our proposed algorithm so that negative transfer is avoided.

3.2. Learning target task with both target and selected source data

In the second stage of our approach, an ensemble of learners for the target task are constructed. Although the source data selected from the first stage is closer to target task in terms of similarity between source task and target task, it's not guaranteed that all source data transferred to target task will be beneficial for learning target task. In order to reduce the impact from the irrelevant data, bootstrapping is used on the combination of the target data and selected source data. The details of building and ensemble of learners for the target task are given in Algorithm 2.

Algorithm 2. Construct an Ensemble for the Target Task
Input: Target training data D^T, source data R_S selected by algorithm 1, iteration number K, eliminate rate m and error threshold value e.
Output: A ensemble E of learners for the target task
Step 1. Combine R_S and D^Tinto training dataset D
Step 2. Train a standard learner L using the target data only
Step 3. Generate training samples by randomly resampling with replacement from D. The size of training samples is the same as D^T.
Step 4. Train K learners L_1, L_2,…, L_k. Each $L_k$is trained using data generated in step 3.
Step 5. Judge whether the learner L_k is eligible for a candidate base model according to the error indicator. If the error indicator value of the learner is larger than the given threshold value e, the learner will be discarded, and then the step 3 to 5 is repeated. Otherwise, the learner will be reserved as a candidate base learner.
Step 6. Judge whether the iteration number is reached. If not, return to step 3.
Step 7. Based on the eliminating rate, eliminate some candidate base learners with the worst performance.
Step 8. Produce the ensemble model by integrating the reserved base models.

At the end of iteration, the average value of the ensemble model is adopted as the final regression result.

IV. Experiments

4.1. Data description and error evaluation

We adopted data provided by GEFCOM2012[28], that contains electric load data from 20 geographical zones in the United States. The original dataset includes data from January 1,2004 to July 7 2008. Because the objective of the paper is short-term load forecasting, we only use data from March 1 ,2008 to June 30 ,2008. The training data used was from March 1,2008 to May 31,2008. The testing data used was from June 1,2008 to June 30,2008. Since the sample time interval was 60 minutes, there are 24 sample data every day. For the training time period, there were 2,208 training data in all and 720 data points in the testing dataset.

From the original GEFCOM2012 dataset, several pre-processing steps were performed. Originally the relationship between zones and weather stations in the dataset were not given. We followed the method according to [28] to map the weather station for each zone. Basically, a testing week (the last week of training data) was used to decide which sites to use for each zone. Temperature data from all 11 weather stations was combined with the load demand data separately. A traditional machine learning model was applied to the above datasets to forecast the load demand of the last day. In our case, we chose GBDT to forecast. The weather station with the best results as evaluated by MAPE was chosen for the corresponding zone.

Based on the analysis in [28], data from zone 2, zone 3, zone 4, zone 9 and zone 10 were not used due to data duplication. As a result, only data from 15 out of 20 zones was used in this study.

In Figure1, we plot part of the original data from four zones. As the figure shows, the power load trend similarities shared by different zones indicate that the transfer learning approach is a promising method.

To produce accurate load forecasting, the following input variables were included in the model:

- Demands around the same time period for the last two days;
- The maximum demand in the last 24 hours;
- The minimum demand in the last 24 hours;
- The average demand in the last seven days;
- The maximum temperature in the last 24 hours;
- The minimum temperature in the last 24 hours;
- The average temperature in the last seven days.
- Day of the week.
- Time of the year
- Holiday (weekend is taken as holiday)

The mean absolute percentage error (MAPE) was calculated to examine the forecasting accuracy, which is defined as follow:

$$\text{MAPE} = \frac{\sum_{i=1}^{N}\left|\frac{(P_i - A_i)}{A_i}\right|}{N} \times 100\% \qquad (4)$$

where N is the forecasting period, and P_i and A_i are the i_{th} predicted and actual values respectively.

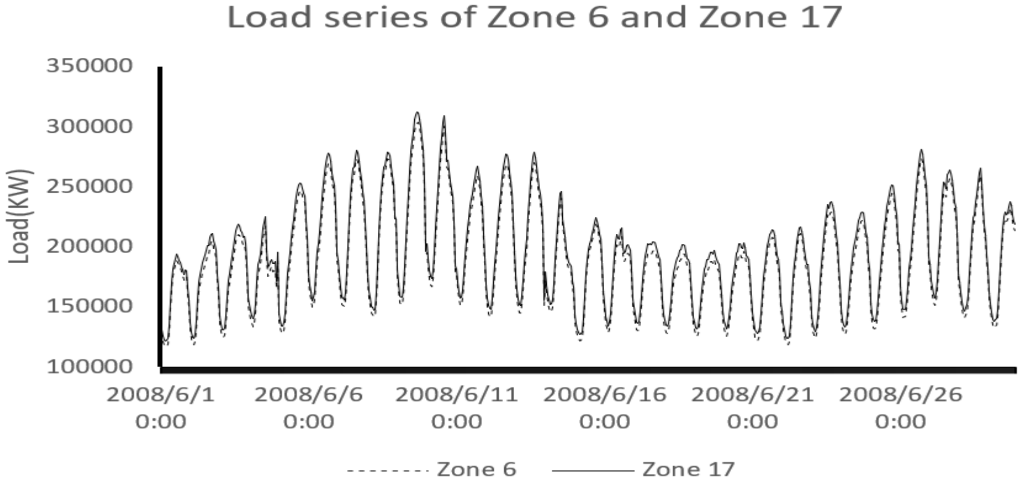
4.2. Method comparison

To evaluate the effectiveness of the proposed method in this paper, one traditional machine learning method and two transfer learning methods were selected as counterparts for comparison purposes. The following are simple introductions to these three counterparts:

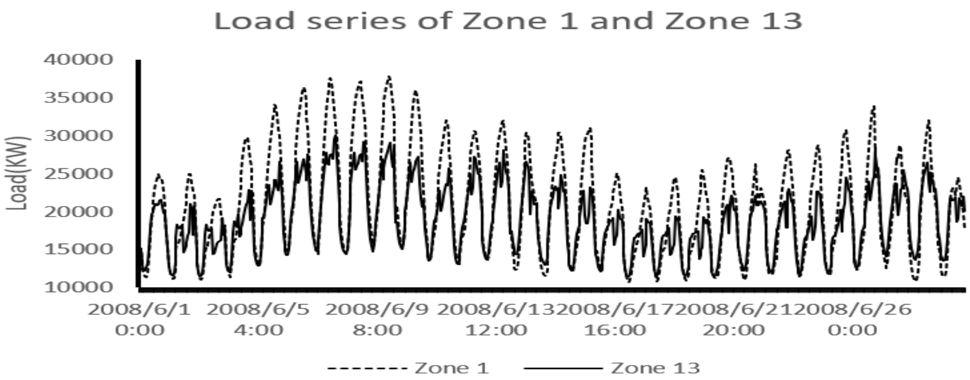
(1) AdaBoostRegressor: We applied AdaBoostRegressor to the dataset at each zone independently. The results serve as a baseline for traditional machine learning.

(2) Two stage TrAdaBoost.R2: This is an algorithm developed in [16]. Using original work in Two stage TrAdBoost.R2, all source data sets are combined into a single dataset when there is more than one source.

S



(a) Zone 6 and Zone 17



(b) Zone 1 and Zone 13

Figure 1. Electric power load data for four regions (Sample interval: 60 minutes)

(3) Multi-Source TrBagging: Similar to two-stage TrAdaBoost.R2, we simply combined all source datasets into a single dataset. Then the second stage of our proposed method was executed by skipping similarity checking in the first stage.

All our experiments were carried out in Python 3.5 using a computer with Intel Core i7-7500U CPU, 2.90 GHz, and 8 GB RAM.

With different choices for target and source tasks, we compared the performance of our transfer method to that of Two stage TrAdaBoost.R2 and Multi-Source TrBagging. We first discuss cases in which data from two sources were transferred into the target task. Then we discuss the cases with more than two sources.

We randomly picked one zone as the target location and picked the source locations from the remaining zones. Each experiment was repeated 10 times, and the results were averaged. The number of samples transferred from all sources was the same as number of interactions. Scikit-learn was used for implementation. The maximum depth of the tree was set to 6. The number of bagging iterations was set to 100. The eliminate rate was 0.8.

Table 1 reports the results of 9 experiments, that transferred data from two sources. The table shows that our method improved prediction accuracy in all 9 cases. Compared to the baseline, our method achieved an improvement over 11% at most while choosing zone 17 as the target location and both zone 7 and zone 8 as the source locations. Our method outperformed the other two transfer learning approaches. In fact, Multi-Source TrBagging brought negative transfer in 5 out of 9 cases. Two stage TrAdaBoost.R2 was worse than Multi-Source TrBagging, bringing negative transfer in 7 out of 9 cases.

There are two factors that affect the improvement a transfer method can bring. One is the effectiveness of the transfer approach and the other is the effectiveness of the data transferred from the sources. Both Two stage TrAdaBoost.R2 and Multi- Source TrBagging utilize all samples from sources. Even though they reduce the impact of unrelated data at the algorithm level, negative transfer still happened in more than half of the experiments. Based on research from [17], resampling often outperforms

Table 1. MAPE performance comparison of transfer methods. All methods use decision tree regressor as base learner. Negative transfers are marked with '*' and best performances are highlighted with boldface font.

Experiments	Baseline (No Transfer)	Transfer methods		
Target zone ID (Source zone IDs)	AdaBoostRegressor	Our method	Multi-Source TrBagging	Two stage TrAdaBoost.R2
1(11,12)	11.92%	**11.26%**	15.36%*	14.38%*
1(5,8)	11.92%	**11.58%**	12.81%*	14.46%*
1(13,14)	11.92%	**11.22%**	11.7%	11.27%
6(16,17)	12.04%	**11.22%**	11.26%	14.11%*
6(19,20)	12.04%	**11.37%**	12.39%*	13.66%*
6(1,5)	12.04%	**11.23%**	11.29%	12.96%*
17(1,5)	11.97%	**11.81%**	12.1%*	16.86%*
17(7,8)	11.97%	**10.65%**	12.19%*	13.73%*
17(6,7)	11.97%	**10.92%**	10.71%	11.28%

reweighting, which explains why Multi-Source TrBagging is better than Two stage TrAdaBoost.R2.

Table 2 reports the results of the experiments with data transferred from more than two sources. The result shows that our method performs best when there are three sources, and the prediction accuracy can't be improved more as the number of source data is more than four. Our method outperformed baseline and the other two transfer learning approach in all cases. While choosing zone 17 as the target location and zone 1, zone 5 and zone 6 as the source locations, our method outperformed AdaBoostRegressor 10.4% at most. As more data was transferred from other sources, both Two stage TrAdaBoost.R2 and Multi- Source TrBagging gained better prediction results. Multi-Source TrBagging outperformed the baseline 7% in the best case while choosing zone 6 as the target location and zone 16, zone 17, zone 19 and zone 20 as the source locations. Two stage TrAdaBoost.R2 outperformed the baseline 1% in the best case while choosing zone 1 as the target location and zone 11, zone 12 and zone 13 as the source locations. The performance of Multi-Source TrBagging was close to our proposed method in 2 out of 12 experiments. Negative transfer is still inevitable, even though there are more data transferred from the sources. Multi-Source TrBagging brought negative transfer in 6 out of 12 cases. Two stage TrAdaBoost.R2 brought negative transfer in 10 out of 12 cases. Negative transfer was avoided completely by the proposed algorithm.

Finally, we compared the running time of our method to that of the others in Table 3. As shown in Table 3, the time spent by our method was much less than that needed by Two stage TrAdBoosting.R2 and Multi-Source TrBagging. Because these two approaches take all sample data from all sources, they take more time than the proposed algorithm.

Table2. MAPE performance comparison of transfer methods with data transferred from more than two sources. All methods use decision tree regressor as base learner. Negative transfers are marked with '*' and best performances are highlighted with boldface font.

Experiments	Baseline (No Transfer)	Transfer methods		
Target zone ID (Source zone IDs)	AdaBoostRegressor	Our method	Multi-Source TrBagging	Two stage TrAdaBoost.R2
1(11,12)	11.92%	**11.26%**	15.36%*	14.38%*
1(11,12,13)	11.92%	**11.18%**	15.07%*	11.82%
1(11,12,13,14)	11.92%	**11.2%**	11.26%	12.23%*
1(11,12,13,14,15)	11.92%	**11.34%**	11.17%	12.24%*
6(16,17)	12.04%	**11.22%**	11.26%	14.11%*
6(16,17,19)	12.04%	**11.21%**	12.19%*	15.13%*
6(16,17,19,20)	12.04%	**11.22%**	12.8%*	16.14%*
6(5,16,17,19,20)	12.04%	**11.24%**	12.47%*	12.39%*
17(1,5)	11.97%	**11.81%**	12.1%*	16.86%*
17(1,5,6)	11.97%	**10.73%**	11.52%	13.38%*
17(1,5,6,7)	11.97%	**11.2%**	11.9%	11.91%
17(1,5,6,7,8)	11.97%	**10.77%**	11.56%	12.82%*

V. CONCLUSION

In this paper, we proposed a two-stage bagging based multiple source transfer regression framework, which is appropriate for short-term electricity load forecasting. By transferring knowledge from multi-source with the multi-similarity approach proposed in the paper, the chance to transfer useful knowledge from the source to the target increased dramatically. Experiment results on a real-world dataset suggest that the proposed algorithm can significantly improve the forecasting performance by importing knowledge from multiple sources. Meanwhile, we also investigated the effect of negative transfer and showed that potential negative transfer can be prevented by the proposed method.

In the future, we will work on following aspects. First, similarity measurement for time series can be analyzed, for instance: Dynamic Time Warping (DTW) for time series data. Second, to enhance the prediction accuracy, deep learning approaches could collaborate with the ensemble transfer learning framework.

Table3. Average Time Spent to Run the Experiments on a 64-bit Windows Machine with Intel Core i7 CPU and 8G RAM.

Experiments	Transfer methods		
Target zone ID (Source zone IDs)	Our method	Multi-Source TrBagging	Two stage TrAdaBoost.R2
1(11,12)	92.05s	141.53s	108.92s
1(11,12,13)	89.45s	174.63s	150.17s
1(11,12,13,14)	98.34s	224.85s	210.95s
1(11,12,13,14,15)	122.19s	293.26s	277.17s
6(16,17)	98.44s	134.82s	102.99s
6(16,17,19)	98.28s	184.12s	148.13s
6(16,17,19,20)	100.56s	230.28s	191.39s
6(5,16,17,19,20)	164.06s	402.47s	327.76s
17(1,5)	94.75s	141.64s	103.01s
17(1,5,6)	92.87s	171.42s	150.39s
17(1,5,6,7)	98.01s	250.09s	218.88s
17(1,5,6,7,8)	118.93s	353.15s	317.19s

REFERENCES

[1] Soliman, Abdel Hady, D. Ph., and A. M. Al-K. "Electrical Load Forecasting." Electrical Load Forecasting 27.5(2010):299-305.

[2] Zheng, Dehua, et al. "Short-term Wind Power Forecasting Using a Double-stage Hierarchical ANFIS Approach for Energy Management in Microgrids." Protection & Control of Modern Power Systems 2.1(2017):13.

[3] Huang, S. R. "Short-term Load Forecasting Using Threshold Autoregressive Models." IET Proceedings - Generation Transmission and Distribution 144.5(1997):477-481.

[4] Yang, Hong Tzer, and C. M. Huang. "A New Short-term Load Forecasting Approach Using Self-Organizing Fuzzy ARMAX Models." IEEE Transactions on Power Systems 13.1(2002):217-225.

[5] Infield, D. G, and D. C. Hill. "Optimal Smoothing for Trend Removal in Short Term Electricity Demand Forecasting." IEEE Transactions on Power Systems 13.3(1998):1115-1120.

[6] Yingzhe, et al. "A Novel DBN Model for Time Series Forecasting." IAENG International Journal of Computer Science 44.1(2017): 79-86.

[7] Swaroop, R., and H. Abdulqader. "Load Forecasting for Power System Planning using Fuzzy-Neural Networks." Lecture Notes in Engineering & Computer Science (2012): 504-508.

[8] Lin, Z., C. Ma, and X. Liu. "Study on A Rolling Prediction Model of Power System Load Forecasting Based on Evolutionary SVM." Lecture Notes in Engineering & Computer Science 2167.1(2007).

[9] Pan, Sinno Jialin, and Q. Yang. "A Survey on Transfer Learning." IEEE Transactions on Knowledge & Data Engineering 22.10(2010):1345-1359.

[10] V Kudva, K. Prasad, and S. Guruvare. "Hybrid Transfer Learning for Classification of Uterine Cervix Images for Cervical Cancer Screening." Journal of Digital Imaging 33 (2020): 619-631.

[11] Cai, Chenjing, et al. "Transfer Learning for Drug Discovery." Journal of Medicinal Chemistry 63 (2020): 8683-8694.

[12] Cai, W, et al. "Neighborhood-Enhanced Transfer Learning for One-Class Collaborative Filtering." Neurocomputing 341.May 14(2019):80-87.

[13] He, Dongrui, and Wu. "Transfer Learning for Brain-Computer Interfaces: A Euclidean Space Data Alignment Approach. " IEEE sTransactions on Bio Medical Engineering 67 (2020): 399-410.

[14] M.T. Rosenstein, Z. Marx, and L.P. Kaelbling, “To Transfer or Not to Transfer,” Proc. Conf. Neural Information Processing Systems (NIPS ’05) Workshop Inductive Transfer: 10 Years Later, Dec. 2005.

[15] Dai, Wenyuan, et al. "Boosting for transfer learning." International Conference on Machine Learning (2007): 193-200

[16] Pardoe, David, and Peter Stone. "Boosting for Regression Transfer." International Conference on Machine Learning (2010): 863-870.

[17] Seiffert, C., et al. "Resampling or Reweighting: A Comparison of Boosting Implementations." International Conference on Tools with Artificial Intelligence (2008): 445-451.

[18] Liu, Xiaobo, et al. "Ensemble Transfer Learning Algorithm." IEEE Access (2018): 2389-2396.

[19] Bendavid, Shai, and Reba Schuller. "Exploiting Task Relatedness for Multiple Task Learning." Lecture Notes in Artificial Intelligence (2003): 567-580.

[20] Yao, Yi, and Gianfranco Doretto. "Boosting for Transfer Learning with Multiple Sources." Computer Vision and Pattern Recognition (2010): 1855-1862.

[21] Huang, Pipei, G. Wang, and S. Qin. “Boosting for Transfer Learning from Multiple Data Sources.” Pattern Recognition Letters (2012): 568--579.

[22] Zhang, Qian, et al. "Instance Transfer Learning with Multisource Dynamic TrAdaBoost." The Scientific World Journal, 2014, (2014-7-24) 2014.4(2014):282747.

[23] Yulai, et al. "Power Load Forecasting based on Multi-task Gaussian Process." IFAC Proceedings Volumes 47.3(2014):3651-3656.

[24] Fiot, Jeanbaptiste, and Francesco Dinuzzo. "Electricity Demand Forecasting by Multi-Task Learning." IEEE Transactions on Smart Grid 9.2 (2018): 544-551.

[25] Zhang, Yulai, and Guiming Luo. "Short Term Power Load Prediction with Knowledge Transfer." Information Systems (2015): 161-169.

[26] Wu, Di, et al. "Boosting Based Multiple Kernel Learning and Transfer Regression for Electricity Load Forecasting." European Conference on Machine Learning (2017): 39-51.

[27] Bing, Lidong, et al. "Abstractive Multi-Document Summarization via Phrase Selection and Merging." International Joint Conference on Natural Language Processing (2015): 1587-1597.

[28] Taieb, Souhaib Ben, and Rob J. Hyndman. "A Gradient Boosting Approach to the Kaggle Load Forecasting Competition." International Journal of Forecasting 30.2 (2014): 382-394.