Comparison of Three Methods for a Weather Based Day-Ahead Load Forecasting

Citation for published version:
Zou, M, Gu, J, Fang, D, Harrison, G, Djokic, S, Wang, X & Zhang, C 2019, Comparison of Three Methods for a Weather Based Day-Ahead Load Forecasting. in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe).*, 8905631, Proceedings of 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2019, Institute of Electrical and Electronics Engineers Inc., 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2019, Bucharest, Romania, 29/09/19. https://doi.org/10.1109/ISGTEurope.2019.8905631

Digital Object Identifier (DOI):
10.1109/ISGTEurope.2019.8905631

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)

Comparison of Three Methods for a Weather Based Day-Ahead Load Forecasting

Mingzhe Zou[1], Jiachen Gu[2], Duo Fang[1], Gareth Harrison[1], Sasa Djokic[1]
[1]School of Engineering, [2]School of Informatics
The University of Edinburgh
Edinburgh, UK
sasa.djokic@ed.ac.uk

Xinying Wang, Chen Zhang
Artificial Intelligence Research Centre
China Electric Power Research Institute
Beijing, China
xinyingking@gmail.com

***Abstract*—Day-ahead load forecasting plays an increasingly important role for the operation of networks and generation dispatch. The accuracy of the forecasting depends on many factors, including the quality and size of historical data, selected forecasting model and available information on influential factors (e.g., weather data). This paper compares three state-of-the-art models in terms of their ability to capture complex variations in load profiles and provide accurate day-ahead load forecasting: multilayer perceptron (MLP), gradient boosting regression trees (GBRT) and stacked bidirectional long short-term memory (SB-LSTM). The models are implemented on one dataset from China and another from Scotland. The performance is evaluated using mean absolute error (MAE), mean absolute percentage error (MAPE) and other energy based indices. The presented results show that GBRT outperforms MLP with the same expert extracted load characteristics as additional inputs, but SB-LSTM provides the most accurate forecasting results, without extracting any artificial data feature from the two considered demand datasets.**

***Index Terms*--Day-ahead load forecasting, deep learning, multilayer perceptron, recurrent neural network, regression tree ensemble.**

I. Introduction

Load forecasting aims to provide an accurate estimation of energy consumption based on historical demand recordings and other relevant data, providing important information for the development of modern electricity networks. Long-term forecasting is important for power system planning, while short-term forecasting, especially day-ahead forecasting, is essential for operational studies [1].

Typical load profiles exhibit periodicity, but also feature strong hourly, daily, weekly and seasonal variations. Weather (meteorological) factors, such as temperature, precipitation, solar irradiation, etc., as well as socio-behavioral factors, all have impact on variations of demands, increasing uncertainty and complexity of load modeling problem. There are many load forecasting approaches in existing literature, including time series analysis [2], Gaussian process [3], neural network [1], [4], fuzzy regression [5], tree-based model [6] and support vector regression [7]. Regardless of the technique used to model load profiles and provide predictions, model selection is an important step, which relies on understanding of the inherent features and characteristics of different models.

Compared to traditional machine learning models (e.g., artificial neural network with only one hidden layer), deep learning models usually have a cascade of multiple layers, in order to transform the data and extract their features more efficiently. In simple problems, or if there are limited data, deep learning methods may not even be as accurate as classic machine learning methods. However, their performance is much stronger in complex problems requiring processing of large amounts of data. In particular, long short-term memory (LSTM) method, [8], which is a type of recurrent neural network (RNN) under deep learning framework, is well-suited for the classification and regression of time series data.

In addition to deep learning methods, regression tree ensemble models are also gaining popularity, as their hyperparameters can be easily tuned and their training is suitable for parallelization. Furthermore, they are not sensitive to the size of datasets, which makes them attractive when there are not enough data [9].

This paper analyses three state-of-the-art methods for a day-ahead load forecasting and compares their performance on two actually recorded demand datasets. Section II presents multilayer perceptron (MLP), gradient boosting regression trees (GBRT) and stacked bidirectional LSTM (SB-LSTM) methods. Section III compares day-ahead load forecasting results, while Section IV gives main conclusions.

II. Three Considered Models

This section presents three models for load forecasting based on: classical artificial feedforward neural network; regression tree and ensemble learning; and recurrent neural network. Regression analysis aims to estimate relationships between variables, where inputs and outputs are defined as predictor and response variables. Regression analysis can be used for forecasting, if variables of past and current states are set as predictor variables, while variables in the future state are set as response variables. Neural network-based methods are widely used in classification problems as supervised learning approaches, and they can also be implemented for regression analysis and load forecasting by setting their outputs as response variables instead of categories.

A. Multilayer perceptron

MLP is a classical feedforward artificial neural network architecture consisting of at least three layers: an input layer, an output layer and a hidden layer, Figure 1. Hidden layer typically uses a nonlinear activation function to map the weighted inputs to its outputs; For extremely complex problems, the depth of the network should usually increase by adding more hidden layers, but a deeper MLP is more likely to have overfitting issue. To resolve that, (random) droppingout of hidden and visible neurons in a certain layer has been presented as an efficient regularization technique [10]. L2 regularization can also improve the accuracy by adding the squared values of weight coefficients to the loss function as penalty factors [11].

In classical MLP, layers are fully-connected (also known as "dense layers"), which means the outputs of all neurons in the previous layer are connected to every neuron in the next layer. In the training process, MLP learns through changing weights of all neurons in the steepest direction that reduces the error between the network outputs and the actual outputs, which is known as gradient descent.

B. Gradient Boosting Regression Trees

Unlike neural network models, tree-based models are white box models, rather than black box models. A regression tree can be grown through an iterative process that recursively divides data into partitions, which is inherently a statistical method based on conditional probabilities in features' class space. Since the splitting allocates the data to small groups according to the given predictors, regression trees are usually well explainable.

A typical regression tree is a top-down structure, where at each step constructing algorithm chooses a variable that can most efficiently split the dataset; efficiency can be evaluated by various metrics, e.g. information gain and variance [12]. A fully-grown regression tree is very likely to suffer overfitting issues (shallow tree) and generally cannot perform well on any other datasets, except the training set.

Ensemble learning can improve flexibility of regression tree model by growing multiple regression trees to obtain better prediction, instead of relying only on one shallow tree [13]. Gradient boosting can be used for creating accurate ensembles that are typically formed of tree-based models. At each boosting stage, the algorithm will try to identify weak regression trees that have the maximum predicting errors and then improve these learners by adding extra estimators fitted to residuals. At the next boosting stage, the new regression trees will attempt to reduce the errors of their predecessors. As residuals are negative gradients of the squared errors, gradient boosting is actually a gradient reduce algorithm [14].

C. Stacked Bidirectional Long Short-term Memory

RNN is a class of neuron network models specialized for sequence data and is widely used in natural language processing (NLP), speech recognition, machine translation and so on. Unlike feedforward neural networks (e.g. MLP), RNN is designed to memorize sequence data. Figure 2 shows the typical architecture of a traditional RNN. It can be seen that each RNN cell uses both the previous internal state h_{t-1} and its current input x_t to predict y_t, with internal state h_t also updated and passed to the next step.

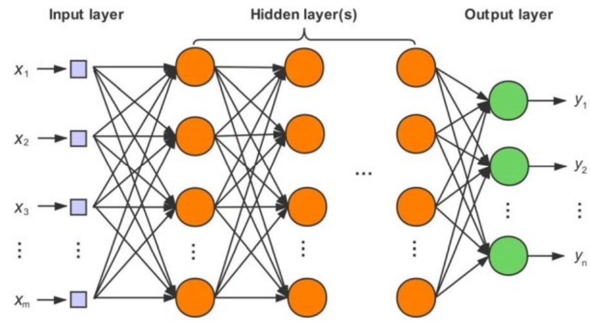


Figure 1. Typical MLP architecture

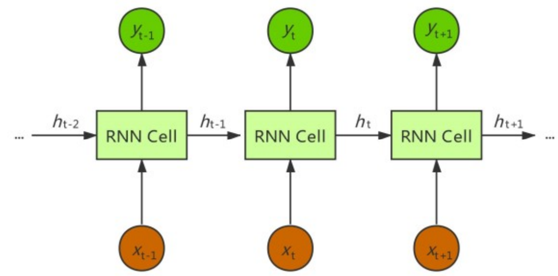


Figure 2. Typical traditional RNN architecture

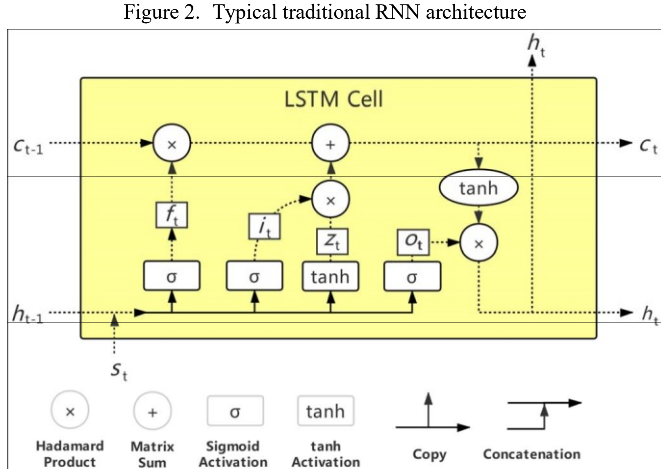


Figure 3. LSTM cell architecture (solid arrows represent weighed vectors, dot arrows represent unweighted vectors)

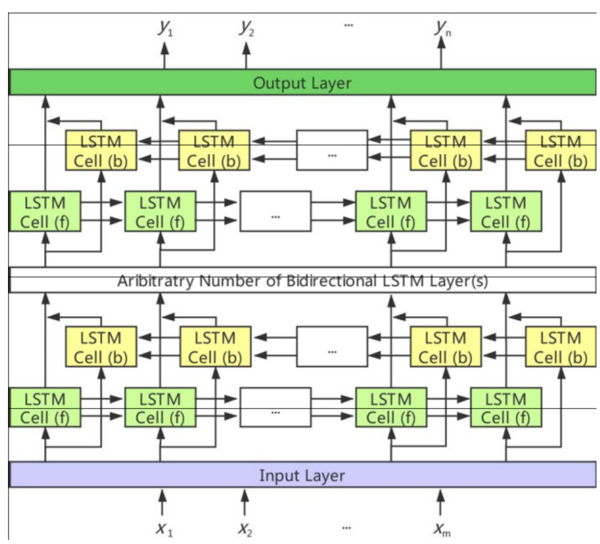


Figure 4. SB-LSTM architecture: (f) - "forward" and (b) - "backward"

When processing long time series data, traditional RNN method tends to have gradient exploding/vanishing issues [15]. To deal with that, long short-term memory (LSTM) is developed, which is relatively insensitive to gaps of duration between important sequence events [8]. Every cell of LSTM has added three gates to control whether to remember or forget certain information and whether to output at a specific point. The architecture of LSTM cell is illustrated in Figure 3, where s_t is the current input and y_t is its output. Like traditional RNN, LSTM can also pass the internal states, but an LSTM cell has two different states: c is the cell state that changes extremely slow, thus in most case, it is very similar to the previous cell state; h stands for the hidden state, which varies a lot from nodes to nodes and usually depends on the current input, as well as on the previous hidden state. Three gate signals are generated by applying activation function to weighted matrix concatenating the current input and the previous hidden state:

Forget gate: Decides which part of the cell state information ought to be forgotten; uses h_{t-1} and s_t as inputs and it is activated by sigmoid activation function σ:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, s_t] + b_f\right) \tag{1}$$

Remember gate: Controls which part of the input and hidden state to remember; signal i is activated by sigmoid activation function and another candidate vector z is activated by *tanh* function:

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, s_t] + b_i) \\ z_t = \tanh(W_z \cdot [h_{t-1}, s_t] + b_z) \\ c_t = f_t \odot c_{t-1} + i_t \odot z_t \end{cases} \tag{2}$$

Output gate: Decides which part of the information to output; the output information is again activated by *tanh* activation function:

$$\begin{cases} o_t = \sigma(W_o \cdot [h_{t-1}, s_t] + b_o) \\ h_t = o_t \odot \tanh(c_t) \end{cases} \tag{3}$$

In (1)-(3), W_f, W_i, W_z, W_o, and b_f, b_i, b_z and b_o are weight and bias parameters of each gate. These three gates enable LSTM to remember important information and forget the irrelevant one over a long sequence. This significantly increases magnitudes of hyperparameters, which makes LSTM hard to tune and train. Like MLP, all the parameters of weights and biases in LSTM are also learned by minimizing differences between the network outputs and the ground truth.

The original LSTM has only one hidden layer of one direction, which may not perform well for deep information mining. The stacked bidirectional LSTM (SB-LSTM) has multiple hidden bidirectional LSTM layers, each containing two hidden layers of opposite directions to the same outputs, Figure 4. The bidirectional structure is introduced in [16] and it allows the outputs to have both forward and backward information. The stacked structure also allows the network model to be deeper, therefore it could learn more accurately on more challenging problems [17]. However, too many layers make the training time longer and the model may also have higher probability of overfitting.

III. Model Implementation and Comparison of Results

Two datasets are used for analysis of MLP, GBRT and SB-LSTM models. The first is five-year measurements of average 15-minute active power demands from a city in China and synchronous daily minimum ($T_{\min}$) and maximum ($T_{\max}$) temperature recordings in the same area. Because of different temporal resolutions from load data, temperature data are firstly up-sampled to 96 values per day using spline interpolation method [18], assuming that $T_{\min}$ and $T_{\max}$ occur around 5 am and 2 pm, respectively [19]. The second dataset is from a substation in a town in Scotland, containing five-year average 30-minute active power demands and 60-minute meteorological/weather data: temperature (T), solar irradiation (S), rainfall (R) and wind speed (WS). Again, meteorological data (T, S, R, WS) are all up-sampled to 30-minute resolution to be synchronous with demand datasets.

Each of five-year dataset is divided into training set, validation set and test set, where minimum demand week and maximum demand week in year 5 are selected for testing and contiguous 4-year-length data exactly before the test week are used as training/validation set. The training set and validation set are obtained by splitting the 4-year-length data by the ratio of 80:20 in contiguous blocks in every 100 days. The dataset should not be divided randomly, because the recordings are sequence data which have significant temporal dependencies.

A. Calendar Variables

Load recordings have strong autocorrelations and periodicities, and the demand at specific time of a day has the strongest correlations with demands at the identical time of other days. Since recordings have high temporal dependencies and they are strongly driven by time, standard calendar variables, i.e., year, month, day, hour and day of the week (*YY*, *MM*, *DD*, *HH*, *DW*), as well as holiday indicator (*HLD*) and British summer time indicator (*BST*) are selected. In addition, solar analemma (Sun azimuth angle *SA* and elevation angle *SE*) can precisely represent time and season variations and are calculated as in [20]. Moon's rotation also has impacts on Earth and human behaviors [21], and moon phase (*MP*) is taken into consideration as an additional variable [22].

B. Generalization of MLP Model

In MLP model, to forecast the day-ahead load profile, multiple load values at all moments in the next day (i.e., 96 values for China dataset and 48 points for Scotland dataset) are predicted separately. For every forecasted load value, the model uses following inputs: 1) Calendar variables (*YY*, *MM*, *DD*, *HH*, *DW*, *HLD*, *BST*, *SA*, *SE* and *MP)*; 2) Meteorological data: for China dataset, only temperature (T), while full meteorological data (T, S, R, WS) for Scotland dataset. 3) Load recordings at the identical moment from the past 28 days. The model output is predicted load value at the forecasted moment, where full day-ahead forecast is obtained by applying the model multiple times for all moments the next day.

Calendar variables, excluding analemma and moon phase data, are firstly encoded by one hot encoder, as they are categorical features and the accuracy can be improved by feeding neurons with encoded data [23]. Analemma, moon phase, meteorological and load recordings are scaled and mapped into the range [-1, 1] using min-max normalizations. The minimum and maximum values are assumed to be the minimum and maximum recordings in the training/validation set. Z-score standardization method is not used. Three hidden layers and three dropout layers are implemented in MLP model (each hidden layer consists of 256 neurons). Leaky rectified linear units (LReLU) is selected as activation function following each hidden layer, due to its good performance in deep learning and advantages over normal ReLU in handling negative values [24]. L2 regularization is added to prevent overfitting and Adam optimizer algorithm is used [25]. Objective is to minimize mean absolute error (MAE) between model and recordings [26].

C. Generalization of GBRT Model

The GBRT model uses the same inputs and outputs as the MLP model, but it does not need to use one hot encoder, or min-max normalization to preprocess the data before feeding the model, because they can be regarded as individual features with specific physical meanings in the tree-based model. There are 512 learners implemented in the GBRT and the minimum leaf size is set as 5, which means there should be at least 5 responses of each leaf node. Least-square boosting (LSBoost) algorithm is adopted to fit the regression [27].

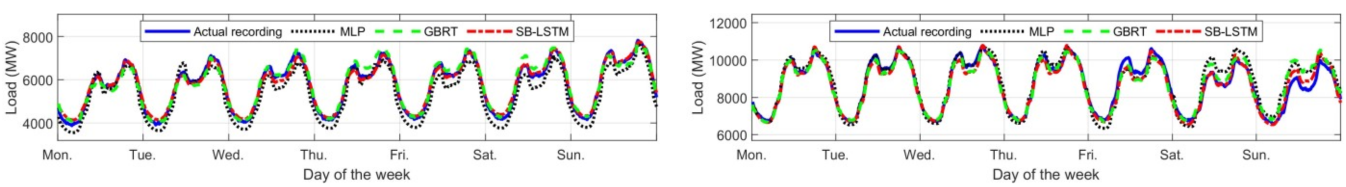
D. Generalization of SB-LSTM Model

One advantage of SB-LSTM is that the characteristics of the load series need not to be artificially extracted, so this class of model can process sequence data and learn its deep features directly. Therefore, there will be no need to forecast different moments separately. In SB-LSTM model, the inputs are the full sequence of load measurements from the previous 28 days, the meteorological data in the forecasted day and the calendar variables in the forecasted day. The output is the load profile in the forecasted day. Like the MLP model, one hot encoder and min-max normalization are used to preprocess the data before model generalization. The SB-LSTM model is implemented with three bidirectional LSTM layers and three dropout layers, while L2 regularization and Adam optimizer are again used to minimize MAE.

E. Comparisons of Results

The examples of forecasting results of day-ahead load profiles are shown in Figures 5 and 6 for China dataset and Scotland dataset, respectively. They are both day-by-day (one day after another) day-ahead forecasting results for Monday to Sunday in the same weeks. Tables I to Table IV provides numerical results for quantifying the benefits of different models, which should be evident through the comparisons of MAE and mean absolute percentage error (MAPE) [26]. The absolute and percentage errors in the total, over-estimated and under-estimated energy consumptions (denoted as E_T, E_O and E_U respectively) are also compared with the actual demands, assuming that the mean load values are demanded over each averaging window (15 minutes for China dataset, and 30 minutes for Scotland dataset).

The variations of load profiles on different days may be caused by a lot of factors, such as weather changes and dayof-the-week effects [28]. All models have appreciated the impacts of meteorological factors and different days of the week, and they all can capture the main shapes and trends of load profiles. The errors of GBRT and SB-LSTM methods are both lower than MLP method errors, both regarding MAE and MAPE results. From the presented results, SB-LSTM method can provide the most accurate predictions in the two datasets, and is more flexible capturing actually deep dependencies and features of sequenced load data.



(a) (b)

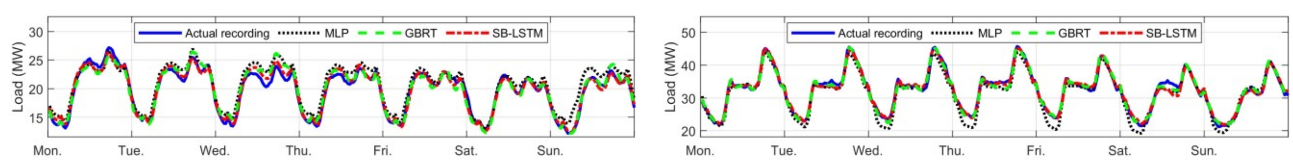
Figure 5. Examples of day-ahead load forecastings in China dataset: (a) the minimum demand week and (b) the maximum demand week.

TABLE I. COMPARISON OF MODEL RESULTS AND ACTUAL MEASUREMENTS IN CHINA DATASET (FOR 7 DAYS IN FIGURE 5A)

Minimum Load	MAE (MW)	MAPE (%)	E_O (GWh)	E_O (%)	E_U (GWh)	E_U (%)	E_T (GWh)	E_T (%)	Total Demand (GWh)
MLP	370.174	6.751	1.733	2.738	-60.456	-6.718	-58.723	-6.096	963.262
GBRT	151.039	2.575	16.762	2.785	-8.613	-2.383	8.149	0.846	963.262
SB-LSTM	107.915	2.027	10.973	2.014	-7.157	-1.711	3.817	0.396	963.262

TABLE II. COMPARISON OF MODEL RESULTS AND ACTUAL MEASUREMENTS IN CHINA DATASET (FOR 7 DAYS IN FIGURE 5B)

Maximum Load	MAE (MW)	MAPE (%)	E_O (GWh)	E_O (%)	E_U (GWh)	E_U (%)	E_T (GWh)	E_T (%)	Total Demand (GWh)
MLP	299.849	3.548	35.189	4.049	-15.185	-2.561	20.004	1.368	1462.043
GBRT	222.017	2.543	22.031	3.550	-15.268	-1.815	6.763	0.463	1462.043
SB-LSTM	165.530	1.864	8.757	1.817	-19.052	-1.944	-10.296	-0.704	1462.043



(a) (b)

Figure 6. Examples of day-ahead load forecastings in Scotland dataset: (a) the minimum demand week and (b) the maximum demand week.

TABLE III. COMPARISON OF MODEL RESULTS AND ACTUAL MEASUREMENTS IN SCOTLAND DATASET (FOR 7 DAYS IN FIGURE 6A)

Minimum Load	MAE (MW)	MAPE (%)	E_O (MWh)	E_O (%)	E_U (MWh)	E_U (%)	E_T (MWh)	E_T (%)	Total Demand (MWh)
MLP	1.145	6.125	178.218	6.371	-14.191	-2.760	164.026	4.953	3311.580
GBRT	0.647	3.223	63.895	3.652	-44.728	-2.864	19.167	0.579	3311.580
SB-LSTM	0.448	2.296	45.073	2.402	-30.143	-2.100	14.930	0.451	3311.580

TABLE IV. COMPARISON OF MODEL RESULTS AND ACTUAL MEASUREMENTS IN SCOTLAND DATASET (FOR 7 DAYS IN FIGURE 6B)

Maximum Load	MAE (MW)	MAPE (%)	E_O (MWh)	E_O (%)	E_U (MWh)	E_U (%)	E_T (MWh)	E_T (%)	Total Demand (MWh)
MLP	1.215	4.080	23.567	1.859	-180.554	-4.325	-156.987	-2.884	5442.538
GBRT	0.551	1.763	37.532	1.584	-55.046	-1.791	-17.514	-0.322	5442.538
SB-LSTM	0.525	1.660	44.182	1.553	-44.028	-1.695	0.154	0.003	5442.538

IV. CONCLUSIONS

This paper presents analysis and comparison of day-ahead load forecasting with MLP, GBRT and SB-LSTM methods, based on historical load recordings and meteorological (weather) data. MLP is a relatively straightforward and naïve model, in which input signals can only travel in one direction to the output layer with no feedback and memory structure. It is suitable for modeling nonlinear relationship between its input and output, but selections of input features are crucial and pattern extraction is difficult in complex problems; in presented load forecasting study MLP did not capture the full temporal correlations and changes in the daily load profiles.

GBRT is an ensemble model, which combines many regression trees to learn the characteristics of load profiles. Although every individual regression tree is weak and shallow, together they provide more accurate predictions. Another reason why GBRT is better than MLP is that it can handle categorical features naturally and there is no need to use one hot encoder, or data normalization. Since the inputs of GBRT are also artificial extracted features (i.e., recordings at separate moments), so many hidden but important patterns may be lost, its performance is lower than SB-LSTM method.

In SB-LSTM, it is not necessary to transfer the dynamic sequence modeling problem to the static correlation modeling problem. Therefore, its input can naturally preserve the full temporal correlation information. For load profile modeling, SB-LSTM not only considers load data at the same moment of past days, but analyzes all load recordings at all moments and their hidden correlations and deep patterns. Therefore, it is the most robust and promising model among all three models.

There are days where MLP, GBRT and SB-LSTM all fail to have more accurate predictions, because load profiles in these days are inherently and significantly different from other days learned by the models. This may be due to more complex factors, planned/unplanned network maintenance and possibly demand-side-management actions, especially during the hot summer days, when temperatures are very high in Chinese dataset. These all bring higher uncertainty to the network operating conditions and make the forecasting less accurate.

REFERENCES

[1] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Trans. on Smart Grid,* 10 (1), pp. 841-851, 2019.

[2] R. Weron, *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. The Wiley Finance Series, 2006.

[3] G. Xie, X. Chen, and Y. Weng, "An Integrated Gaussian Process Modeling Framework for Residential Load Prediction," *IEEE Transactions on Power Systems,* vol. 33, no. 6, pp. 7238-7248, 2018.

[4] M. Brenna, F. Foiadelli, M. Longo, and D. Zaninelli, "Solar radiation and load power consumption forecasting using neural network," in *2017 6th International Conference on Clean Electrical Power*, 2017, pp. 726-731.

[5] S. Kyung-Bin, B. Young-Sik, H. Dug Hun, and G. Jang, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE Trans. on Power Systems,* vol. 20, no. 1, pp. 96-101, 2005.

[6] H. Guo, "Accelerated Continuous Conditional Random Fields For Load Forecasting," *IEEE Trans. on Knowledge and Data Eng.,* 27 (8), 2015.

[7] E. Ceperic, V. Ceperic, and A. Baric, "A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines," *IEEE Trans. on Power Systems,* vol. 28, no. 4, pp. 4356-4364, 2013.

[8] S. Hochreiter and j. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[9] Z. Zhou and J. Feng, "Deep Forest: Towards An Alternative to Deep Neural Networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," arXiv, 2012..

[11] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *ICML '04 Proceedings of the twenty-first international conference on Machine learning*, 2004.

[12] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 35, no. 4, pp. 476-487, 2005.

[13] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *J. of Artificial Intelligence Res.,* 11 (1), pp. 169-198, 1999.

[14] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," Procs of the 12th Int. Conf. on Neural Information Processing Systems, Denver, CO, 1999.

[15] G. Chen, "A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation," arXiv, 2016.

[16] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Processing,* 45 (11), 1997.

[17] M. Hermans and B. Schrauwen, "Training and analyzing deep recurrent neural networks," Procs of the 26th Int. Conf. on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada, 2013.

[18] C. A. Hall and W. W. Meyer, "Optimal error bounds for cubic spline interpolation," *J. of Approximation Theory,* 16 (2), pp. 105-122, 1976.

[19] D. R. Satterlund, R. C. Chapman, and R. D. Beach, "Modeling the Daily Temperature Cycle," *Northwest Science,* vol. 57, pp. 22-31, 1983.

[20] J. LeSage, "Analyzing the Sun's Path," MATLAB Central File Exchange, 2016.

[21] M. Zimecki, "The lunar cycle: Effects on human and animal behavior and physiology," *Adv. in Hygiene and Exp. Medicine,* 60, pp. 1-7, 2006. [22] M. Mahooti, "Moon Phases," MATLAB Central File Exchange, 2019.

[23] D. Harris and S. Harris, *Digital Design and Computer Architecture, Second Edition*. Morgan Kaufmann Publishers Inc., 2012, p. 712.

[24] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.

[25] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015.

[26] M. Shcherbakov, A. Brebels, N. Shcherbakova, A. Tyukov, T. Janovsky and V. A. Kamaev, *A survey of forecast error measures*. 2013.

[27] L. Breiman, "Random Forests," *Machine Learning,* 45 (1), 2001.

[28] J. Zhang, Y. Lai, and J. Lin, "The day-of-the-Week effects of stock markets in different countries," *Finance Res. Lett.,* 20, pp. 47-62, 2017.