A Short-Term Wind Power Forecasting Approach With Adjustment of Numerical Weather Prediction Input by Data Mining

Qianyao Xu, *Student Member, IEEE*, Dawei He, Ning Zhang, *Member, IEEE*, Chongqing Kang, *Senior Member, IEEE*, Qing Xia, *Senior Member, IEEE*, Jianhua Bai, and Junhui Huang

***Abstract*—This paper proposes a novel short-term wind power forecasting approach by mining the bad data of numerical weather prediction (NWP). Today's short-term wind power forecast (WPF) highly depends on the NWP, which contributes the most in the WPF error. This paper first introduces a bad data analyzer to fully study the relationship between the WPF error with several new extracted features from the raw NWP. Second, a hierarchical structure is proposed, which is composed of a *K*-means clustering-based bad data detection module and a neural network (NN)-based forecasting module. In the NN module, the WPF is fully adjusted based on the output of the bad data analyzer. Simulations are performed comparing with two other different methods. It proves that the proposed approach can improve the short-term wind power forecasting by effectively identifying and adjusting the errors from NWP.**

***Index Terms*—Artificial neural network, data adjustment, feature selection, numerical weather prediction, wind power forecast error.**

NOMENCLATURE

The notation used is provided below.

Indices:

t	Time period.
s	Superscript indicating a value regarding wind speed.
p	Superscript indicating a value regarding wind power.
i	Element i in a vector.
h	Superscript indicting the forecast time horizon.
N^h	Number of forecast points for the forecast time horizon h.

Sets:

S_t	Wind power forecast (WPF) data set in time period t.

Parameters:

WSF_t	Wind speed forecast (WSF) in time period t.
WSF'_t	Original WSF value in time period t.
WPO_t	Wind power output (WPO) measurement in time period t.
WPO'_t	Original WPO measurement in time period t.
WPF_t	WPF in time period t.
WFC	Wind farm capacity.
ε_t	WPF error in time period t.
PV_t^s	Peak-valley difference of the WSF in S_t.
PV_t^p	Peak-valley difference of the WPO measurement in S_t.
μ_t^s	Average value of the WSF in S_t.
μ_t^p	Average value of the WPO measurement in S_t.
$\mathrm{RU}_t^s, \mathrm{RD}_t^s$	Maximum ramp up/down of the WSF in S_t.
$\mathrm{RU}_t^p, \mathrm{RD}_t^p$	Maximum ramp up/down of the WPO measurement in S_t.
σ_t^s	Standard deviation of the WSF in S_t.
σ_t^p	Standard deviation of the WPO measurement in S_t.
α	Factor used to adjust the threshold value, within [0,1].
RMSE^h	Root-mean-square error (RMSE) for the forecasts with time horizon h.
WPF_ω^h	WPF result for the ωth forecast point with time horizon h.
$\widehat{\mathrm{WPO}_\omega^h}$	Actual WPO for the ωth forecast point with time horizon h.
$\widehat{\varepsilon_t}$	Estimated error rate as the output of data adjustment engine.

Functions:

$\max(\cdot)$	Function returning the maximum value of a vector.
$\min(\cdot)$	Function returning the minimum value of a vector.
$\mathrm{num}(\cdot)$	Function returning the element number of a vector or a set.
$I(\cdot)$	Function to judge whether a new WPF belongs to a certain abnormal WPF (AWPF) cluster.

Manuscript received November 30, 2014; revised March 22, 2015; accepted April 23, 2015. Date of publication June 02, 2015; date of current version September 16, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 51307092 and Grant 51325702 and in part by the Research Projects Sponsored by State Grid Corporation of China. Paper no. TSTE-00676-2014. *(Corresponding author: Chongqing Kang.)*

Q. Xu, N. Zhang, C. Kang, and Q. Xia are with the State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: cqkang@tsinghua.edu.cn).

D. He is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.

J. Bai is with the State Grid Energy Research Institute, Beijing 102209, China.

J. Huang is with Jiangsu Electric Power Company, Nanjing 210024, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSTE.2015.2429586

I. Introduction

WIND POWER has a rapid development throughout the world recently. However, it brings about great influences and challenges into power system operation due to its intermittent and nonschedulable nature [1]. To guarantee a reliable system operation, the independent system operator (ISO) has to schedule sufficient spinning reserve for wind power. It is acknowledged by most of the researchers that a more accurate WPF could help in reducing the cost of supplying spinning reserve without losing the reliability of power system operation [2]–[4].

The WPO is affected by various factors such as wind speed, wind direction, temperature, turbine type, turbine position, terrain roughness, air density, and wake effect [5]. Numerous approaches to forecast wind power in a single site have been developed, and generally they can be classified into the following three categories: 1) model-driven approaches; 2) data-driven approaches; and 3) ensemble approaches.

The solutions in the first category mainly focus on the physical model analysis [6], and reveal the influence of the relevant factors on WPO from the physical point of view. These solutions are capable of establishing a specialized WPF model for a specialized wind farm. Meanwhile, they do not require large amount of historic data to train the model. They can perform very well in the long-term WPF as well [7]. However, these approaches need abundant meteorological knowledge and detailed physical characteristics of wind turbines to achieve an accurate model. The accuracy of the modeling is critical to the forecast performance.

To overcome the drawbacks of the model-driven solutions, many data-driven mathematical statistics approaches or artificial intelligence approaches are proposed [8]–[11]. The data used for model training in these approaches are numerical weather prediction (NWP)-related indices such as wind speed, wind direction, and air temperature. Several famous statistical methods are: persistence method (PM) [12], auto-regressive and moving average (ARMA) [13], auto-regressive integrated moving average (ARIMA) [14], Gaussian process (GP) [15], Kalman filtering (KF) [16], etc. The artificial intelligence methods include neural network (NN) [17], support vector machine (SVM) [18], evolutionary algorithms (EA) [19], and wavelet analysis (WA) [20]. The approaches in this category are self-adaptive to types of turbine types, diverse geographical terrains, and different wind farm locations, avoiding the relatively weak influence of other factors, e.g., air density. However, they all require a large amount of historic data as well as a sufficient training process. The model cannot guarantee a reliable estimation for long look ahead of time and are, therefore, usually used for short-term WPF.

Later, the research moves forward to the combination of model-driven and data-driven solutions [21]. The ensemble approaches are not only able to take advantages of each single approach but also to some extent avoid some drawbacks by selecting a reasonable model combination. Comparing to the approaches of the other two categories, the ensemble approaches would theoretically give relatively better results [22]. Several studies have proved its validity [23]. However, as discussed in [23], although all of the above-mentioned works contribute a lot to the development of WPF area, the short-term WPF is still a young area and needs more contributions from different aspects.

All the solutions mentioned above only focus on the improvement of the forecast engine itself. This paper follows up the author's idea in [24] and tries to improve the wind power forecasting accuracy in a different manner.

Among all the factors that influence the accuracy of WPF, the WSF derived from NWP is acknowledged as the most significant [24]. However, the WSF usually comes from a third party such as National Weather Service Center, and its accuracy is still not high enough to support the next-stage WPF. There are already a few works on the wind speed data adjustment [25]. In [25], the authors evaluated NWP error point by point in the scale of per hour; however, these NWP errors is more likely to follow certain patterns that lasts for a longer time, e.g., 12 h. In [24], the authors observe the NWP waveform, and manually select four major types of errors, which can be categorized as miss-peak faults and wrong-peak faults. However, the number of error types is obviously not supposed to be limited to four and a generalized data-mining-based solution should be studied.

This paper proposes a generalized bad data identification method for the raw data of NWP aiming to improve the performance of WPO forecast. The *K*-means algorithm [26] acts as the NWP clustering solution. The Bayes information criterion (BIC) [27] is employed to identify the bad data. Finally, the NN is applied as the main forecasting engine. Case study is carried out using the data of wind farms from Global Energy Forecasting Competition 2012-Wind Forecasting (GEFCOM2012-WF) [28], [29].

This paper is organized as follow. Section II presents the core idea and framework of the solution. Section III gives the detailed explanation of each module in the proposed solution. Section IV carries out the results of the simulation and Section V concludes the paper and summarizes the future work.

II. Idea and Framework

A. Dataset and Definitions

To better present the idea of this paper, several definitions are given first. In WPF, three types of datasets are usually used: 1) the historic WPO; 2) the historic WSF; and 3) the future WSF.

The WPF error rate is usually defined as the ratio of the absolute forecast error and the total installed wind power capacity

$$\varepsilon_t = \frac{|\mathrm{WPF}_t - \mathrm{WPO}_t|}{\mathrm{WFC}} \times 100\%. \tag{1}$$

An AWPF is defined as a forecast with an error rate greater than an assigned threshold value, whereas other forecasts with error rates less than the threshold value are regarded as a normal WPF (NWPF). For each forecast, in addition to the forecasted wind power, a corresponding data struct is generated, which includes the past T_1-hour's WSF, the future T_2-hour's WSF,

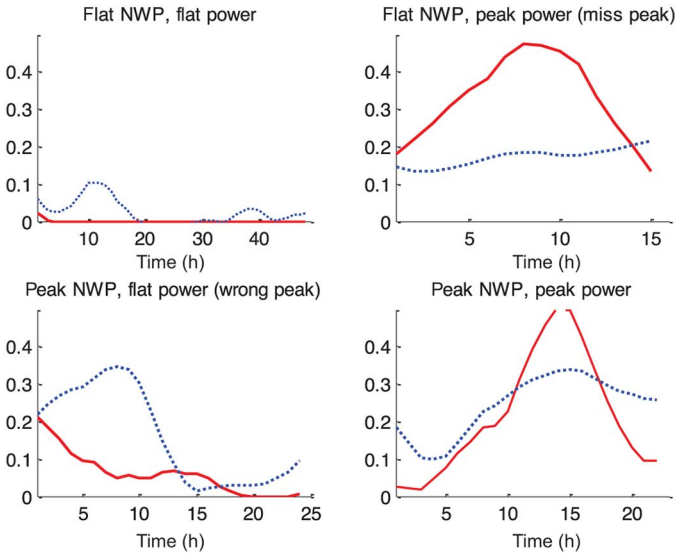

Fig. 1. Four different types of error patterns [24]. Black solid line is wind power curve and blue-dotted line is regularized NWP curve.

and the past T_3-hour's WPO. Equation (2) gives a more clear illustration of the data struct for forecasting wind power in time period t, denoted as S_t. In addition, each S_t contains a binary attribute, i.e., AWPF or NWPF. In this paper, T_1, T_2, and T_3 are all set as 12

$$S_t = \left\{ \begin{array}{c} \mathrm{WSF}_{t-T_1}, \ldots, \mathrm{WSF}_{t-1} \\ \mathrm{WSF}_t, \ldots, \mathrm{WSF}_{t+T_2-1} \\ \mathrm{WPO}_{t-T_3}, \ldots, \mathrm{WPO}_{t-1} \end{array} \right\}. \tag{2}$$

B. Core Idea of the Proposed Work

In [24], an observation on WSF with a 24-h horizon has been implemented and four obvious types of error patterns have been extracted as shown in Fig. 1. These errors may happen because of the abnormal temperature, air pressure, air density, or even some other special patterns. No matter what reason, once the WSF has large deviation, the accuracy of the WPF would be spoiled.

Instead of identifying the above four patterns, we try to propose a general way of finding out the errors in NWP raw data using data-mining technique and embed this method into the WFO procedure.

The overall flowchart of this idea is illustrated in Fig. 2. For every WPF point, the system first generates a corresponding *struct* as the preprocessing. Then, the data of the *struct* will run through an abnormal detection engine and is identified as AWPF or NWPF. The NWPFs are directly sent to the WPF engine to obtain the WFO results, whereas AWPFs are sent to the data adjustment engine based on their types of error pattern to calculate the estimated WPF deviation. This estimated deviation will be used to compensate the result directly from the WPF engine.

To implement the framework, each module in Fig. 2 is required to be well trained before the real application. Next section will explain each module in detail.

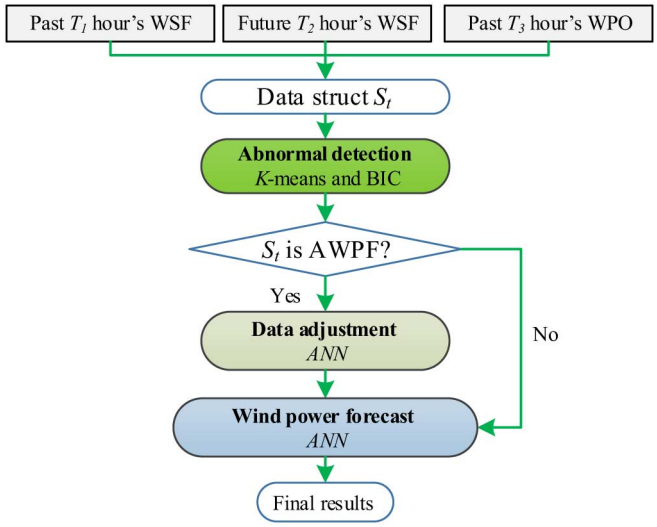

Fig. 2. Flowchart of the short-term wind power forecasting system.

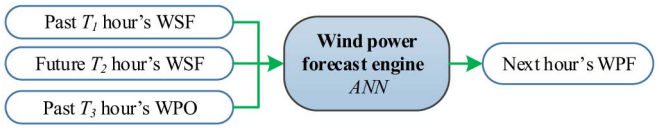

Fig. 3. Framework of WPF engine.

III. Method Introduction

A. WPF Engine

The WPF engine forms the basis of the whole forecasting system. As shown in Fig. 3, the artificial NN (ANN) is adopted to run as the forecast engine. The inputs are the same as those in Fig. 2. The output of forecast engine is next 1-hour's WPF. In this paper, the forecasting process is run in a rolling base, hour by hour, as shown in Fig. 4. This means that at each time, only one hourly WPF result could be achieved, and when forecasting later wind power, the previous achieved WPF results would be treated as the input WPO data.

To improve the efficiency and performance of the ANN, data preprocessing is implemented. In detail, wind speed usually ranges from 0 to 25 m/s for a wind turbine; therefore, the input speeds are normalized to [0, 1] using a linear transformation as in (3). Moreover, the wind power is normalized to [0, 1] as well by taking the WFC as the base value, as in (4)

$$\mathrm{WSF}_t = \frac{\mathrm{WSF}'_t}{25} \tag{3}$$

$$\mathrm{WPO}_t = \frac{\mathrm{WPO}'_t}{\mathrm{WFC}}. \tag{4}$$

Empirically, one-hidden layer with multilayer perceptrons (MLPs) is a nonlinear function that is proved to be able to approximate any normal function to an arbitrary degree of accuracy; therefore, it is sufficient for a 12-h WPF. Moreover, model selection techniques are usually required to specify the number of neurons for each layer. However, since the NN design is not the core work in this paper, the network is finalized experimentally with 36 neurons for input layer, 4 neurons for hidden

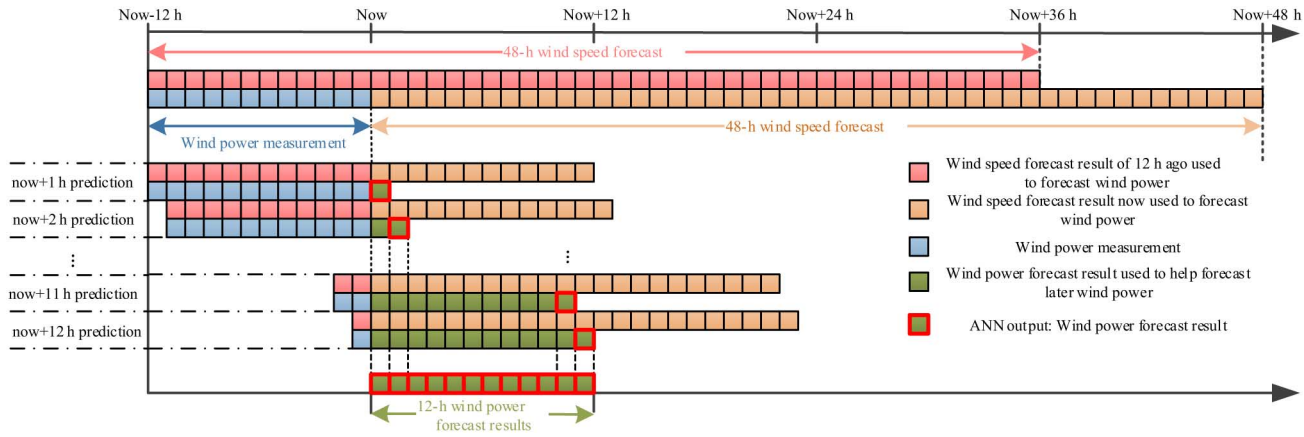

Fig. 4. Illustration of WPF process.

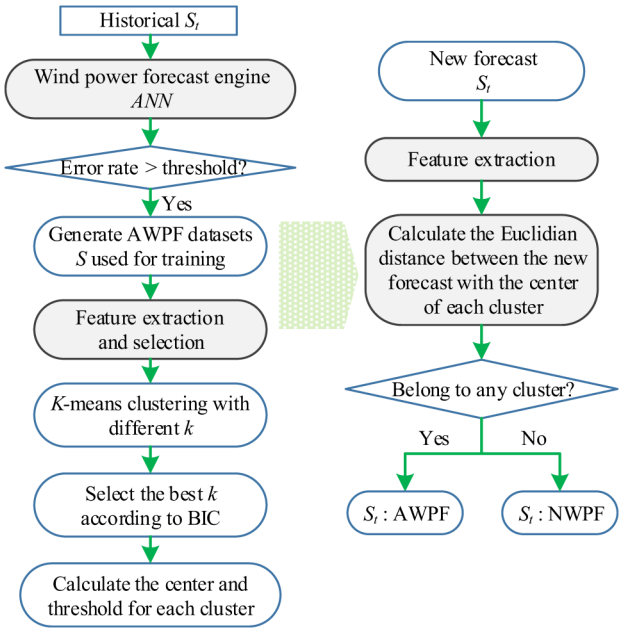

Fig. 5. Framework of abnormal detection module.

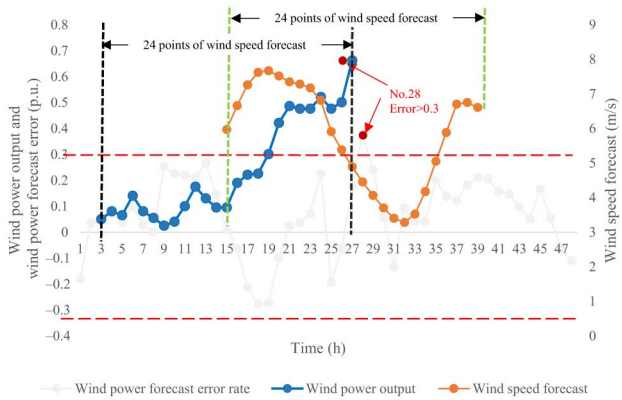

Fig. 6. Example of AWPF dataset.

layer, and 1 neuron for output layer. The input layer and hidden layer use log-sigmoid transfer function, and the output layer uses linear transfer function. The forecaster is trained with back propagation (BP).

B. Abnormal Detection

The abnormal detection module is implemented based on the K-means clustering method and the BIC. Fig. 5 gives the flowchart of the abnormal detection module.

First, the AWPF data set S needs to be generated for K-means learning. As shown in Fig. 5, all the data used to generate the AWPF datasets come from the historic WPO and WPF. The forecast engine is run for each historic time point and gets the WPF of that time. Moreover, the WPO forecast is compared with the corresponding WPO of that time point. Based on the definition of the AWPF, the forecast result whose error rate is higher than a threshold, i.e., 30%, will be selected, with its corresponding S_t. An example is shown in Fig. 6.

Next, a feature vector that can represent the raw AWPF dataset is extracted for each forecast. The feature vector should possess the capability to facilitate the clustering algorithm to distinguish the AWPF effectively and efficiently. Usually, those features associated with the same type of abnormal forecast are supposed to be close with each other. In this paper, several features are extracted as shown below. The features extracted from the real WSFs in time period t include

$$\mathrm{WSF}_{t-12},\ldots,\mathrm{WSF}_{t-1},\mathrm{WSF}_t,\ldots,\mathrm{WSF}_{t+11} \quad (5)$$

$$\begin{aligned}\mathrm{PV}_t^s = &\max(\mathrm{WSF}_{t-12},\ldots,\mathrm{WSF}_t,\ldots,\mathrm{WSF}_{t+11})\\ &-\min(\mathrm{WSF}_{t-12},\ldots,\mathrm{WSF}_t,\ldots,\mathrm{WSF}_{t+11})\end{aligned} \quad (6)$$

$$\mu_t^s = \frac{\sum_{i=t-12}^{t+11}\mathrm{WSF}_i}{\mathrm{num}\,(\mathrm{WSF}_{t-12},\ldots,\mathrm{WSF}_t,\ldots,\mathrm{WSF}_{t+11})} \quad (7)$$

$$\mathrm{RU}_t^s = \max\left(\mathrm{WSF}_{i+1}-\mathrm{WSF}_i\,|i=t-12,\ldots,t+10\right) \quad (8)$$

$$\mathrm{RD}_t^s = \max\left(\mathrm{WSF}_i-\mathrm{WSF}_{i+1}\,|i=t-12,\ldots,t+10\right) \quad (9)$$

$$\sigma_t^s = \sqrt{\frac{\sum_{i=t-12}^{t+11}(\mathrm{WSF}_i-\mu_t^s)^2}{\mathrm{num}\,(\mathrm{WSF}_{t-12},\ldots,\mathrm{WSF}_t,\ldots,\mathrm{WSF}_{t+11})}}. \quad (10)$$

The features extracted from the real WPOs include

$$\mathrm{WPO}_{t-12}, \mathrm{WPO}_{t-11}, \ldots, \mathrm{WPO}_{t-1} \tag{11}$$

$$\begin{aligned}\mathrm{PV}_t^p = &\max(\mathrm{WPO}_{t-12}, \mathrm{WPO}_{t-11}, \ldots, \mathrm{WPO}_{t-1}) \\ &- \min(\mathrm{WPO}_{t-12}, \mathrm{WPO}_{t-11}, \ldots, \mathrm{WPO}_{t-1})\end{aligned} \tag{12}$$

$$\mu_t^p = \frac{\sum_{i=t-12}^{t-1} \mathrm{WPO}_i}{\mathrm{num}\left(\mathrm{WPO}_{t-12}, \mathrm{WPO}_{t-11}, \ldots, \mathrm{WPO}_{t-1}\right)} \tag{13}$$

$$\mathrm{RU}_t^p = \max\left(\mathrm{WPO}_{i+1} - \mathrm{WPO}_i \,|\, i = t-12, \ldots, t-2\right) \tag{14}$$

$$\mathrm{RD}_t^p = \max\left(\mathrm{WPO}_i - \mathrm{WPO}_{i+1} \,|\, i = t-12, \ldots, t-2\right) \tag{15}$$

$$\sigma_t^p = \sqrt{\frac{\sum_{i=t-12}^{t-1} \left(\mathrm{WPO_i} - \mu_t^p\right)^2}{\mathrm{num}\left(\mathrm{WPO}_{t-12}, \mathrm{WPO}_{t-11}, \ldots, \mathrm{WPO}_{t-1}\right)}}. \tag{16}$$

Furthermore, the *K*-means algorithm is applied here to implement the clustering based on the feature vectors generated in the previous step.

The number of clusters is denoted as k, which will be learnt by the algorithms itself based on the BIC. The cluster center of each cluster l is denoted by $C_l = \left(C_l^1, C_l^2, \ldots, C_l^m\right)$, where m is the total number of features in that cluster. The threshold radius of the lth cluster is denoted as e_l, which is the maximum Euclidean distance between set elements and the set center. In (17), $F_{l,i}^j$ is the value of feature j for the ith point in the lth cluster

$$e_l = \max\left(\sqrt{\sum_{j=1}^{m} \left(F_{l,i}^j - C_l^j\right)^2} \quad \forall \text{ point} \quad i \in \text{cluster } l\right). \tag{17}$$

The threshold value of each cluster will be utilized as the rules to identify the error patterns in the real application

$$I(S_t) = \begin{cases} S_t \in \text{cluster } l, & \sqrt{\sum_{j=1}^{m} \left(F_t^j - C_l^j\right)^2} \le \alpha e_l \\ S_t \notin \text{cluster } l, & \text{otherwise.} \end{cases} \tag{18}$$

BIC for each l and C_l will be calculated [30]

$$\mathrm{BIC}(k) = \log \prod_i \left[\frac{N_{C(i)}}{N} \frac{1}{\sqrt{2\pi}\sigma_{C(i)}} \exp\left(-\frac{\left\|x_i - \boldsymbol{m}_{C(i)}\right\|}{2{\sigma_{C(i)}}^2}\right)\right] - \frac{p \cdot \log N}{2}. \tag{19}$$

In (19), N denotes the number of points, $C(i)$ denotes the cluster to which x_i is assigned, $N_{C(i)}$ denotes the number of points assigned to $C(i)$, $\boldsymbol{m}_{C(i)}$ denotes the center vector of $C(i)$, $\sigma_{C(i)}$ denotes the standard deviation of points in $C(i)$ with respect to $\boldsymbol{m}_{C(i)}$, and p denotes the number of parameters in the statistical model, which equals to $k \cdot m$, i.e., the multiplication of number of clusters and the feature space dimension. $\log \prod_i \left[\frac{N_{C(i)}}{N} \frac{1}{\sqrt{2\pi}\sigma_{C(i)}} \exp\left(-\frac{\left\|x_i - \boldsymbol{m}_{C(i)}\right\|}{2{\sigma_{C(i)}}^2}\right)\right]$ is the log-likelihood of the *K*-means fit, the higher of which reflects a more precise clustering. $(p \cdot \log N)/2$ reflects the complexity of the model, which increases with k. As the model becomes more complex, i.e., k increasing, both the first and the second terms in (19) will increase. The resulting BIC will reach a peak point at some k. This very k will be selected and its corresponding clustering result will be used

The complete procedure of the algorithm is presented in the pseudo-code as follows.

Initiate a set K of candidate k
for $k \in K$
  Make initial guesses for the means $m_1, m_2, \ldots, m_k$
  Set the counts $n_1, n_2, \ldots, n_k$ to zero
  Until interrupted
    Acquire the next example, x
    If m_i is closest to x
      Increment n_i
      Replace m_i by $m_i + 1/n_i(x - m_i)$
    endif
  enduntil
  calculate $BIC(k)$ according to (19)
endfor
Select the k with max $BIC(k)$,
and return corresponding clustering results

The number of clusters as well as the center and threshold of each cluster is calculated as a basis of the bad data identification. The abnormal detection engine will evaluate each WPF whether it belongs to some type of AWPF by calculating the Euclidean distance between the S_t of the forecast and the center of each cluster, and moreover compare the distance with the threshold.

C. Data Adjustment Engine

Once S_t is labeled as the AWPF, it will be sent to the data adjustment engine first. The data adjustment engine is also constructed using NN. The procedure is shown in Fig. 7. The input of the data adjustment engine is the features extracted from

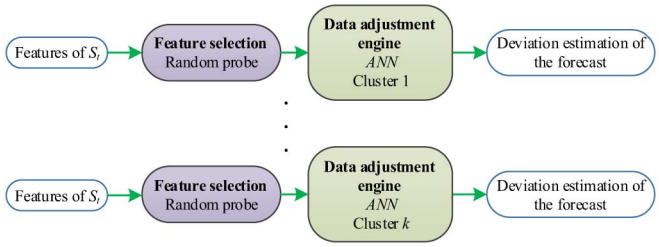

Fig. 7. Framework of data adjustment.

the abnormal detection engine. The used network structure of ANN here is identical with the one in the WPF engine, and the number of neurons in the input layer equals the number of the selected features.

Since the NN naturally has the over-fitting problem, which is known as the bias-variance dilemma, the feature selection is usually required. A proper design should aim at reducing the ratio of the complexity to the number of training sets as low as possible. Many related works exist [31]–[34]. In this paper, the random probe [35] is applied because of its two desirable properties, i.e., it does not rely on any assumption about the probability distribution of the features, and it can provide the rank for all the candidate features. In addition, it is easy to implement, computationally efficient, and has been proved successful in a variety of areas.

In detail, "probe features," which is a set of irrelevant features is first generated by exchanging the components of the candidate feature vectors randomly. The entire feature set now contains the generated probe features and the candidate features. Moreover, ranking is performed by orthogonal forward regression, based on Gram–Schmidt orthogonalization. This rank will be used to select features.

The probability that a probe feature is able to rank better than a candidate feature can be estimated; therefore, a threshold rank has to be specified to enable as few as irrelevant features to be selected as final features, because all variables that are ranked below the threshold are discarded.

The next step is to decide the threshold rank. The probe features are generated by randomly exchanging the components of candidate feature vectors, and the number of probe features is denoted as n_p. Therefore, they have the same probability distributions as the original candidate feature. The probability that a probe feature ranks better than a candidate feature is calculated as follows: the estimated probability that the probe rank is smaller than or equal to a given rank r is the ratio n_{rp}/n_p, where n_{rp} is the number of the random probe whose rank is smaller than or equal to r. During the ranking process, when a rank r is reached such that $n_{rp}/n_p > \delta$, where δ is the risk ratio chosen by the designer, and in this paper, it is set as 0.05, the procedure is terminated and the threshold rank r_0 is set equal to $r-1$. A general presentation of the above-mentioned method, put into the perspective of alternative variable selection methods, is provided in [33].

Before moving forward to the validation, one question needs to be answered. Why is the data adjustment engine trained separately for different types of error patterns instead of all together? The reason is that if it is trained together, there is no need to cluster the error patterns at all, and the solution will be almost the same as what is described in [25], a point-by-point forecast. As the authors claimed, the method used in [25] performs very well for the datasets used in their work. However, it does not provide the same contribution in this work experimentally. On one hand, the mechanism for different types of AWPFs vary from each other. On the other hand, there is high possibility that certain features may have completely contrary influences on the error rates. Taking Fig. 1 as an example, the bottom-left and the bottom-right figures have similar features of wind speed ramp-up rate; however, their wind power measurements in the corresponding ramp-up duration are totally different. Therefore, the data adjustment engine in this paper is trained separately for different types of AWPFs to distinguish the influence of features on forecast errors.

TABLE I
PARAMETERS OF THE PROPOSED APPROACH

Name	Value
Threshold error rate of error event	0.3
α	1

IV. METHOD VALIDATION

In this section, the case study is carried out using the data of wind farms from GEFCOM2012-WF to validate the proposed method [28], [29]. The data set contains the wind power measurement data and NWP data of seven wind farms, including wind speed and direction (update the 48-h-ahead forecast every 12 h) for more than 1 year time period. The time interval is 1 h, and in this paper, the target is to forecast the future 12-h WPOs of the first wind farm (WF1) when the future NWP data is available. The data from July 01, 2009 to December 12, 2010 (530 days) are adopted in this paper to conduct the case study. As the WSF data is updated every 12 h, there would be a total of 1060 sets and each set contains the 12-h actual wind power measurement and 48-h WSF results. In this paper, 200 sets are used for training the WPF engine, 800 sets are used for training the data adjustment engine, and the rest are used for algorithm evaluation. Parameters of the proposed method are shown in Table I.

A. Indices for Evaluating the Prediction Accuracy

This paper will use the index of RMSE to study the prediction accuracy. For the forecast time horizon h hour, the corresponding RMSE is calculated as

$$\mathrm{RMSE}^h = \sqrt{\frac{\sum_{\omega=1}^{N^h} \left(\widehat{\mathrm{WPO}}_\omega^h - \mathrm{WPF}_\omega^h\right)^2}{N^h}}. \tag{20}$$

B. WPF Engine

The detailed algorithm has been explained in Section III-A. The analysis of results for the historic data is given in Fig. 8. It is the error rate density for different forecast horizon from 1-h-ahead to 12-h-ahead. More than 60% of forecast error rates are within $[-0.1, 0.1]$ for 1–12-h prediction horizon. The closer the forecasting time point to the current time point is, the higher the accuracy trends to be. Therefore, the WPF engine should be run in a rolling way hour by hour.

C. Abnormal Detection

As discussed in Section III-B, the datasets used for abnormal detection are generated by WPF engine. Moreover, the

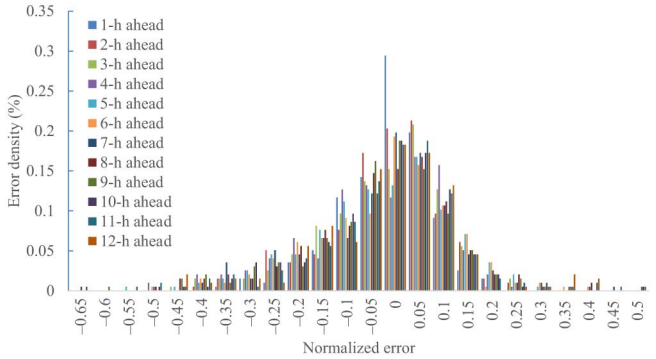

Fig. 8. Forecast error density for the final prediction results with each prediction horizon from 1 h ahead to 12 h ahead.

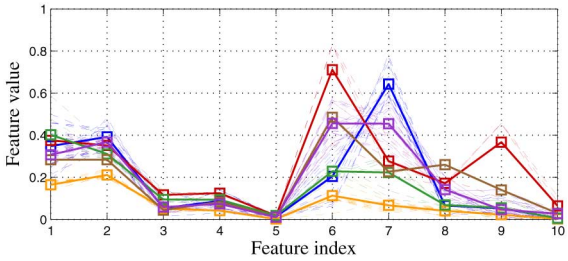

Fig. 9. Using *K*-means clustering method to classify AWPFs into six clusters. Ten features are selected as the final features, corresponding to (6)–(10) and (12)–(16).

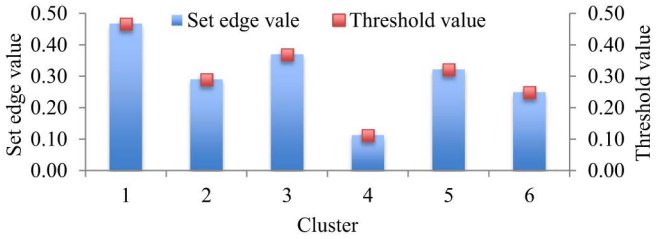

Fig. 10. Edge value of each AWPF cluster and the corresponding threshold value.

extracted features of each datasets are calculated. Finally, putting these features into the *k*-means clustering algorithm, based on the BIC, a total of six clusters are generated as shown in Fig. 9 (only the centers are shown in the figure). Ten features are selected as the final features, including (6)–(10) and (12)–(16). The edge of each cluster is the maximum Euclidean distance between the element in the cluster with the cluster center, and the threshold value is the level used to identify that whether a WPF is AWPF, by (18). In this paper, we set the threshold value as same as the set edge, as shown in Fig. 10. The threshold value can be set lower to reduce the risk that an NWPF is wrongly identified as AWPF. However, a too low threshold value may not dig out NWPFs efficiently, and this is a tradeoff to set an appropriate threshold value.

D. Data Adjustment Engine

The data adjustment engine is trained based on the features selected in abnormal detection stage. The output is error rate. After the training, once a forecast is defined as an AWPF, the potential error rate can be calculated using data adjustment engine. Then, the final results can be adjusted using

$$\text{Final WPF output} = \frac{\text{WPF}_t}{1+\widehat{\varepsilon_t}}. \tag{21}$$

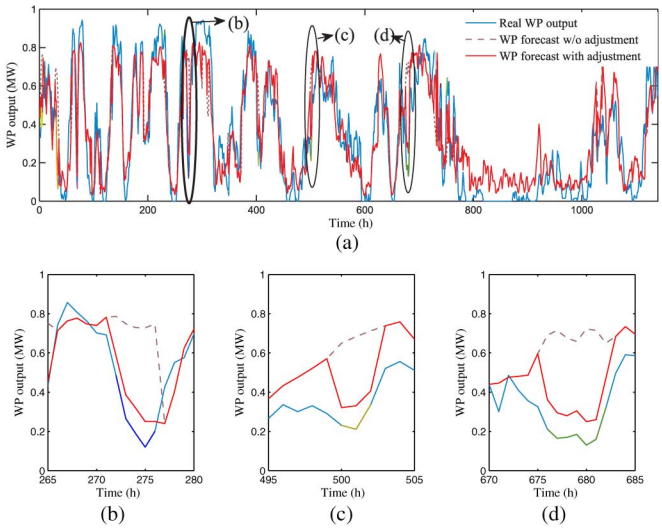

Fig. 11. Comparison between the forecasted wind power by WPF engine and the one after adjustment with the actual measured wind power. (a) WPF results with and without data adjustment. (b) Adjustment 1. (c) Adjustment 2. (d) Adjustment 3.

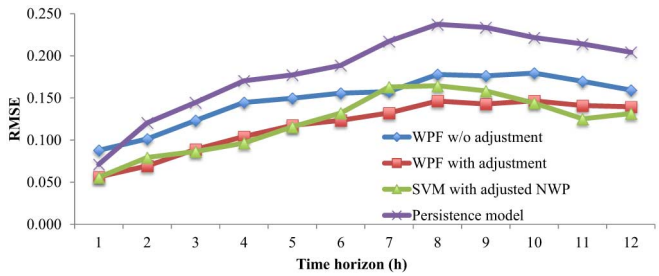

Fig. 12. RMSE of the preliminary WPF result, the final WPF result after adjustment, persistence model forecast result, and SVM forecast result with adjusted NWP.

E. Results Analysis

The WPF results of the test datasets are shown in Fig. 11. Both the forecast results by the WPF engine and the one by the adjustment engine are compared with the measured wind power. Fig. 11(b)–(d) are three extracted segments from Fig. 11(a), and the results show that AWPFs can be identified effectively. WPF with adjustment has higher accuracy compared with that without adjustment.

The RMSE results are shown in Fig. 12. The blue curve is the RMSE of preliminary wind power prediction derived from base forecaster and the black curve is the RMSE of final wind power prediction after adjustment. It shows that the prediction accuracy is higher for larger prediction horizon. Detailed results of comparison of different methods are shown in Table II.

It should be noted that the overall performance of the proposed approach is not as appealing as the top entries of GEFCOM2012-WF. The reason is that this paper focuses on

TABLE II
RMSE COMPARISON OF DIFFERENT APPROACHES

Time horizon (h)	WPF w/o adjustment	WPF with adjustment	SVM with adjusted NWP	Persistence model
1	0.09	0.06	0.06	0.07
2	0.11	0.07	0.08	0.12
3	0.12	0.09	0.09	0.14
4	0.14	0.10	0.10	0.17
5	0.15	0.12	0.12	0.18
6	0.16	0.12	0.13	0.19
7	0.16	0.13	0.16	0.22
8	0.18	0.15	0.16	0.24
9	0.18	0.14	0.16	0.23
10	0.18	0.15	0.14	0.22
11	0.17	0.14	0.13	0.21
12	0.16	0.14	0.13	0.20

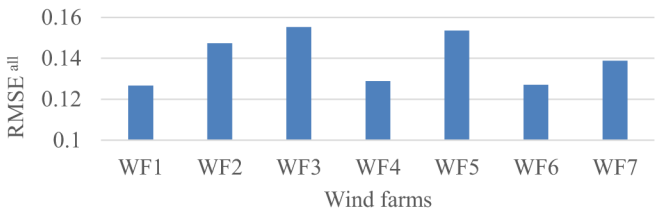

Fig. 13. $\mathrm{RMSE}^{\mathrm{all}}$ of all seven wind farms using the proposed approach.

the accuracy improvement brought on by the AWPF detection. To avoid distraction from the proposed idea, we have not devoted much effort to fine-tune the forecast engine in this paper. This is again following the presentation method in [29]. Note that the AWPF detection method can be used with other forecast engines including the ones used by the winners of GEFCOM2012-WF.

Furthermore, the proposed method is applied to conduct WPF for the other six wind farms in the GEFCOM2012-WF, and the value of $\mathrm{RMSE}^{\mathrm{all}}$ for each wind farm is calculated by (22). The results are shown in Fig. 13

$$\mathrm{RMSE}^{\mathrm{all}} = \sqrt{\frac{1}{12}\sum_{h=1}^{12}\left(\mathrm{RMSE}^{h}\right)^{2}}. \tag{22}$$

V. CONCLUSION

The main contribution of this paper is to introduce the NWP data adjustment to short-term wind power forecasting. It is implemented by identifying and clustering the errors of NWP using data-mining techniques, and then the abnormal raw NWP data will be adjusted before it is sent to the WPF engine. The simulation results demonstrate that the proposed approach can effectively reduce the total WFO error.

However, there are still challenges remaining for the proposed work. First, the physical explanation and reason of different error patterns is still unclear. A purely data-driven solution without model explanation is not solid enough to guarantee the generality of the proposed algorithm. Second, the threshold of selecting AWPF is determined through trial and error. It requires a more adaptive method and robust criterion for the practical application. Finally, the proposed approach has too many parameters, and it needs some further parameter-setting analysis in real application.

REFERENCES

[1] K. Porter and J. Rogers, "Status of centralized wind power forecasting in North America," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/SR-550-47853, 2010.
[2] N. Chen, Q. Wang, L. Yao, L. Zhu, Y. Tang, and F. Wu, "Wind power forecasting error based dispatch method for wind farm cluster," *J. Mod. Power Syst. Clean Energy*, vol. 1, no. 1, pp. 65–72, 2013.
[3] X. Zhang, G. He, S. Lin, and W. Yang, "Economic dispatch considering volatile wind power generation with lower-semi-deviation risk measure," in *Proc. IEEE 4th Int. Conf. Elect. Utility Deregul. Restruct. Power Technol. (DRPT)*, 2011, pp. 140–144.
[4] C. Peng, S. Lei, Y. Hou, and F. Wu, "Uncertainty management in power system operation," *CSEE J. Power Energy Syst.*, vol. 1, no. 1, pp. 28–35, Mar. 2015.
[5] L. Li, Y. Mei Wang, and Y. Qian Liu, "Impact of wake effect on wind power prediction," in *Proc. 2nd IET Renew. Power Gener. Conf. (RPG'13)*, Sep. 2013, pp. 1–4.
[6] D. Carvalho, A. Rocha, M. Gómez-Gesteira, and C. Santos, "A sensitivity study of the WRF model in wind simulation for an area of high wind energy," *Environ. Model. Softw.*, vol. 33, pp. 23–34, 2012.
[7] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renew. Sustain. Energy Rev.*, vol. 13, no. 4, pp. 915–920, 2009.
[8] M. Khalid and A. Savkin, "Closure to discussion on 'a method for short-term wind power prediction with multiple observation points'," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1898–1899, May 2013.
[9] J. Tastu, P. Pinson, P.-J. Trombe, and H. Madsen, "Probabilistic forecasts of wind power generation accounting for geographically dispersed information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 480–489, Jan. 2014.
[10] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014.
[11] P.-J. Trombe, P. Pinson, and H. Madsen, "A general probabilistic forecasting framework for offshore wind power fluctuations," *Energies*, vol. 5, no. 3, pp. 621–657, 2012.
[12] H. Bludszuweit, J. Dominguez-Navarro, and A. Llombart, "Statistical analysis of wind power forecast error," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 983–991, Aug. 2008.
[13] S. Rajagopalan and S. Santoso, "Wind power forecasting and error analysis using the autoregressive moving average modeling," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PES'09)*, Jul. 2009, pp. 1–6.
[14] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, "Arima-based time series model of stochastic wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 667–676, May 2010.
[15] N. Chen, Z. Qian, I. Nabney, and X. Meng, "Wind power forecasts using gaussian processes and numerical weather prediction," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 656–665, Mar. 2014.
[16] P. Louka *et al.*, "Improvements in wind speed forecasts for wind power prediction purposes using kalman filtering," *J. Wind Eng. Ind. Aerodyn.*, vol. 96, no. 12, pp. 2348–2362, 2008.
[17] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, Feb. 2014.
[18] J. Zeng and W. Qiao, "Short-term wind power prediction using a wavelet support vector machine," *IEEE Trans. Sustain. Energy*, vol. 3, no. 2, pp. 255–264, Apr. 2012.
[19] E. Elattar, "Short term wind power prediction using evolutionary optimized local support vector regression," in *Proc. 2nd IEEE PES Int. Conf. Exhib. Innovative Smart Grid Technol. (ISGT Europe)*, Dec. 2011, pp. 1–7.
[20] J. Shi, Y. Liu, Y. Yang, and W.-J. Lee, "Short-term wind power prediction based on wavelet transform-support vector machine and statistic characteristics analysis," in *Proc. IEEE Ind. Commer. Power Syst. Tech. Conf. (ICPS)*, May 2011, pp. 1–7.

[21] N. Amjady, F. Keynia, and H. Zareipour, "Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 265–276, Jul. 2011.
[22] S. Fan, J. Liao, R. Yokoyama, L. Chen, and W.-J. Lee, "Forecasting the wind generation using a two-stage network based on meteorological information," *IEEE Trans. Energy Convers.*, vol. 24, no. 2, pp. 474–482, Jun. 2009.
[23] A. Costa *et al.*, "A review on the young history of the wind power short-term prediction," *Renew. Sustain. Energy Rev.*, vol. 12, no. 6, pp. 1725–1744, 2008.
[24] G. Qu, J. Mei, and D. He, "Short-term wind power forecasting based on numerical weather prediction adjustment," in *Proc. 11th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2013, pp. 453–457.
[25] G. Sideratos and N. Hatziargyriou, "An advanced statistical method for wind power forecasting," *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 258–265, Feb. 2007.
[26] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer, 2006.
[27] S. Watanabe, "A widely applicable bayesian information criterion," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 867–897, Mar. 2013.
[28] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *Int. J. Forecast.*, vol. 30, no. 2, pp. 357–363, 2014.
[29] T. Hong, P. Wang, and L. White, "Weather station selection for electric load forecasting," *Int. J. Forecast.*, vol. 31, no. 2, pp. 286–295, 2015.
[30] D. Pelleg *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 727–734.
[31] N. Amjady and F. Keynia, "Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro-evolutionary algorithm," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 306–318, Feb. 2009.
[32] I. Drezga and S. Rahman, "Short-term load forecasting with local ann predictors," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 844–850, Aug. 1999.
[33] I. Drezga and S. Rahman, "Input variable selection for ann-based short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1238–1244, Nov. 1998.
[34] M. Dong and C. Lou, "Adaptive electric load forecaster," *Tsinghua Sci. Technol.*, vol. 20, no. 2, pp. 164–174, Apr. 2015.
[35] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, "Ranking a random feature for variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1399–1414, Mar. 2003.

Qianyao Xu (S'10) received the Bachelor's degree in electrical engineering from Tsinghua University, Beijing, China, in 2010, and is currently pursuing the Ph.D. degree in electrical engineering at the same university.

His research interests include power planning and wind power operation.

Dawei He received the B.Eng. degree in electrical engineering from Tsinghua University, Beijing, China, in 2010, and is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Ning Zhang (S'10–M'12) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2007 and 2012, respectively.

He is currently a Lecturer with the same university. His research interests include stochastic characteristic analysis and simulation of renewable energy, power system planning, and scheduling with renewable energy.

Chongqing Kang (M'01–SM'07) received the bachelor's degree in both electrical engineering and environment engineering, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 1993 and 1997, repectively.

He is a Professor of Electrical Engineering with Tsinghua University, Beijing, China. His research interests include renewable energy, power system planning, electricity marketing, and optimization theory.

Qing Xia (M'01–SM'08) received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 1989.

He is a Professor of Electrical Engineering with Tsinghua University, Beijing, China. His research interests include electricity market, generation scheduling optimization, power system planning, and load forecasting.

Jianhua Bai received the bachelor's and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China.

He is a Senior Engineer with State Grid Energy Research Institute, Beijing, China. His research interests include energy and power system planning.

Junhui Huang received the bachelor's degree in electrical engineering from Southeast University, Nanjing, China.

He is the Director of the Power System Planning Center, Economic Research Institute, Jiangsu Electric Power Company, Nanjing, China. His research interests include energy and power system planning.