

Research Article

Multistep Wind Speed and Wind Power Prediction Based on a Predictive Deep Belief Network and an Optimized Random Forest

Zexian Sun ¹, Hexu Sun ¹, and Jingxuan Zhang ²

¹School of Artificial Intelligence, Hebei University of Technology, China

²College of Electrical Engineering, North China University of Science and Technology, China

Correspondence should be addressed to Hexu Sun; hxsun@hebust.edu.cn

Received 11 March 2018; Revised 24 May 2018; Accepted 4 June 2018; Published 24 July 2018

Academic Editor: Marco Mussetta

Copyright © 2018 Zexian Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A variety of supervised learning methods using numerical weather prediction (NWP) data have been exploited for short-term wind power forecasting (WPF). However, the NWP data may not be available enough due to its uncertainties on initial atmospheric conditions. Thus, this study proposes a novel hybrid intelligent method to improve existing forecasting models such as random forest (RF) and artificial neural networks, for higher accuracy. First, the proposed method develops the predictive deep belief network (DBN) to perform short-term wind speed prediction (WSP). Then, the WSP data are transformed into supplementary input features in the prediction process of WPF. Second, owing to its ensemble learning and parallelization, the random forest is used as supervised forecasting model. In addition, a data driven dimension reduction procedure and a weighted voting method are utilized to optimize the random forest algorithm in the training process and the prediction process, respectively. The increasing number of training samples would cause the overfitting problem. Therefore, the k-fold cross validation (CV) technique is adopted to address this issue. Numerical experiments are performed at 15-min, 30-min, 45-min, and 24-h to indicate the superiority and signal advantages compared with existing methods in terms of forecasting accuracy and scalability.

1. Introduction

The uncertainty of wind energy indeed imposes major challenges on power system operation and planning, such as power system security assessment and reserve management [1]. A reliable wind power forecasting plays an important role in preventing damage to wind turbines and maintaining the stability and security of the power system.

Numerous forecasting models which are generally classified into four categories have been proposed for short-term wind power forecasting.

(1) Physical-based method includes spatial and temporal factors in a full fluid-dynamics model of the atmosphere [2]. NWP is a popular physical approach utilizing complex mathematical models but always suffers from the difficulties in gaining information and its limited spatial resolution.

(2) Statistical-based method characterizes the history data to yield precise performance for short-term forecasting

tasks. The dynamical ensemble LSSVR [3], the closed-loop forecasting engine including KIM and EMD[4], and adaptive robust multikernel regression model [5] have been proposed to yield the research. Adaptive neurofuzzy inference systems were developed to perform a nonlinear mapping between inputs and outputs [6].

(3) Artificial intelligence-based method represents the complex nonlinear relationship between inputs and outputs without predefined mathematical models. These models include Ridgelet ANN [7], bagging neural network [8], and Multilayer Perceptron Networks [9].

(4) Hybrid-based method based on two or more architectures benefits from the advantages of each approach to reach the optimal results. A hybrid approach based on the combination between least square support vector machine and gravitational search algorithms is proposed [10].

Most methods reported in the current literature have three disadvantages.

(1) In most artificial intelligence methods, there is only one single hidden layer [11–14]. Due to the finite ability of searching optimal solution in the parameter spaces of ANN with more than one hidden layer, these models cannot provide accurate outputs.

(2) Some methodologies require tediously hand-engineered features [13]. Due to the lack of sufficient knowledge for a specific domain, the selected features may not be suitable for corresponding models.

(3) The conventional features in the previous supervised forecasting models contain the NWP data of the wind farms [15–19]. However, the NWP model usually runs once or twice a day, which is often applied for medium- to long-term forecasts [20].

Deep learning has attracted tremendous attention in recent years of academia and industrial communities [21]. It has been successfully used in the field of speed recognition [22], image processing [23], and health status assessment [24, 25]. However, it has not been actively utilized in the wind power or wind speed forecasting fields. Deep belief network, due to its strong ability of learning, has been performed in short-term WSP [26]. The stacked denoising autoencoder combined with rough set was applied to extract features from wind speed series [20]. In the literature [27], deep autoencoders act as base-regressors, whereas deep belief network is used as a metaregressor. A DNN-MRT scheme was proposed to predict wind power. Therefore, employing deep learning to address the problems mentioned above has a great potential.

In this paper, a hybrid intelligent method for short-term wind power forecasting is proposed. First, DBN as a probabilistic generative model consisting of multiple layers of restricted Boltzmann machines (RBMs) provides the unsupervised learning features from the wind speed series data in the pretraining phase, and backpropagation neural network (BPNN), as the top layer of the DBN, uses the labeled data to fine-tune the parameters of the DBN by stochastic gradient decent. Performance of DBN is affected by its parameters, which are often limited by hand-engineered features. The bat algorithm combined the major advantages between particle swarm optimization and genetic algorithm and Harmony Search is applied to yield optimal parameters in the DBN. Second, random forest is suitable for handling large data due to its parallelization [28]. It has been combined with the Spark [28], heuristic bootstrap sampling method [29], kernel principal component analysis [30], and other technologies to perform fault diagnosis and regression tasks [31, 32]. Owing to the improvement of the forecasting accuracy for high-dimensional and large-scale wind turbine data, we propose an optimized random forest method which consists of a dimension reduction procedure and the weighted voting process for the short-term WPF.

The contributions of the paper are as follows.

(1) Predictive deep belief network is applied as the deep learning architecture to perform the short-term wind speed prediction. The DBN model has better generalized ability than the traditional shallow architectures.

(2) The bat algorithm is incorporated with DBN to improve the training accuracy and reduce the costing time in wind speed prediction.

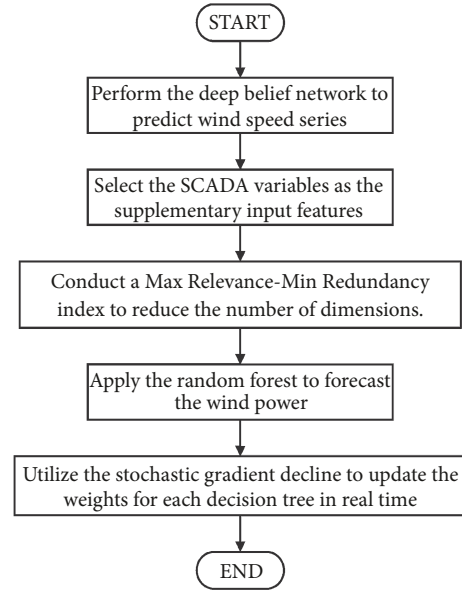


FIGURE 1: The process of the proposed method for multistep wind speed and wind power prediction.

(3) Benefiting from the dimension reduction and weighted voting procedures, the random forest model avoids the lack of prior knowledge for feature selection and enhances the capacities of tackling new condition data.

This paper is organized as follows. In Section 2, the DBN and its learning algorithm for short-term WSP are presented. Section 3 presents the optimized random forest and the proposed model for short-term WPF. Numerical results and conclusion are presented in Sections 4 and 5, respectively.

2. Brief Architecture about the Proposed Method

In this section, the proposed method would be briefly introduced and its process is shown in Figure 1. Wind speed, as the most important variable to the normal running of wind turbine, is hard to predict because of its randomness and fluctuation. Therefore, due to the merged advantages of the deep learning architectures and the bat algorithm, the WSP model based on predictive deep belief network and bat algorithm is firstly performed to capture the characteristics of wind speed series. Then owing to its ensemble learning and parallelization the random forest is used as the WPF model. In addition, a data driven dimension reduction procedure and a weighted voting method are utilized to optimize the random forest algorithm in the training process and the prediction process, respectively.

3. DBN for Wind Speed Prediction

Due to the intermittence and volatility of the wind energy, it is difficult to gain the efficient future wind speed series. Thus, this section would introduce a novel WSP method to perform the wind speed forecasting.

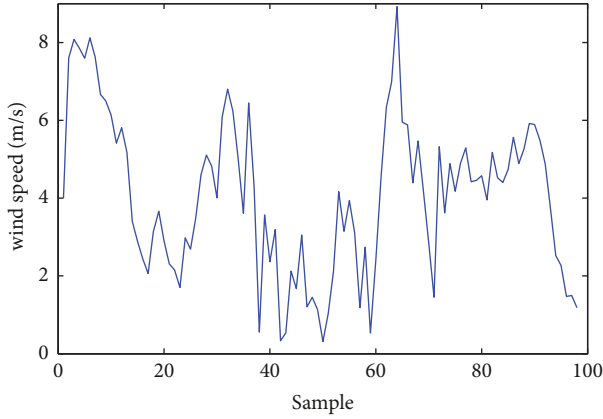


FIGURE 2: The wind speed series recorded on October 30, 2017, in northern China.

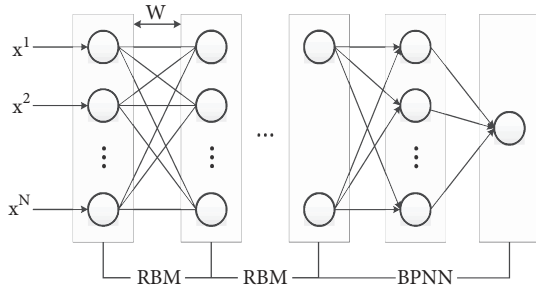


FIGURE 3: Architecture of DBN composed of a number of stacked RBMs and BPNN.

3.1. Short-Term Wind Speed Forecasting. The wind speed series can be described as $X = \{x_1, x_2, \dots, x_t\}$, where x_i is the average wind speed in the past 15 minutes as illustrated in Figure 2. For short-term wind speed prediction, it is to forecast the future value of $x_{t+\tau}$ by utilizing the previous N data, where t is the index of the wind speed series and τ is the forecast horizon. Therefore, it is a significant task to build the function f to describe the wind speed explicitly:

$$f(x_{t-1}, \dots, x_{t-N}, \theta) = x_{t+\tau} \quad (1)$$

where θ is the parameter vector.

3.2. Traditional Deep Belief Network. The architecture of DBN is shown in Figure 3. It is composed of several stacked RBMs. The unit number of first input layer is N , which is the same as the number of previous data. The numbers of units for the hidden layers are gradually reduced. The input layer of BPNN, as well as the hidden layer of the last RBM, consists of two units. The output layer has one unit. Then, the optimal mapping function can be found through the parameter space which is influenced by the bat algorithm and avoid the effect of personal experience.

There are two steps included in the training process of traditional DBN.

(1) The pretraining phase: As the hardcore of DBN, RBM is a stochastic generative model which performs this procedure as a technology of greedy layerwise unsupervised

learning. Each RBM consists of two layers, namely, the visible layer and the hidden layer, as shown in Figure 2. The probability distribution over an RBM is defined by the connection weights between visible units and hidden units through an energy-based model $E(v, h; \theta)$:

$$E(v, h; \theta) = -bv - hWv - ch \quad (2)$$

where $\theta = (W, b, c)$ represents the model parameters, W denotes the weights connecting hidden and visible units, and b and c are the biases of the visible units and the hidden units, respectively. The conditional probability over visible units and hidden units is defined as

$$P(h | v) = \frac{\exp(-E(v, h; \theta))}{Z} \quad (3)$$

$$Z = \sum_h \exp(-E(v, h; \theta)) \quad (4)$$

Considering the specific structure of the RBM, the neurons are binary; the probabilistic version of activating a unit is a logistic function which is given by

$$p(h_i = 1 | v) = \text{sigm}(c_i + W_i v) \quad (5)$$

$$p(v_i = 1 | h) = \text{sigm}(b_i + W_i h) \quad (6)$$

where sigm is the logistics sigmoid function.

The objective by training the model is maximized log-likelihood, which is defined by

$$L(\theta, D) = \sum_{x \in D} \log P(x, \theta) \quad (7)$$

In the commonly studied case, the contrastive divergence is adopted to calculate the derivative of the log probability about the model parameter

$$\frac{\partial \log P(x)}{\partial W} = Ep[x \cdot h] - Ep'[x \cdot h] \quad (8)$$

$$\frac{\partial \log P(x)}{\partial c} = Ep[x] - Ep'[x] \quad (9)$$

$$\frac{\partial \log P(x)}{\partial b} = Ep[h] - Ep'[h] \quad (10)$$

where Ep denotes an expectation under the data distribution and Ep' denotes an expectation under the model distribution.

Then update rules for the parameters can be written as

$$W = W + \alpha \frac{\partial \log P(x)}{\partial W} \quad (11)$$

$$b = b + \alpha \frac{\partial \log P(x)}{\partial b} \quad (12)$$

$$c = c + \alpha \frac{\partial \log P(x)}{\partial c} \quad (13)$$

where α is the learning rate.

(2) The fine-tune phase: For the DBN model, the gradient descent associated with greedy layer-wise training method

is performed in the pretraining phase, and this predictive fine-tune is carried out by adopting BPNN. The parameters space of the DBN model would be fine-tuned by minimizing the error between the predicted value and actual value. The parameters can be updated as

$$W_l = W_l - \alpha \frac{\partial}{\partial W_l} L(W, b; X) \quad (14)$$

$$B_l = B_l - \alpha \frac{\partial}{\partial B_l} L(W, b; X) \quad (15)$$

where W_l and B_l are the weights and bias of the l th layer, α is the learning rate, and $L(W, b; X)$ is the cost function.

The DBN model has the capability of forecasting the wind speed in the current condition space after the fine-tune procedure. When it comes to the new dataset, the parameters space can be further fine-tuned by simply performing the new dataset instead of training from scratch.

3.3. Optimization of the DBN Model. To find the optimal mapping function in Section 3.1, an improved DBN is employed. The connecting weights, the offsets of the units, and the learning rate need to be decided when the model is applied. As mentioned above, the model parameters are initialized by random experiments and then updated by iterative training, which may take a long time and need the mutual experience. To address this problem, a bat algorithm is employed to help search the optimal parameters space, which is composed of the connecting weights between visible units and hidden units, the biases of the visible units and the hidden units, the learning rate in the RBM, and the parameters in the BPNN. The detailed steps are shown as follows.

Step 1. Initialize $iter = 1$, bat position x_i , velocity v_i , frequency Q_i , pulse rate r_i , and loudness A_i for each of the n bats.

Step 2. Compute the fitness function value for every bat and select the best bat

$$f_i = \frac{1}{m} \sum_{j=1}^m (R_j - R_j^a)^2 \quad (16)$$

$$x_{best} = \arg \min (f_i) \quad (i = 1, 2, \dots, n) \quad (17)$$

where m is the size of output vector, R_j is the predicted value and R_j^a is the measured value, and x_{best} is the best solution in the current situation.

Step 3. Compute the new solution x_i^t , the new velocity v_i^t , and the new fitness f_i .

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \beta \quad (18)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*) f_i \quad (19)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (20)$$

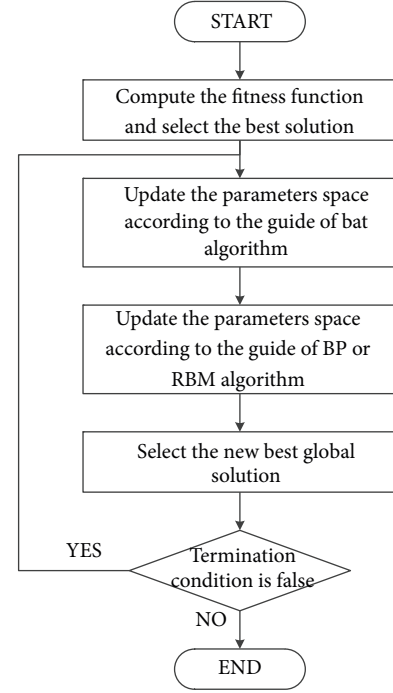


FIGURE 4: Using bat algorithm to design an optimized DBN.

where β is a uniform random value within $[0, 1]$ and x_* is the current global best solution, and f_{\max} and f_{\min} denote the maximum and minimum frequency, respectively.

Step 4. BPNN or RBM updates the parameters space according to its own learning method.

Step 5. Compare the performance of parameters spaces between the BPNN or RBM and the bat algorithm using (16), and select the best one as the new best global solution.

The second step to the fifth step would be repeated until the termination condition is true. Figure 4 shows the framework of the DBN model developed in this section.

4. Optimized Random Forest for Wind Power Forecasting

Random forest has been widely used for prediction problems. The proposed WPF model is based on WSP and optimized random forest, which consists of attribute reduction and weighted voting procedures. Figure 5 shows the training and prediction process of the WPF model. First, Pearson's correlation coefficient is conducted on the vector space to select the favor features. Second, this paper adopts the WSP and the selected SCADA variables as the input attributes. Finally, a real-time weighted voting is constructed in the prediction process.

4.1. Random Forest Algorithm. Random forest is an ensemble predictor based on regression tree model. It generates k different decision trees by training different data subsets.

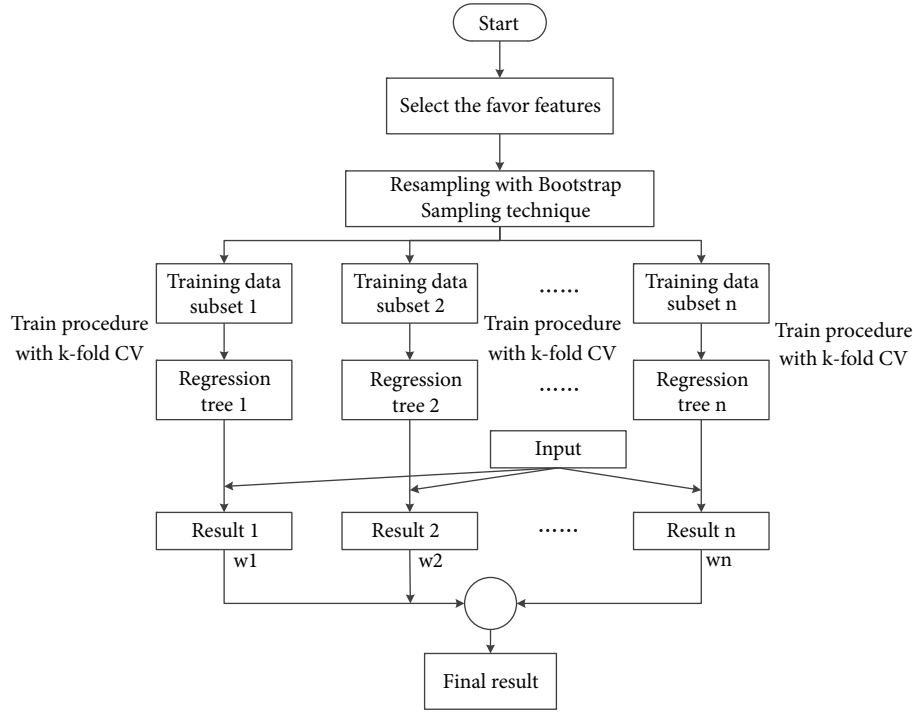


FIGURE 5: Process of training and prediction of the WPF model.

Each regression tree provides a prediction for every testing data, and the final result depended on the vote of these trees.

The original training data is denoted as $S = \{(c_i, t_j), i = 1, 2, \dots, L; j = 1, 2, \dots, D\}$, where c is a sample and t is the feature sets. Namely, the training data contains D variables and L samples. The steps of constructing each decision tree of random forest are as follows.

Step 6. Select training subsets in a bootstrap manner.

Step 7. d features are randomly selected from D variables.

Step 8. Construct the tree to the maximum depth without pruning.

The above three steps are repeated until k decision trees are collected into a random forest model.

4.2. The Dimension Reduction for High-Dimensional Data. To improve the prediction accuracy of the random forest model, we present a Max Relevance-Min Redundancy (MRMR) index by conducting Pearson's correlation coefficient to reduce the number of dimensions. In the training procedure of each decision tree, the top d features according to the MRMR index value are selected as the favor features, and then we randomly select the $(l - d)$ feature variables from the remaining $(D - d)$ ones. Therefore, the dimension is reduced from D to l . The process in the dimension reduction is presented in Figure 6.

First, in the training process, Pearson's correlation coefficient based on the covariance matrix is adopted to evaluate the relationship between two vectors α_i and β_j .

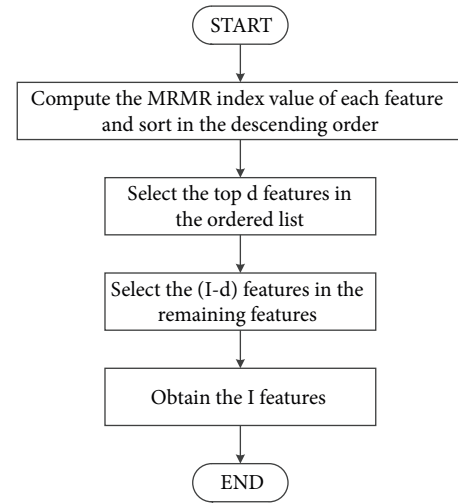


FIGURE 6: Dimension reduction in the training process.

$$P(\alpha_i, \beta_j) = \frac{\text{cov}(\alpha_i, \beta_j)}{\sqrt{\text{var}(\alpha_i) \times \text{var}(\beta_j)}} \quad (21)$$

The relevance of each feature would be calculated by

$$R(t_j, Y_L) = \frac{1}{D} \sum_{t_j \in t} P(t_j, Y_L) \quad (22)$$

where t_j is the feature, D is the size of dimension, Y_L is the target feature, and the most relevant feature can be obtained by

$$t_{\max} = \arg \max (R(t_j, Y_L)) \quad (23)$$

(1) Input: a training dataset, the whole feature set F_w , favor feature size s , the favor feature set $F_s = \phi$, the remaining feature set $F_r = F_w - F_s$

(2) for $k=1$ to s
 search the favor feature from the remaining feature set according to

$$F_{opt} = \begin{cases} \arg \max_{F_j \in F_r} [P(F_j, Y_L) - \frac{1}{k-1} \sum_{F_i \in F_s} P(F_j, F_i)] & k \neq 1 \\ \arg \max_{F_j \in F_r} [P(F_j, Y_L)] & k = 1 \end{cases}$$

(3) update the favor and the remaining feature sets $F_s = F_s \cup F_{opt}$ $F_r = F_r - F_{opt}$
 end for
 Output: F_s

ALGORITHM 1: The process of selecting the favor features.

Second, the redundancy for each input variable is calculated. The min redundancy feature can be obtained by

$$R(t_i, t_j) = \frac{1}{D^2} \sum_{t_i, t_j \in t} P(t_i, t_j) \quad (24)$$

$$t_{\min} = \arg \min R(t_i, t_j) \quad (25)$$

The MRMR index is a simple form of $R(t_j, Y_L)$ and $R(t_i, t_j)$, and the candidate feature can be selected by

$$\varphi = R(t_j, Y_L) - R(t_i, t_j) \quad (26)$$

$$F_{opt} = \arg \max \varphi \quad (27)$$

Given the framework discussed above, we adopt the MRMR index to select the favor features; the detailed steps are shown in Algorithm 1. Compared with the traditional RF model, the dimension reduction method balances the accuracy and diversity of the RF algorithm and prevents the overfitting efficiently.

4.3. Weighted Prediction Process. In the prediction process, limited to the training data, it likely leads to the performance reduction of the traditional RF algorithm when the new condition data is applied. To overcome this drawback, a real-time update for weighted voting approach is proposed to decrease the prediction error for the testing data.

Each sample of testing data is predicted by all the decision trees in the RF model. The prediction result is weighted average value of all decision trees. The weighted regression result is defined as

$$Y_j = \frac{\sum_{i=1}^k w_{ji} y_{ji}}{\sum_{i=1}^k w_{ji}} \quad (28)$$

where w_{ji} is the weight for the i th decision tree on the j th sample, y_{ji} is the prediction result for the i th decision tree on the j th sample, and Y_j is the final prediction result.

The objection of the predictor learning is to minimize the prediction error. The error function in this paper is defined as

$$Err = 0.5 \times (y_{ji} - Y_j)^2 \quad (29)$$

The stochastic gradient decline is conducted to update the weights for decision trees as follows.

$$\begin{aligned} \Delta w_{ji} &= -\eta \frac{\partial Err}{\partial w_{ji}} \\ &= \eta (y_{ji} - Y_j) \end{aligned} \quad (30)$$

$$\times (y_{ji} - Y_j) \left[\frac{y_i \sum_{i=1}^k \omega_{ji} - \sum_{i=1}^k \omega_{ji} y_{ji}}{(\sum_{i=1}^k \omega_{ji})^2} \right]$$

$$\omega_{j,i+1} = \omega_{ji} - \Delta w_{ji} \quad (31)$$

In the weighted voting procedure of RF model, a reasonable weight associated with each tree is applied to improve the global prediction accuracy and reduce the generation error especially for the new condition data.

4.4. The Short-Term Wind Power Forecasting Model. To further improve the inferential ability of the proposed method, the k -fold cross validation partitions the training data into k different subsets. Then $(k-1)$ of these subsets are selected to train the model and the other one is used for testing the performance of the trained model. The root mean square error is taken as the criterion to select the optimal model. To clearly describe the process of the WPE, the detailed steps of training and prediction procedure are given as follows.

The training procedure includes the following.

Step 1. Select the training features from the input variables according to the dimension reduction method mentioned in Section 4.2. Twelve SCADA variables are selected as the candidate features for RF model, which are as shown in Table 1.

Step 2. Based on the input features selected in Step 1, the regression tree model is built on its traditional way with k -fold cross validation without pruning.

Step 3. The above two steps are repeated until the size of random forest model reaches the threshold.

TABLE 1: The list of candidate features.

no	The list of candidate features
1	generator speed (CCU)
2	Rotor speed
3	Wind speed
4	generator speed (PLC)
5	The temperature of bearing A
6	The temperature of bearing B
7	The temperature of gearbox
8	The ambient temperature
9	The temperature of nacelle
10	The temperature of gearbox bearing
11	Blade 1 actual value
12	Blade 2 actual value
13	Blade 3 actual value

TABLE 2: The size of training sets and testing sets.

	The size of training sets	The size of testing sets
Case 1	13873	195
Case 2	13783	285
Case 3	10000	150
Case 4	10000	150

The prediction procedure includes the following.

Step 1. Each regression tree of the RF model applies the testing data which consists of the future wind speed series obtained from the WSP and the SCADA variables to perform short-term WPF.

Step 2. Compute the final prediction result based on the prediction values from all regression trees according to (28), and then update the error weight for each regression tree in real-time according to (30)-(31).

5. Numerical Results

To verify the performance of the proposed method, numerical experiments have been carried out on four sets of historical SCADA data. The task is to forecast the 15-min, 30-min, 45-min, and 24-h wind speed and the corresponding wind power. The size of training sets and testing sets in four cases are as shown in Table 2.

5.1. Performance Index. The root mean square error (RMSE), the mean absolute error (MAE), the average percentage error (APE), the bias (BIAS), and the standard deviation of the error (SDE) are the measurements to compute the error scores.

$$RMSE = \sqrt{\frac{1}{m} \sum_{j=1}^m (y_j - y_t)^2} \quad (32)$$

$$MAE = \frac{1}{m} \sum_{j=1}^m |y_j - y_t| \quad (33)$$

$$BIAS = \frac{1}{m} \sum_{j=1}^m (y_j - y_t) \quad (34)$$

$$SDE = \sqrt{\frac{1}{m} \sum_{j=1}^m (y_j - y_t - BIAS)^2} \quad (35)$$

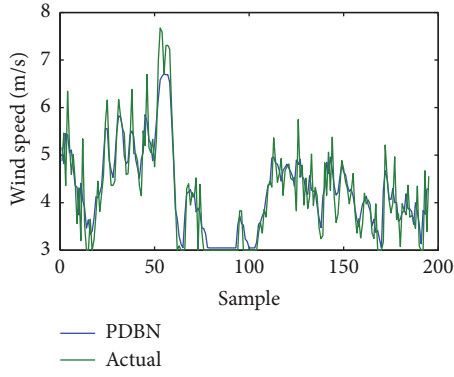
$$APE = \frac{1}{m} \sum_{j=1}^m \frac{|y_j - y_t|}{|y_t|} \quad (36)$$

where y_j is the predicted value from the model and y_t is the true value; m is the number of testing samples.

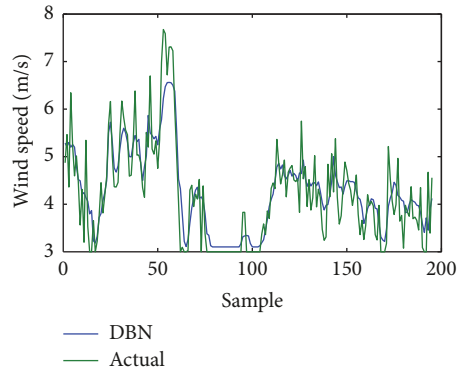
5.2. Short-Term Wind Speed Prediction Results. The structure of the predictive DBN is determined by human experience, which is 10-7-5-4-2-4-1. How to search the optimal structure is still an open problem. In this section, the proposed method is compared with several wind speed forecasting models, which have been proposed in the literature including BPNN, ELMAN, persistence method, and traditional DBN. To evaluate the generalization performance, the experiments with the forecast horizon ranging from 15 min to 24 hours are carried out on four sets: Case 1-Case 4.

The actual wind speed data and their predicted values using predictive DBN, traditional DBN, BPNN, ELMAN, and persistence model are shown in Figure 7. Figures 7(a)–7(j) show the predicted wind speed series in the future 15 minutes and 30 minutes from predictive DBN, traditional DBN, BPNN, ELMAN, and persistence model in Case 1 and Case 2. Similarly, the results for the 45-minute and 24-hour horizon in Case 3 and Case 4 are shown in Figures 7(k)–7(t), where the green and blue lines denote the actual and predicted wind speed series by these models, respectively. In the first two cases, the estimated values from predictive DBN are very close to the real data, but the others have obvious errors. However, as we can see from Figures 7(k)–7(t), the performance of these models decrease gradually as the inference step increases, where the predictive DBN also generates the minimum prediction errors.

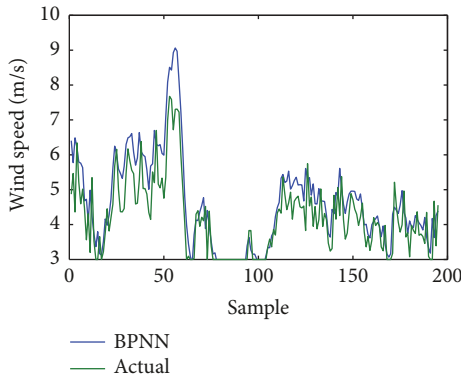
To further show the performance of these models, the RMSE, MAE, BIAS, SDE, and APE, as the indexes to evaluate the approximate performance, are computed and demonstrated in Table 3. As shown in Table 3, the deep learning architectures procure much better results than shallow networks and persistence model. The proposed model shares nearly the same generalization capacities with the traditional DBN to predict the future wind speed, while the forecasting ability of the persistence model degrades with the growth of the time scale. Predictive DBN, as a deep learning network, has improved the RMSE, MAE, BIAS, SDE, and APE by 1.505, 1.427, 2.439, 1.134, and 0.625 over multiple time



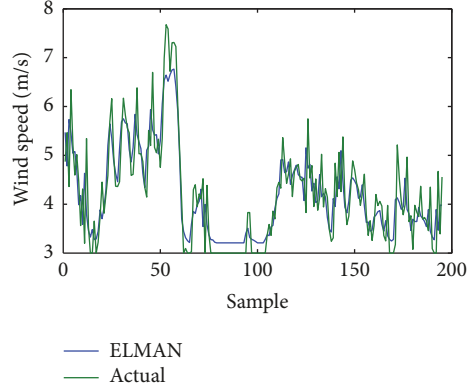
(a) The forecasting result of predictive DBN for 15-min



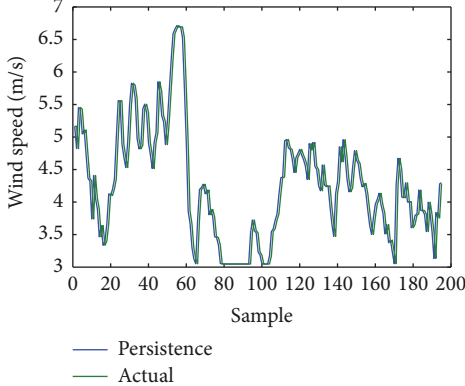
(b) The forecasting result of traditional DBN for 15-min



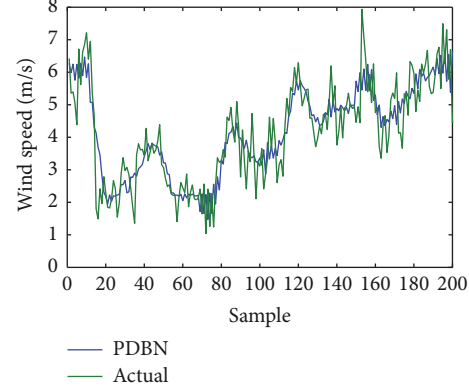
(c) The forecasting result of BPNN for 15-min



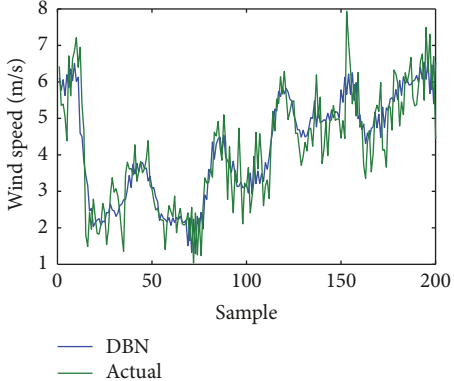
(d) The forecasting result of ELMAN for 15-min



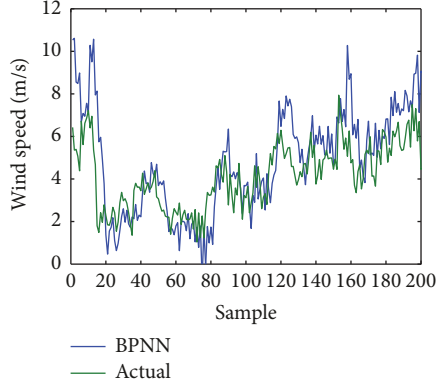
(e) The forecasting result of persistence model for 15-min



(f) The forecasting result of predictive DBN for 30-min

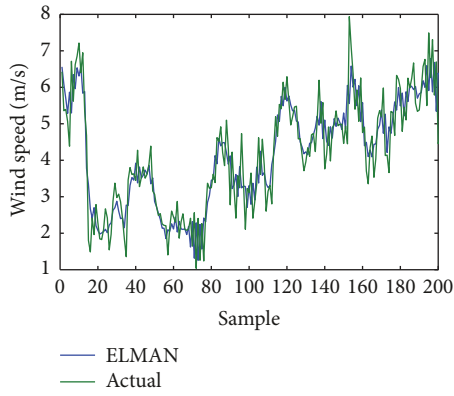


(g) The forecasting result of traditional DBN for 30-min

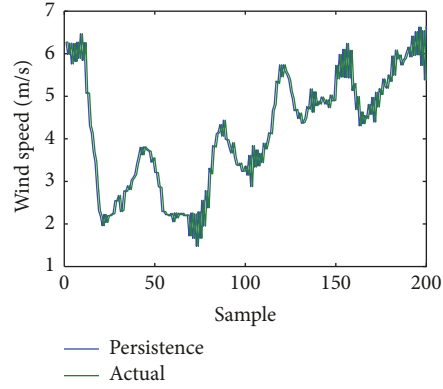


(h) The forecasting result of BPNN for 30-min

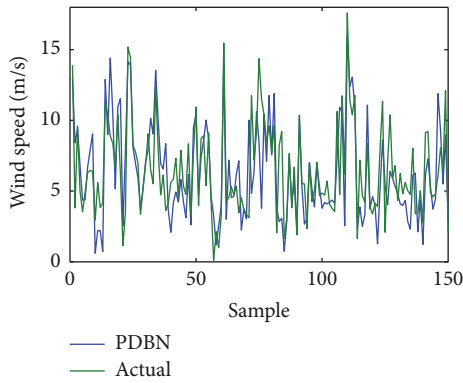
FIGURE 7: Continued.



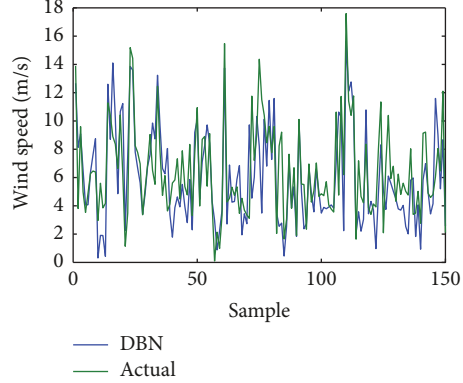
(i) The forecasting result of ELMAN for 30-min



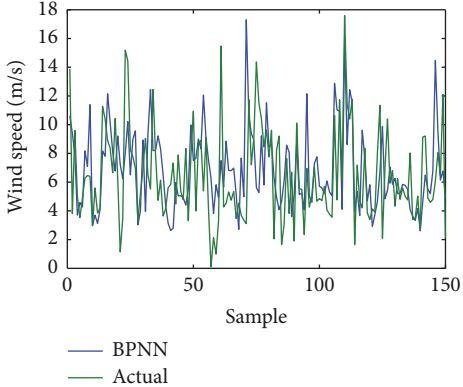
(j) The forecasting result of persistence model for 30-min



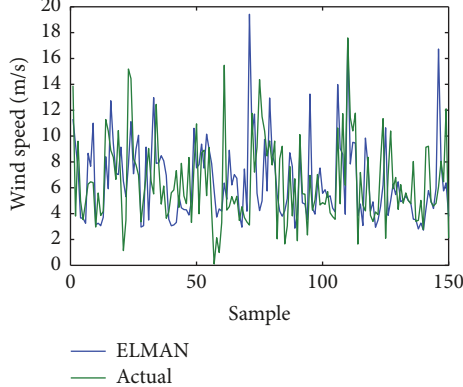
(k) The forecasting result of predictive DBN for 45-min



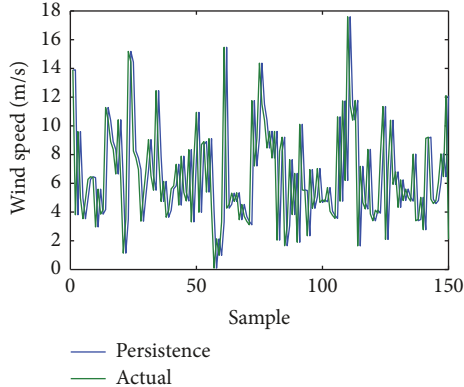
(l) The forecasting result of traditional DBN for 45-min



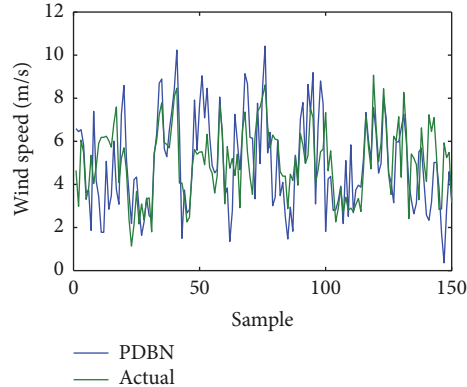
(m) The forecasting result of BPNN for 45-min



(n) The forecasting result of ELMAN for 45-min



(o) The forecasting result of persistence model for 45-min



(p) The forecasting result of predictive DBN for 24-h

FIGURE 7: Continued.

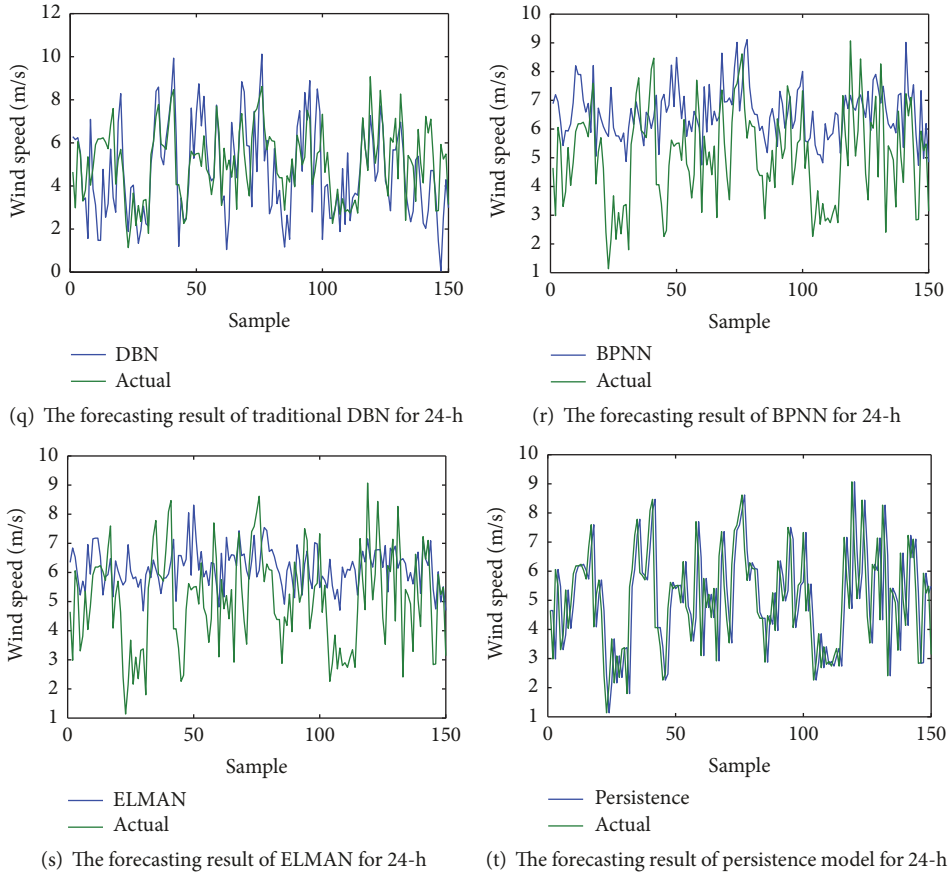


FIGURE 7: The results of the short-term wind speed prediction.

TABLE 3: The performance of wind speed forecasting models for different horizons.

Index	Cases	Predictive DBN	BPNN	ELMAN	Persistence	Traditional DBN
RMSE	Case 1	0.446	0.721	0.479	0.462	0.494
	Case 2	0.654	0.831	0.716	0.673	0.669
	Case 3	2.416	3.307	3.491	4.14	2.435
	Case 4	1.869	2.076	1.913	1.949	1.879
MAE	Case 1	0.34	0.555	0.362	0.337	0.386
	Case 2	0.508	0.684	0.566	0.519	0.523
	Case 3	1.786	2.494	2.553	3.08	1.809
	Case 4	1.421	1.749	1.558	1.581	1.425
BIAS	Case 1	0.051	0.403	0.065	0.004	0.064
	Case 2	0.037	0.526	0.015	0.003	0.038
	Case 3	-0.193	-0.4	0.259	0.078	0.186
	Case 4	-0.093	1.484	-1.085	0.1	0.024
SDE	Case 1	0.443	0.597	0.475	0.452	0.49
	Case 2	0.653	0.673	0.716	0.416	0.668
	Case 3	2.408	3.283	3.491	4.139	2.417
	Case 4	1.867	1.952	1.976	1.949	1.868
APE	Case 1	0.078	0.111	0.092	0.53	0.089
	Case 2	0.156	0.177	0.163	0.083	0.162
	Case 3	0.544	0.933	0.874	0.739	0.549
	Case 4	0.298	0.48	0.437	0.393	0.302

TABLE 4: The performance of wind power forecasting models for different horizons.

Models	Experiments	RMSE	MAE	BIAS	SDE	APE
The optimized random forest	Case 1	33.41	21.48	-9.26	32.11	0.97
	Case 2	21.84	14.37	-7.73	19.39	0.67
	Case 3	72.72	54.61	42.06	59.32	0.52
	Case 4	71.85	44.04	15.28	70.21	0.69
Random forest without weights updating	Case 1	34.58	24.61	-18.47	29.23	7.89
	Case 2	24.47	15.46	-10.08	23.23	0.69
	Case 3	140.1	72.12	49.79	130.96	0.71
Random forest without favor features	Case 4	120.8	63.2	27.36	84.37	1.34
	Case 1	52.03	36.80	2.52	51.97	10.93
	Case 2	37.81	23.87	-6.72	37.21	1.07
	Case 3	177.67	83.16	55.5	168.78	0.76
	Case 4	289.05	160.76	-59.77	282.81	2.55

TABLE 5: The performance of wind power forecasting models with 10-fold CV for different horizons.

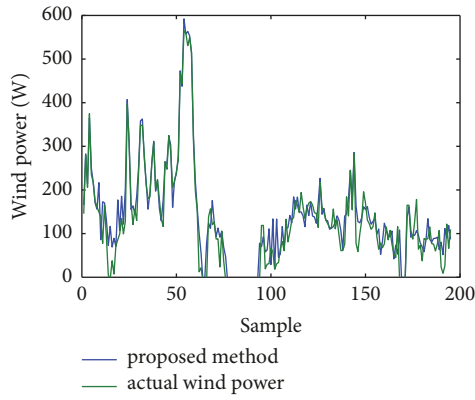
Models	Experiments	RMSE	MAE	BIAS	SDE	APE
The optimized random forest	Case 1	31.27	19.56	-8.39	32.08	0.95
	Case 2	20.59	13.65	-7.43	18.66	0.59
	Case 3	71.49	54.28	41.34	58.62	0.52
	Case 4	71.82	43.63	14.08	69.62	0.68
Random forest without weights updating	Case 1	34.26	23.87	-17.84	29.68	8.01
	Case 2	24.25	14.63	-9.78	22.78	0.65
	Case 3	133.45	67.54	45.65	128.57	0.67
	Case 4	111.34	60.49	27.36	82.43	1.12
Random forest without favor features	Case 1	50.36	31.24	2.69	49.51	9.58
	Case 2	35.67	22.76	-6.41	35.17	0.88
	Case 3	156.62	74.52	46.21	139.28	0.63
	Case 4	267.45	150.32	-47.53	271.47	2.13

horizons compared to BPNN; this demonstrates the ability of deep learning network to handle the highly varying wind speed series. As for the proposed model, predictive DBN outperforms traditional DBN with the total improvement of the RMSE, MAE, BIAS, SDE, and APE by 0.092, 0.088, 0.062, 0.072, and 0.026. This is mainly due to the employment of bat algorithm, which can not only reduce the training time, but also provide more accurate network architectures.

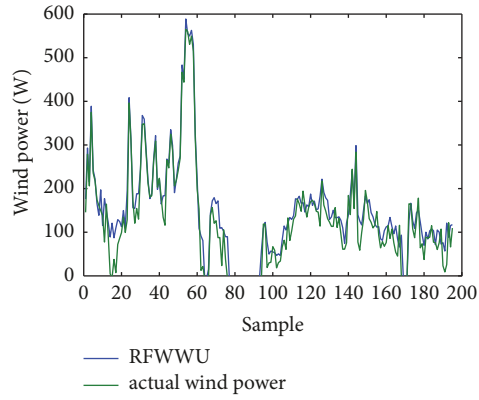
5.3. Short-Term Wind Power Forecasting Results. Graphical representation about the predicted and actual power value in four cases is shown in Figure 8. Figures 8(a)–8(c) denote the predicted wind power and actual value in Case 1 from the proposed method, random forest without the weights updating (RFWWU), and random forest without the favor features (RFWFF), respectively. Similarly, Figures 8(d)–8(f), Figures 8(g)–8(i), and Figures 8(j)–8(l) represent the predicted results from those models in Case 2, Case 3, and Case 4.

6. Discussion

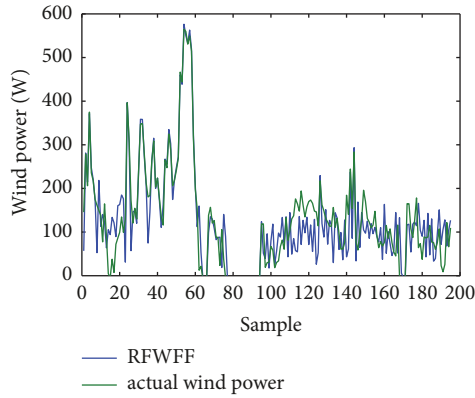
As seen clearly, the DBN together with optimized random forest can provide notably accurate future wind power series. In Case 1, Case 2, and Case 3, the lines of predicted values and the actual data are almost overlapping. It makes a little error in Case 4 because of the wind fluctuation and the longer time horizon, as can be seen from Figure 7(p). To further inspect the generalization ability of the proposed model, Table 4 reports the RMSE, MAE, BIAS, SDE, and APE indexes of the proposed model and unoptimized random forest. What is more, Table 5 shows the forecasting indexes by utilizing tenfold cross validation in training procedure of random forest. The training data would be divided into ten subsets. Each subset of the training data is used once for testing while being used nine times for training. CV method contributes to searching the optimal weights of the RF algorithm. From Table 5, the effectiveness of CV can be easily seen by comparing the indexes with Table 4. Most



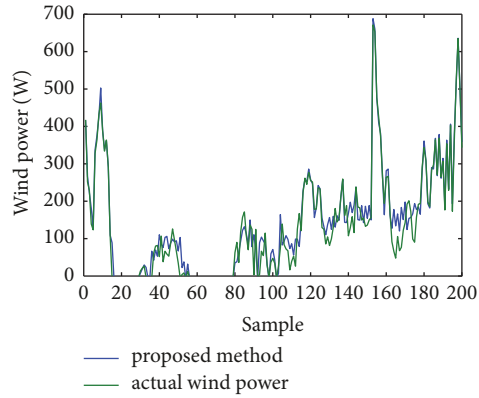
(a) The forecasting result of proposed method for 15-min



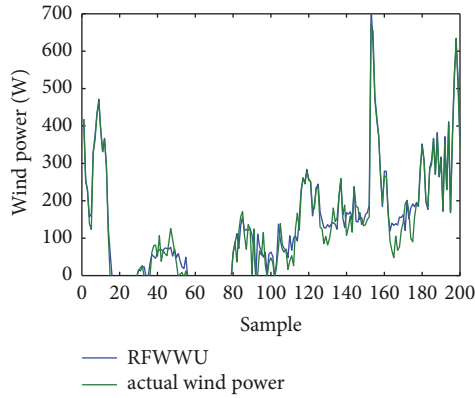
(b) The forecasting result of RFWWU for 15-min



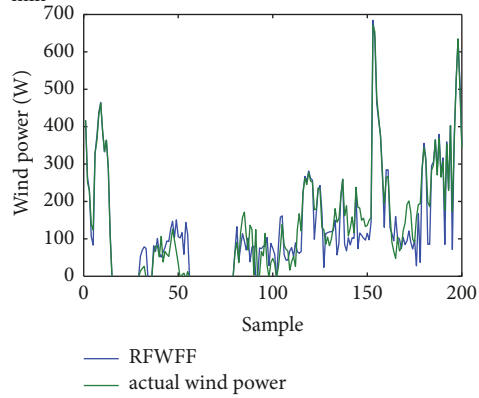
(c) The forecasting result of RFWFF for 15-min



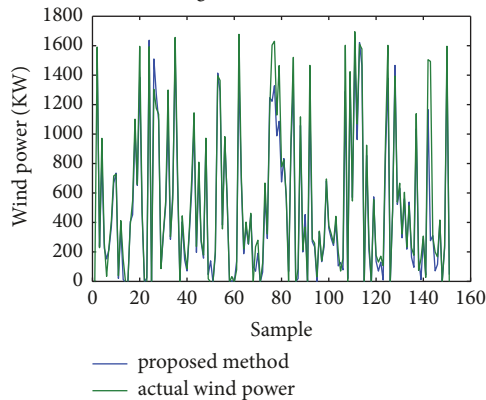
(d) The forecasting result of proposed method for 30-min



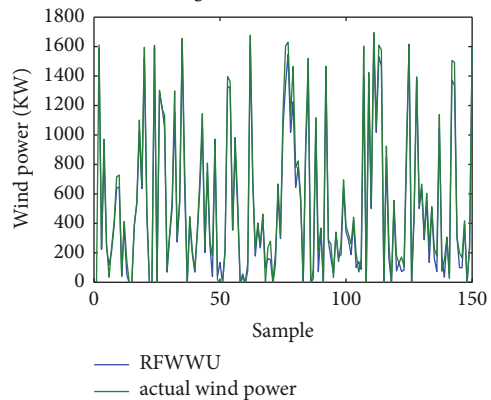
(e) The forecasting result of RFWWU for 30-min



(f) The forecasting result of RFWFF for 30-min



(g) The forecasting result of proposed method for 45-min



(h) The forecasting result of RFWWU for 45-min

FIGURE 8: Continued.

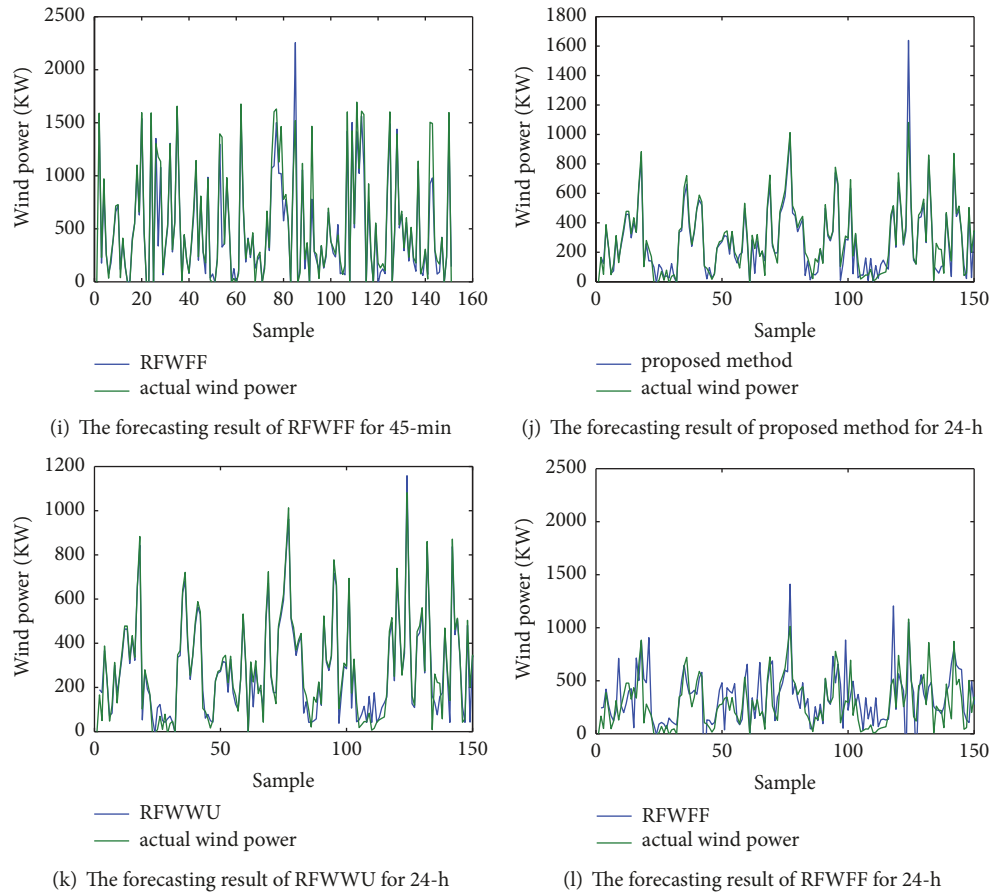


FIGURE 8: The results of the short-term wind power forecasting.

TABLE 6: The MRMR index and the favor features.

Attribute name	1	2	3	4	5
generator speed (CCU)	0.7404	-0.099	0.79038	-----	-----
Rotor speed	0.7398	-0.098	0.7901	0.4401	0.265
Wind speed	0.846	-----	-----	-----	-----
generator speed (PLC)	0.7404	-0.099	0.79036	0.4404	-----
The temperature of bearing A	0.024	-0.102	0.064	-0.002	-0.036
The temperature of bearing B	0.409	-0.098	0.427	0.238	0.144
The temperature of gearbox	0.509	-0.138	0.557	0.296	0.165
The ambient temperature	-0.183	-0.132	-0.173	-0.148	-0.136
The temperature of nacelle	-0.36	-0.069	-0.354	-0.243	-0.188
The temperature of gearbox bearing	0.675	-0.097	0.690	0.404	0.261
Blade 1 actual value	-0.523	0.2064	-----	-----	-----
Blade 2 actual value	-0.523	0.2063	-0.658	-0.3	-0.121
Blade 3 actual value	-0.523	0.2062	-0.658	-0.3	-0.121

indexes of the all cases degrade with CV in the training procedure except for some individual situations. From this observation, using the CV technology in the training process of random forest is a reasonable choice. It can be observed that the predicted accuracy of optimized random forest is greater than that of RFWWU and RFWFF for all cases. The performance of RFWWU is just a little bit worse

than the optimized random forest. The predicted values from RFWFF have the largest errors among those models, which from another point of view prove the importance of favor features. The results of MRMR index and the favor features are illustrated in Table 6. The favor features size is set to 5, and the bold one is chosen to be a favor feature.

7. Conclusions

In this paper, we proposed a multistep wind speed and wind power forecasting model combined with predictive deep belief network and optimized random forest to produce the future 15-min, 30-min, 45-min, and 24-h wind speed and wind power series. The DBN models the wind speed series by its deep learning ability, and the bat algorithm is employed to further enhance its inference performance. The performance of the predictive DBN is verified by four cases, and the results demonstrate that the DBN makes a major prediction accuracy increase. Wind speed has a pivotal effect on the wind power generation. Benefited by the approximated ability of the DBN model, random forest algorithm can procure better future wind power data with higher precise wind speed series than traditional WPF models. On the other hand, the competitive generalization property can be further obtained with the dimension reduction and weights updating in real-time.

However, there are still many problems for the wind speed and wind power forecasting model. One is that the learning time will be booming when it has a large scale. Therefore, a parallel deep learning algorithm is developing in recent years. The second one is the parameters selection; in this paper, we utilize the bat algorithm to search the parameter space. However, it is very difficult to determine the optimal number of layers, number of the units in each layer, and the number of epochs. These setting parameters affect the learning ability of the deep network significantly. Some automatic selection models should be developed to fix this problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Hexu Sun has proposed the idea of the optimized model. Zexian Sun has implemented the software code. Jingxuan Zhang has collected the experimental data.

Acknowledgments

This work obtains the support of Hebei Province Science and Technology Plan Project—Construction and Application of Wind Power “Smart Capsule” Cloud Management Platform Based on Big Data Technology (17214304D)—and funded project of Hebei Natural Science Foundation—Data Scheduling Optimization and Demand Forecasting of Discrete Manufacturing Industry Based on Big Data (F2018202206).

References

- [1] S. Fang and H.-D. Chiang, “Improving supervised wind power forecasting models using extended numerical weather variables and unlabelled data,” *IET Renewable Power Generation*, vol. 10, no. 10, pp. 1616–1624, 2016.
- [2] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen, and E. Feitosa, “A review on the young history of the wind power short term prediction,” *Renewable & Sustainable Energy Reviews*, vol. 12, no. 6, pp. 1725–1744, 2008.
- [3] L. Rongsheng, P. Minfang, Z. Haiyan, W. Xun, and S. Meie, “Ultra-short-term wind power prediction based on dynamical ensemble least square support vector regression,” *Journal of Hunan University (Natural Sciences)*, vol. 44, no. 4, pp. 79–86, 2017.
- [4] N. Amjady and O. Abedinia, “Short term wind power prediction based on improved kriging interpolation, empirical mode decomposition, and closed-loop forecasting engine,” *Sustainability*, vol. 9, no. 11, 2017.
- [5] Y. Wang, Q. Hu, D. Meng, and P. Zhu, “Deterministic and probabilistic wind power forecasting using a variational Bayesian-based adaptive robust multi-kernel regression model,” *Applied Energy*, vol. 208, pp. 1097–1112, 2017.
- [6] O. Karakuş, E. E. Kuruoğlu, and M. A. Altinkaya, “One-day ahead wind speed/power prediction based on polynomial autoregressive model,” *IET Renewable Power Generation*, vol. 11, no. 11, pp. 1430–1439, 2017.
- [7] N. Amjady, F. Keynia, and H. Zareipour, “Short-term wind power forecasting using ridgelet neural network,” *Electric Power Systems Research*, vol. 81, no. 12, pp. 2099–2107, 2011.
- [8] W. Wu and M. Peng, “A data mining approach combining K-means clustering with bagging neural network for short-term wind power forecasting,” *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 979–986, 2017.
- [9] C. G. Raji and S. S. Vinod Chandra, “Long-Term Forecasting the Survival in Liver Transplantation Using Multilayer Perceptron Networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2318–2329, 2017.
- [10] X. H. Yuan, C. Chen, Y. B. Yuan, Y. H. Huang, and Q. X. Tan, “Short-term wind power prediction based on LSSVM-GSA model,” *Energy Conversion and Management*, vol. 101, pp. 393–401, 2015.
- [11] E. Cadenas and W. Rivera, “Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks,” *Journal of Renewable Energy*, vol. 34, no. 1, pp. 274–278, 2009.
- [12] Q. Cao, B. T. Ewing, and M. A. Thompson, “Forecasting wind speed with recurrent neural networks,” *European Journal of Operational Research*, vol. 221, no. 1, pp. 148–154, 2012.
- [13] W. Zhang, J. Wang, J. Wang, Z. Zhao, and M. Tian, “Short-term wind speed forecasting based on a hybrid model,” *Applied Soft Computing*, vol. 13, no. 7, pp. 3225–3233, 2013.
- [14] K. Bhaskar and S. N. Singh, “AWNN-assisted wind power forecasting using feed-forward neural network,” *IEEE Transactions on Sustainable Energy*, vol. 3, no. 2, pp. 306–315, 2012.
- [15] D. Lee and R. Baldick, “Short-term wind power ensemble prediction based on gaussian processes and Neural networks,” *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 501–510, 2014.
- [16] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, “Wind power forecasts using Gaussian processes and numerical weather prediction,” *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 656–665, 2014.
- [17] J. P. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, “Hybrid intelligent approach for short-term wind power forecasting in Portugal,” *IET Renewable Power Generation*, vol. 5, no. 3, pp. 251–257, 2011.

- [18] S. Li, P. Wang, and L. Goel, "Wind Power Forecasting Using Neural Network Ensembles with Feature Selection," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1447–1456, 2015.
- [19] Q. Xu, D. He, N. Zhang et al., "A short-term wind power forecasting approach with adjustment of numerical weather prediction input by data mining," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1283–1291, 2015.
- [20] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough Deep Neural Architecture for Short-Term Wind Speed Forecasting," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2770–2779, 2017.
- [21] M. Ma, C. Sun, and X. Chen, "Discriminative Deep Belief Networks with Ant Colony Optimization for Health Status Assessment of Machine," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 12, pp. 3115–3125, 2017.
- [22] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 303–315, 2016.
- [23] S. Shao, W. Sun, P. Wang, R. X. Gao, and R. Yan, "Learning features from vibration signals for induction motor fault diagnosis," in *Proceedings of the 2016 International Symposium on Flexible Automation (ISFA '16)*, pp. 71–76, Cleveland, Ohio, USA, August 2016.
- [24] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2727–2736, 2017.
- [25] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2016.
- [26] C.-Y. Zhang, C. L. P. Chen, M. Gan, and L. Chen, "Predictive Deep Boltzmann Machine for Multiperiod Wind Speed Forecasting," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1416–1425, 2015.
- [27] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," *Applied Soft Computing*, vol. 58, pp. 742–755, 2017.
- [28] J. Chen, K. Li, Z. Tang et al., "A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 919–933, 2017.
- [29] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *IEEE Access*, vol. 5, pp. 16568–16575, 2017.
- [30] C. Dai, Z. Liu, K. Hu, and K. Huang, "Fault diagnosis approach of traction transformers in high-speed railway combining kernel principal component analysis with random forest," *IET Electrical Systems in Transportation*, vol. 6, no. 3, pp. 202–206, 2016.
- [31] A. M. Shah and B. R. Bhalja, "Fault discrimination scheme for power transformer using random forest technique," *IET Generation, Transmission & Distribution*, vol. 10, no. 6, pp. 1431–1439, 2016.
- [32] C. González, J. Mira-McWilliams, and I. Juárez, "Important variable assessment and electricity price forecasting based on regression tree models: Classification and regression trees, Bagging and Random Forests," *IET Generation, Transmission & Distribution*, vol. 9, no. 11, pp. 1120–1128, 2015.

