

Research Article

Multiperiod-Ahead Wind Speed Forecasting Using Deep Neural Architecture and Ensemble Learning

Lei Chen,^{1,2} Zhijun Li ,¹ and Yi Zhang³

¹School of Artificial Intelligence, Hebei University of Technology, Tianjin 300130, China

²Graduate School, North China University of Science and Technology, Tangshan, Hebei Province 063210, China

³College of Electrical Engineering, North China University of Science and Technology, Tangshan, Hebei Province 063210, China

Correspondence should be addressed to Zhijun Li; lzj@hebut.edu.cn

Received 9 March 2019; Revised 22 May 2019; Accepted 11 June 2019; Published 27 June 2019

Academic Editor: Juan P. Amezcua-Sanchez

Copyright © 2019 Lei Chen 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.

Accurate forecasting of wind speed plays a fundamental role in enabling reliable operation and planning for large-scale integration of wind turbines. It is difficult to obtain the accurate wind speed forecasting (WSF) due to the intermittent and random nature of wind energy. In this paper, a multiperiod-ahead WSF model based on the analysis of variance, stacked denoising autoencoder (SDAE), and ensemble learning is proposed. The analysis of variance classifies the training samples into different categories. The stacked denoising autoencoder as a deep learning architecture is later built for unsupervised feature learning in each category. The ensemble of extreme learning machine (ELM) is applied to fine-tune the SDAE for multiperiod-ahead wind speed forecasting. Experimental results are made to demonstrate that the proposed model has the best performance compared with the classic WSF methods including the single SDAE-ELM, ELMAN, and adaptive neuron-fuzzy inference system (ANFIS).

1. Introduction

Wind has been one of the rapidly growing energies due to its clean and free nature. However, the intrinsic volatility and deviation of wind energy impose major challenges on the operation and planning of power system. Wind speed plays a pivotal role in the wind power output [1]. Since wind power has a functional relationship with wind speed [2–5], accurate WSF can enhance the performance of wind power prediction. Prediction of multiperiod-ahead wind speed and resulting multiperiod-ahead wind power production enable energy scheduling strategies to maintain operational stability and improve cost performance of the microgrid with wind turbines [6]. Numerous approaches have been procured to capture the characteristics of the wind speed series. Generally speaking, WSF models can be classified into three types: physical methods, statistical methods, and artificial intelligence (AI) methods.

Physical models combine the meteorological attributes to highlight the forecasting performance. Numerical weather

prediction (NWP) is a well-known physical approach [7]. A novel wind speed model was applied by utilizing the Kalman filter method to minimize the forecasting errors of a NWP model [8]. However, NWP model runs only few times one day, limited to its uncertainties on initial atmosphere conditions, and thus is not suitable for fitting the short-term wind speed series [9], which evidently restricts the practical application.

Statistical methods mainly design a mathematical architecture to represent the inner characteristics of wind speed series [10, 11]. The statistical models such as autoregressive integrated moving average (ARIMA) models are constructed to deal with the linear and nonlinear features of wind speed series [12]. Hourly wind speed series forecasting is conducted by combining the ARIMA and generalized autoregressive conditional heteroskedasticity (GARCH) [13], and an assessment on the forecasting ability was compared with the physical models [7]. To evaluate the wind energy at certain site, a large number of probability density functions are performed to analyze the statistical characteristics of

wind speed series. The frequency analysis of wind speed was processed by two PDFs including the Pearson type V and the Burr [14]. The bimodal Weibull & Weibull PDF was applied to depict the distribution of wind speed series, which proved better fitting performance than the single Weibull function [15].

However, the statistical models would present the weak generalization on learning the nonlinear features, especially for the wind speed series.

With the capacity of mapping the nonlinear features, the intelligent methods are receiving great attentions in the WSF area. The evident advantages of AI methods are to predict the future wind speed series without any predefined mathematical tools. As the representative of AI methods, artificial neural network, due to the ability of handling noisy data, has offered a variety of approaches for multiperiod-ahead WSF, such as the modified EMD-based ANN [16], the radial basis function neural network (RBFNN), [17], and the feed-forward neural network [18]. The RBFNN as an adaptive linear element neural network and BP neural network were employed for wind speed prediction [19]. A wind speed forecasting strategy was proposed based on a two-hidden-layer BPNN [20].

Compared to the traditional neural network, deep learning algorithms can capture the deep features of the wind speed series. The deep learning algorithms have been successfully applied in the computer vision and speech signal processing [21]. Therefore, deep learning algorithms have been performed to tackle the complex characteristics of the wind speed series and get magnificent achievements in this domain [22–24]. A deep learning model for multistep wind speed forecasting was established by combining the LSTM network with ELM network. The experimental results demonstrate that the deep learning algorithms have the best performance among the benchmark models. In the WSF field, most approaches produce the predicted wind speed values directly from raw wind speed, but have not effectively analyzed the nonstationary volatility of the wind speed series before building the architectures.

In this paper, a novel multiperiod-ahead WSF method is developed. To mitigate the impacts of volatile wind speed series, the analysis of variance is applied to classify the time series into several categories, and the samples grouped together share near fluctuant level. Furthermore, SDAE is adopted to perform the unsupervised learning to extract hidden low-level nonlinear features and remove the noise data. Meanwhile, the fine-tuning process of SDAE is optimized by the ELM-based ensemble learner. Finally, the 15-min, 1-h, 4-h, 8-h, and 24-h wind speed series have been employed to evaluate the proposed model. The contributions of this paper are as follows.

- (1) The training samples are classified into several categories according to the variance of the wind speed series. The inherent volatility of wind speed series are fully taken into consideration in the subsequent research.
- (2) The SDAE is performed in each category which is generalized from (1) to remove the noise

data and reduce the dimension of the time series.

- (3) Nowadays, the top networks for the fine-tuning phase in the training procedure of deep learning architecture have mainly single learner whose performance is easily affected by noise data and new condition data. To get the optimal inferential accuracy, we propose an ensemble predictor of ELMs based on bootstrap sampling approach to quantify the irregular wind speed series.

The rest of paper is organized as follows. In the next section, a brief formulation for multiperiod-ahead WSF is introduced. The analysis of variance, SDAE architecture, and corresponding ensemble learners are presented in Section 3. The numerical results are shown in Section 4. A conclusion is drawn in Section 5.

2. Related Background

2.1. Multiperiod-Ahead Wind Speed Forecasting. The wind speed series can be denoted as $S = \{s_1, s_2, \dots, s_{i-1}, s_i, s_{i+1}, \dots\}$, where s_i is the average wind speed in the past 10-min [9]. For multiperiod-ahead WSF, the future wind speed value $s_{i+\tau}$ is obtained by utilizing the previous N data, where i is the index of wind speed series and τ represents the predicted horizon. We assume that the wind speed is a realization of explicit function of the form

$$f(s_{i-1}, \dots, s_{i-N}, \Theta) = s_{i+\tau} \quad (1)$$

where Θ is the parameters space and f is the function modeling the wind speed. The parameters space is estimated by minimizing the loss function which is denoted as

$$\Theta = \arg \min_{\theta \in \Theta} \sum_i (s_p - s_a)^2 \quad (2)$$

where s_a is the real wind speed value and s_p is the predicted value from function f .

2.2. Autoencoder Neural Network. As a three-layer network, the autoencoder maps the input data into the output value to perform the reconstructed procedure. It aims at minimizing the reconstruction error by performing unsupervised learning. Figure 1 depicts the structure of autoencoder.

The autoencoder utilizes the deterministic function f_θ to obtain the hidden representation from the input data according to

$$h = f_\theta(x) = \text{sig}(Wx + b) \quad (3)$$

where sig is the activation function, W is the weight matrix, and b is the bias vector. Then the output layer maps h into the reconstructed vector z by

$$z = g_v(h) = \text{sig}(W'h + b') \quad (4)$$

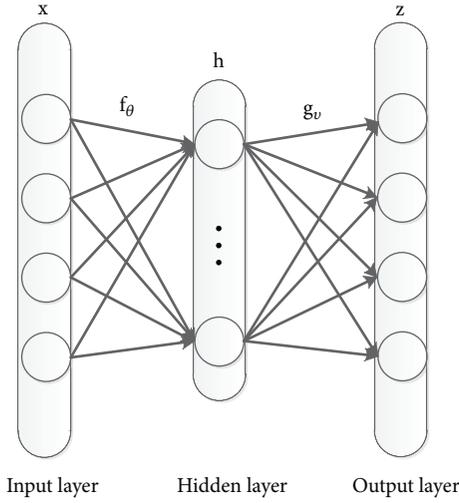


FIGURE 1: The structure of autoencoder.

where $v = \{W', b'\}$. The optimal parameters space is searched by minimizing the loss function as given by the following.

$$\theta^*, v^* = \arg \min_{\theta, v} \frac{1}{n} \sum_{i=1}^n \|x^{(i)} - g_v(h)\|^2 \quad (5)$$

However, the training process of the autoencoder only focuses on copying the input to output, which will arouse the overfitting problem and cannot guarantee the efficiency of the extracted features.

3. Wind Speed Forecasting Strategy

3.1. Analysis of Variance for Wind Speed Series. The wind speed series is highly varying due to the characteristics of randomness and fluctuation. In the proposed methodology, the variance of each wind speed series is first carried out to categorize the training samples according to

$$mean = \frac{\sum_{i=1}^n s_i}{n} \quad (6)$$

$$variance = \frac{\sum_{i=1}^n (s_i - mean)^2}{n} \quad (7)$$

where n is the number of s_i .

Then the variance series is sorted in ascending order as illustrated in Figure 2.

As shown in Figure 2, efficacious intervals are defined between the minimum and maximum variance, which can be denoted as intervals = $\{[\alpha_1, \alpha_2], [\alpha_2, \alpha_3], \dots, [\alpha_{m-1}, \alpha_m]\}$. We can derive the following rule from the value of α_i .

If the variance of one training sample meets the condition $\alpha_{i-1} \leq \text{variance} \leq \alpha_i$, the training sample is added to the corresponding category.

Samples with similar fluctuant level being gathered into one group can alleviate the influence of volatility in wind speed series efficiently. Then each predictive SDAE whose parameters space is built in corresponding category can have better approximated capacity.

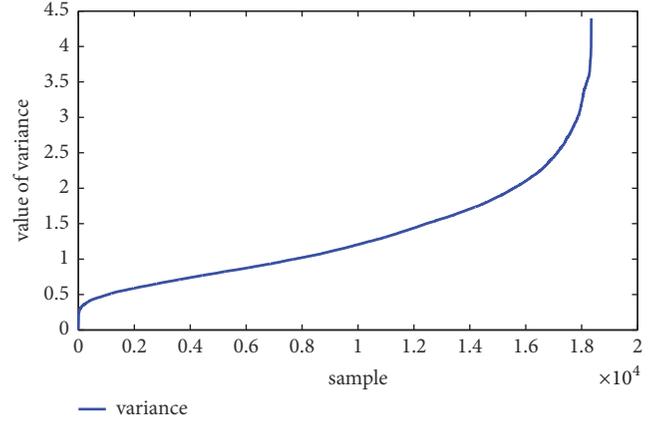


FIGURE 2: The variance series for the training sample in ascending order.

3.2. Stacked Denoising Autoencoders. An autoencoder (AE) is an artificial neural network with one single input layer, hidden layer, and output layer. The contribution of AE is to learn the latent representation for the input vectors by minimizing the reconstructed error in an unsupervised way.

The training procedure of a denoising autoencoder (DAE) is similar to AEs except for the corrupted input, as shown in Figure 3.

The corrupted version of inputs which adds noise to the raw inputs by stochastic mapping $\tilde{x} \sim q_D(\tilde{x} | x)$ is encoded by

$$c(\tilde{x}) = \text{sig}(W\tilde{x} + b) \quad (8)$$

where \tilde{x} is the corrupted input by stochastic mapping, $c(\tilde{x})$ is the latent representation of the hidden layer, sig is the logistics activate function, W is the weight matrix connecting input layer and hidden layer, and b is the bias vector of units belonging to the hidden layer.

The decoding part is done by mapping the $c(\tilde{x})$ into the original feature space by

$$z = \text{sig}(W'c(\tilde{x}) + b') \quad (9)$$

where z is reconstructed inputs, W' is the weight matrix connecting hidden layer and output layer, and b' is the bias vector of units belonging to the output layer.

The errors between reconstructed inputs and raw inputs must exist. The optimal DAE model can be obtained by

$$\theta = \arg \min_{\theta \in \Theta} \|x - z\|^2 \quad (10)$$

where x is the raw inputs, z is the reconstructed inputs, and θ is the parameters space containing weight matrix and bias vectors.

SDAE is composed of stacked DAEs that aim at learning more robust features, as presented in Figure 4. After the former layer is trained, the raw inputs are utilized to generate the latent representation as the input to the next DAE. The above-mentioned is the process of pretraining. It should be pointed out that this process is unsupervised and label information is not required.

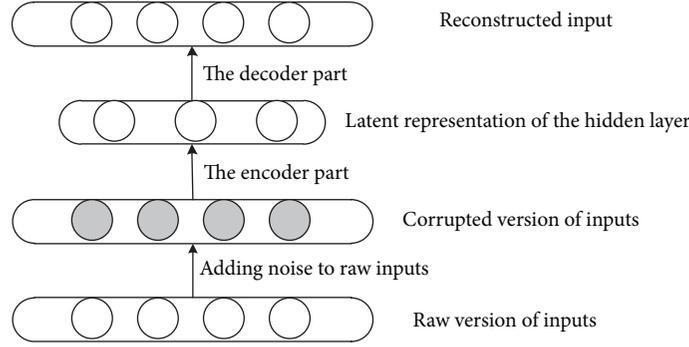


FIGURE 3: The reconstruction process of DAE.

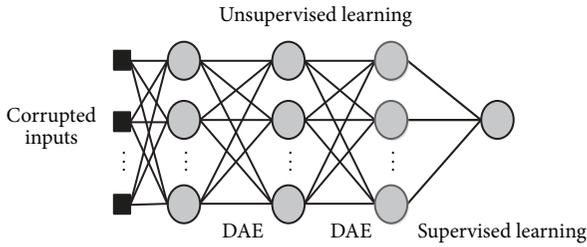


FIGURE 4: The architecture of SDAE.

When the pretraining procedure has been done, the top layer would be used to fine-tune the parameters space with the loss function. The fine-tuning stage which applies the desired forecasting target values for the supervised learning of the parameters in the neural network [23]. In order to promote the ability of fitting big data, the ensemble of ELMs have been applied.

3.3. The Ensemble of ELMs. Most conventional top networks of deep learning architectures only have one single learner, which may result in the degradation of the performance in the situation of handling big data with noise and new condition data. To tackle this problem, the ensemble of ELMs based on bootstrap sampling technology are applied to perform the fine-tuning procedure.

Given n output samples from SDAE $[y_1, y_2, \dots, y_n]$ and the corresponding target data $[t_1, t_2, \dots, t_n]$, the outputs from the ELM with an activation function $sig()$ can be described as

$$\sum_{j=1}^L \beta_j sig(w_j^T \cdot y_i + b_j) = t_i, \quad i = 1, \dots, n \quad (11)$$

where $w_j = [w_{j1}, \dots, w_{jN}]^T$ is the weight matrix connecting nodes between input layer and hidden layer. In ELM, the weight matrices w_j and b_j are randomly generated without considering the input data. N is the number of units in input layer. $\beta_j = [\beta_{j1}, \dots, \beta_{jM}]^T$ is the weight matrix connecting nodes between hidden layer and output layer, and M is the number of units in output layer. b_j is the bias of the

j th hidden node. The outputs from ELM can be rewritten as

$$f(y_i) = h(y_i) \beta, \quad i = 1, \dots, n \quad (12)$$

where $h(y_i)$ is the output from hidden nodes. To get the better approximation, ELM aims to solve the following formulation.

$$\min \frac{1}{2} \|\beta\|^2 + \frac{C}{2} \sum_{i=1}^N \|e_i\|^2 \quad (13)$$

$$h(y_i) \beta = y_i^T - e_i^T, \quad i = 1, \dots, n \quad (14)$$

As described in [25], the weight matrix connecting nodes between hidden layer and output layer can be easily determined as

$$\beta = H^\dagger T = \left(\frac{I}{C} + H^T H \right)^{-1} H^T T \quad (15)$$

where I is the identity matrix.

Given the superiority within the ELM, we apply it in the ensemble learner which synthesizes all the forecasting results of base learners to generate the final result. In the ensemble learner, bootstrap sample technique is procured to improve diversity among the base learners. Overall, the ensemble learner is applied to perform the fine-tuning process of the SDAE and enhance the forecasting performance. The training process of the ensemble learner is briefly described in Algorithm 1.

3.4. The WSF Strategy. The proposed model is composed of the analysis of variance, SDAE, and the ELM-based ensemble learner and creates the efficient hybrid system profiting by the inherent advantages from the mentioned methods.

The working principle of the proposed model is divided into two parts: the training procedure and prediction procedure, as illustrated in Figure 5.

The following steps are a summary of training procedure:

- (1) Compute the variance of each training sample and sort it in the ascending order.
- (2) Classify the training samples into several categories according to the variance and record the intervals.
- (3) Build the SDAE architecture for feature learning in each category.

```

Input: the data output from SDAE  $[y_1, y_2, \dots, y_n]$ 
(1) Initialization: set the size of ensemble learners  $Q$ , the number of units in visible nodes,
    hidden nodes and output nodes in each ELM, the length of training samples for each ELM, threshold  $\epsilon$ .
(2) The training procedure:
    while(the size of ensemble  $\neq Q$ ){
        (1) select the training samples in bootstrap way
        (2) train the base predictor, namely, ELM.
        (3) Filter the test data  $Y_{test} = Y - Y_{train}$  and compute the error
        (4) if(error <  $\epsilon$ )then
            Add it to the ensemble learner.
    }
Output: the ensemble learner
    
```

ALGORITHM 1: Training the ensemble learner.

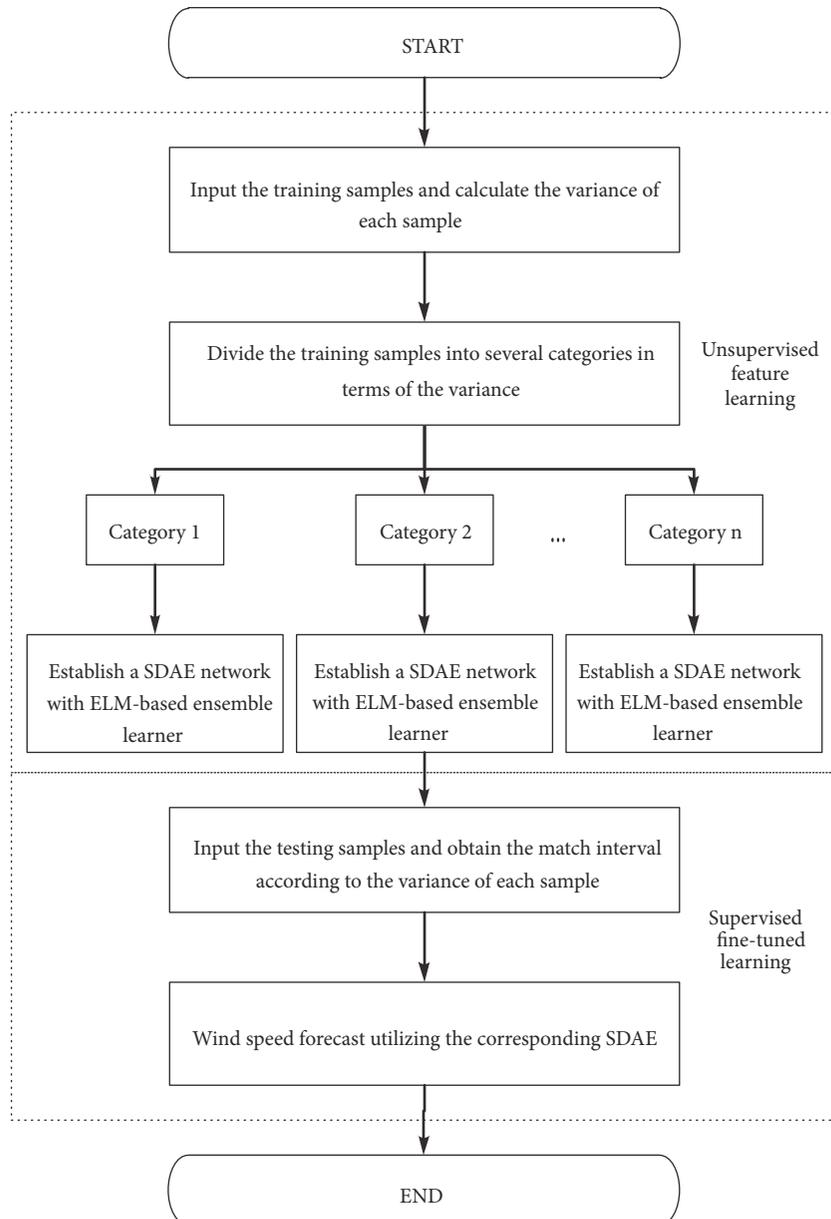


FIGURE 5: The training procedure of the proposed model.

- (4) Train the ELM-based ensemble learner and take the output from SDAE as input for the ensemble predictor.

The main steps in the predicted procedure are shown as follows:

- (1) Obtain the variance for the single testing data and match it into the corresponding interval.
- (2) Perform the unsupervised learning by SDAE for the input data.
- (3) Make a prediction utilizing the ensemble learner with the input data composed of outputs from SDAE
- (4) Compute the average of the fitting outputs of the ensemble learners as the forecasting result.

4. Numerical Results

4.1. Performance Criteria. After the analysis of variance, training of the deep learning architecture, and the ELM-based ensemble learner being finished, performance criteria in terms of the root mean square error (RMSE), the mean absolute error (MAE), the bias (BIAS), the standard deviation of the error (SDE), and the average percentage error (APE) are estimated on the testing data.

$$RMSE = \sqrt{\frac{1}{m} \sum_{j=1}^m (y_j - y_t)^2} \quad (16)$$

$$MAE = \frac{1}{m} \sum_{j=1}^m |y_j - y_t| \quad (17)$$

$$BIAS = \frac{1}{m} \sum_{j=1}^m (y_j - y_t) \quad (18)$$

$$SDE = \sqrt{\frac{1}{m} \sum_{j=1}^m (y_j - y_t - BIAS)^2} \quad (19)$$

$$APE = \frac{1}{m} \sum_{j=1}^m \frac{|y_j - y_t|}{|y_t|} \quad (20)$$

where y_j is the predicted value from the proposed model, y_t is the actual value of wind speed, and m is the size of testing data.

4.2. Data Collection. The 15-min, 1-h, 4-h, 8-h, and 24-h wind speed series in six months from three different wind farms in the northern part of China have been collected to analyze the performance of the forecasting model. The original wind speed series consists of average observations in the past five minutes. For 15-min horizon, the experimental series are obtained by every 1 data item. For 1-h, 4-h, 8-h, and 24-h, the training and testing sets are collected by every 3, 15, 31, and 95 data items, respectively. The sizes of multiperiod training samples and testing samples are listed in Table 1.

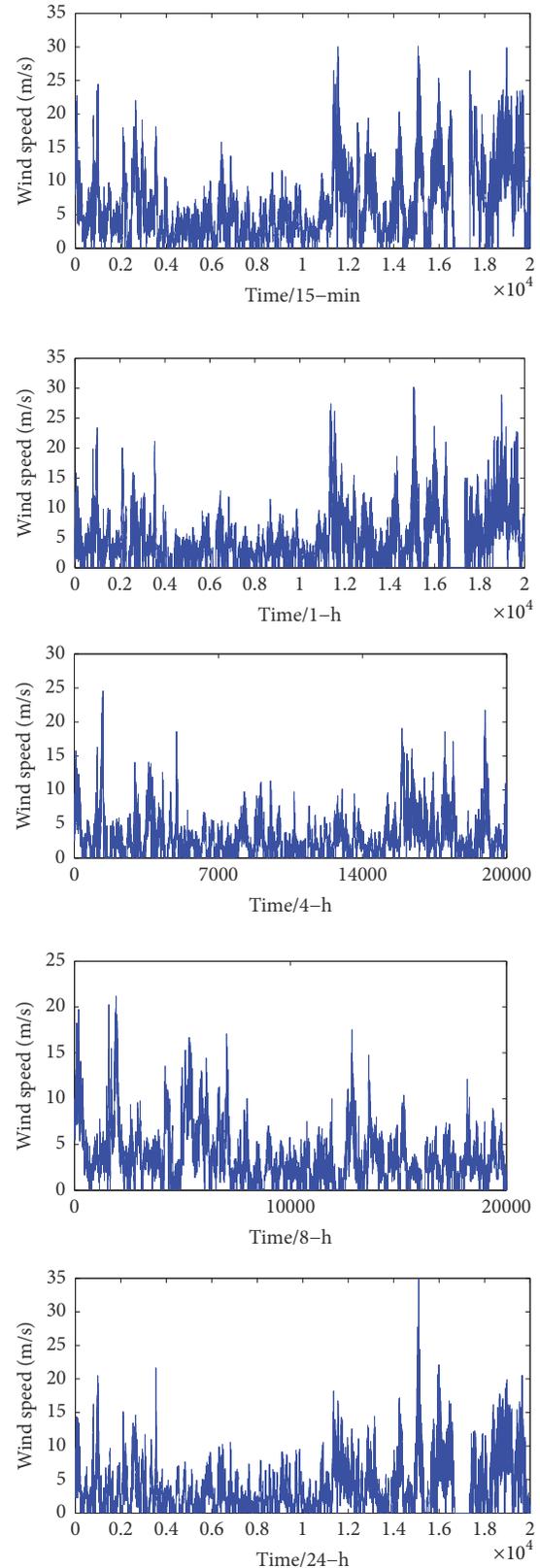


FIGURE 6: The trend of training sets for WSF.

Figure 6 shows the trends of different training samples. However, noise components inevitably exist in the raw wind

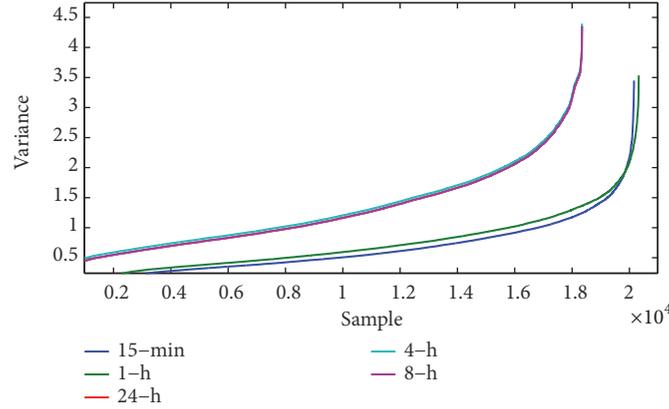


FIGURE 7: The variance of the five training samples.

TABLE 1: The sizes of multiperiod training samples and testing samples.

	Training samples	Testing samples
15-min	20000	600
1-h	20000	600
4-h	20000	600
8-h	20000	600
24-h	20000	600

speed series. In order to reduce the negative effects caused by the noise data and depict the characteristics of randomness and intermittency of wind speed, the analysis of variance has been effectively employed. Consequently, SDAE can be built to further perform features learning from the training samples.

4.3. Experimental Setting. In the three experiments, to capture the characteristics of wind speed series, we use the historical values $\{s_{k-i}, \dots, s_{i-1}\}$ to forecast the wind speed s_i . To further improve the fitting ability of the proposed method, the k-fold cross validation partitions the training set 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. The root mean square error is taken as a criterion to select the optimal model.

In the proposed model, the analysis of variance classifies the training samples into r categories and the size of ensemble model would be set as n. The number of neurons in each SDAE is 10-8-6-4. As a result, for the SDAE with 10 inputs, there are 80, 48 and 24 connecting weights in each layer, respectively. The ELM as the base learner in the ensemble model has 8 hidden neurons. Totally, there are $(152 \times r) + (33 \times n) \times r$ nonlinear parameters trained by the proposed learning algorithm.

To further demonstrate the improvements derived from the analysis of variance and ensemble learning, the single SDAE-ELM model, which has the same architecture with the

base learner in the proposed model, is selected as one of the benchmark models. Thus, there are 185 weights to be trained when single SDAE-ELM model is used to predict the wind speed series.

The ELMAN neural network as a recurrent neural network can memorize the dependencies under the wind speed series by feeding the outputs from the hidden layer to the hidden layer. Therefore, the ELMAN neural network which is composed of four layers, namely, input layer, hidden layer, recurrent layer, and output layer, is suitable for modeling and predicting the fluctuation of wind speed series. In this paper, 14 neural nodes in the hidden layer with sig function as the activate function are designed. The predicted procedure of the ELMAN neural network can be expressed as

$$h_{jt}(k) = f_h \left[\sum_{i=1}^r (w_{ij}^i x_{it}(k) + b_i) + \sum_{j=1}^n w_j^r h_{jt}(k-1) \right] \quad (21)$$

$$y_{t+1}(k) = f_o \left[\sum_j w_j^o h_{jt}(k) + b_j \right] \quad (22)$$

where h_{jt} denotes the output of the hidden layer; x_{it} is the input time series at time t; w_{ij}^i, w_j^r, w_j^o are the connecting weights between the input layer and hidden layer, between the recurrent layer and hidden layer, and between the hidden layer and output layer, respectively.

The ANFIS method combining the advantages of fuzzy inference system and neural network is a powerful tool modeling the time series. In this paper, we assign 5 fuzzy sets in each input. Hence, there are 5^n linear parameters to be trained during the learning process, and n is the length of input which is set to 10 in this paper.

4.4. The Analysis of Variance for the Training Samples. The analysis of variance, as a strong nonparametric tool, is conducted to extract the components with similar volatility level from the raw wind speed series. Figure 7 illustrates the

TABLE 2: Some of the intervals of the variance on training samples and the corresponding size of the category.

15-min		1-h		4-h		8-h		24-h	
intervals	size								
[0, 0.25)	3306	[0, 0.23)	3532	[0, 0.20)	2653	[0, 0.17)	39	[0, 0.28)	41
[0.25, 0.5)	6545	[0.23, 0.53)	6158	[0.20, 0.49)	5758	[0.17, 0.68)	1625	[0.28, 0.56)	1691
[0.5, 0.75)	4210	[0.53, 0.75)	4317	[0.49, 0.69)	4594	[0.68, 0.86)	3921	[0.56, 0.84)	3870
[0.75, 0.8)	628	[0.75, 0.8)	711	[0.69, 0.79)	697	[0.86, 1.09)	3761	[0.84, 1.12)	3607
.....
[2.91, 3.01)	8	[2.09, 3.12)	11	[2.56, 3.35)	34	[3.52, 3.75)	42	[3.8, 4.02)	27
[3.01, 3.45]	17	[3.12, 3.15]	28	[3.35, 3.35]	24	[3.75, 4.24]	8	[4.02, 4.39]	6

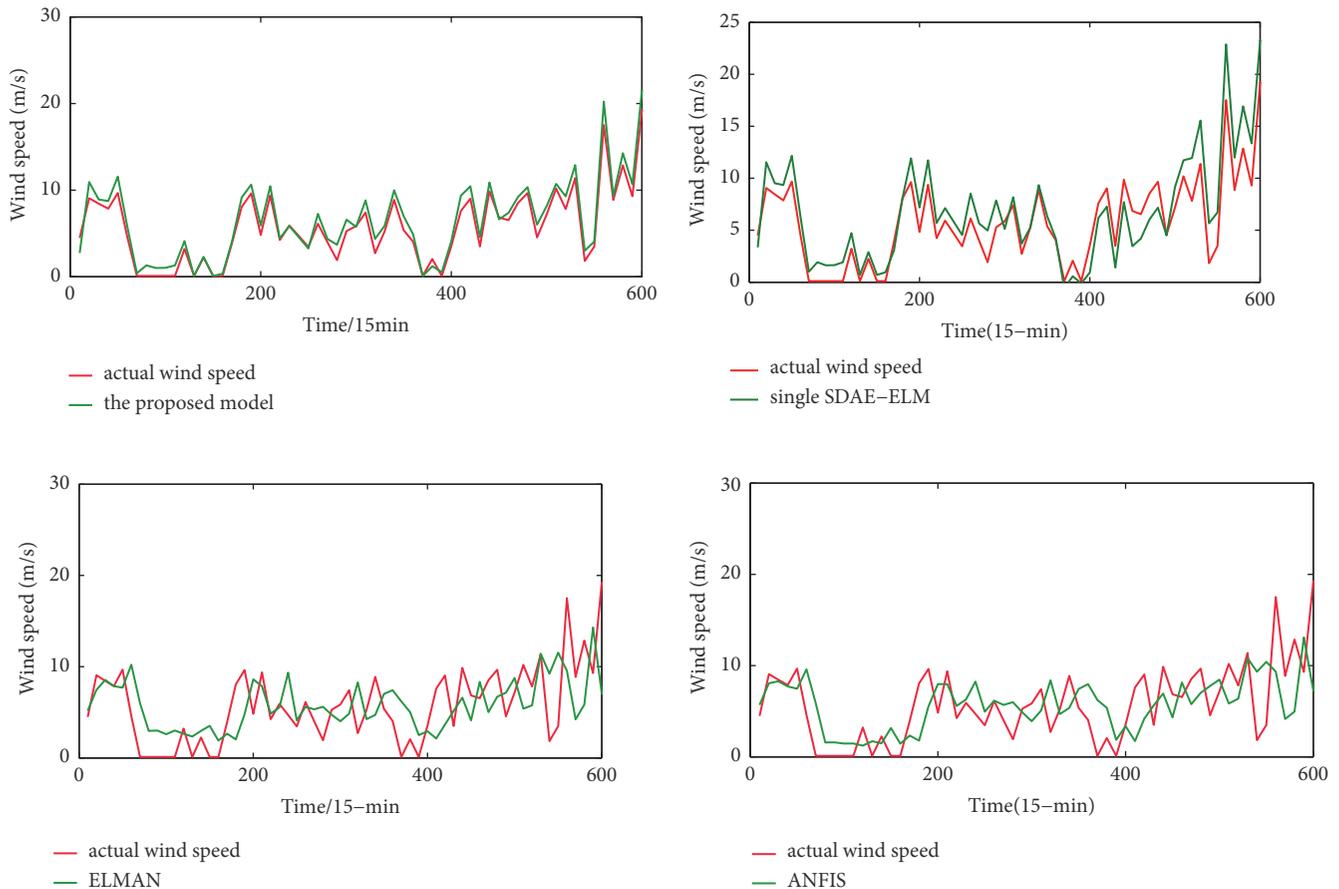


FIGURE 8: The comparison of predicted performance between the proposed model and several classical models for 15-min wind speed forecasting.

variance of the wind speed series in ascending order for 15-min, 1-h, 4-h, 8-h, and 24-h horizons.

To further improve forecasting accuracy, we map the similar components into one group. A SDAE architecture and ELM-based ensemble learner are performed in each group. Table 2 shows some of the intervals of the variance and the size of the corresponding group.

4.5. Multiperiod-Ahead Wind Speed Forecasting. To evaluate the feasibility and high accuracy of the proposed model,

several simulations are conducted on the 15-min, 1-h, 4-h, 8-h, and 24-h horizons. Figures 8–12 exhibit the predicted wind speed series and the corresponding actual values from the proposed model and classical WSF models including the single SDAE-ELM, ELMAN, and ANFIS models.

See Figures 8–12 for the accuracy and stability of the proposed model. The predicted series generated from different models for the multiperiod-ahead wind speed forecasting are shown in Figures 8–12, which demonstrate

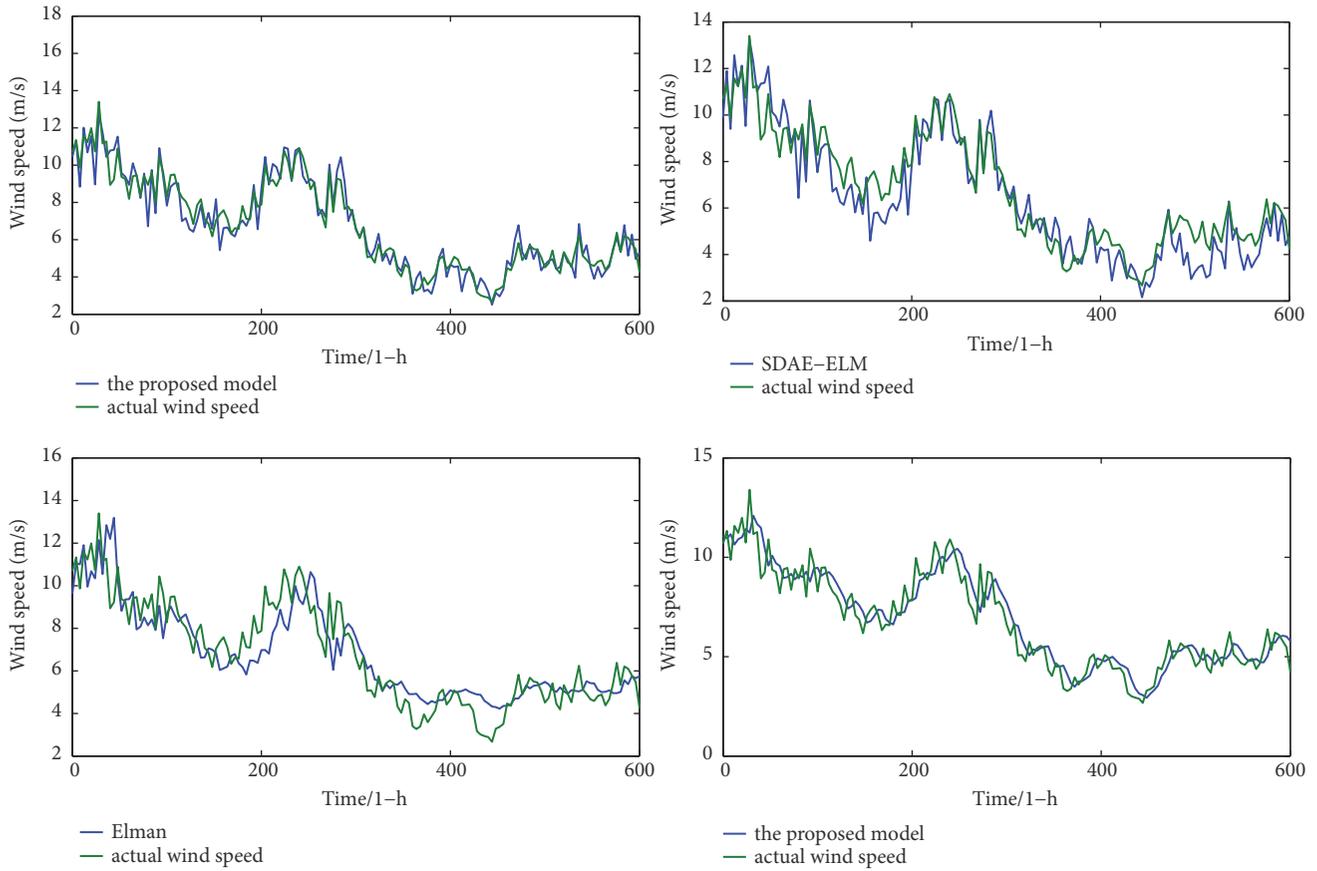


FIGURE 9: The comparison of predicted performance between the proposed model and several classical models for 1-h wind speed forecasting.

TABLE 3: Performance criteria of forecasting models.

Indices	The proposed model					The single SDAE-ELM				
	RMSE	MAE	BIAS	SDE	APE	RMSE	MAE	BIAS	SDE	APE
15-min	1.30	1.11	-1.00	0.83	0.64	2.19	1.79	-1.26	1.80	1.23
1-h	1.38	0.98	-0.67	0.76	0.68	1.59	1.26	-1.12	1.21	0.87
4-h	1.54	1.33	-0.87	1.27	0.96	1.67	1.68	-1.59	1.40	1.35
8-h	1.98	1.52	-0.9	1.99	1.24	2.22	1.95	-1.30	2.12	1.29
24-h	1.87	1.42	-0.09	1.87	0.30	2.64	2.24	-1.30	2.30	0.51
Indices	ELMAN					ANFIS				
	RMSE	MAE	BIAS	SDE	APE	RMSE	MAE	BIAS	SDE	APE
15-min	4.28	3.29	0.10	4.28	3.43	4.13	3.13	0.22	4.12	3.29
1-h	4.52	3.35	0.98	4.76	3.58	4.38	2.86	0.88	3.74	2.98
4-h	4.67	3.59	1.76	4.69	3.68	4.88	4.09	1.34	4.58	3.76
8-h	5.12	3.97	-1.65	5.12	3.98	5.34	4.28	-1.59	5.13	4.01
24-h	1.91	1.56	-1.08	1.58	0.44	2.29	1.82	0.92	2.09	0.50

that the predicted performance produced by the proposed model in each time scale is the highest. In other words, the proposed model has the lowest predicted error compared with the three classical WSF models. In addition, the classical models perform worse in the long-time

forecasts, since the fluctuation of the long-time series is very high.

Here, the aforementioned five statistical metrics are utilized to demonstrate better approximated capacity of the proposed model. Table 3 records the performance criteria for the

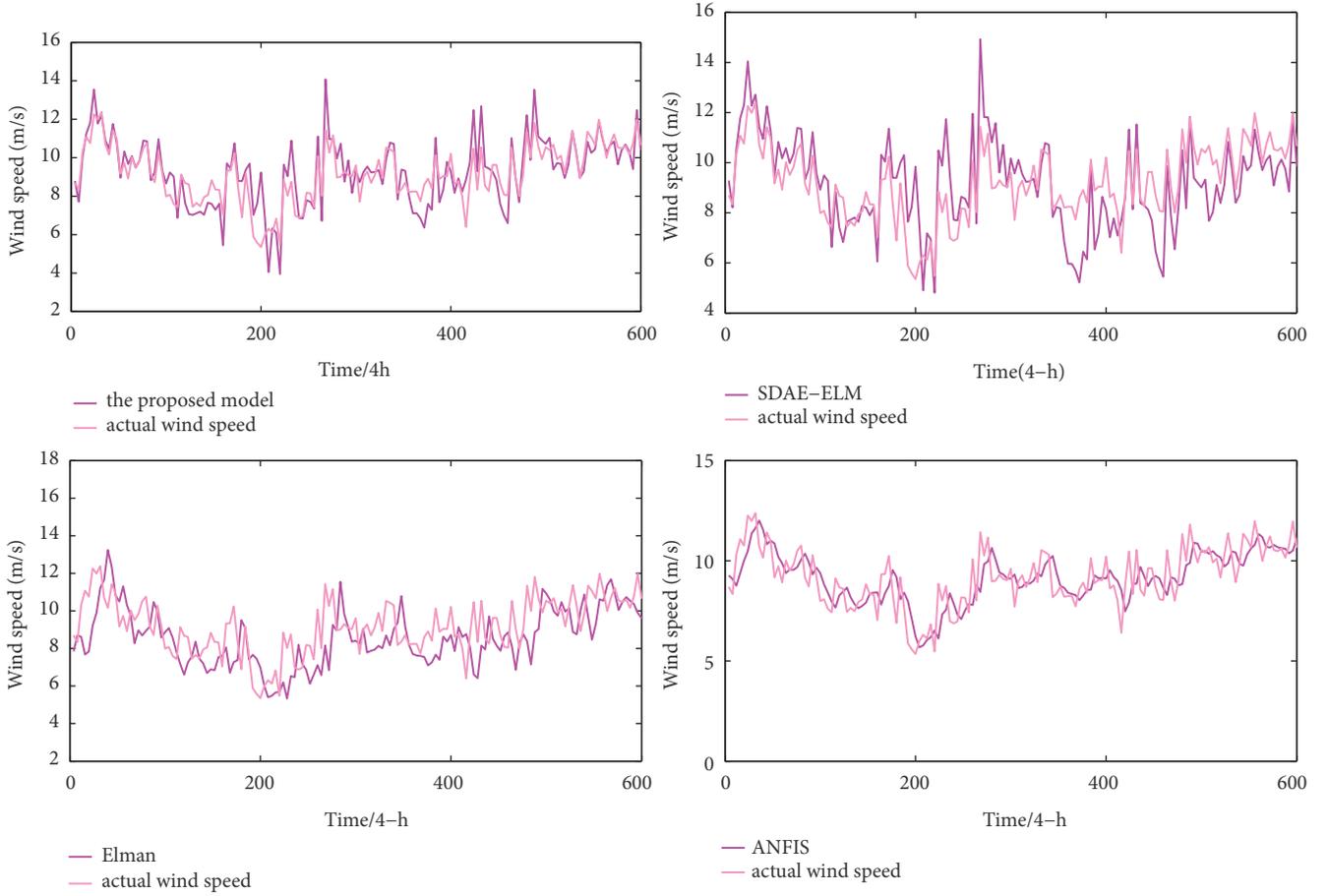


FIGURE 10: The comparison of predicted performance between the proposed model and several classical models for 4-h wind speed forecasting.

proposed model, the single SDAE-ELM, ELMAN, and ANFIS for different horizons. As shown in Table 3, the proposed model achieves higher inference than the other classical models. Compared with the single SDAE-ELM which has the same architecture as the base learner in the proposed model, in the forecasting horizons of 1-h, the proposed model reduces the RMSE from 1.59 to 1.38, the MAE from 1.26 to 0.98, the BIAS from -1.12 to -0.67, and the APE from 0.87 to 0.68. This phenomenon is also suitable for the comparison with ELMAN and ANFIS. Additionally, the 4-h forecasting results have been recorded in Table 3. The proposed model produces 0.13, 0.35, 0.72, 0.13, and 0.39 improvements of RMSE, MAE, BIAS, SDE, and APE in terms of the comparison with the single SDAE-ELM, respectively. Besides, the promotion of the proposed model for performance criteria compared with classical models on 15-min, 1-h, 4-h, 8-h, and 24-h WSF is shown in Tables 4–8. Consequently, the performances of the forecasting models degrade as the horizon is prolonged. However, in the 24-h studied horizon, the proposed model in this paper generates the closest value to the real wind speed among the other benchmark models, which demonstrates that the proposed

TABLE 4: The promotion of the proposed model for performance criteria compared with classical models on 15-min WSF.

	single SDAE-ELM	ELMAN	ANFIS
RMSE	0.89	2.98	2.83
MAE	0.68	2.18	2.02
BIAS	0.26	1.1	1.22
SDE	0.97	3.45	3.29
APE	0.59	2.79	2.65

model is more efficient and precise for multiperiod-ahead wind speed forecasting. Finally, in order to further determine the optimal forecasting model, the residuals of the above-mentioned models in the different horizons are given in Figure 11.

As exhibited in Figure 13, most residuals from the proposed model are less than the benchmark models except for very few data. Overall, the residuals of the proposed model can be nearly cancelled out in the short-term WSF. In terms of the long-term horizon, the proposed model achieves

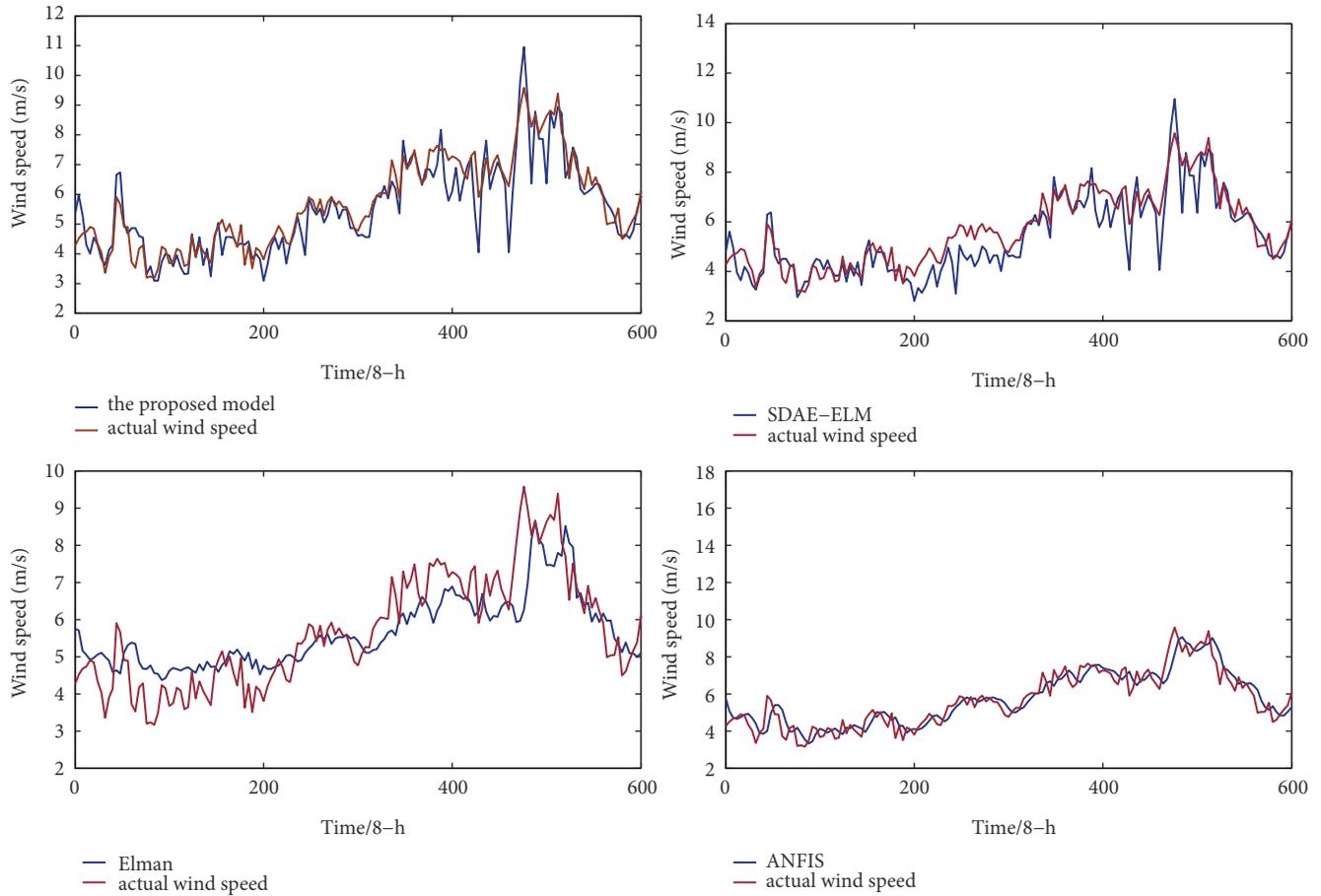


FIGURE 11: The comparison of predicted performance between the proposed model and several classical models for 8-h wind speed forecasting.

TABLE 5: The promotion of the proposed model for performance criteria compared with classical models on 1-h WSF.

	single SDAE-ELM	ELMAN	ANFIS
RMSE	0.21	3.14	3
MAE	0.28	2.37	1.88
BIAS	0.14	0.31	0.21
SDE	0.45	4	2.98
APE	0.19	2.9	2.3

TABLE 6: The promotion of the proposed model for performance criteria compared with classical models on 4-h WSF.

	single SDAE-ELM	ELMAN	ANFIS
RMSE	0.13	3.13	3.34
MAE	0.35	2.26	2.76
BIAS	0.72	0.99	0.47
SDE	0.13	3.42	3.31
APE	0.39	2.72	2.8

the evidently fitting capacity. Accordingly, the proposed approach can perform much better given the forecasting accuracy.

5. Conclusions

This paper proposes a hybrid architecture employing the analysis of variance, stacked denoising autoencoders, and ELM-based ensemble predictors to conduct multiperiod-ahead wind speed forecasting. This model has been evaluated

on three subsets of data collected from a wind farm. Experimental studies reveal that the proposed model has the best approximation compared to the other alternatives including the single SDAE-ELM, ELMAN, and ANFIS models in terms of RMSE, MAE, BIAS, SDE, and APE for different horizons. The superiority of the proposed model is attributed to the analysis of variance, the deep learning architecture, and the ensemble technology. The first categorizes the training and testing samples by similar volatility level to reduce the probability of noise data in the subsequent procedure.

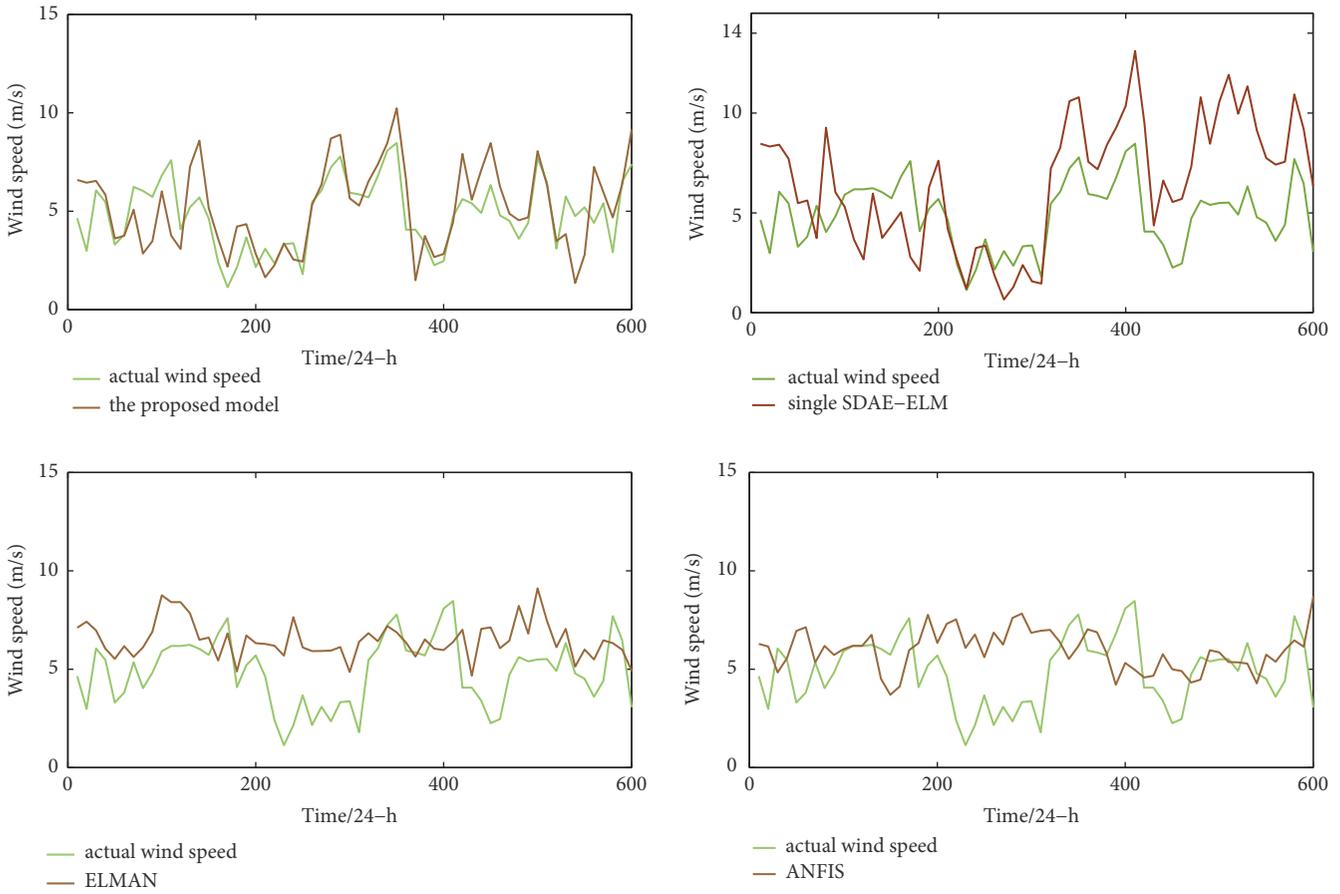


FIGURE 12: The comparison of predicted performance between the proposed model and several classical models for 24-h wind speed forecasting.

TABLE 7: The promotion of the proposed model for performance criteria compared with classical models on 8-h WSF.

	single SDAE-ELM	ELMAN	ANFIS
RMSE	0.24	2.54	3.32
MAE	0.43	2.45	2.76
BIAS	0.4	0.75	0.69
SDE	0.13	3.03	3.04
APE	0.05	2.74	2.77

The latter effectively performs the unsupervised learning to extract the stochastic features in each wind speed category. Eventually, benefiting from the strong robustness, high accuracy, and stability, the ELM-based ensemble predictor cancels the error of the base learner out. In addition, the application of the proposed model in other fields deserves further investigation.

TABLE 8: The promotion of the proposed model for performance criteria compared with classical models on 24-h WSF.

	single SDAE-ELM	ELMAN	ANFIS
RMSE	0.77	0.04	0.42
MAE	0.82	0.14	0.4
BIAS	1.21	0.99	1.01
SDE	0.43	0.29	0.22
APE	0.21	0.14	0.2

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.

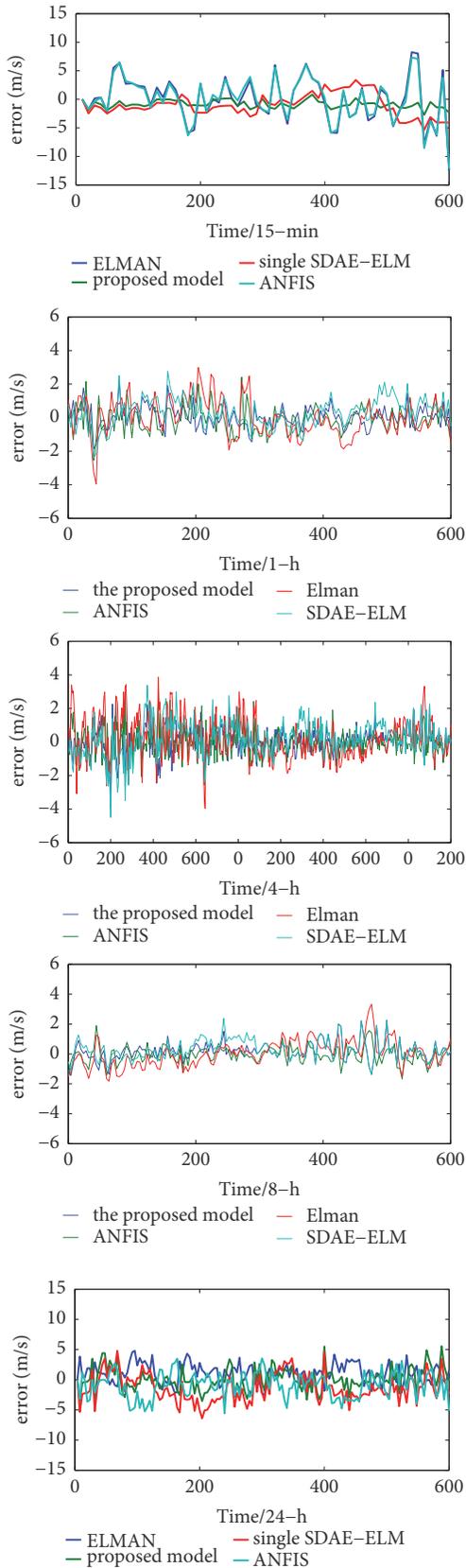


FIGURE 13: The residuals of the above-mentioned models in the different horizons.

Acknowledgments

This work is supported by National Natural Science Foundation of China: Robust Distributed Model Predictive Control for Load Frequency of Interconnected Power Systems under Wind Power Intervention under Grants 61803154, and Hebei Province Science and Technology Plan Project: Research on Energy Storage Control of Wind-solar Complementary Generation Grid-connected System under Grant 15214511.

References

- [1] C. Li, L. Wang, G. Zhang, H. Wang, and F. Shang, "Functional-type single-input-rule-modules connected neural fuzzy system for wind speed prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 751–762, 2017.
- [2] 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.
- [3] 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.
- [4] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.
- [5] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis, and P. S. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 352–361, 2004.
- [6] M. Mao, J. Ling, L. Chang, N. D. Hatziaargyriou, J. Zhang, and Y. Ding, "A novel short-term wind speed prediction based on MFEC," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 4, pp. 1206–1216, 2016.
- [7] S. Buhan, Y. Özkazanç, and I. Çadirci, "Wind pattern recognition and reference wind mast data correlations with nwp for improved wind-electric power forecasts," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 991–1004, 2016.
- [8] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output," *Applied Energy*, vol. 99, pp. 154–166, 2012.
- [9] T. Niu, J. Wang, K. Zhang, and P. Du, "Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy," *Journal of Renewable Energy*, vol. 118, pp. 213–229, 2018.
- [10] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable & Sustainable Energy Reviews*, vol. 31, pp. 762–777, 2014.
- [11] E. Erdem and J. Shi, "ARMA based approaches for forecasting the tuple of wind speed and direction," *Applied Energy*, vol. 88, no. 4, pp. 1405–1414, 2011.
- [12] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Journal of Renewable Energy*, vol. 34, no. 5, pp. 1388–1393, 2009.
- [13] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 775–782, 2009.

- [14] V. Lo Brano, A. Orioli, G. Ciulla, and S. Culotta, "Quality of wind speed fitting distributions for the urban area of Palermo, Italy," *Journal of Renewable Energy*, vol. 36, no. 3, pp. 1026–1039, 2011.
- [15] O. Jaramillo and M. Borja, "Wind speed analysis in La Ventosa, Mexico: a bimodal probability distribution case," *Journal of Renewable Energy*, vol. 29, no. 10, pp. 1613–1630, 2004.
- [16] Z. H. Guo, W. G. Zhao, H. Y. Lu, and J. Wang, "Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model," *Journal of Renewable Energy*, vol. 37, no. 1, pp. 241–249, 2012.
- [17] 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.
- [18] 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.
- [19] G. W. Chang, H. J. Lu, Y. R. Chang, and Y. D. Lee, "An improved neural network-based approach for short-term wind speed and power forecast," *Journal of Renewable Energy*, vol. 105, pp. 301–311, 2017.
- [20] W. Sun and Y. Wang, "Short-term wind speed forecasting based on fast ensemble empirical mode decomposition, phase space reconstruction, sample entropy and improved back-propagation neural network," *Energy Conversion and Management*, vol. 157, pp. 1–12, 2018.
- [21] M. Långkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.
- [22] H. Wang, G. Li, G. Wang, J. Peng, H. Jiang, and Y. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Applied Energy*, vol. 188, pp. 56–70, 2017.
- [23] 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.
- [24] 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.
- [25] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics Part B-Cybern*, vol. 42, no. 2, pp. 513–529, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

