Constructing Probabilistic Load Forecast From Multiple Point Forecasts: A Bootstrap Based Approach

Jiawei Zhang[1] Yi Wang[2] Mingyang Sun[3], Ning Zhang[2*] and Chongqing Kang[2]

***Abstract*—Probabilistic load forecast presents more information on the possible deviation of forecast than the point forecast. There are sufficient regression models that can make point forecasts. An intuitive question can be raised: *Is there a way to combine the point forecasts to construct a probability or interval forecast?* In this paper, a bootstrap based ensemble approach is put forward to construct forecast intervals from multiple point forecasts. Specifically, multiple point forecasting models are first trained based on the bootstrap sampled training datasets and different forecasting models. Then, bootstrap is applied again to the multiple point forecasts. Finally, the quantiles are estimated according to the distribution of the sampled point forecasts. Two common machine learning methods, random forest (RF) and gradient boosting regression tree (GBRT), are combined to test the feasibility of the proposed forecasting framework. Compared with quantile RF (Q-RF) and quantile GBRT (Q-GBRT), numerical experiments demonstrate its advantage over Q-RF and Q-GBRT.**

***Index Terms*—Probabilistic load forecast, bootstrap, ensemble forecasting, random forest, gradient boosting regression tree (GBRT)**

I. Introduction

Load forecasting plays a crucial role in the development of electric power industry. Inaccurate load forecast may result in higher power system operating cost and renewable energy waste. Traditional point load forecasting method only gives a single value at a certain time. Due to the volatility of the load, a probabilistic forecast in form of intervals or quantiles is essential to deliver more information about the uncertainties of the future load [1].

Recently, a lot of machine learning methods or algorithms are put forward to make a better forecast on power loads. For instance, support vector machine (SVM), as well as least square SVM (LS-SVM), are widely used methods for short time forecasting [2], [3]. Also, the decision tree algorithms are also introduced to the forecast problem. To improve the accuracy of decision trees, the data scientists involved the thoughts of bagging and boosting into the tree algorithms and gave birth to the two ensemble methods Random Forest (RF) and gradient boosting regression tree (GBRT), which are widely applied on power industry [4], [5]. With the integration of distributed renewable energy, electric vehicle, etc., the electrical load becomes more complex and shows greater uncertainty. Thus probabilistic electric load forecast methods become more important especially to the power system planning and operations [1].

There are plenty of works on the probabilistic load forecasting. Most of these methods can be categorized into three categories. The first is to construct probabilistic models through analyzing the relations between various features. For example, Charytoniuk *et al.* [6] provided a nonparametric method for forecasting power demands. This method focused on discovering relations between customer demand, temperature, and time.The second is to apply time series theories to probabilistic forecasts. For example, a semi-parametric regression model was proposed by the second top team in the forecast competition GEFCom2014 to make a probabilistic forecast [7]. In the meantime, Markov Chain is also used to obtain probabilistic forecast [8]. The third is to construct distributions through quantile regressions is a method widely used in power load forecasts. For example, He *et al.*introduced quantile regressions methods applied to machine learning algorithms such as SVM, neural networks to provide the lower and upper bound of the forecasts. Some statistic theory like copula and kernel functions are used in this method [9], [10].

In the probabilistic load forecast, the randomness in the train data sample and the uncertainty in the models themselves determine the stochastic distribution of the forecast results. For example, a two-or-three-year sample is not comprehensive enough to cover all the situations in the reality. Its randomness lowers the confidence level of the sample data. Also, the inertial properties of algorithms such as stabilities may influence the confidence interval of a forecast.

In order to evaluate the uncertainty of models, Liu *et al.* proposed a quantile regression algorithm averaging on sister forecasts, where the sister forecasts are different models trained on different features [11]. However, this method is weakened when lack of features.

To summarize, the contributions of this paper are as follows:

1) Proposing a bootstrap based probabilistic load forecasting ensemble method to produce probabilistic forecast without complex model rebuilding process;
2) Conducting case study on a real dataset to verify the superiority of the proposed method.

The rest of this paper is organized as follows. Section II presents our methodology of constructing probabilistic Distributions for point forecast methods. Section III details the evaluation and comparison of the methodology. Section IV shows a case study and its results. Finally, Section V gives the conclusions of this paper.

[1] Jiawei Zhang is with the Department of Physics, Tsinghua University, Beijing, China
[2] Yi Wang, Ning Zhang, and Chongqing Kang are with the Department of Electrical Engineering, Tsinghua University, Beijing, China
[3] Mingyang Sun is with the Department of Electrical & Electronic Engineering, Imperial College London, London, UK.
[*] Corresponding author: ningzhang@tsinghua.edu.cn
This work was supported by National Key R&D Program of China (No. 2016YFB0900100)

II. Methodology

The proposed method mines future uncertainties from two aspects: 1) we use bootstrap on training dataset to discover the randomness of the dataset itself; 2) we use various forecasting methods (such as RF and GBRT) to discover the uncertainty from the regression models.

A. Motivation

How to depict the randomness of the dataset and the uncertainty of the model is the first step to deal with. Supposed that we have M algorithms, bootstrap method samples the dataset with a replacement for $M \times B$ times, construct $M \times B$ sample sets differ in detail. After training these sample sets through M algorithms, we get $M \times B$ models. Since B may not be large due to the computing power, we shall bootstrap these models again for B' times. Finally, we obtain B' groups of forecasts with $M \times B$ items for each, which are sufficient to describe the forecast properties.

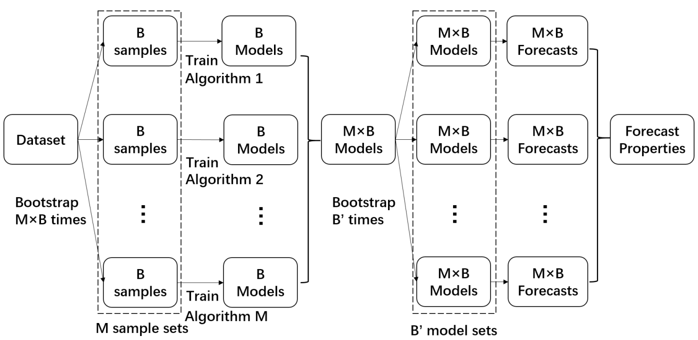

Fig. 1. The flow chart of motivation

B. Bootstrap

Bootstrap is a method that reduces bias while constructs intervals for parameters or estimators. Bradley Efron first introduced this method to the statistics in 1979 [12]. Thanks to the development of computation technology, bootstrap became popular in the following decades.

The core spirit of bootstrap is the process of randomly sampling with replacement. For example, Let

$$\mathbf{X} = (X_1, \cdots, X_n) \tag{1}$$

be a sample of the certain stochastic event. Since n is finite, there must be some information lost when sampled from the whole sample space. For example, in practice, we are supposed to choose the data two or three years before to train our models for electric load forecast, but actually, the two-or-three-year episode is incapable to cover all kinds of stochastic patterns in the real world power system.

The bootstrap method helps a lot to get different samples of the train sets by sampling with replacement. After B times of sampling with replacement, we obtain B train sets in form of

$$\mathbf{X}_j^* = (X_{j1}^*, \cdots, X_{jn}^*),\ j = 1, \cdots, B \tag{2}$$

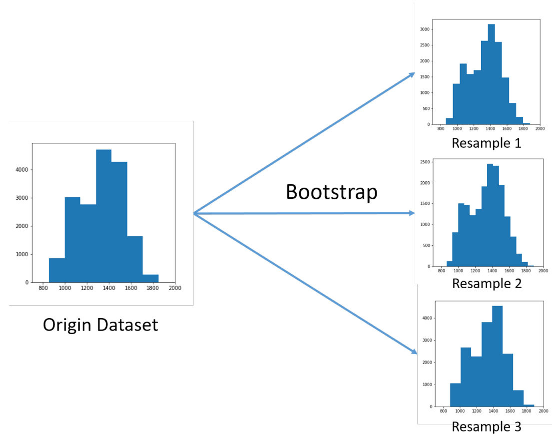

Fig. 2. An example of bootstrap

Note that the length of $\mathbf{X}_j^*$ is equivalent to one of the sample set $\mathbf{X}$. These new sample sets can simulate the randomness of the historical data. i.e. some records in the original data set are dismissed or repeated in the B new train sets, while these B new sample set are likely to resemble the original sample on the whole.

C. Bootstrap on train data set

Now apply the thought of bootstrap into power load forecast. Let $X_i,\ i = 1, 2, \cdots, n$ be the vectors in the train set consisting target actual power load or power demand and features such as historical power load, time, temperature and so on.

If there are M point forecast algorithms such as random forests and GBRT available to build forecast models, we sample the train set for $M \times B$ times and then train $M \times B$ models with these $M \times B$ new train sets. Note that we should tune the parameters using cross-validation to make sure these M algorithms perform well individually before this procedure.

As for the choice of B, it depends on the computation performance and the expectation on time consumption. In practice, the larger choice of B, the better their models perform. Fortunately, though lack of high-performance computer, the following steps can help to build acceptable intervals of the forecast results. Also, you can choose different B value for each algorithm, if you have some knowledge of the property and efficiency of these algorithms before this step.

D. Bootstrap on models

The classical bootstrap method samples the set of vectors with replacement, to build new sample sets. The $M \times B$ models we build through bootstrap can also be considered as *vectors* in functional space. The ensemble of these models contributes to a new model set $\mathbf{F}$ consists of $M \times B$ items.

$$\mathbf{F} = (F_1, \cdots, F_n),\ n = M \times B \tag{3}$$

Sample with replacement for B' times, we get B' new model sets.

$$\mathbf{F}_j^* = (F_{j1}^*, \cdots, F_{jn}^*),\ j = 1, \cdots, B',\ n = M \times B \tag{4}$$

Each of these model sets provides a distribution of forecast results, and these new model sets reflect the uncertainty of the models themselves.

$$\mathbf{Q} = (Q_1, \cdots, Q_m),\ m = B' \tag{5}$$

where Q_j, $j = 1, \cdots, B'$ stands for the distribution of forecast by $\mathbf{F}^*_j$, $j = 1, \cdots, B'$.

Since the time consumed on the forecast process is much shorter than that in the training process, it is acceptable for us to set a larger B'.

E. Estimate bounds

we can naively assume that the mean value of the forecast distribution is a good estimate of the point forecast. By the collection of the mean values in Q_j, a distribution of the point forecast will be constructed. The interval of the distribution is narrow and its mean value can be determined as the point forecast result.

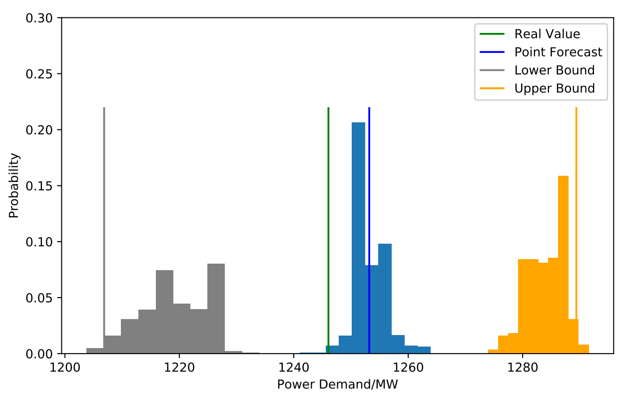

Fig. 3. An example of bound estimate

By counting the 2.5% quantiles and 97.5% quantiles in the Q_js' distributions obtained before, we construct the distributions of the two significant statistics, the 2.5% and 97.5% quantiles.

In Fig. 3, the distribution on the left (gray) represents the distribution of 2.5% quantiles counted from Q_js, while the middle (blue) and right (orange) ones show the distributions of the mean values and the 97.5% quantiles of Q_js'.

The bound of the forecast can be determined as follows. We take the 2.5% quantile of the 2.5% quantile distribution and the 97.5% quantile of the 97.5% quantile distribution as the lower and upper bound of the forecast, which are the gray line and the orange line shown in Fig. 3.

Fig. 4 summarizes the flow chart about the methodology mentioned above.

It should be noted that some bagging algorithms such as RF also utilize bootstrap to build new sample sets. However, these models trained after bootstrap are weak-learners, which means that the predictions from this models are not that accurate. Through casting votes or calculating the average, these weak-learners combine to be a strong-learner that gives out point forecasts with high accuracy [13]. The proposed method differs RF and other similar ensemble methods in two aspects: First, traditional point ensemble methods average multiple forecasts as the final forecast; while the proposed method use bootstrap to get the distribution from multiple forecasts. Second, RF produces *good enough* forecast based on weak-learners; while the proposed method needs *strong-learners* to guarantee the credibility of the point forecasts.

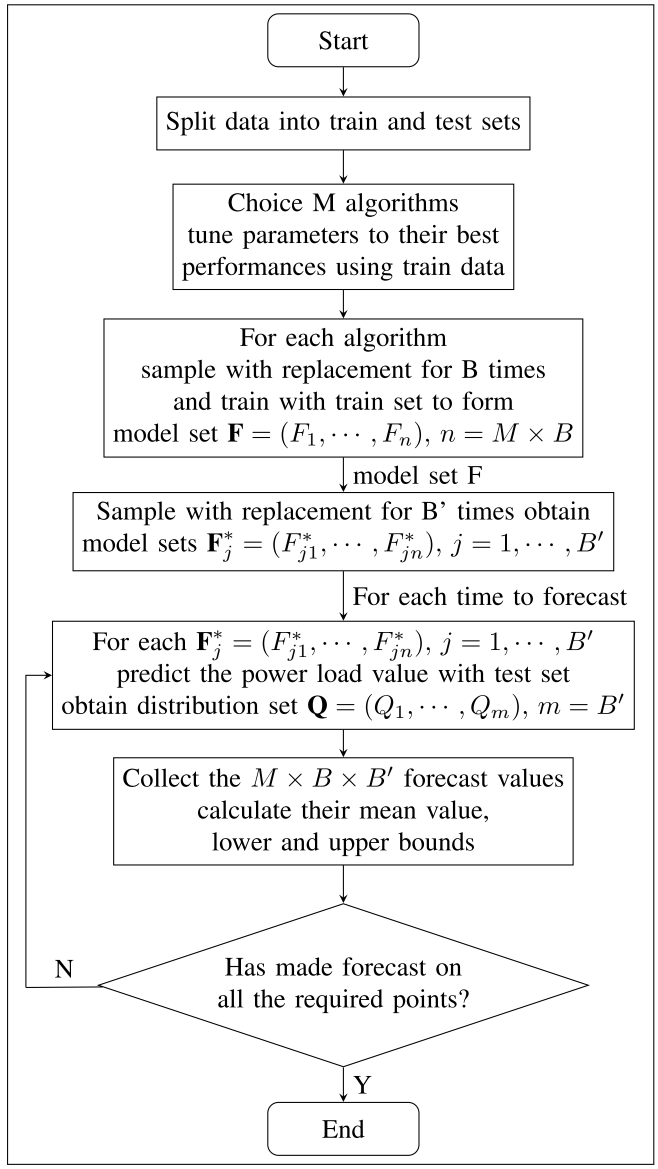

Fig. 4. The flow chart of the proposed methodology

III. Evaluation Criteria and Comparison

In this section, we will introduce several evaluation criteria for the power load forecast methods. Also, two quantile methods we are supposed to compare with are briefly discussed here.

A. Criteria

1) MAPE: Mean absolute percentage error (MAPE) is a common criterion to evaluate the accuracy of a point forecast or estimate.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{\hat{y}_i - y_i}{y_i}| \times 100\% \qquad (6)$$

The $\hat{y}_i$ denotes the forecast or estimate value, while the y_i denotes the true value under the index i. Though bootstrap can offer a distribution of a forecast, the accuracy of point forecast is still an important index of the model.

2) Minimum cover index (MCI): Hypothesis test is a phenomenal concept widely used in statistics, and its spirit is to construct a test to distinguish the the null hypothesis and the alternative hypothesis.

We define the null hypothesis and the alternative hypothesis at a significance of α as

H_0: $\hat{y}_{lb} \cdot (1-\alpha) < y_{real} < \hat{y}_{ub} \cdot (1+\alpha)$

H_1: Otherwise.

where $\hat{y}_{lb}$ and $\hat{y}_{ub}$ stands for the estimated lower bound and upper bound, respectively.

To simplify the problem, the test function is defined as follows,

$$\varphi(x) = \begin{cases} 1 & \theta \in \Theta_0 \\ 0 & \theta \in \Theta_1 \end{cases} \qquad (7)$$

To sum up the test function for each time point in the predicted period, and divide it by the length of total time points calculated, the result represents the cover capability of an algorithm.

Define the cover capability ψ as follows,

$$\psi(\alpha) = (\sum_i \varphi(t_i))/n,\ i = 1, \cdots, n \qquad (8)$$

where n is the length of the time period.

We define Minimum cover index as the minimum α when the cover capability reaches 95%.

3) Average interval width (AIW): Average interval width evaluates the ability to reduce intervals of an algorithm. In real practice, for example, too wide bounds of a forecast may cause wastes on reserved generators in power plants. Average interval width W is defined as

$$W(\alpha) = \sum_i (\hat{y}_{ubi} \cdot (1+\alpha) - \hat{y}_{lbi} \cdot (1-\alpha))/n, \qquad (9)$$

where α is the MCI defined in III-A2, $\hat{y}_{ubi}$ and $\hat{y}_{lbi}$ represent the estimated upper and lower bounds for i^{th} item, respectively, $i = 1, \cdots, n$.

B. Comparison

Quantile regression methods are typical algorithms to construct probabilistic distributions of a forecast. Here, we use two quantile regression methods as follows:

- Quantile Random Forest (Q-RF): Q-RF uses RF to regress the quantiles of the probabilistic forecast.
- Quantile GBRT (Q-GBRT): Q-BGRT uses gradient boosting regression tree algorithm to regress the quantile of the probabilistic forecast.

IV. Numerical Experiments

A. Dataset

ISO New England data set contains sufficient information about the electricity market in the New England region, the USA, such as time, day ahead demand, real time demand, Locational Marginal Price (LMP), dry-bulb temperature and dew-point temperature. Since we focus on power load forecast, real time demand is selected as the value to predict, while the information about time, real time demand, dry-bulb temperature and dew-point temperature on the same hour in the past seven days are chosen as features.

In detail, the features of the forecasting contain three parts as follows. Note t here is the time to be predicted represented in hour.

- The time information vector:

$$X_{ti} = (Month, Date, Weekday, Hour)_i \qquad (10)$$

- The previous load vector:

$$X_{load\,i} = (load_{t-24\cdot 1}, \cdots, load_{t-24\cdot 7}) \qquad (11)$$

- The previous dry-bulb temperature vector:

$$X_{dry_i} = (dry_{t-24\cdot 1}, \cdots, dry_{t-24\cdot 7}) \qquad (12)$$

- The previous dew-point temperature vector:

$$X_{dew\,i} = (dew_{t-24\cdot 1}, \cdots, dew_{t-24\cdot 7}) \qquad (13)$$

The combination of these three sub-vectors is the feature vector

$$X_i = (X_{ti}, X_{load\,i}, X_{dry_i}, X_{dew\,i}) \qquad (14)$$

B. Numerical Results

RF and GBRT are famous ensemble algorithms based on decision trees. To simplify the problem, we only choose this two machine learning methods and the bootstrap times B are set equally to 50 for each algorithm. In the meantime, B' is set to 500 when estimating the bounds.

We choose two year features with their real-time demand as training set(from 05/21/2014 to 05/21/2016), and make forecasts for the following 7×24 hours.

Take the forecast from 05/22/2016 to 05/28/2016 (Sunday to Saturday) as an example, and its result is shown in Fig. 5. The mean value of the distribution is taken as a point forecast. The MAPE of the mentioned week is 1.99%, which is also acceptable as a point forecast.

Moreover, most of the real-time demand values fall within the bounds of the forecast, while few points are outside the bound especially in the valleys of the demand. The width of the forecast is narrow when the demand increases and decreases rapidly and wide at the peaks and valleys for each day. From Fig. 6., the cover capability defined in Eq. 8. will be greater than 95% when α is above 0.03, the MCI of 0.03 only contributes an addition on the bound width around 60 MWs.

Under the MCI of 0.03, we construct the distribution of bound width from 05/22/2016 to 05/28/2016 in Fig. 7. The

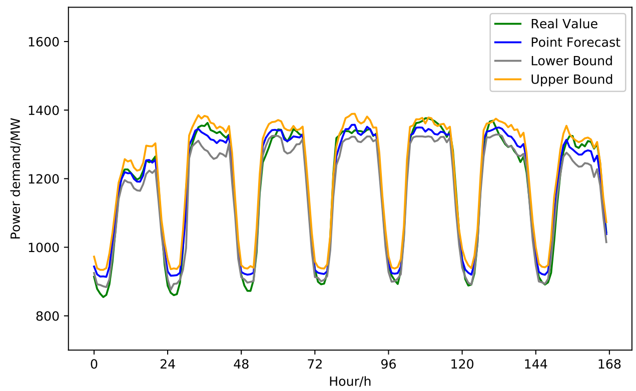

Fig. 5. The forecasts from 05/22/2016 to 05/28/2016

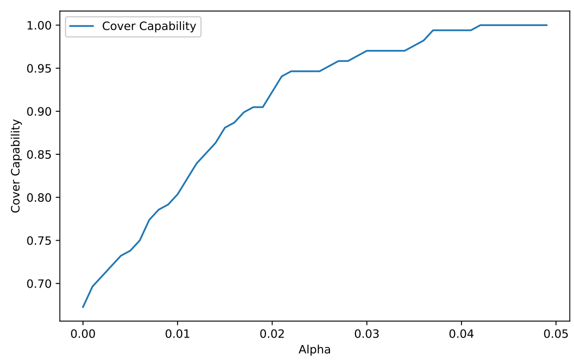

Fig. 6. The cover capability of the algorithm

mean band with is 127 here, approximately 10% of the average demand.

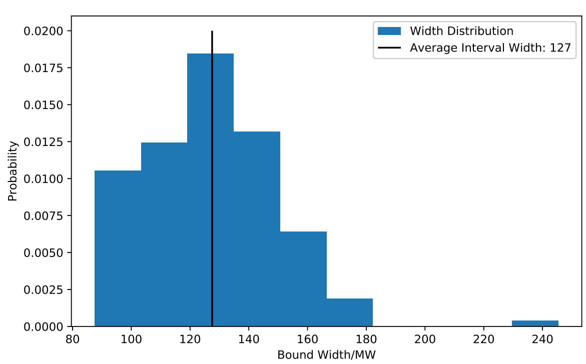

Fig. 7. The distribution of bound width

C. Comparison

1) Q-RF: The number of trees is set to 500, which is equivalent to the one in the numerical experiments. Also, 10%

TABLE I
THE PERFORMANCE OF THREE MODELS

Methods	MAPE/%	MCI	AIW/MW
Proposed Model	**1.99**	**0.03**	**127**
Q-RF	2.91	0.03	262
Q-GBRT	2.33	0.04	168

and 90% quantiles are chosen as bounds, and the 50% quantile represents the point forecast of power load here. The package *quantregForest* in R provides us with the quantile random forest method here [14].

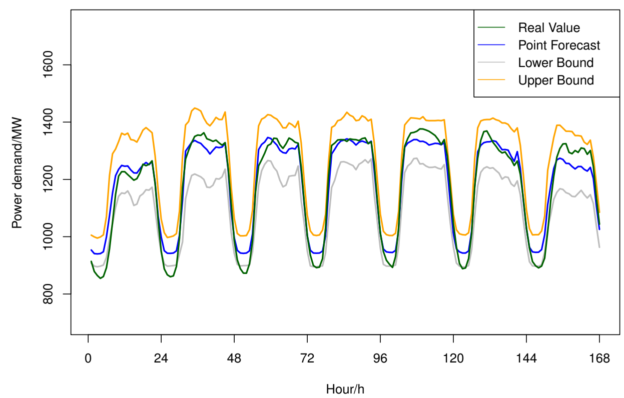

Fig. 8. Q-RF forecasts from 05/22/2016 to 05/28/2016

2) Q-GBRT: Similar to the settings in the quantile random forest, 10% and 90% quantiles are chosen as bounds, while the half quantile represents the point forecast. The package *sklearn* provide the *GradientBoostingRegressor* function with quantile regression model here [13].

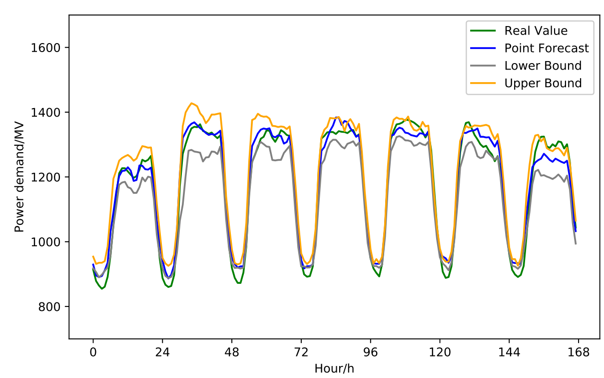

Fig. 9. Q-GBRT forecasts from 05/22/2016 to 05/28/2016

D. Discussion

Since the intervals for the quantile regression in IV-C are wide, We choose 10% and 90% quantiles in the quantile method instead of 2.5% and 97.5% quantiles same as in the IV-B. If the performance in our model dwarfs the ones' in the quantile models on the criteria mentioned above, we can declare confidently that our model is better in this case.

Fig. 8 and Fig. 9 show the results of the two quantile methods. The performance of three models is presented in Table I.

Comparing these two methods with the bootstrap method put forward in this paper, our model performs better on the point forecast, requires smaller MCI to cover all the points and has the narrowest bound width in this case.

V. Conclusions and Future Work

The paper proposes a method to construct probability load forecast from point algorithms, without changing the details about the algorithms. The bootstrap method helps to build different train data sets to simulate different kinds of possible cases, while no more information outside the original train set is wrapped in. Also, this model provides an idea to build probability load forecast combining different kinds of algorithms, which is hard to realize by quantile regression. To demonstrate the efficiency of the model, MAPE, the cover capability and the average interval width are selected as a criterion. The result shows that the proposed method can increase the accuracy while narrow the width.

In this paper, all train set data are involved in the bootstrap procedure to build new train sets. However, in fact, for example, when forecasting the load in July, the historical information in the adjacent months, June, July, and August play significant rolls. We can increase the probability of items to be re-sampled by bootstrap in these months to emphasize their power to the forecast.

Furthermore, we can extend the point forecast algorithms to quantile regression algorithms using the bootstrap method. For instance, we can build the forecast intervals for 40% and 60% quantiles using the method in this paper, and the lower bound of 40% quantiles and upper bound of 60% are selected as the bounds of the forecast. In this case, we take both the advantage of quantile regressions which give out probabilistic forecasts and avoid the too wide interval such as 10% and 90% quantiles in classical quantile regression methods.

References

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[2] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4356–4364, Nov 2013.

[3] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy load forecasting using empirical mode decomposition and support vector regression," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 549–556, March 2013.

[4] G. Dudek, *Short-Term Load Forecasting Using Random Forests*. Springer International Publishing, 2015.

[5] C. Persson, P. Bacher, T. Shiga, and H. Madsen, "Multi-site solar power forecasting using gradient boosted regression trees," *Solar Energy*, vol. 150, pp. 423–436, 2017.

[6] W. Charytoniuk, M. S. Chen, P. Kotas, and P. Van Olinda, "Demand forecasting in power distribution systems using nonparametric probability density estimation," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1200–1206, 1999.

[7] V. Dordonnat, A. Pichavant, and A. Pierrot, "Gefcom2014 probabilistic electric load forecasting using time series and semi-parametric regression models," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1005–1011, 2016.

[8] M. J. Sanjari and H. B. Gooi, "Probabilistic forecast of pv power generation based on higher order markov chain," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2942–2952, July 2017.

[9] Y. He, Q. Xu, J. Wan, and S. Yang, "Short-term power load probability density forecasting based on quantile regression neural network and triangle kernel function," *Energy*, vol. 114, pp. 498–512, 2016.

[10] Y. He, R. Liu, H. Li, S. Wang, and X. Lu, "Short-term power load probability density forecasting method using kernel-based support vector quantile regression and copula theory," *Applied Energy*, vol. 185, pp. 254–266, 2017.

[11] B. Liu, J. Nowotarski, T. Hong, and R. Weron, "Probabilistic load forecasting via quantile regression averaging on sister forecasts," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2015.

[12] B. Efron, "Bootstrap methods: Another look at the jackknife," vol. 7, no. 1, pp. 1–26, 1979.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] N. Meinshausen, "quantregforest: Quantile regression forests," *R Package Version*, 2016.