Energy Conversion and Management 136 (2017) 439–451

Contents lists available at ScienceDirect

Energy Conversion and Management

journal homepage: www.elsevier.com/locate/enconman

A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting

Wenyu Zhang, Zongxi Qu *, Kequan Zhang, Wenqian Mao, Yining Ma, Xu Fan

College of Atmospheric Sciences, Key Laboratory of Arid Climatic Change and Reducing Disaster of Gansu Province, Lanzhou University, Lanzhou 730000, China

ARTICLE INFO

Article history:
Received 19 August 2016
Received in revised form 30 December 2016
Accepted 8 January 2017
Available online 25 January 2017

Keywords:
Wind speed prediction
Combined model
Weight coefficient optimization
Complete ensemble empirical mode decomposition adaptive noise

ABSTRACT

Wind energy, which is stochastic and intermittent by nature, has a significant influence on power system operation, power grid security and market economics. Precise and reliable wind speed prediction is vital for wind farm planning and operational planning for power grids. To improve wind speed forecasting accuracy, a large number of forecasting approaches have been proposed; however, these models typically do not account for the importance of data preprocessing and are limited by the use of individual models. In this paper, a novel combined model – combining complete ensemble empirical mode decomposition adaptive noise (CEEMDAN), flower pollination algorithm with chaotic local search (CLSFPA), five neural networks and no negative constraint theory (NNCT) – is proposed for short-term wind speed forecasting. First, a recent CEEMDAN is employed to divide the original wind speed data into a finite set of IMF components, and then a combined model, based on NNCT, is proposed for forecasting each decomposition signal. To improve the forecasting capacity of the combined model, a modified flower pollination algorithm (FPA) with chaotic local search (CLS) is proposed and employed to determine the optimal weight coefficients of the combined model, and the final prediction values were obtained by reconstructing the refined series. To evaluate the forecasting ability of the proposed combined model, 15-min wind speed data from four wind farms in the eastern coastal areas of China are used. The experimental results of this study show that: (a) the proposed CEEMDAN-combined model can take advantages of individual models and has the best performance among single models and the benchmark model; (b) the proposed CLSFPA is superior to FPA according to test functions and is effectively applied in optimizing the combined model; (c) the proposed algorithms are effective in high-precision wind speed predictions.

1. Introduction

Wind power, which has a large generating capacity, has been gaining an increasing share of the world's attention. However, the key problem with wind power utilization is that intermittent wind energy sources cause power instability and significant fluctuations when large quantities of wind energy penetrate the power grid [1–3]. Moreover, wind turbines may be damaged by fluctuating wind speed, so reliable and precise wind speed prediction is vital for wind power generation systems. To improve the precision of wind speed predictions, numerous methods have been proposed and developed in recent decades. These approaches can be divided into three general types: physical models, conventional statistical models, and artificial intelligence models [1,2,4–6]. Physical models use current meteorological and geographical data based on the atmosphere of physical processes for forecasting but are poor at short-term wind speed simulation. Conventional statistical models, in contrast, draw on vast historical data based on mathematical models usually involving conventional time series analysis, such as ARMA, ARIMA models [7,8], and achieve more accurate short-term wind speed predictions than physical models. However, the fluctuating and intermittent characteristics of wind speed sequences require more complicated functions to capture non-linear relationships rather than assuming a linear correlation structure [9]. Given

Abbreviations: complete ensemble empirical mode decomposition adaptive noise, CEEMDAN; no negative constraint theory, NNCT; flower pollination algorithm with chaotic local search, CLSFPA; China National Renewable Energy Center, CNREC; auto regressive moving average, ARMA; autoregressive integrated moving average, ARIMA; artificial neural network, ANN; radial basis function, RBF; support vector machine, SVM; back propagation, BP; wavelet neural network, WNN; particle swarm optimization, PSO; chaotic particle swarm optimization, CPSO; genetic algorithm, GA; differential evolution, DE; genetic simulated annealing algorithm, GSA; feed-forward neural network, FNN; Elman neural network, ENN; general regression neural network, GRNN; intrinsic mode function, IMF; square sum of the error, SSE; average absolute forecast error of n times forecast results, MAE; root mean-square forecast error, RMSE; average of absolute error, MAPE; Diebolde-Mariano (DM) test, DM test.

* Corresponding author.
E-mail address: quzx14@lzu.edu.cn (Z. Qu).

http://dx.doi.org/10.1016/j.enconman.2017.01.022
0196-8904/

the development of statistical models along with the advent of artificial intelligence techniques, artificial intelligence models, including Artificial Neural Networks (ANNs) and other mixed methods, have been proposed and are widely used in the field of wind speed forecasting[1–3,6,9]. For instance, De Giorgi et al. [10] adopted the ANNs to forecast wind speeds; compared to the linear time-series-based model, the ANNs provide a robust approach for wind prediction. Due to the chaotic nature of wind time series, Guo et al. [11] proposed a forecasting approach employing a back propagation neural network (BPNN) and seasonal exponential adjustment for forecasting wind speed series. Wang et al. [12] exploited a radial basis function (RBF) neural network for wind speed forecasting, and the effectiveness of this method was proved by a practical case. Zhou et al. [13] proposed a model based on a support vector machine (SVM), which has good precision of prediction in short-term wind speed forecasting. Wang et al. [14] successfully exploited a wavelet neural network (WNN) based on both back propagation (BP) neural network and wavelet analysis theory for wind speed prediction, resulting in good performance.

However, a large amount of forecasting approaches have been proposed for wind speed prediction; all of these methods have improved the precision of wind speed predictions to some extent. However, wind speed time series are highly noisy and unstable; therefore, using the primary wind speed series directly to establish prediction models is subject to substantial errors [32,33]. To build an effective prediction model, the features of original wind speed datasets must be fully analyzed and considered. The Ensemble Empirical Mode Decomposition (EEMD) [34] is an advanced, effective decomposition technology; it makes up for the deficiency of EMD [35] and has been widely used in the decomposition of wind speed series. For example, Jiang et al. [36] also proposed a hybrid forecasting model based on EEMD in which the original data series are decomposed into certain signals with different frequencies and then the grey support vector machine (GSVM) is employed for forecasting. Hu et al. [37] proposed a hybrid method based on the EEMD to disassemble the original wind speed datasets into a series of independent Intrinsic Mode Functions (IMFs), and use SVM to predict the values for IMFs in different frequencies. Zhou et al. [38] additionally proposed a hybrid forecasting method based on EEMD and the generalized regression neural network (GRNN). Although EEMD offers improvement and is widely used in many fields, the added noise cannot be completely canceled, which is a defect. Torres et al. [39] proposed complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), a novel modification method based on EEMD. Ye et al. [40] performed a comparative test on EMD and three developed versions (EEMD, CEEMD and CEEMDAN) and combined SVR for wind speed forecasting; the results showed that CEEMDAN-SVR outperforms the other three methods.

The aforementioned EMD-based forecasting models only employ a single forecasting model to predict all of the decomposition signal sequences. However, different sub-series have different characteristics, meaning that a simple forecasting individual model has its unavoidable drawbacks and is not always able to adapt to all properties of the decomposition series [15]. Therefore, a combination forecasting model, which can take advantage of each single model, would be helpful to avoid the drawbacks of individual models and further improve forecasting precision. To further study the combined models, many evolutional optimization algorithms are applied to help determine the optimal weights of the combination model. Xiao et al. [16] proposed a combined model based on chaotic particle swarm optimization (CPSO) to optimize the weight coefficients for wind speed forecasting. Yang et al. [17] used Differential Evolution (DE) to optimize the weight coefficients of a combination model for electricity demand forecasting. Xiao et al. applied the cuckoo search algorithm (CSO) to determine the optimal weights of the combination model. In this paper, the Flower Pollination Algorithm with Chaos Local Search (CLSFPA), which is a new modified meta-heuristic algorithm, is proposed to optimize the weight coefficients of the combined model. Flower Pollination Algorithm (FPA), proposed by Yang in 2012 [18], is a novel bionics evolution algorithm that imitates flower pollination behavior in nature. The performance of the FPA against the GA and PSO algorithm tested by Yang show that the FPA has strong optimization ability and performs better than both the GA and PSO algorithm. However, when addressing more complex problems, FPA suffers from the problem of falling into local optimum solutions and slow convergence rates. To improve the performance of the FPA with respect to optimized problems, many new modifications have been proposed. Dubey et al. [19] presented a modified FPA that combines the FPA and an evolution DE method for a wind–thermal system and improves the capabilities of synergy and joint search. Chakraborty et al. [20] proposed FPA-GSA to optimize the performances of feed-forward neural network (FNN); the results showed that FPA-GSA outperform both the FPA and Gravitational Search Algorithm (GSA). Tahani et al. [21] proposed a FPA/SA algorithm that combines FPA and Simulated Annealing (SA) to increase system efficiency and decrease costs. This study aimed to enhance exploration capacities and improve the local searching ability of the flower pollination algorithm. A new modified FPA with chaotic local search (CLS) method is developed to improve the local search ability of the FPA; the performance and feasibility of the proposed algorithm were tested by five testing functions. In this paper, the CLSFPA is employed to determine the weights of the combined model based on five forecasting models and address decomposition signal series with different characteristics. First, the original data are decomposed into different IMF components with corresponding frequencies by CEEMDAN, and then each component is predicted by a weight-determined combined method, which is optimized by CLSFPA for improved forecasting accuracy.

The rest of the paper is organized as follows. Section 2 introduces individual forecasting models, the combined forecasting model theory and CEEMDAN. Section 2.4 describes FPA and CLAFPA. Section 3 describes the main procedure of the proposed CEEMDAN-CLSFPA combined Model. Section 4 describes the performance metric used for this study. Section 5 discusses the forecasting results of the proposed combined model and the comparative results. Finally, Section 6 concludes by outlining the important results of this paper.

2. Related methodology

In this paper, the five single neural network forecasting models, CEEMDAN, CLSFPA and NNCT, which are applied in the proposed combined model, are introduced as follows.

2.1. Individual forecasting algorithms

In the past decades, many ANN forecasting models have been proposed and used in wind speed forecasting. In this paper, four popular ANNs, BPNN, RBFNN, ENN, GRNN and WNN, which have shown good performance in wind speed forecasting, are used to construct the combined model. The details are shown in Appendix A.

2.2. CEEMDAN

The EMD technique, proposed by Huang et al. [22], is an adaptive data analysis method and has been applied in analyzing non-linear and non-stationary data. It decomposes complex signals into Intrinsic Mode Functions (IMFs), which satisfy the following

conditions: (a) In the entire data sequence, the number of extrema and the number of zero crossings in an entire sampled data set must either be equal or differ at most by one and (b) the mean value at any point of the envelope defined by the local maxima and the envelope defined by the local minima is zero. With the hypothesis of decomposition and the definition of the IMF above, the EMD process of a raw data series $n_{(t)}$ $(t = 1, 2, \ldots, T)$ can be formulated as

$$n(t) = \sum_{k=1}^{m} imf_m(t) + r_m(t) \tag{1}$$

where $n(t)$ denotes any nonlinear and non-stationary signal, $imf_m(t)$ is the m-th IMF of the signal, and $r_m(t)$ is the residual item, which can be a constant or the signal mean trend.

The EMD process can decompose non-stationary and non-linear signals into numbers of components, but the mode mixing problem [23] is encountered frequently in practical applications. Due to the aforementioned drawback of EMD, the advent of the EEMD method [24] was proposed. The EEMD is a noise-assisted data analysis method that defines true IMF components as the mean of an ensemble of trials, with each trial adding a white noise of finite amplitude to the decomposition results of the signal. As a result, the white noise series, incorporated into the original signal, can provide a uniform reference scale to facilitate the EMD process and consequently help extract the true IMFs. After fulfilling its duty, the added white noise can cancel itself via ensemble averaging.

However, the EEMD has problems with high computational cost and contains residual noise. The number of trial increases may cause the number of sifting process increases and different realizations of signal plus noise may produce a different number of modes; as a result, final averaging is difficult. To decrease the number of trials while keeping the capacity to solve the mode mixing problem, a CEEMDAN was developed in [25]. Compared to EMD, EEMD and CEEMD, the CEEMDAN has the following advantages: (1) An extra noise coefficient vector w is introduced to control the noise level at each decomposition; (2) reconstruction is complete and noise free; and (3) it requires less trials than EEMD and CEEMD. The detailed steps of CEEMDAN are shown in Fig. 2(b).

In our study, we select the optimal standard deviation from 0.1 to 0.5 and set the number of ensemble members as 200. Fig. 1 presents the original wind speed series decomposed by CEEMDAN.

2.3. Combined model based on NNCT

The traditional constant weight combination model is always based on positive weight determination, but if the combination forecasts have significant similarities or differences in forecasting accuracy, the weights for the best forecasts are assigned a value of 1 and the worst is assigned a value of 0. To overcome this problem, no negative constraint theory (NNCT), which was proposed by Xiao et al. [16], was applied to the weight-determined combination model.

Definition 1. An improved combination model based on the proposed no negative constraint theory (NNCT), is given as follows:

$$\min \ \mathbf{S} = \mathbf{W}^T \mathbf{E} \mathbf{W} = \sum_{t=1}^{T} \sum_{j=1}^{m} \sum_{i=1}^{m} w_i w_j e_{it} e_{jt} \tag{2}$$

$$\textit{st} \ \mathbf{C}^T \mathbf{W} = [-2, 2] \tag{3}$$

where $\mathbf{W} = (w_1, w_2, \ldots, w_m)^T$ is the weight vector and $\mathbf{C} = (1, 1, \ldots, 1)^T$ is a column vector where all elements are 1. $\mathbf{E} = (\mathbf{E}_{ij})_{m \times m}$ is called the error information matrix and $\mathbf{E}_{ij} = e_i^T e_j$,

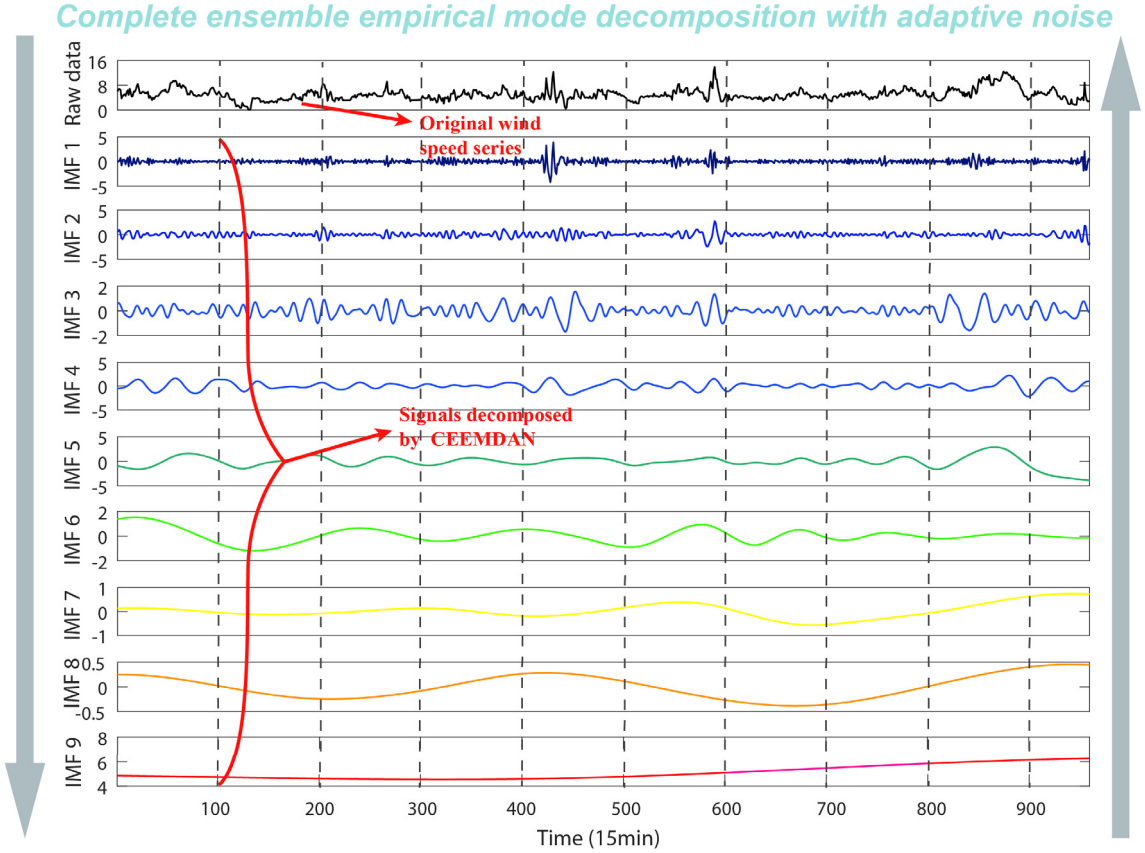

Fig. 1. The IMFs, from high frequency to low frequency, of the CEEMAD method.

where $e_i = (e_{i1}, e_{i2}, \ldots, e_{iN})$ .The experiment results show that the combination model can achieve desirable results when the weight vector is set from −2 to 2.

2.4. Optimization algorithm-CLSFPA

To enhance the prediction ability of the combined model, a modified optimization algorithm named the CLSFPA is proposed to determine the optimal weight coefficients of the combined model. In this section, the basic Flower Pollination Algorithm (FPA) and the FPA modified by Chaotic Local Search (CLS) are introduced followed by a test of CLSFPA.

2.4.1. FPA

The Flower Pollination Algorithm (FPA) is a new metaheuristic algorithm that imitates the flow pollination process of flowering plants. Two processes in flower pollination are implemented by cross-pollination or self-pollination. In cross-pollination, pollen is spread among different flowers by species that can fly long distances, such as bees, birds, bats and butterflies. These species are regarded as global pollinators, which obey the parameters of Levy flight. Self-pollination is the second pollination undertaken for the purpose of fertilization. This pollination behavior can be described by the following definitions.

Definition 1. (1) Cross-pollination can be regarded as a global search, where pollinators carry pollen via Levy flights. (2) The process of self-pollination can be recognized as a local search. (3) Flower constancy is equal to a reproduction ratio that is proportional to the similarity between two flowers. (4) Local pollination and global pollination are controlled by the value of the variable P in [0,1].

Definition 2. The algorithm is formulated as follows:

(1) The process of cross-pollination and flower constancy can be formulated as follows:

$$x_i^{iter+1} = x_i^{iter} + \gamma L(x_i^{iter} - x_i^{best}) \tag{4}$$

where x_i^{iter} represents pollen I at iteration $iter$, and x_i^{best} is the current best solution. Here, γ represents a scaling factor to control the step size and L represents the Levy flight. The Levy distribution is the strength of pollination and formulated as

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{S^{\lambda+1}} \quad (S \gg S_0 > 0) \tag{5}$$

where $\Gamma(\lambda)$ represents the standard gamma function, and this distribution is valid for large steps s > 0. λ is the distribution factor that ranges from 0.3 to 1.99.

(2) The process of self-pollination and flower constancy can formulated as follows:

$$x_i^{iter+1} = x_i^{iter} + \varepsilon(x_j^{iter} - x_k^{iter}) \tag{6}$$

where x_j^{iter} and x_k^{iter} are pollen from different flowers of the same plants species. ε is derived from a uniform distribution in [0,1].

(3) The switch probability p is presented to switch between local or global of pollination. According to [18], a value of P that equals 0.8 may work better for most applications.

2.4.2. CLSFPA

To retain the advantages of FPA, such as strong global exploitation and exploration capacities and the ability of overcome drawbacks, such as slow convergence speed and the propensity to become trapped in local optimum, the chaotic local search (CLS) is used to enrich the searching behavior and accelerate the local convergence speed of the FPA algorithm. When FPA finishes an iteration to find an optimal solution, CLS performs a local search for better solutions around the optimum.

A well-known logistic equation applied in CLS is defined as follows.

$$cx_i^{iter+1} = \mu cx_i^{iter}(1 - cx_i^{iter}) \tag{7}$$

where cx_i is the ith chaotic variable, $iter$ represents the iteration number, the above equation exhibits chaotic dynamics when $\mu = 4$, cx_i is range in (0,1) and $cx_0 \notin \{0.25, 0.5, 0.75\}$.

For further details regarding CLS, please refer to Liu [26]. The main procedure of the CLSFPA is shown in Fig. 2(c). The pseudo-code of the proposed CLSFPA algorithm is outlined as follows:

Algorithm: CLSFPA
Objective:

Minimize and maximize the objective function $f(x)$, $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$

Parameters:
iter-iteration number. *Maxiter*-the maximum number of iteration.
I-a population pop.
p-the switch probability

1. /*Initialize a population of I flowers $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ with random positions and initialize $iter$ = 0.*/
2. **WHILE** $iter < Maxiter$
3. **FOR** i = 1 to I
4. **IF** rand > p then
5. Draw the vector L which obeys the Levy distribution, L has d dimension
6. Do global pollination via $x_i^{iter+1} = x_i^{iter} + \gamma L(x_i^{iter} - x_i^{best})$
7. **ELSE**
8. Draw ε from a uniform distribution in [0,1]
9. Randomly chose x_j^{iter} and x_k^{iter} from the population
10. Do local pollination via $x_i^{iter+1} = x_i^{iter} + \varepsilon(x_j^{iter} - x_k^{iter})$
11. **END IF**
12. /*Evaluate the fitness of the new solutions x_i^{iter}.*/
13. /*If new solutions are better, update x_i^{iter} by x_i^{iter+1}.*/
14. **END FOR**
15. /*Jump out of local optimum by using chaotic local search.*/
16. Calculate $cx_i^{iter} = \frac{x_i^{iter} - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$
17. Compute the next iteration chaotic variable by $cx_i^{iter+1} = \mu cx_i^{iter}(1 - cx_i^{iter})$
18. Transform cx_i^{iter+1} for the next iteration $x_i^{iter+1} = x_i^{\min} + cx_i^{iter+1}(x_i^{\max} - x_i^{\min})$
19. /*Evaluate x_i^{iter}, replace x_i^{iter} by x_i^{iter+1} if the newly generation is better.*/
20. /*Find the current best solution $gbest$*/
21. $iter = iter + 1$
22. **END WHILE**

2.4.3. Benchmark functions for validation of the modified CLSFPA algorithm

As shown in Table 2, five typical benchmark functions, namely, ***Sphere***, **Ackley,** ***Rosenbrock***, ***Rastrigin*** and ***Schaffer*** are applied to validate the performance of CLSFPA and standard FPA algorithm. The test of FPA and CLSFPA on all benchmark functions was implemented in MATLAB R2015a on Windows 8 with 2.40 GHz Intel Core i7 5500U, 64 bit and 8 GB RAM. The experiment parameters of FPA and CLSFPA are set as Table 1.

Table 3 indicates the following:

(1) For the ***Sphere*** function, during the independent running of the test 50 times by the FPA and CLSFPA algorithm, the convergence speed is uneven, the maximum, minimum and average value of the iteration are 1567 and 1070, 1044 and 683, 1390.9 and 839.5, respectively. The convergence rates of the two algorithms are both 100%.
(2) For the ***Rosenbrock*** function, the convergence speed of the test offered by the FPA algorithm is uneven, the average value of iteration is 369.3, the maximum value of iteration is 434, and the convergence rate is only 81%. However, the average and maximum iteration value of CLSFPA are 141.1 and 221, respectively. The convergence rate is 100%.
(3) For the ***Rastrigin*** function, the maximum, minimum and average iteration value of the FPA and CLSFPA algorithm are 885 and 399, 699 and 197, 809 and 285.9, respectively. In addition, the convergence speed of the FPA and CLSFPA algorithm are 93% and 100%, respectively.
(4) For the ***Ackley*** function, the maximum, minimum and average iteration value of the FPA and CLSFPA algorithm are 657 and 301, 539 and 243, 597.7 and 275.8, respectively. In addition, the convergence speed of the FPA and CLSFPA algorithm are 100% and 100%, respectively.
(5) For the ***Schaffer*** function, the maximum, minimum and average iteration value of the FPA algorithm are 2194, 1560and 1892.5, respectively. The maximum, minimum and average iteration values of the CLSFPA algorithm are 928, 192 and 674.2, respectively. The convergence rate of the FPA and CLSFPA algorithm are 87% and 100%.

Remark. The validation results of four benchmark test functions demonstrate that the FPA and CLSFPA can successfully converge to four test functions. However, to achieve the same convergence accuracy 10^{-6}, the maximum, minimum and average iteration values of the CLSFPA algorithm are all much smaller than those of the standard FPA algorithm. Therefore, it is demonstrated that the convergence speed and performance of the CLSFPA algorithm are superior to the standard FPA algorithm.

3. CEEMDAN-CLSFPA combined model

In this section, we present a novel combined model that integrates the CEEMDAN decomposition technique, several neural network forecasting models, and CLSFPA optimization. The main processes of the proposed combined model are demonstrated in Fig. 2. The detailed procedures of the combined model are as follows:

Table 1
The experiment parameters of FPA and CLSFPA algorithms.

Experiment parameters	Algorithm	
	FPA	CLSFPA
Maximum generation	10,000	10,000
Population size	50	50
P	0.8	0.8
Convergence tolerance	10^{-6}	10^{-6}
Maximum generation of CLSFPA	–	20

3.1. Step 1: CEEMDAN process

To build an effective prediction model, the features of original wind speed datasets must be fully analyzed and considered. In our study, CEEMDAN is introduced to decompose the original wind speed series into a collection of IMFs with corresponding frequencies and the residue component.

3.2. Step 2: Combination forecasting on IMFs

The decomposition result shows that different signals have different characteristics, meaning that a simple individual forecasting model can no longer adapt to all properties of the IMFs. A combination method is proposed to address this issue, and it can be stated that if there are M types of forecasting models with properly selected weight coefficients for solving a certain problem, then several model results are added together.

Assume that P_{model} (Model = 'BPNN', 'RBFNN', 'ENN', 'GRNN', 'WNN') is the combination forecasting result of each IMF using these methods. Then, the output of the combined model with NNCT, which is optimized by CLSFPA, can be expressed as $Output^{IMFs}_{CLSFPA-NNCT} = w_1 \times P_{model1} + w_2 \times P_{model2} + w_3 \times P_{model3}$. Here w_i $(i = 1, 2, \ldots, N)$ is the value for the weight coefficient of the model N. $w_i \in [-2, 2]$ is given by NNCT. To optimize the weight coefficients w_i $(i = 1, 2, \ldots, N)$ for individual models, a modified optimization algorithm CLSFPA, described in Section 2.3, is employed. Before optimization, the target function of the optimization must be confirmed. In this paper, the target function is set by Eq. (11). The optimization process terminates when it reaches the preset minimum of the target function or the maximum of the iteration number. In addition, the dimension of the nest is 5, the search boundary is set to $[-2,2]$ and the number of iterations of the CLSFPA is 500.

3.3. Step 3: Assemble forecasting result

The combination forecasting results of each IMF is obtained through step 2. Then, combine the forecasting results of each IMF component to obtain the final result.

4. The performance metric

4.1. Model evaluation criteria

To evaluate the forecast capacity of the proposed models, four evaluation criteria are considered to assess performance, as shown in Table 4. Smaller values reflect better forecasting performance.

Here y_n and $\hat{y}_n$ indicate the actual and predicted values at time n, respectively. N is the sample size.

4.2. The benchmark model

The ARIMA model is widely used because it can characterize nonlinear data. In this paper, the ARIMA model is used as a benchmark model to perform a comparison with the proposed model. A general ARIMA model [ARIMA (p, d, q)] for the y time series is predefined as,

$$y_k = \sum_{t=1}^{p} f_k y_{t-k} + \sum_{n=1}^{q} \sigma_n e_{t-k} + \varepsilon_t \quad (8)$$

Table 2
Benchmark functions for validation of the proposed modified CLSFPA algorithm.

Function name	Domain	Optimum value	Test function
Sphere	[−5.12,5.12]	0	$\sum_{i=1}^{d} x_i^2$
Rosenbrock	[−2.084,2.084]	0	$\sum_{i=1}^{d-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$
Rastrigin	[−5.12,5.12]	0	$\sum_{i=1}^{d-1}[x_i^2 - 10(2\pi x_i) + 10]$
Ackley	[−32.768,32.768]	0	$f(x) = -20\exp\left[-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right] + 20 + e$
Schaffer	[−5.12,5.12]	0	$\left(sin^2\sqrt{\sum_{i=1}^{d} x_i^2} - 0.5\right) / \left[1 + 0.001\left(\sum_{i=1}^{d} x_i^2\right)\right]^2 + 0.5$

Table 3
Validation result of FPA and the proposed CLSFPA algorithm.

Benchmark function	Algorithm	Value of iteration			Convergence rate (%)
		Max	Min	Average	
Sphere	FPA	1567	1044	1390.9	100
	CLSFPA	1070	683	839.5	100
Rosenbrock	FPA	434	308	369.3	81
	CLSFPA	221	96	141.1	100
Rastrigin	FPA	885	699	809	93
	CLSFPA	399	197	285.9	100
Ackley	FPA	657	539	597.7	100
	CLSFPA	301	243	275.8	100
Schaffer	FPA	2194	1560	1892.5	87
	CLSFPA	928	192	674.2	100

where f_k denotes the kth autoregressive parameter, y_t denotes the observed value at time t, σ_n denotes the nth moving average parameter and ε_k is the error at time k.

In ARIMA (p,d,q), p presents the number of autoregressive parts, d presents the number of non-seasonal differences and q is the order of the moving average term.

4.3. Bias-variance framework

To estimate the availability of the wind-speed forecasting models, a Bias-variance framework [27] was employed to assess the accuracy and stability of the proposed combined model and single models. Let $x_t - \hat{x}_t$ be the difference between observed value x_t and predicted value $\hat{x}_t$, and the average difference over all points is

$$\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t) = \frac{1}{T}\sum_{t=1}^{T}x_t - \frac{1}{T}\sum_{t=1}^{T}\hat{x}_t \tag{9}$$

where T is all the forecasting data and t is the tth performance evaluation of the data. The expectation of the total number of forecasting values is $E(\hat{x}) = \frac{1}{T}\sum_{t=1}^{T}\hat{x}_t$, and the expectation of the actual value is $x = \frac{1}{T}\sum_{t=1}^{T}x_t$. The Bias-variance framework can be decomposed as follows:

$$\begin{aligned} E(\hat{x} - x)^2 &= E(\hat{x} - E(\hat{x}) + E(\hat{x}) - x)^2 \\ &= E(\hat{x} - E(\hat{x}))^2 + (E(\hat{x}) - x)^2 \\ &= Var(\hat{x}) + Bias^2(\hat{x}) \end{aligned} \tag{10}$$

where $Bias^2(\hat{x})$ indicates the prediction accuracy of the forecasting model and $Var(\hat{x})$ demonstrates the stability.

4.4. Diebolde-Mariano (DM) test

Hypothesis testing is a statistical inference approach that can determine if there is a real difference between two models by quantifying the level of confidence. In our study, the Diebold-Mariano test is applied to evaluate the efficiency of the combined model. The hypothesis test is defined as follows:

Let y_n indicates the actual value and let $\hat{y}_{n+h}^{(1)}$ and $\hat{y}_{n+h}^{(2)}$ indicates two competing series of $h \subseteq$ step forecasting. The prediction errors of the two models are

$$\varepsilon_{n+h}^{(1)} = y_{n+h} - \hat{y}_{n+h}^{(1)} \tag{11}$$

$$\varepsilon_{n+h}^{(2)} = y_{n+h} - \hat{y}_{n+h}^{(2)} \tag{12}$$

The precision of each prediction is measured by a proper loss function $L(\varepsilon_{n+h}^{(i)})\ i = 1, 2$. Two popular loss functions are as follows:

Square error loss:

$$L(\varepsilon_{n+h}^{(i)}) = (\varepsilon_{n+h}^{(i)})^2 \tag{13}$$

Absolute deviation error loss:

$$L(\varepsilon_{n+h}^{(i)}) = |\varepsilon_{n+h}^{(i)}| \tag{14}$$

The Diebold-Mariano test statistic can be defined as

$$DM = \frac{\frac{1}{T}\sum_{n=1}^{T}(L(\varepsilon_{n+h}^{(1)}) - L(\varepsilon_{n+h}^{(2)}))}{\sqrt{S^2/T}} \tag{15}$$

where S^2 is an estimator of the variance of $d = L(\varepsilon_{n+h}^{(1)}) - L(\varepsilon_{n+h}^{(2)})$.

To determine whether one model is more accurate than another, we test the equal accuracy hypothesis. The null hypothesis is:

$$H_0 : L(\varepsilon_{n+h}^{(1)}) = L(\varepsilon_{n+h}^{(2)}) \tag{16}$$

against the alternative

$$H_1 : L(\varepsilon_{n+h}^{(1)}) \neq L(\varepsilon_{n+h}^{(2)}) \tag{17}$$

The null hypothesis denotes that the two forecasting models have the same accuracy. The alternative hypothesis denotes that two models have different levels of accuracy. The DM statistics converge to a standard normal distribution $N(0,1)$, if $|DM| \leqslant 1.96$ under the null hypothesis. If the DM statistic falls outside the range of $[-1.96, 1.96]$, the null hypothesis is rejected.

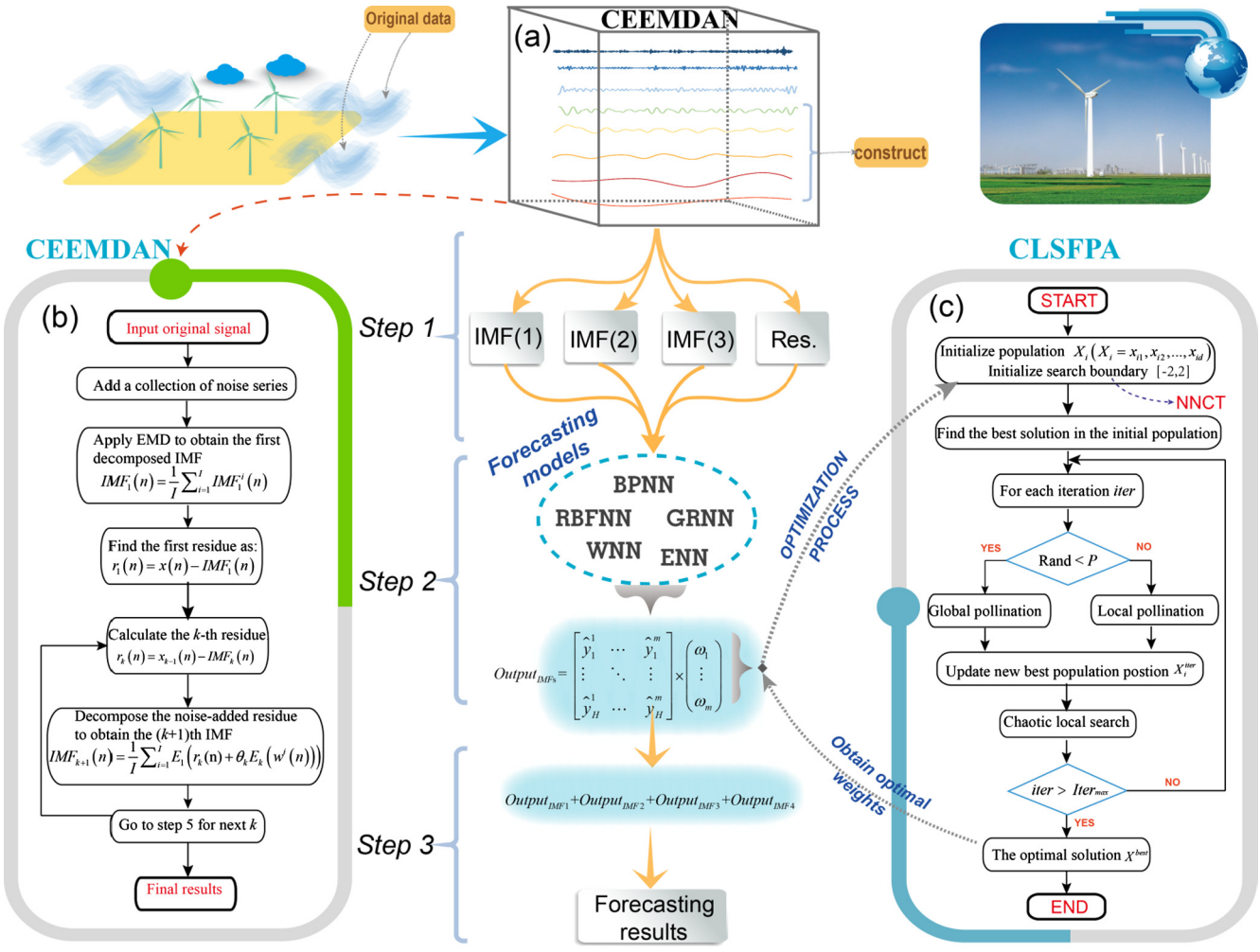

Fig. 2. The main procedure of the proposed combined model. Step 1: CEEMDAN process. Step 2: Combination forecasting on IMFs. Step 3: Assemble forecasting results.

Table 4
Four evaluation rules.

Metric	Equation	Definition
SSE	$SSE = \sum_{n=1}^{N}(y_n - \hat{y}_n)^2$	The square sum of the error
MAE	$MAE = \frac{1}{N}\sum_{n=1}^{N}\lvert y_n - \hat{y}_n\rvert$	The average absolute forecast error of n times forecast results
RMSE	$RMSE = \left(\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2\right)^{1/2}$	The root mean-square forecast error
MAPE	$MAPE = \frac{1}{N}\sum_{n=1}^{N}\left\lvert\frac{y_n - \hat{y}_n}{y_n}\right\rvert \times 100\%$	The average of absolute error

5. Results and analysis

5.1. Data description

In this paper, forecasting performance is assessed based on wind speed data from Shandong Province, which had abundant wind energy resources with an installed capacity of 651,000 kW in 2015. A significant number of large-scale wind farms, which are located around the coastal areas in Shandong, have recently been constructed, including Laizhou, Penglai, Yantai and Weihai, as shown in Fig. 3. In our study, the original wind speed data were collected by wind turbines with a 70 m hub height from wind farms in Laizhou, Penglai, Yantai and Weihai; these data were employed to test if the proposed models can be applied on different occasions. The wind speed data were sampled at an interval of 15 min from June 20, 2013 to July 15, 2013. During this period, the mean temperature and humidity of four wind farms were 28.43 °C and 74.32%; 27.82 °C and 83.38%; 28.02 °C and 82.08%; 27.91 °C and 86.30%, respectively. For each case, the wind speed series were divided into a training set (70%) and a test set (30%). Thus data for 7 days, providing a total of 672 samples of 15 min of data, were selected to train the forecasting models and identify the weights of single models; the next 288 samples for 3 days were employed to test the performance of the developed combined model.

5.2. Data decomposition by CEEMDAN

In the proposed CEEMDAN-CLSFPA combined model, the CEEMDAN is applied to decompose the raw wind speed series into several independent IMFs, as shown in Fig. 1. However, too many IMFs may result in more complex simulation and additional computational cost [28]. On the one hand, too many IMFs may accumulate the estimation error of each IMF in the process of ensemble prediction. On the other hand, too many IMFs may lead to excessive amounts of time being consumed in the individual forecasting step. To avoid these problems, the IMFs are reconstructed into three IMFs and one residual. Fig. 2(a) shows the schematic of the reconstructed results of the IMFs that decomposed via CEEMDAN.

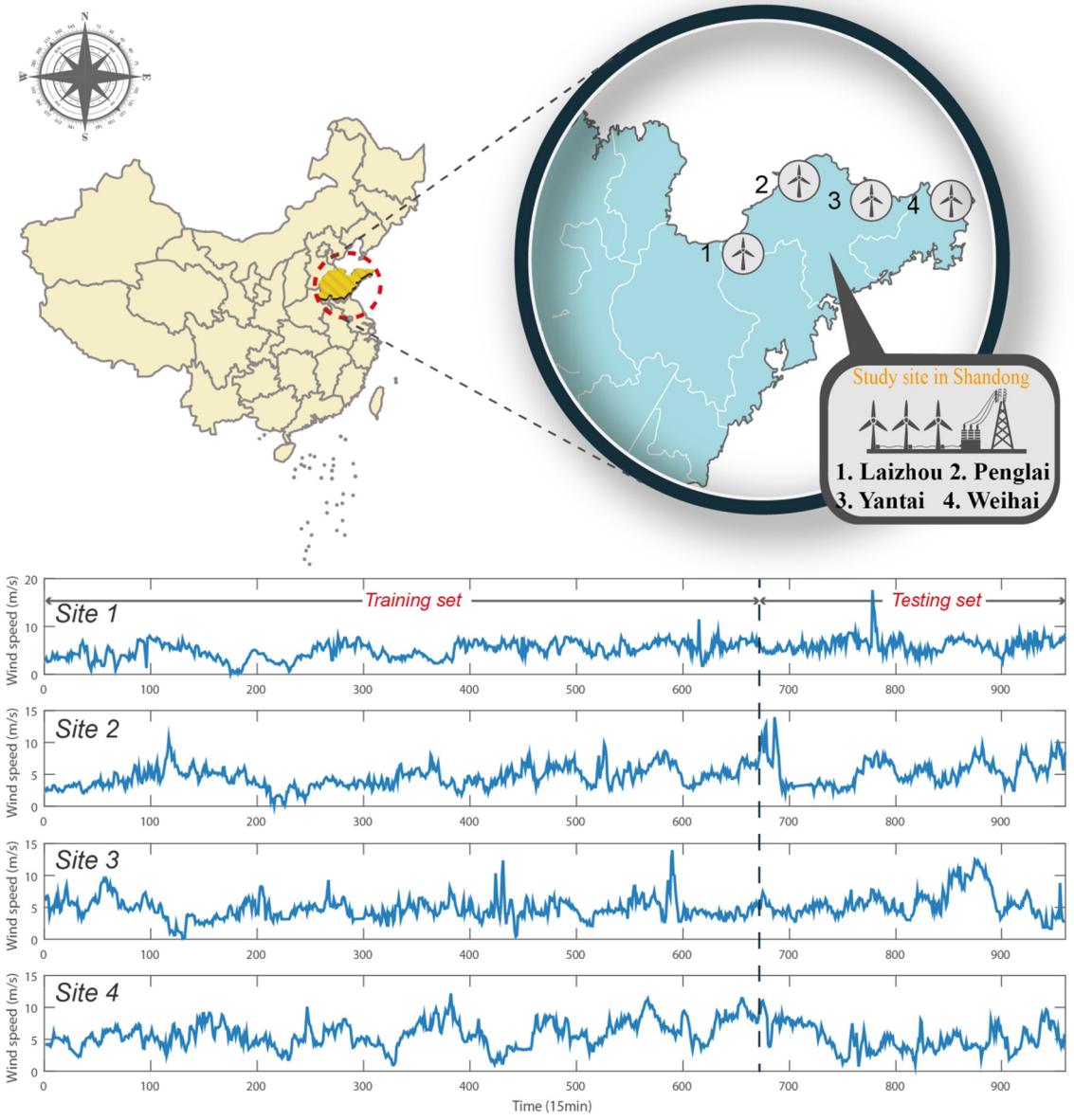

Fig. 3. Specific location of the study sites and original wind speed series from four sites.

5.3. The process of combination forecast on IMFs

The BPNN, RBFNN, ENN, RNN and WNN forecasting models based on the CEEMDAN-CLSFPA combined model are used to forecast each IMF and a residue, which is reconstructed in Section 5.2. The number of input nodes, hidden nodes and output nodes of the five neural networks are set to 4, 9, and 1, respectively. In this paper, we use the rolling one-step forecast operation method based on the wind speed data of four sites to test the forecasting models. The experiment parameters of five neural networks are shown in Table 5.

The forecasting results of single models and the proposed combined model are presented in Table 6. The SSE is used as a model-evaluation index to evaluate model performance. It can be obviously observed that each single model exhibits the best forecast performance at a specific IMF. For example, the forecasting results in Laizhou reveal that the BPNN presents the best results for Residue. The RBFNN, however, exhibits the lowest SSE values among all single models for IMF3, while the lowest value for IMF1 and IMF2 is achieved by ENN. The experimental results for Penglai are shown as follows. Among all of the single models, when the ENN was applied, the value of SSE was lower than those of the other methods at IMF1 and IMF2. At IMF3, the WNN provides the optimal results. At Residue, the result from the RBFNN is the best.

Table 5
Experiment parameters of ANNs.

Model	Experimental parameters	Default value
BPNN	The learning velocity	0.01
	The maximum number of trainings	1000
	Training requirements precision	0.00004
RBF	Spread of radial basis functions	0.5
	Training requirements precision	0.00004
ELMAN	The maximum number of trainings	1000
GRNN	Spread of radial basis functions	0.15
WNN	Learning rate	0.1
	Training requirements precision	0.00004

The results in Yantai show that the ENN performs the best at IMF1 while the GRNN performs better than the others at IMF3. Meanwhile, the RBFNN exhibits optimal results at other signals. The forecasting results for Weihai show that the most accurate result belongs to the BPNN at IMF2. When the ENN is used, the results are more accurate from IMF1 to Residue. The results show that the WNN only performs desirably at IMF3.

From the above analysis, we find that no single model provides the best results for all signals, but each model has its strengths at specific IMFs. Therefore, the best-suited model is chosen based on different conditions, which indicates that a combination model can take advantage of each single model to overcome the one-sidedness of single models. In our study, a combination model based on CLSFPA and NNCT is proposed to search for the optimal combination weights of forecasting models. According to the NNCT, the searching boundary is set as $[-2,2]$ and the SSE criteria are taken as the fitness function of the CLSFPA. The optimal weights and the final results of the combination model are shown in Table 6, and the error comparisons are shown in Fig. 4. A comparison of the IMF forecasting results from each single model indicates that the perfect forecasting result is achieved by the proposed combined model. With respect to BPNN, RBFNN, ENN, GRNN and

Table 6
IMFs forecasting results of the individual model and combined models for four sites.

Site	Models	IMF1		IMF2		IMF3		Residue	
		Weights	SSE	Weights	SSE	Weights	SSE	Weights	SSE
Laizhou	BPNN	0.1751	178.8471	0.0854	46.3403	0.5793	19.1916	1.2085	1.1267
	RBFNN	0.3967	96.6678	0.3384	23.3976	0.9868	7.9042	0.1017	27.0066
	ENN	0.1669	93.4068	0.5325	11.4812	−0.0579	15.8382	−0.0214	19.7202
	GRNN	0.0706	139.7958	−0.186	43.8606	−0.3605	39.3694	−0.0009	52.3923
	WNN	0.1987	105.6706	0.3065	15.7071	−0.1402	44.9135	−0.284	5.3918
	Combined model	–	86.3523	–	8.4982	–	5.7073	–	0.9426
Penglai	BPNN	0.1848	49.8701	0.1759	14.1897	0.0297	2.5611	0.3596	26.6982
	RBFNN	0.3418	38.4458	−0.1403	29.3421	0.0246	3.1967	0.8679	3.4151
	ENN	0.8985	33.9866	0.9857	6.0717	−0.144	2.8356	0.8679	7.4806
	GRNN	−0.8069	47.8585	0.191	29.7824	0.1986	5.568	−0.2766	61.2491
	WNN	−0.1611	50.9649	−0.064	13.2673	0.9644	0.2041	−0.0438	14.0416
	Combined model	–	29.6803	–	4.6231	–	0.1109	–	1.241
Yantai	BPNN	0.0253	98.8704	0.429	14.5991	−0.0118	26.6801	−0.3433	125.9677
	RBFNN	−0.518	55.9716	0.6918	9.0493	0.5188	4.3425	1.0481	4.6334
	ENN	1.0859	42.2601	−0.7158	10.9666	−0.1066	14.3987	−0.0089	31.7397
	GRNN	0.2784	66.1009	0.3594	12.8182	0.5439	3.2842	0.0284	141.0416
	WNN	0.0365	49.1773	0.2384	18.766	0.1389	5.1075	0.2784	35.0571
	Combined model	–	39.0327	–	6.564	–	2.138	–	0.5998
Weihai	BPNN	0.2081	53.3913	0.5154	10.9486	0.2583	15.8428	0.47273	1.1038
	RBFNN	0.3979	39.5912	0.5342	13.2401	0.6476	5.5549	−0.0579	6.1667
	ENN	0.0841	37.2049	−0.5247	32.2164	−0.4494	14.0037	0.256	1.084
	GRNN	0.0075	42.2154	0.6343	25.9355	0.06038	7.6729	0.0198	3.8487
	WNN	0.3669	37.4896	0.071	22.8589	0.5951	4.5001	0.3001	5.5865
	Combined model	–	33.0675	–	5.7732	–	2.5923	–	0.3575

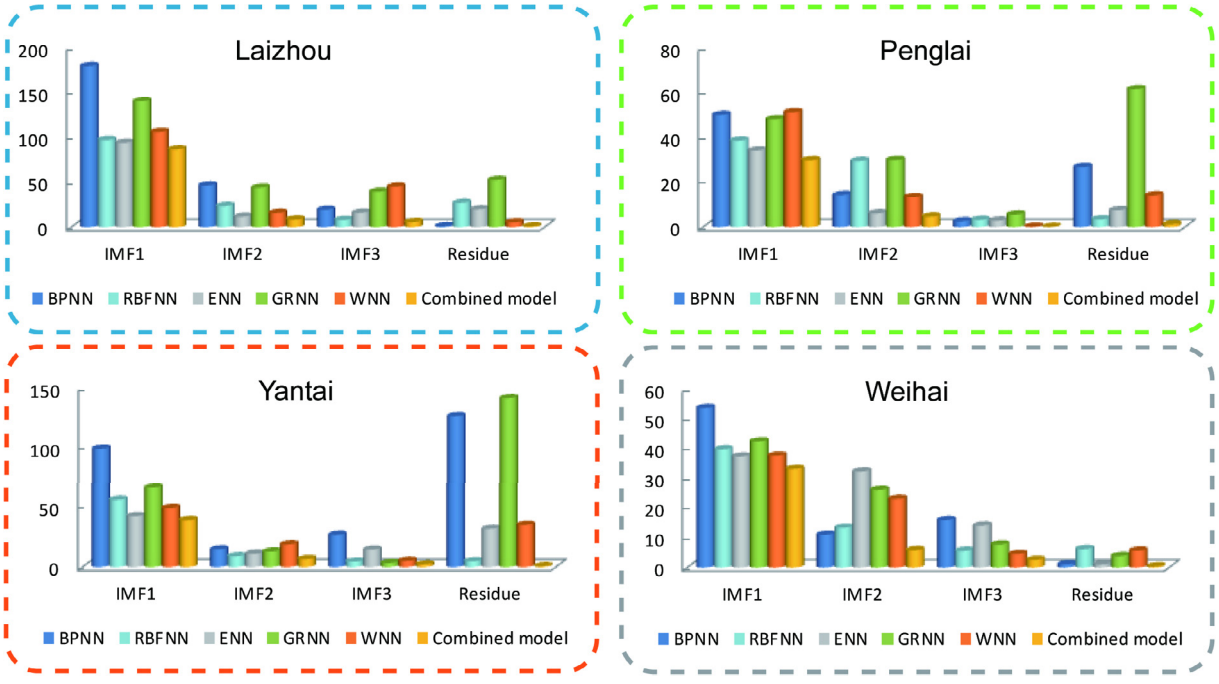

Fig. 4. Error comparison among combined model and single models by forecasting each IMFs.

WNN, the average SSE value of four sites on IMF1 was reduced by 50.62%, 18.44%, 9.05%, 36.44% and 22.68% compared to that of the combined model. For IMF2, the average SSE value of four sites was reduced by 70.42%, 66.07%, 58.08%, 77.35% and 63.94%, respectively. For IMF3, the average SSE value of four sites was reduced by 83.59%, 49.76%, 77.59%, 81.13% and 80.72%, respectively. For residue, the average SSE value of four sites was reduced by 97.97%, 92.38%, 94.77%, 98.79% and 94.77%, respectively. Additionally, the remarkable decrease of SSE with respect to the combined model is 99.52%. Consequently, the combined model has significant improvement at each IMF component compared with BPNN, RBFNN, ENN, GRNN and WNN. Moreover, the prediction results from IMF1 to residue shows that the higher the frequency, the more difficult the forecast.

In the process above, the three independent IMFs and one residual decomposed by CEEMDAN are predicted by the combined model. In step 3, the final results are obtained by assembling the forecasting results of each IMF.

5.4. The forecasting results of the combined model and comparisons

To test the proposed combination model, four evaluation criteria are applied to make a comparison with other single models and a benchmark model; furthermore, a bias-variance framework and DM test are applied for a more comprehensive evaluation of the proposed model.

5.4.1. Experiment 1: The comparison results of the combined model and CEEMDAN-based models

In this experiment, the five CEEMDAN-based forecasting models are used for comparison with the proposed combined model. The CEEMDAN-based forecasting models include CEEMDAN-BPNN, CEEMDAN-RBFNN, CEEMDAN-ENN, CEEMDAN-GRNN and CEEMDAN-WNN; each use a single forecasting model to forecast all of the signals decomposed by CEEMDAN. The forecasting results of a combined model and the comparison with other models are shown in Table 7.

Table 7
The results of the combined model and other CEEMDAN-based models at four sites.

Site	Evaluation criteria	Models					
		Combined model	CEEMDAN-BPNN	CEEMDAN-RBFNN	CEEMDAN-ENN	CEEMDAN-GRNN	CEEMDAN-WNN
Site 1	SSE(m/s)	85.0860	189.6071	149.1640	111.9167	366.3764	152.4022
	MAE(m/s)	0.3758	0.4832	0.4665	0.4428	0.6152	0.4249
	RMSE(m/s)	0.5435	0.8113	0.7197	0.6234	1.1279	0.7274
	MAPE(%)	6.2124	7.6704	7.5637	7.2928	9.6484	6.6991
Site 2	SSE(m/s)	56.6668	77.0082	61.7844	56.3158	113.4218	58.0481
	MAE(m/s)	0.2254	0.2839	0.3178	0.2428	0.4557	0.3099
	RMSE(m/s)	0.4436	0.5171	0.4632	0.4422	0.6276	0.4490
	MAPE(%)	4.1119	5.2548	6.3142	4.4058	8.9693	6.1416
Site 3	SSE(m/s)	49.1160	230.2971	81.3609	97.1212	227.5302	115.2462
	MAE(m/s)	0.2636	0.5778	0.3677	0.42067	0.5410	0.3770
	RMSE(m/s)	0.4130	0.8942	0.5315	0.58071	0.8888	0.6326
	MAPE(%)	5.3155	10.758	7.2045	8.1100	8.8945	7.1468
Site 4	SSE(m/s)	34.6359	67.6939	58.8575	80.0470	86.6140	60.8717
	MAE(m/s)	0.2509	0.3376	0.3367	0.37163	0.3954	0.3070
	RMSE(m/s)	0.3468	0.4848	0.4521	0.5272	0.5484	0.4597
	MAPE(%)	5.9528	8.2826	8.0790	9.2252	9.7642	7.5459
Average	SSE(m/s)	56.3762	141.1516	87.7917	86.3502	198.4856	96.6421
	MAE(m/s)	0.2789	0.4206	0.3722	0.3695	0.5018	0.3547
	RMSE(m/s)	0.4367	0.6769	0.5416	0.5434	0.7982	0.5672
	MAPE(%)	5.3982	7.9915	7.2904	7.2585	9.3191	6.8834

Table 8
The results of the combined model and other single models at four sites.

Site	Evaluation criteria	Models						
		Combined model	BPNN	RBFNN	ENN	GRNN	WNN	ARIMA
Site 1	SSE (m/s)	**85.0860**	624.0785	643.1805	496.2678	814.9715	802.0102	497.5054
	MAE (m/s)	**0.3758**	0.9968	0.9025	0.8696	1.0957	1.0049	0.8474
	RMSE (m/s)	**0.5435**	1.4721	1.4944	1.3127	1.6822	1.6688	1.3143
	MAPE (%)	**6.2124**	16.0058	14.2396	14.0554	17.7470	16.4271	14.1165
Site 2	SSE (m/s)	**56.6668**	338.5491	173.3188	234.2436	209.2737	173.7457	158.2251
	MAE (m/s)	**0.2254**	0.8257	0.5803	0.6595	0.6294	0.6060	0.5629
	RMSE (m/s)	**0.4436**	1.0842	0.7758	0.9019	0.8524	0.7767	0.7412
	MAPE (%)	**4.1119**	14.8171	11.4657	12.5574	12.1541	12.3742	11.3388
Site 3	SSE (m/s)	**49.1160**	484.7171	329.7178	644.5422	571.2236	996.9458	235.3224
	MAE (m/s)	**0.2636**	0.8600	0.7561	0.9752	1.0236	1.2135	0.6586
	RMSE (m/s)	**0.4130**	1.2973	1.0700	1.4960	1.4083	1.8605	0.9039
	MAPE (%)	**5.3155**	14.9398	13.1845	15.7243	18.6124	21.9849	12.9417
Site 4	SSE (m/s)	**34.6359**	291.0106	251.5576	235.5946	295.6705	293.7579	239.0791
	MAE (m/s)	**0.2509**	0.7207	0.6756	0.6623	0.7485	0.7589	0.6681
	RMSE (m/s)	**0.3468**	1.0052	0.9346	0.9045	1.0132	1.0099	0.9111
	MAPE (%)	**5.9528**	18.1685	16.7691	16.2083	19.4098	20.0995	16.1841
Average	SSE (m/s)	**56.3762**	434.5888	349.4437	402.6621	472.7848	566.6149	282.5330
	MAE (m/s)	**0.2789**	0.8508	0.7286	0.7917	0.8743	0.8958	0.6843
	RMSE (m/s)	**0.4367**	1.2147	1.0687	1.1538	1.2390	1.3290	0.9676
	MAPE (%)	**5.3982**	15.9828	13.9147	14.6364	16.9808	17.7214	13.6453

The ensemble forecasting results of each model show that the proposed combined model improves forecasting accuracy significantly based on a comparison with CEEMDAN-BPNN, CEEMDAN-RBFNN, CEEMDAN-ENN, CEEMDAN-GRNN and CEEMDAN-WNN according to four evaluation criteria. Particularly, the average MAPE of the combined model was reduced by 32.4509%, 25.9549%, 25.6294%, 42.0743% and 21.5767%, respectively.

5.4.2. Experiment 2: The comparison results of the combined model and individual models

In this experiment, the five single models and a benchmark model are used to perform a comparison with the combined model. The individual models include BPNN, RBFNN, ENN, GRNN and WNN. The ARIMA model is employed as the benchmark model. The comparison forecasting results for the combined model and other models are shown in Table 8; the lowest SSE, MAE, RMSE and MAPE values among all the forecasting models at the four sites are highlighted in bold. Clearly, the proposed combined model provides the highest accuracy for all wind farms; the average MAPE of the four sites improved the percentages of BPNN, RBFNN, ENN, GRNN, WNN and ARIMA, which are 66.2253%, 61.2055%, 63.1182%, 68.2103%, 69.5388% and 60.4394%, respectively (see Fig. 5).

5.4.3. Experiment 3: Tested with the bias-variance framework and Diebold-Mariano test

This part focuses on the effectiveness of the proposed combined model using the bias-variance framework and the Diebold-Mariano test presented in Section 4. The Diebold-Mariano test is a comparison test for predictive accuracy. The bias-variance framework is employed to evaluate the models' accuracy and stability; bias is regarded as the difference between the predicted value and the observed value, and variance is regarded as the variability of the forecasting model.

The values of the Diebold-Mariano statistic based on the absolute deviation error loss function are shown in Table 9. The results show that the Diebold-Mariano value of five single models and a benchmark model are all higher than the upper limits at the 0.01

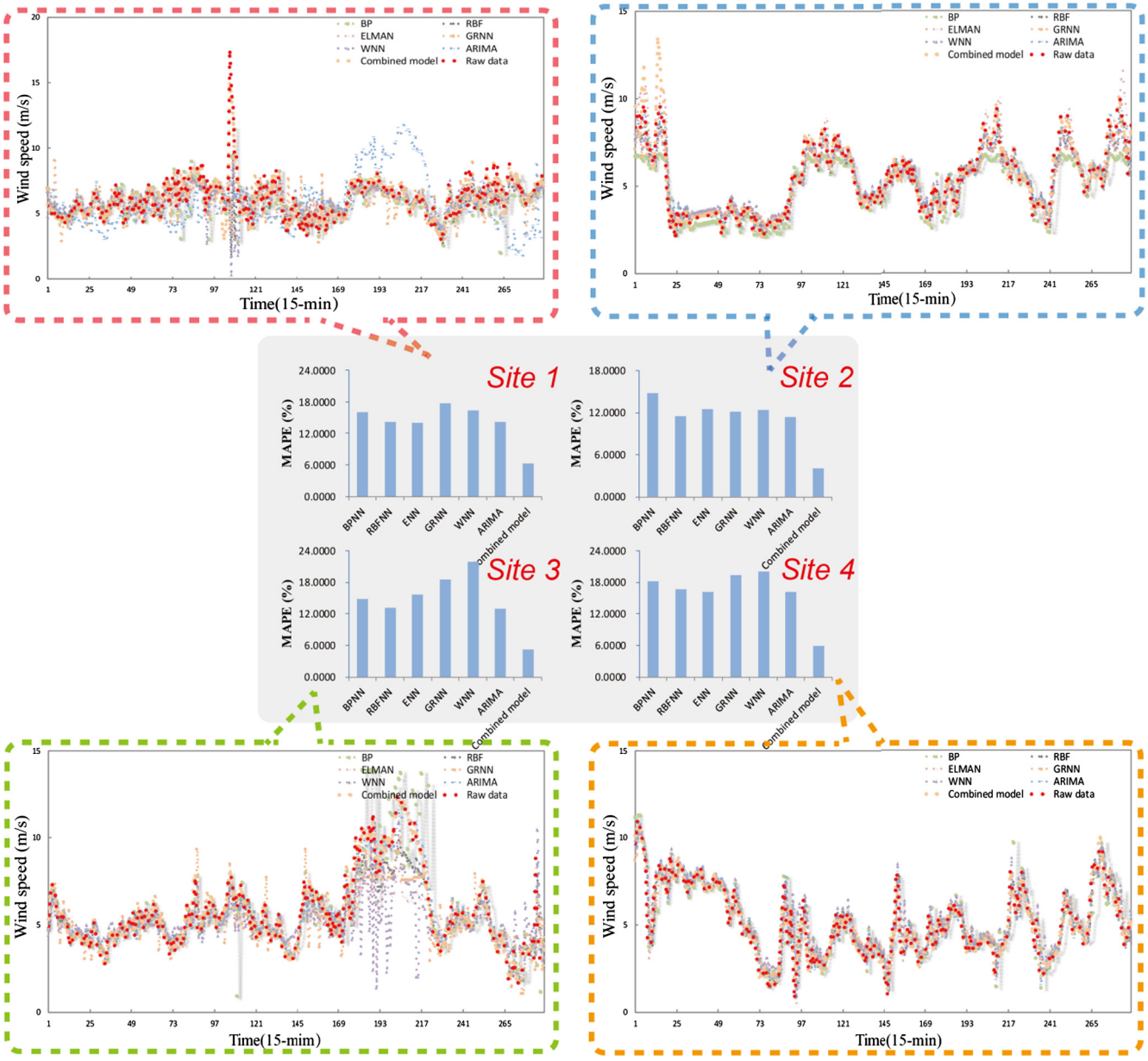

Fig. 5. Forecasting results of the combined model and individual models.

Table 9
Bias-variance and Diebold-Mariano test of seven models for the mean value at four sites.

Model	D-M statistic	Bias-variance	
		Bias	Var
Combined model	–	0.017535	0.193725
BPNN	6.461000*	0.187900	1.178275
RBFNN	5.451025*	0.146975	1.347175
ENN	5.47625*	0.127525	1.583225
GRNN	6.260875*	0.174075	1.869575
WNN	5.372000*	0.057650	2.062275
ARIMA	6.684875*	0.095250	1.178275

** 5% significance level.
*** 10% significance level.
* 1% significance level.

significance level, which imply the proposed combined model is significantly better than each single model and benchmark model. The absolute values of the biases of the combined model are smaller than those of the other individual models, which indicate that the proposed combined model can provide more accurate results. Similarly, the results of the variances reveal that the combined model is more stable. Overall, the developed combined model has higher accuracy and stability and is superior to the other single models with respect to wind speed forecasting.

As shown in Fig. 5, the forecasting curves and error bars of all the models show that the proposed combined model has the lowest prediction errors and fits better than other the models at four sites. This phenomenon shows that the proposed combined model can take advantage of each single model. Meanwhile, the weight coefficients of the combined model can be scientifically calculated while maintaining high precision and stability using the CLSFPA optimization algorithm. Moreover, incorporating the CEEMDAN algorithm into the developed combined model is an efficient decomposition method that can decompose the wind speed series into more simple shapes for forecasting. Although the proposed developed combined model is not simple, it is able to provide better forecasting accuracy and is quite an efficient tool in actual application.

6. Conclusion

Reliable and precise wind speed forecasting is vital for wind power generation systems. Because wind speed series are highly noisy and unstable, the primary wind speed datasets are very hard to predict precisely. Thus, the CEEMDAN, an advanced and effective decomposition technique, is applied to decompose raw wind speed series into several IMF signals for easy analysis and forecasting. The forecasting results show that the individual forecasting models applied for forecasting wind speed have limited capacity and are not suitable for all IMFs. The appropriate combination model can give full play to the strengths of each of the individual models and make each individual model perform in its specific manner. To this end, a developed combined forecasting model based on BPNN, RBFNN, ENN, GRNN and WNN is proposed for forecasting each decomposition signals with different characteristics. Furthermore, an improved Flower Pollination Algorithm, called CLSFPA, is proposed to obtain the optimal weights of the combined model. The main contributions of this model are summarized as follows: (1) A novel CEEMDAN-CLSFPA combined model is proposed for short-term wind speed forecasting. (2) A modified optimization algorithm-CLSFPA is proposed to optimize the weights of the combined model. (3) The CEEMDAN technique is adopted to decompose the original wind speed series. (4) The no negative constraint theory is employed in the combined model.

The comparison results obtained in this study show that the proposed combined model performs better than the five CEEMDAN-based models, five individual models and a benchmark model. Meanwhile, based on the bias-variance framework and Diebold-Mariano test, the results indicate that the combined model has higher stability and prediction precision compared with other single models. Thus, the proposed combined model provides an effective and powerful mining and forecasting tool for wind power plants and is suitable for high-precision wind speed prediction for the safety of wind power conversion.

Acknowledgements

This work was supported by the National Natural Science Foundation project (41225018).

Appendix A

A.1. BP neural network

The back-propagation (BP) neural network, which is a type of multilayer feed-forward neural network, has widely applications. It is based on a gradient descent method that minimizes the sum of the squared errors between the output values and the actual values. The process of the back-propagation algorithm is divided into two phases, updating and learning.

$$k_{ij}(t) = w_{ij}(t-1) - \Delta w_{ij}(t) \tag{A1}$$

$$\Delta w_{ij}(t) = \eta \partial E / \partial w_{ij}(t-1) + \alpha \cdot \Delta w_{ij}(t-1) \tag{A2}$$

where w_{ij} denotes the weights between nodes i and j. η is the learning speed, α is the impulse parameter, t is the current iterative steps, and E is the error super curve face.

A.2. RBF neural network

The radial basis function (RBF) neural network is a type of feed-forward network developed by Broomhead and Lowe [29]. It has three layers of architecture, where there are no weights between the input hidden layers, and each hidden unit implements a radial-activated function.

The Gaussian activation function is used in each neuron at the hidden layer, which can be formulated as

$$\phi_j(x) = \exp\left(-\frac{\sum_{i=1}^{M}(x_i - \mu_j)^2}{2\sigma_j^2}\right) \tag{A3}$$

where x_i is the ith input sample, μ_i is the mean value of the jth hidden unit presenting the center vector, σ_j is the covariance of the jth hidden unit denoting the width of the RBF kernel function, and M is the number of training samples.

The network output layer is linear so that the kth output is an affine function that can be expressed as

$$y_k = \sum_{j=1}^{L} \phi_j w_{jk} + \theta_k \tag{A4}$$

where w_{jk} is the weight between nodes k and j, θ_k is the biased weight of the kth output, and L is the number of hidden nodes.

A.3. Elman neural network

The Elman neural network (ENN) is a well-known recurrent network. The ENN model consists of four layers, the input layer, the hidden layer, the context layer and the output layer. There are a lot of parallel connections, which are the context units, between the input and hidden layers, as well as between the hid-

den and output layers. Dynamic system modeling is sensitive to input and is effectively used for the self-connections of the context node [30]. The numbers of hidden units is equal to the context units, which is the constraint of this network. The state equations can be expressed as follows:

$$x^{t+1} = f(w_{in}p^{t+1} + wx^t) \tag{A5}$$

$$y^{t+1} = f_{out}(w_{out}p^{t+1} + r) \tag{A6}$$

where w_{in}, w and w_{out} represent the weights of the input, internal state and output, respectively, x^t, p^t and y^t are the input, the internal state and the output at time t, r is the activation of internal neurons, and f_{out} is the output activation.

A.4. GRNN

The general regression neural network (GRNN) is a powerful computational technique used to solve nonlinear approximation problems based on non-linear regression theory. The advantages of GRNNs include good feasibility, a simple structure and fast convergence rate. It is mainly composed of four layers, an input layer, a pattern layer, a summation layer, and an output layer.

Definition 1. There is a nonlinear transformation in the pattern layer, which transforms the input space into the pattern space. The relationship between the input neuron and the response from the pattern layer can be remembered in each neuron in this layer.

Definition 2. The summation layer has two summations, i.e., simple summation *Ss* and weighted summation *Sw*. Then, the summation layer is transmitted to the output layer.

A.5. Wavelet neural network (WNN)

Wavelet neural network (WNN) is a type of artificial neural network model constructed based on wavelet transform; it includes the time–frequency localization of wavelet analysis and the self-learning ability of neural network. In other words, the traditional sigmoid function is replaced by a mother Wavelet function, which reflectes more accurately in different scales of time series and distributions in the time domain.

A group of mother wavelets can be approximately expressed by the function $y_k(x)$

$$y_k(x) = \sum_{j=1}^{J} w_{kj}\varphi\left(\frac{X_j - b_j}{a_j}\right), \quad (k = 1, \ldots, K) \tag{A7}$$

where $X_J = \sum_{i=1}^{I} w_{ji}x_i$, $(j = 1, \ldots, J)$; a_j, b_j $(j = 1, \ldots, J)$ are the dilation and translation parameters of the wavelet function at hidden node j, respectively, and $\varphi(\bullet)$ is the wavelet base function. In this paper, the basic mother wavelet function takes its real part, namely

$$\varphi(t) = \cos(1.75t)\exp(-t^2/2) \tag{A8}$$

References

[1] Meng A, Ge J, Yin H, Chen S. Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. Energy Convers Manag 2016;114:75–88. http://dx.doi.org/10.1016/j.enconman.2016.02.013.

[2] Liu H, Tian H, Li Y. Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms. Energy Convers Manag 2015;100:16–22. http://dx.doi.org/10.1016/j.enconman.2015.04.057.

[3] Okumus I, Dinler A. Current status of wind energy forecasting and a hybrid method for hourly predictions. Energy Convers Manag 2016;123:362–71. http://dx.doi.org/10.1016/j.enconman.2016.06.053.

[4] Giebel G, Brownsword R, Kariniotakis G, Denhard M, Draxl C. The state-of-the-art in short-term prediction of wind power a literature overview. Tech Report, ANEMOS.plus 2011:1–109.

[5] Giebel G, Landberg L. State-of-the-art on methods and software tools for short-term prediction of wind energy production. Energy 2010:16–9.

[6] Aghajani A, Kazemzadeh R, Ebrahimi A. A novel hybrid approach for predicting wind farm power production based on wavelet transform, hybrid neural networks and imperialist competitive algorithm. Energy Convers Manag 2016;121:232–40. http://dx.doi.org/10.1016/j.enconman.2016.05.024.

[7] Torres JL, Grcia A, de Blas M, de Francisco A. Forecast of hourly averages wind speed with arma models in Navarre. Sol Energy 2005;79:65–77.

[8] Kavasseri RG, Seetharaman K. Day-ahead wind speed forecasting using f-ARIMA models. Renew Energy 2009;34:1388–93. http://dx.doi.org/10.1016/j.renene.2008.09.006.

[9] Wang J, Qin S, Zhou Q, Jiang H. Medium-term wind speeds forecasting utilizing hybrid models for three different sites in Xinjiang, China. Renew Energy 2015;76:91–101. http://dx.doi.org/10.1016/j.renene.2014.11.011.

[10] De Giorgi MG, Ficarella A, Russo MG. Short-term wind forecasting using artificial neural networks (ANNs). Energy Sustain Ii 2009;121:197–208. http://dx.doi.org/10.2495/esu090181.

[11] Guo Z, Wu J, Lu H, Wang J. A case study on a hybrid wind speed forecasting method using BP neural network. Knowl-Based Syst 2011;24:1048–56. http://dx.doi.org/10.1016/j.knosys.2011.04.019.

[12] Wang J, Zhang W, Wang J, Han T, Kong L. A novel hybrid approach for wind speed prediction. Inf Sci (NY) 2014;273:304–18. http://dx.doi.org/10.1016/j.ins.2014.02.159.

[13] Zhou J, Shi J, Li G. Fine tuning support vector machines for short-term wind speed forecasting. Energy Convers Manag 2011;52:1990–8. http://dx.doi.org/10.1016/j.enconman.2010.11.007.

[14] Wang J-J, Zhang W-Y, Liu X, Wang C-Y. Modifying wind speed data observed from manual observation system to automatic observation system using wavelet neural network. Phys Proc 2012;25:1980–7. http://dx.doi.org/10.1016/j.phpro.2012.03.338.

[15] Moghram I, Rahman S. Analysis and evaluation of five short-term load forecasting techniques. IEEE Trans Power Syst 1989;4:1484–91. http://dx.doi.org/10.1109/59.41700.

[16] Xiao L, Wang J, Dong Y, Wu J. Combined forecasting models for wind energy forecasting: a case study in China. Renew Sustain Energy Rev 2015;44:271–88. http://dx.doi.org/10.1016/j.rser.2014.12.012.

[17] Yang Y, Chen Y, Wang Y, Li C, Li L. Modelling a combined method based on ANFIS and neural network improved by DE algorithm: a case study for short-term electricity demand forecasting. Appl Soft Comput 2016. http://dx.doi.org/10.1016/j.asoc.2016.07.053.

[18] Yang XS. Flower pollination algorithm for global optimization. In: Lecture notes on computer sciences. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7445 LNCS; 2012. p. 240–9. doi:http://dx.doi.org/10.1007/978-3-642-32894-7_27.

[19] Dubey HM, Pandit M, Panigrahi BK. Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch. Renew Energy 2015;83:188–202. http://dx.doi.org/10.1016/j.renene.2015.04.034.

[20] Chakraborty D, Saha S, Maity S. Training feedforward neural networks using hybrid flower pollination-gravitational search algorithm. In: International conference on future trends on computer analysis and knowledge management (ABLAZE); 2015. p. 261–6. doi:http://dx.doi.org/10.1109/ABLAZE.2015.7155008.

[21] Tahani M, Babayan N, Pouyaei A. Optimization of PV/Wind/Battery stand-alone system, using hybrid FPA/SA algorithm and CFD simulation, case study: Tehran. Energy Convers Manag 2015;106:644–59. http://dx.doi.org/10.1016/j.enconman.2015.10.011.

[22] Huang N, Shen Z, Long S, Wu M, Shih H, Zheng Q, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc A Math Phys Eng Sci 1998;454(995):903. http://dx.doi.org/10.1098/rspa.1998.0193.

[23] Tong W, Zhang M, Yu Q, Zhang H. Comparing the applications of EMD and EEMD on time–frequency analysis of seismic signal. J Appl Geophys 2012;83:29–34.

[24] Wu Z, Huang NE. Ensemble empirical mode decomposition: a noise-assisted data analysis method. Adv Adapt Data Anal 2009;1:6281–4. doi:http://dx.doi.org/10.1142/S1793536909000047.

[25] Torres ME, Colominas MA, Schlotthauer G, Flandrin P. A complete ensemble empirical mode decomposition with adaptive noise. In: Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing; 2011. p. 4144–7. doi:http://dx.doi.org/10.1109/ICASSP.2011.5947265.

[26] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. Chaos, Solitons Fractals 2005;25:1261–71.

[27] Neville J. Jensen DBT-IC on ILP. Bias/Variance Anal Relat Dom 2007:27–8.

[28] Yu L, Wang Z, Tang L. A decomposition–ensemble model with data-characteristic-driven reconstruction for crude oil price forecasting. Appl Energy 2015;156:251–67. http://dx.doi.org/10.1016/j.apenergy.2015.07.025.

[29] Broomhead DS, Lowe D. Radial basis functions, multi-variable functional interpolation and adaptive networks. Tech Rep., vol. 2; 1988. p. 4148.

[30] Liu H, Tian H, Liang X, Li Y. Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks. Appl Energy 2015;157:183–94. http://dx.doi.org/10.1016/j.apenergy.2015.08.014.