

## Short-term wind speed predicting framework based on EEMD-GA-LSTM method under large scaled wind history

Yaoran Chen<sup>a</sup>, Zhikun Dong<sup>a</sup>, Yan Wang<sup>a</sup>, Jie Su<sup>a</sup>, Zhaolong Han<sup>a,b,c,d,\*</sup>, Dai Zhou<sup>a,b,c,d,\*</sup>, Kai Zhang<sup>e</sup>, Yongsheng Zhao<sup>a,b</sup>, Yan Bao<sup>a,b,c,d</sup>

<sup>a</sup> School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>b</sup> State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>c</sup> Key Laboratory of Hydrodynamics of Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>d</sup> Shanghai Key Laboratory for Digital Maintenance of Buildings and Infrastructure, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>e</sup> Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854, USA



### ARTICLE INFO

#### Keywords:

Wind speed prediction  
Extensive wind history  
Hybrid machine learning  
Data-mining  
EEMD-GA-LSTM

### ABSTRACT

Accurate short-term wind speed prediction is of great significance for early warning and regulation of wind farms. At present, the scale of wind speed time-history data is increasing, and its time resolution is also becoming higher. Traditional machine learning models cannot effectively capture and utilize nonlinear features from the large scaled dataset and this, not only increases the difficulty of model building, but also reduces the prediction accuracy. To overcome such challenges, a machine learning based framework involving data-mining method was proposed in this paper. To begin with, a powerful signal decomposition technique (ensemble empirical mode decomposition) was used to divide the original wind sequence into several intrinsic mode functions to form a potential feature set. Then, a more appropriate sub-feature set together with the corresponding machine learning model were automatically generated through an iteration process. Such process was constructed through a coupled algorithm using the binary coded searching method known as the genetic algorithm and the advanced recurrent neural network with long short term memory unit. The analytical results show that, when compared with the traditional mainstream models, the strategy of using the sequences provided by the signal decomposition technology as the input features can significantly improve the prediction accuracy. On the other hand, faced with the high-dimensional feature set generated from the big data, the selected sub-feature set can not only perform a large dimension reduction, but also further improve the prediction accuracy up to 28.33% in terms of different kinds of evaluation criteria. Therefore, there is a potential application of the proposed method on more accurate short-term wind speed prediction under a considerable dataset of wind history.

### 1. Introduction

A conspicuous global energy pathway is defined as the transition from relying on fossil fuels to low-carbon and renewable sources [1]. Among them, the wind energy has undergone an impressive development, with annually global capacity over 52 GW during the past five years [2]. Meanwhile, as a type of variable renewable energy (VRE), the time-dependent characteristics of wind flow have imposed technical challenges to its electrical power industry, including the difficulty in designing control strategy and the challenge of making efficient warning of extreme conditions for wind power system [3]. It can be said that a crucial step to overcome these challenges is to figure out the inlet wind

information for the near future, but this requires further knowledge about short-term wind speed [4]. Different from the time interval of monitoring for structural wind engineering, usually larger than a half-day, the definition of "short-term" for wind power system is generally less than 1 h, even in real time [3]. Therefore, to this extent, a precise forecasting model of short-term wind speed is considered to be necessary.

So far, the methodologies of wind speed prediction can be mainly classified into two categories (i) physical methods [5,6] and (ii) statistical methods [7–9]. Most of the physical methods are numerical weather forecasting (NWP) [10], based on the local environmental information, for instance, temperature, humidity and geography. Because of the model's complexity, especially with intricate topography [6],

\* Corresponding authors at: School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.

E-mail addresses: han.arkey@sjtu.edu.cn (Z. Han), zhoudai@sjtu.edu.cn (D. Zhou).

Nomenclature		WT	Wavelet Transform
		Symbols	
ANFIS	Adaptive Network-based Fuzzy Inference System	$\varphi$	DNA set
ANN	Artificial Neural Network	$g$	Index of generation
AR	Auto Regression	$i$	Number of LSTM trail for fitness evaluation
AWNN	Adaptive Wavelet Neural Network	$j$	Index of IMF
BPNN	Back-propagation Neural Network	$k$	Number of start point of test set
CNN	Convolutional Neural Network	$n$	Number of IMF
EA	Evolutionary Strategy	$t$	Time point
EEMD	Ensemble Empirical Mode Decomposition	$M$	Population size
ELM	Extreme Learning Machine	$T$	Total number of wind data points
EMD	Empirical Mode Decomposition	$x$	Input matrix for LSTM
ENN	Extension Neural Network	$y$	Output matrix for LSTM
FT	Fourier Transform	$c_j(t)$	IMF series through iteration
GA	Genetic Algorithm	$f(t)$	Forget gate function for LSTM
GRP	Gaussian Process Regression	$i(t)$	Input gate function for LSTM
GRU	Gate Recurrent Unit	$o(t)$	Output gate function for LSTM
HHT	Hilbert-Huang Transform	$r(t)$	Residual series
IMF	Intrinsic Mode Function	$w(t)$	Gaussian white noise series
KFCM	Kernel-based Fuzzy C-Means clustering	$\sigma(\cdot)$	Standard deviation
LSTM	Long Short Term Memory	$E(\cdot)$	Expectation
MSD	Multi-Step Decomposition	$F(\varphi)$	Objective function
NWP	Numerical Weather Forecasting	$P(\varphi)$	Probability to survive during selection
PM	Persistence Model	$X(t)$	Raw wind series
PSO	Particle Swarm Optimization	$X'(t)$	Noised wind series
RNN	Recurrent Neural Network	$\hat{X}(t)$	Predicted wind series
SSA	Singular Spectrum Analysis	MAE	Mean absolute error
SVM	Support Vector Machine	RMSE	Root mean square error
SVR	Support Vector Machine for Regression	MAPE	Mean absolute percentage error
SW	Shared Weight	$P_{MAE}$	Percentage improvements of MAE
VMD	Variational Mode Decomposition	$P_{RMSE}$	Percentage improvements of RMSE
VRE	Variable Renewable Energy	$P_{MAPE}$	Percentage improvements of MAPE
WIND	Wind Integration National Dataset		

**Table 1**

Comparison of recent machine learning models for short-term wind speed prediction.

Machine learning based short-term wind speed predicting models		Publications	Advantages	Drawbacks
Support Vector Machine models		[13–16]	High generalization, solid mathematical foundation	Considerable computational consumption for large scaled wind history
Fuzzy theory models		[17–19]	Applicable for systems that are difficult to be mathematically expressed	Lack of systematic design, sometimes need highly complex rules
Kalman Filter models		[16,20]	Do not need to store large history, low computational cost	Need pre-knowledge of the target system, sensitive to noise
Gaussian Regression Process		[11,15,20]	The prediction is probabilistic with confidence intervals	Not sparse, low efficiency in case of high-dimension input space
Extreme Learning Machine		[21–24]	Very fast to train the model, easy implementation for high-dimension space problem	Controversy on idea of the methodology and feasibility for real application
Artificial neural network based models	Back-propagation Neural Network based model	[25,26]	Basic model of ANN, straightforward to establish the model	Difficult to train a large number of parameters
	Convolutional Neural Network based model	[12,27,28]	Suitable to train high-dimension inputs especially with spacial relations	Complex to build the model for time series problem such as wind speed prediction
Recurrent Neural Network based models	General Recurrent Neural Network	[13,29,30]	Suitable to create dependencies between time steps	Convergence problems on constructing far-step time dependencies
	Advanced Recurrent Neural Network	[31,32]	Use “gates” to handle far-step time dependencies problems in general RNN	Lower performance than LSTM when facing large data
	Long-Short Term Memory	[9,11,12,14,17,24,25]		More parameters to be trained than GRU

such methods are generally time-consuming, so it is challenging to make the real-time prediction. In contrast, the statistical approaches are more suitable for short-term wind speed prediction, since they are based on historical information in the form of time-series data [9]. The most typical statistical models include autoregressive model (AR) and its variants —autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA). Such variants have been widely used and adopted as the benchmarks by former researches [9,11,12]. However, when facing data series with high nonlinearity, the poor expansibility of these traditional methods will hinder it from a better prediction performance.

In recent years, machine learning models have been increasingly used in the field of short-term wind speed prediction. These approaches are applicable to different scenarios and have greatly improved the prediction accuracy compared with the traditional AR-type models. As shown in Table 1, the most widely used models include Support Vector Machine (SVM), fuzzy theory model, Kalman filter model, Gaussian Regression Process (GRP), Extreme Learning Machine (ELM) and different types of Artificial Neural Networks (ANN). Among them, the ANN based models seem to be the best choice as they can directly learn knowledge from the training data (wind history) without needing any priori together with a higher adaptability to real-field applications [33]. The most basic ANN models are back-propagation neural networks (BPNN), where layers are simply stacked in-depth. This setup may result in an excessive training cost when the network becomes deeper. Addressed to this, in Chen et al. (2020) [25], a parameter optimization process is added into the structure of BPNN to help improving the training efficiency. However, another essential drawback for BPNN is that it doesn't present any links between neurons within the same layer. This means the model cannot make full use of the information provided by input time steps, which hinders it from a further better prediction performance for time series problems.

Different from BPNN, another alternative of ANN models, the convolutional neural networks (CNN), are good at finding local co-relations. Shivam et al. (2020) [28] conducted the prediction process by feeding the one-dimensional training series into a residual augmented causal CNN network, showing a noticeable outperformance to naive models. Nevertheless, the structure of such CNN based model seems too complex for the current problem, where more than 20 different layers and over 30,000 un-trained parameters are used, not to mention the number of hyper-parameters. Hence, so far, Recurrent Neural Networks (RNN) is considered the most proper designing of ANN for sequence modeling problems. In RNN, dependencies between each input step can be recursively constructed, and this fits well with [29,13,34] the time continuity for wind speed prediction. However, traditional RNN models may encounter convergence problems such as gradient explosion or gradient vanishing [35] when finding far-step dependencies. To overcome this issue, “gate control” technologies were employed in RNN models, for instance, gate recurrent unit (GRU) [31,32] or long short-term memory (LSTM) [11,12] algorithms. According to existing literatures [25,36,37], the performance of GRU and LSTM are comparable. However, LSTM based models appear to be more reliable since it has undergone more validations based on different wind datasets from various regions around the world [9,12,24], all showing satisfactory prediction outcomes.

In addition to machine learning model, the appropriate pre-processing technique on wind series is also of great importance to the model performance, and it has a crucial step known as the data-mining. The most commonly used data processing method is signal decomposition, including Singular Spectrum Analysis (SSA) [9,19,24], Variational Mode Decomposition (VMD) [9,24,31], Wavelet Transform (WT) [18,27,38] and Ensemble Empirical Mode Decomposition (EEMD) [18,23,39]. From a number of previous researches [18,27,38], it was confirmed that the WT based models have achieved satisfied outcomes, on account of its excellent localization characteristics in both time and frequency domains. However, it is not clear how to select the specific

wavelet function for an arbitrary dataset [40]. Similar embarrassment is encountered when implementing VMD based methods [9,24,31], where the number of modes is a priori value to be determined at the very beginning, yet posing a significant influence on the decomposition results. Considering the EEMD method shows its impressive superiority of automatic adjusting to any non-stationary time-series by introducing the intrinsic mode functions (IMF) [41], it could be the best choice when faced with such setback. Huang et al. (2019) [42] conducted the multi-step wind speed forecasting by summing all sub-predictions from LSTM models of every IMF resulting from the EEMD decomposition, and then rectified the summation through error correction strategy. Similar routine also appeared in Huang's previous work [43], where he considered the predictions from both EEMD-LSTM and EEMD-GRP models as a weighted combination to determine the final prediction. In [44], Qin et al. (2019) added a fuzzy classification process between EEMD and LSTM modules, first categorizing IMF components into different groups, then conducting LSTM predictions separately for each group and finally aggregating all forecasts into ultimate results. These investigations have laid a solid foundation for the applications combining EEMD with LSTM learning models.

Nevertheless, the issue is that, from the opposite side, it is not always beneficial when the number of sub-series is not defined in advance by using EEMD based models. The reason behind is that, nowadays, as the time resolution of the wind dataset becomes higher and the recording period gets longer, the scale of the dataset is expanding and the non-linearity and non-stability of the wind series are growing higher as a consequence. As a result of implementing EEMD method on such a largely scaled dataset, the number of sub-series (i.e. IMFs) will be considerably boosting at the same time. Therefore, there are at least two bottlenecks processes continue to follow the routines in aforementioned literatures [42–44].

To begin with, more IMF components would result in more untrained parameters, hence the total training cost would correspondingly increase. Secondly, if the design idea of the model is to reconstruct the wind speed sequence using predicted sub-series employing machine learning methods, the prediction error of each component will eventually be accumulated to the final error. Consequently, when the number of IMF increases, the model accuracy will decrease in reverse.

To overcome this issue and to further improve the prediction accuracy, a novel framework is proposed in this work, involving the signal decomposition tool (EEMD), a feature selection process (genetic algorithm, GA) and the machine-learning model (LSTM) all together. Here, different from some previous studies in the field [42–44], the components generated from EEMD method are no longer used for wind history reconstruction, but to compose a potential feature set for LSTM to learn from. Next, with the help of GA, such potential feature set will undergo a process of dimension reduction, producing a sub-collection that comprises hallmarks with more valuable information. As a result, the model can effectively counter with the increasing number of IMFs so as to successfully utilize the powerful EEMD method for short-term wind speed prediction under a large scaled dataset.

Considering all the steps explanations above, the main contributions of this work are stated as follows:

1. A novel structure of hybrid machine-learning framework combining EEMD, GA and LSTM for short-term wind speed prediction is proposed to face the background of the increasing scale of dataset. Different from previous researches [42–44], instead of following the design of “decomposition—prediction—reconstruction”, the proposed framework is trying to provide a more concise and valuable feature set out of a number of potential features that are generated by signal decomposition method for the given wind history.
2. The algorithm here proposed is fully automatic and is totally posteriori. Compared with former studies, this framework does not require any predetermined knowledge to the target system (e.g. the noise assessment for Kalman methods [16,20], logical rules for fuzzy based

- models [17,19] and wavelet functions for WT based models [18,27,38]. Here, the whole model is established only on the recorded wind dataset itself. When it comes to the feature selection process, it is also done through a heuristic iteration, trying to establish an entire end-to-end machine-learning model.
3. A rational validation for the framework is established based on the real world case, with more than 100,000 records of wind farm information collected throughout a year at a time-step of every 5 min. In previous examinations, dataset were limited to thousands of data points [24,31] and here, a largely scaled dataset is intentionally adopted to validate the feasibility of the model regarding the current issue. This change in the process is also suggested to avoid common training problems (e.g. over fitting) at the same time.
  4. In this work, a comparative study of the results is comprehensively conducted from a progressive multi-level. Different from former investigations, where comparisons with other models are conducted altogether [42,43], this research was conducted based on progressive aspects. First, the comparison is performed between the current model and the naive machine learning models. Then, it changes to an analysis among models that involve EEMD method. Finally, there is an additional contrast made between the current model and the EEMD-LSTM model but with a different feature set. Through these step-up comparisons, the superiority of signal decomposition (EEMD), machine learning model (LSTM) and feature selection process (GA) could all be tested.

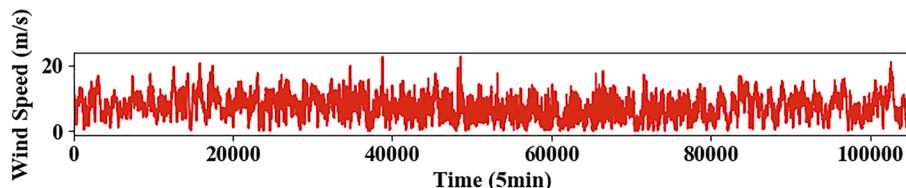
The following parts of this paper are constructed as below: the methodology of the wind speed prediction framework will be stated in [Section 2](#); an introduction of the dataset will be presented in [Section 3](#); the results of the numerical experiment as well as the comparison with parallel models will be discussed in [Section 4](#); finally, [Section 5](#) will summarize solid conclusions and raise the future plan.

## 2. Dataset

A statistical exploration of the dataset is presented in this section. As shown in [Fig. 1](#), the wind speed series being used is a one-year extraction of the WIND Toolkit [45], which is one of the largest publicly available wind datasets that realistically reflected the historical wind characteristics from more than 126,000 wind power sites across the United States [45].

The current dataset recorded the annual wind speed variation of an in-land wind farm in the northern-eastern region of Indiana, US, 2007. Its statistical characteristics are shown in [Table 2](#), where the minimum wind speed throughout the year is 0.054 m/s and the maximum wind speed value is up to 22.593 m/s. The difference between the average wind speed (7.342 m/s) and the median value (7.184 m/s) is small, indicating that the current dataset generally has a fairly balanced distribution.

There are two main reasons for adopting such dataset as the numerical validation of the proposed framework: First, in contrast to other datasets [11,12,39], the temporal resolution of the current dataset is per 5 min, hence the resultant forecasting model can meet the requirements with timely urgent background. Secondly, the scale of this dataset is considerably larger (over tenfold) than previous studies [31,41], hence the reliability of the framework can be more robustly validated.



**Fig. 1.** The original wind speed history from WIND Toolkit [24].

**Table 2**

The statistical information of original wind speed data.

Number of points	Time step (min)	Mean (m/s)	Median (m/s)	Max (m/s)	Min (m/s)	Standard Deviation (m/s)
105,121	5	7.342	7.184	22.593	0.054	3.460

Based on this dataset, the analytical study of short-term wind prediction is performed on Python 3.6 platform with Keras packages [46]. Two servers are in parallel for computation, with details as below:

Server 1: Intel(R) Core(TM) i7-7700 CPU@3.60 GHz, 3.60 GHz, 24 GB RAM;  
Server 2: Intel(R) Core(TM) i7-7500 CPU@2.70 GHz, 2.90 GHz, 16 GB RAM.

## 3. Methodology

In this section, the framework together with the approaches that used for short-term wind speed prediction are presented. As the flowchart shown in [Fig. 2](#), the EEMD, GA and LSTM algorithms are combined together, building the architecture of the current framework. The following sections will give an explanation of how these sub-models respectively works.

### 3.1. Ensemble empirical mode decomposition

In 1998, Huang et al. [47] initially proposed the Empirical Mode Decomposition (EMD) that based on Hilbert-Huang Transform (HHT). This method has been successfully employed throughout decades because of the following advantages: 1. It is suitable for both non-linear and non-stable signals; 2. Unlike WT or FT that needs pre-determined basis, HHT is fully adaptive by originally introducing the intrinsic mode functions (IMFs). However, some of these IMFs may contain oscillations of dramatically different scales, which is known as the “mode mixing”. This drawback will not only make these IMFs lose their physical meaning, but also render the EMD algorithm less robust, being sensitive to tiny perturbation of the dataset [48]. To overcome this problem, in 2009, the Ensemble Empirical Mode Decomposition (EEMD) was subsequently proposed [48]. By adding Gaussian white noise into the raw series, EEMD can attribute signals with different time scales automatically to the appropriate reference scales. As a result, the correlation between resultant IMFs and the raw series is largely improved [48]. The routine of EEMD is presented as follows [48]:

- 1 Adding Gaussian white noise signal  $w(t)$  into the target series  $X(t)$  to form a new signal  $X'(t)$  [48];
- 2 Decompose  $X'(t)$  using EMD method. Obtain IMFs  $c_j(t)$  and the residual  $r_n(t)$  [48];
- 3 Repeat the above steps. Each time, different white noise is added into the same raw series [48];
- 4 Since the mean value of Gaussian white noise is equal to zero, the grand average over all corresponding IMFs will be the final decompositions  $IMF_j(t)$  [48]:

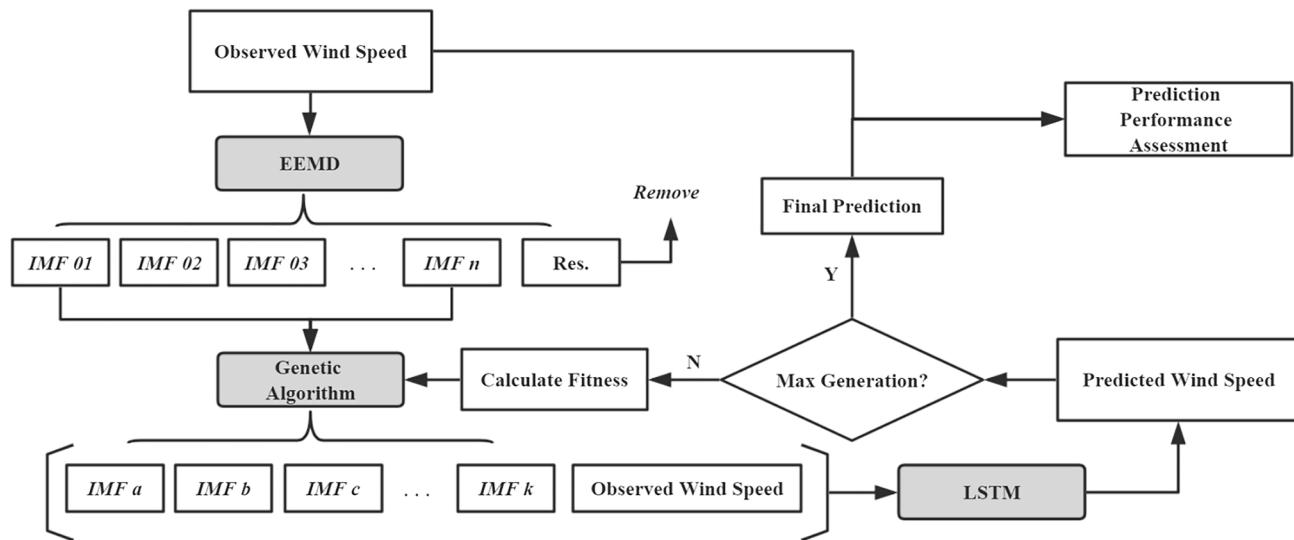


Fig. 2. Flowchart of the proposed approach combining EEMD, GA and LSTM.

$$\dot{X}(t) = X(t) + w(t)$$

(1)

$$\dot{X}(t) = \sum_{j=1}^n c_j(t) + r_n(t)$$

(2)

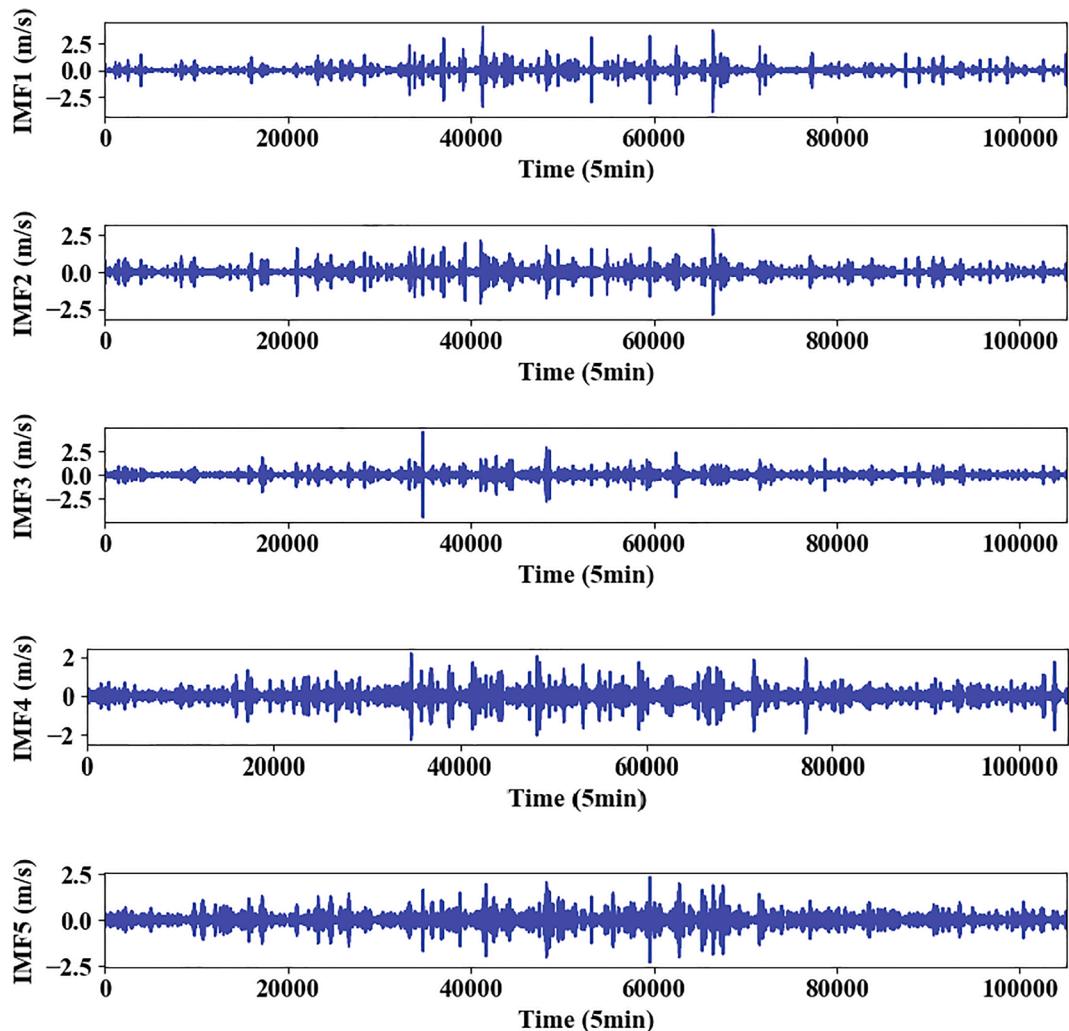


Fig. 3. All 17 intrinsic mode functions of the current wind speed dataset after EEMD processing.

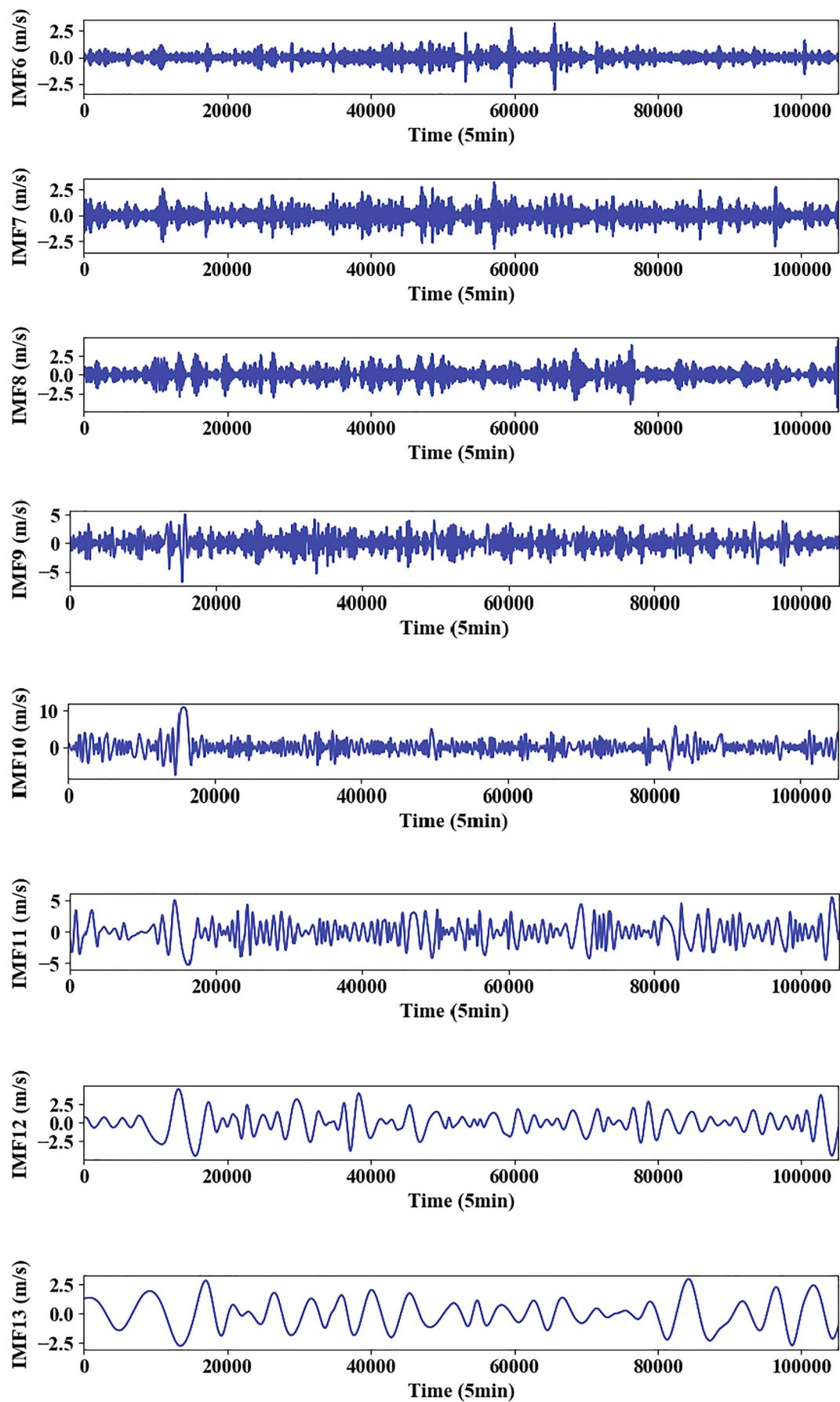


Fig. 3. (continued).

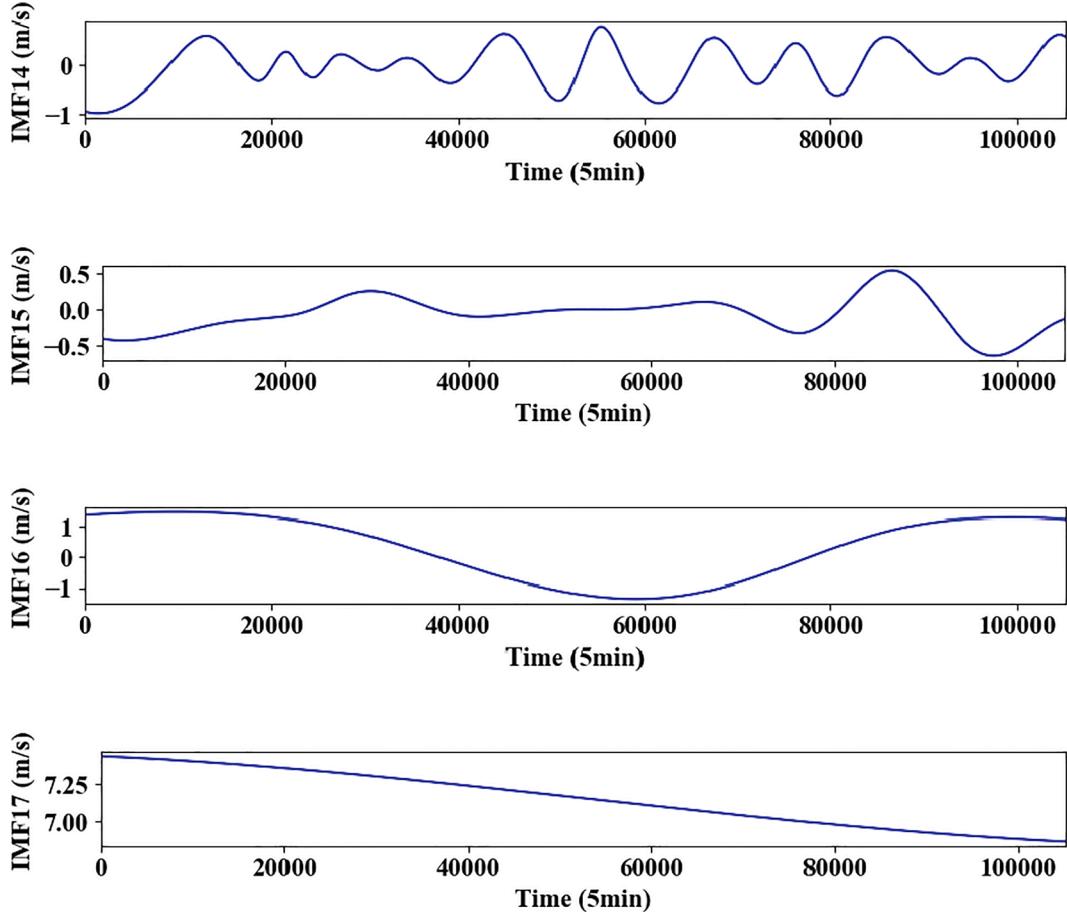


Fig. 3. (continued).

$$X^i(t) = X(t) + w^i(t) = \sum_{j=1}^n c_j^i(t) + r_n^i(t) \quad (3)$$

$$IMF_j(t) = \frac{1}{N} \sum_{i=1}^N c_j^i(t) \quad (4)$$

where  $i$  denotes the current number of trail;  $j$  denotes the number of each IMF;  $n$  is the total IMF number;  $N$  is the max number of iteration (i.e. the ensemble size).

Resolved by above process, in this work, the final decomposed series of current dataset are shown in Fig. 3, including 17 IMFs in total. As for the current 100,000-point series, this number of IMFs agrees well with Wu's prediction of  $\log_2 T$  ( $T$  is the total number of data points) [48].

### 3.2. Binary coded genetic algorithm

Wrapper is an important aspect in feature-selection engineering and deep learning researches. In this work, to improve the performance of the current wind-speed predictor, the binary coded GA is used to find the appropriate IMFs as the features set for LSTM training.

#### 3.2.1. Binary coding

As shown in Fig. 4, all 17 IMFs through EEMD are listed in sequence (from IMF1 to IMF17), followed by a same-sized list comprising only 0 and 1 (binary list). The final selected list of IMFs is then constructed through elemental multiplication between these two lists. In this way, it can be controlled whether each IMF should be discarded or not. If an IMF is needed, the element under the corresponding index of binary list is set to be 1; otherwise, it is set to be 0. As a definition, such binary list,

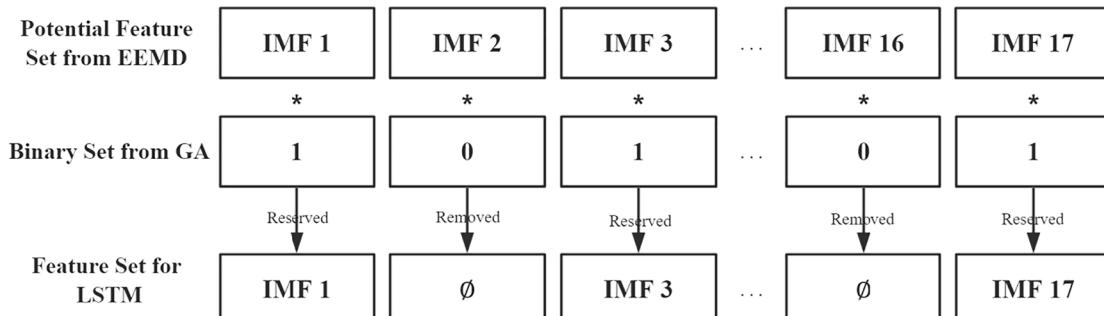


Fig. 4. Binary coding of GA for feature selection.

denoted by  $\varphi$ , is called as the DNA set.

### 3.2.2. Initial trails

An appropriate initial condition is important for searching algorithms like GA, because it can not only provide potential trails from the very beginning but also globally distribute the searching points. Based on such two considerations, in this work, the initial population is comprised of binary sets as follows:

1. All elements are set to be one.
2. All elements are set to be zero.
3. The first-half elements are set to be one and the second-half elements are set to be zero.
4. The first-half elements are set to be zero and the second-half elements are set to be one.
5. The elements whose corresponding IMFs have higher Pearson correlation with raw sequence are set to be one and others zero.
6. The elements whose corresponding IMFs have higher Pearson correlation with raw sequence are set to be zero and others one.
7. Random binary sets obeying normal distribution.

where given a pair of target series( $X, Y$ ), the Pearson correlation is defined as [23]:

$$\rho_{X,Y} = \frac{E((X - E(X))(Y - E(Y)))}{\sigma_X \sigma_Y}. \quad (5)$$

where  $X$  in this work is the raw dataset and  $Y$  represents the IMF;  $E(\cdot)$  and  $\sigma(\cdot)$  denote the expectation and standard deviation.

### 3.2.3. Fitness

The objective function of this work is defined as:

$$F(\varphi) = \frac{1}{\min(MAE(\varphi)_1, MAE(\varphi)_2, \dots, MAE(\varphi)_i)}, \quad (6)$$

where  $MAE(\varphi)_i$  is the mean absolute error between the prediction and observation on test set from  $i^{th}$  trail of the LSTM model using  $\varphi$  as the binary set for feature selection.

The reason for searching minimal from several parallel trails is to mitigate the effects of random initialization of weights and different hyper-parameters while processing the LSTM training, which may lead to unexpected local extreme points despite using the same model structure. A definition of mean absolute error (MAE) can be referred in the Result section.

### 3.2.4. Evolutionary procedure

Inspired by the natural law of “survival of the fittest”, the evolutionary procedure of GA includes: selection, crossover and mutation. An introduction of the main process is described in the following sections.

**3.2.4.1. Selection.** The selection process can be classified under two different conditions. First, the best individual of a generation will maintain itself and directly pass this filter. Besides, other individual who has a higher fitness will possess a higher probability to join in the next generation. In this work, such probability is given by the following expression:

$$P(\varphi_{g,k}) = \frac{F(\varphi_{g,k})^2}{\sum_{m=1}^M F(\varphi_{g,m})^2}. \quad (7)$$

where  $\varphi_{g,k}$  is the  $k^{th}$  individual in  $g^{th}$  generation;  $M$  is the number of individual in one generation (i.e. population size).

It should be noticed that Formula (7) is not a pure process of elimination. It not only provides a higher chance for potential individuals, but also allows individuals who have low fitness to enter into the next generation but with comparatively lower probabilities. This is beneficial

for maintaining the population diversity and increasing the algorithm robustness.

**3.2.4.2. Crossover.** After selection, each DNA candidates will have a probability (i.e. crossover rate) to recombine with another individual from the same generation. The child individual will inherit the DNA information from these two parent sets, with half of its binary code from one parent and the rest from the other.

Inspired by “the law of independent assortment” [49], as for each iteration in crossover, the parent individuals and the cross-points are independently and randomly selected. In another word, among the selected population group, each candidate and every DNA digit has the same chance to participate in the crossover process. Hence, in this part, there is no limitation for the population diversity.

**3.2.4.3. Mutation.** To mitigate the phenomenon of pre-mature and to further enlarge the searching range, a mutation process is also appended after the crossover. Should one element in DNA set be mutated, it will change from 0 to 1 or from 1 to 0. However, it is not necessary to conduct this process on all elements of DNA sets each time, which may cause the convergence problem and increase the computational time. Therefore, another probability, known as the mutation rate, is also considered in this work.

### 3.3. Long-short term memory neural network

The LSTM neural network is an improved version of traditional RNN. It is more robust in dealing problems with both short-term and long-term dependencies. The most innovative contribution of LSTM is the proposal of three gates to control its memory of historical information, including: forget gate, input gate and output gate [35]. Assumed that input series  $x = (x_1, x_2, \dots, x_t, \dots, x_T)$  and the output series  $y = (y_1, y_2, \dots, y_t, \dots, y_T)$  (the subscript denotes the time step), the procedure of calculation shown in Fig. 5 is as follows [35]:

$$f_t = \text{sigmoid}(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (8)$$

$$i_t = \text{sigmoid}(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (9)$$

$$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (10)$$

$$o_t = \text{sigmoid}(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (12)$$

$$h_t = o_t * \text{softsign}(C_t) \quad (13)$$

$$y_t = \text{sigmoid}(W_y \cdot h_t + b_y) \quad (14)$$

where  $f_t$ ,  $i_t$  and  $o_t$  represent the forget gate, input gate and output gate. The definitions of activation functions can be found in Ref. [46].

## 4. Results and discussion

Regarding to the proposed EEMD-GA-LSTM model, a case study on the abovementioned dataset is performed. In this work, the past half-hour wind speed information is used as the raw input to forecast the wind speed in the next 5 min. The sub-sections will be constructed as follows: First, the way of training/test set division and the criteria for performance evaluation will be presented. Next, the output of the feature selection from GA process will be illustrated. Finally, the performance of the proposed model will be evaluated through comparative study with different models.

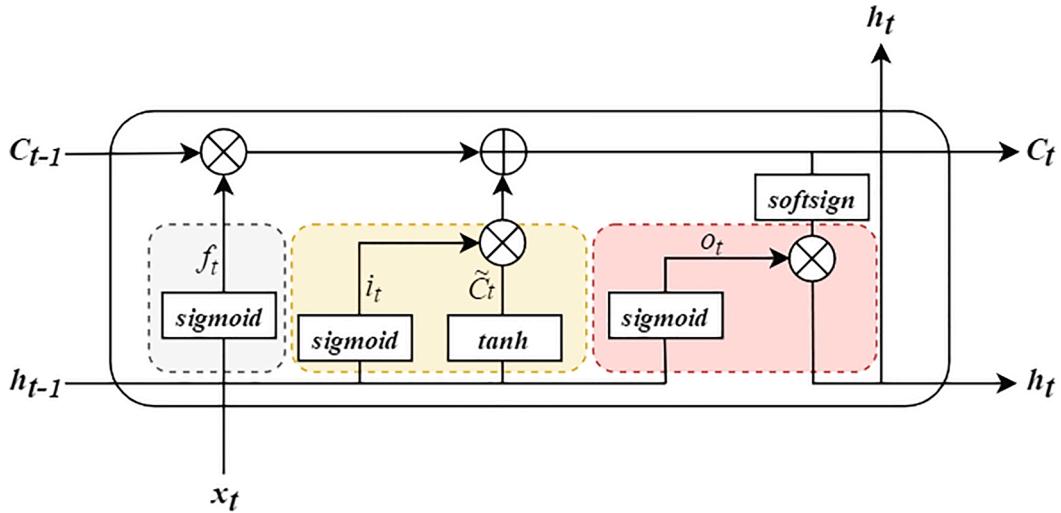


Fig. 5. Architectural of LSTM network containing three gates.

#### 4.1. Performance criteria

The raw dataset is divided into two parts: the training set and the test set. Since this is time series prediction problem, to avoid information leakage, the shuffle technique is not used. In this work, the first 60% of the dataset is selected as the training set and the rest 40% is the test set.

Given  $X = (X_1, X_2, \dots, X_k, \dots, X_T)$  as the observed raw wind speed history and  $\hat{X} = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_k, \dots, \hat{X}_T)$  as the predicted wind speed series, to evaluate the performance of the prediction from the proposed models, three criteria are adopted here to assess the loss in test set, which are mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) [24]:

$$\text{MAE} = \frac{1}{T - k + 1} \sum_{i=k}^T |X_k - \hat{X}_k| \quad (15)$$

$$\text{RMSE} = \sqrt{\frac{1}{T - k + 1} \sum_{i=k}^T (X_k - \hat{X}_k)^2} \quad (16)$$

$$\text{MAPE} = \frac{100}{T - k + 1} * \sum_{i=k}^T \left| \frac{X_k - \hat{X}_k}{X_k} \right|, \quad (17)$$

where  $T$  is the number of total points;  $k$  is the number of point that the test set begin from. To compare the accuracy between models, the corresponding percentage improvements are expressed as below [39]:

$$P_{\text{MAE}} = \frac{|MAE_1 - MAE_2|}{MAE_1} \quad (18)$$

$$P_{\text{RMSE}} = \frac{|RMSE_1 - RMSE_2|}{RMSE_1} \quad (19)$$

$$P_{\text{MAPE}} = \frac{|MAPE_1 - MAPE_2|}{MAPE_1} \quad (20)$$

#### 4.2. Results of feature selection

To balance the searching range and the computational cost, the population size of GA in this work is set to be 32. As shown in Fig. 6, the convergence lines of the mean (Equation (6)) and the best fitnesses among population are plotted below:

It can be referred from Fig. 6, the averaged fitness was continuously climbing in the first 15 generations and slowly converged afterwards, despite of the little oscillation. The best fitness gradually increased in the first 5 generations but nearly unchanged afterwards, showing that the potential selected feature set had been found in an early stage. Although

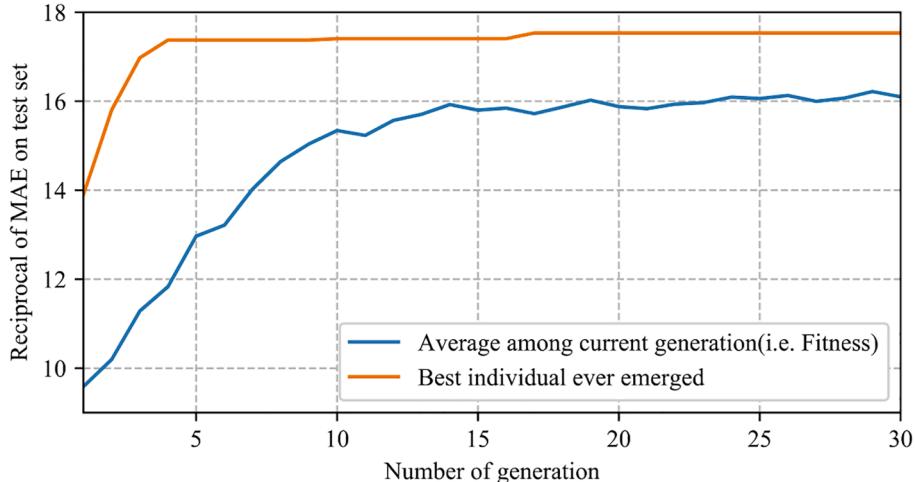


Fig. 6. Changes of mean and best fitness through generations.

in this stage, the averaged fitness had not converged to the best fitness, an early-stop can be implemented since the change of best fitness is evidently foreseen to keep flat. The fitness of the final best individual was 17.534, as the reciprocal, indicating its MAE was 0.057 m/s. The corresponding DNA set is as below:

[1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0]

It can be found that, only 5 IMFs are finally selected as the feature set. Compared with the unselected 17 IMFs in total, it has dropped more than 2/3 sub-series fed into the LSTM model, therefore the model is becoming significantly concise.

#### 4.3. Performance assessment through comparison

In this section, the discussion will be presented in two parts. In the first part, several mainstream models will be used for the comparative analysis. In the second part, an evaluation of the selected feature will be demonstrated, where the performance of the current EEMD-GA-LSTM model will be compared with that of the pure LSTM model as well as the EEMD-LSTM model whose feature set comprises all IMFs.

##### 4.3.1. Comparative study with parallel models

The performance of the proposed model is analyzed through comparative study with the actual data together with the results from several benchmarks, including: Persistence model, ARIMR, BPNN, ELM, RNN and GRU models. These individual models will be assessed twice: without EEMD or with EEMD for pre-processing.

To begin with, the proposed model is compared with a set of non-EEMD models. As shown in Fig. 7, the prediction results on a part of test set adopting different models are plotted together with the actual

wind history, followed by a 25-min zoom-in graph as a closer look to distinguish the difference in detail. As it can be seen from the figure, compared with the real series, major of results from naive benchmarks are right-shifted, where the most representative model is the PM that purely adopted the record in one time step former as the prediction. However, for the proposed model, the predicted curve (black) were intertwining with the real history (red) up and down, without presenting an evident phase difference. To quantificationally assess the prediction accuracy, three performance criteria of these models, namely MAE, RMSE and MAPE, are listed in Table 3. As it can be found, the prediction bias of conventional models (i.e. PM and ARIMR) is inferior to NN models while RNN based models (i.e. RNN and GRU) are better than naive NN models (i.e. ELM and BPNN). Moreover, compared to any other model, the performance of the proposed model is largely improved, where the MAE, RMSE and MAPE of the current model are 0.0570 m/s, 0.1337 m/s and 1.0622%, respectively. The percentages of improvements in terms of  $P_{MAE}$ ,  $P_{RMSE}$  and  $P_{MAPE}$  are shown in Table 4, with an overall enhancement of 56.23% can be observed throughout all entries. To illustrate the difference in a more intuitive way, the corresponding bar graphs of the tables are shown in Fig. 8 (MAE and RMSE) and Fig. 9 (MAPE).

Next, the EEMD method is added as a data-processing technique to the above benchmark models (excluding conventional statistical models: PM and ARIMR). A graphical comparison of the output is shown in Fig. 10, where all predicted curves generally replicated the trend of the real data, but to different degrees. Among them, the curve for EEMD-ELM based model seems to have the minimal non-linearity and the poorest performance, mainly because of the excessive simplicity of the ELM model structure. Compared with other curves, the current model showed a better prediction especially in some of the local extreme values

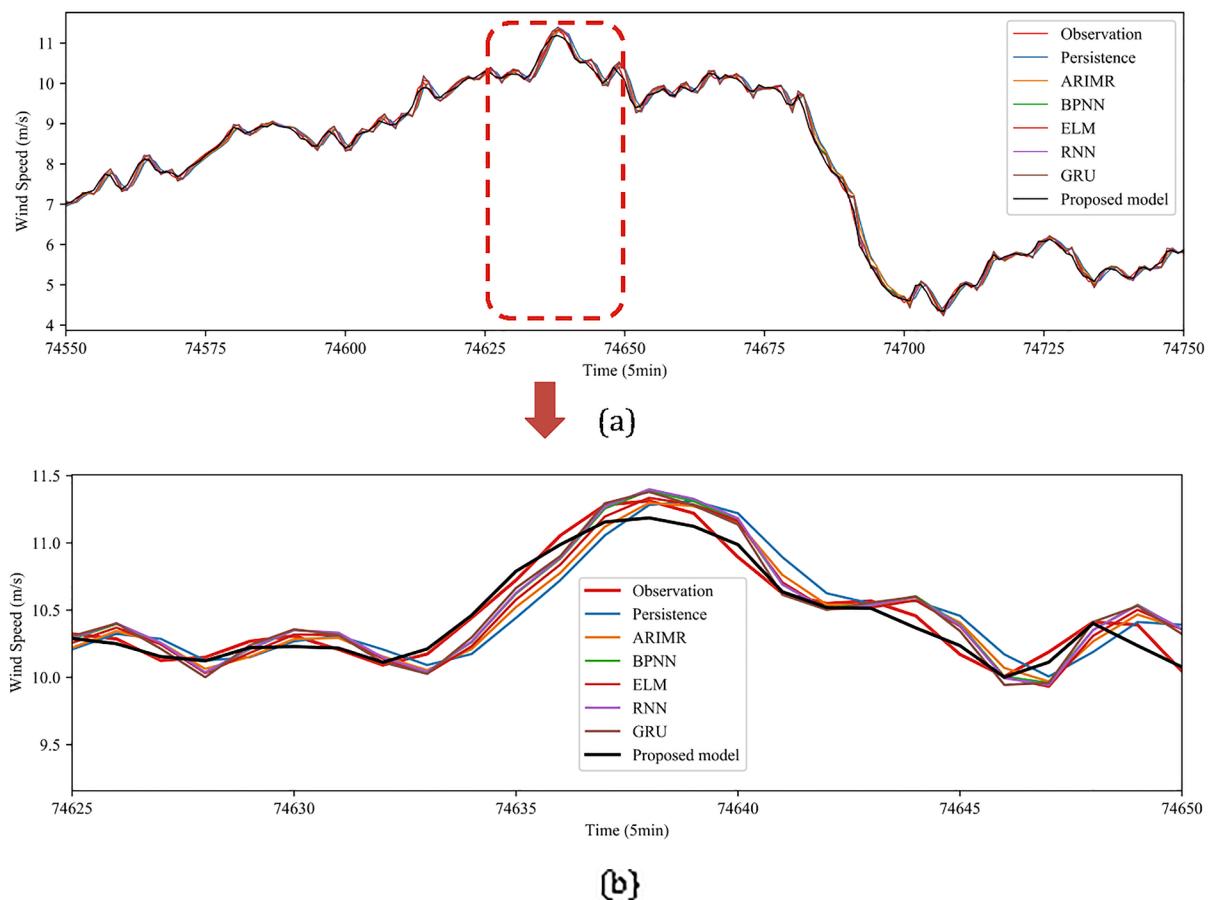


Fig. 7. Comparison forecasted values with non-EEMD benchmarks: (a). 200-min portion extracted from test set; (b). closer 25-min zoom-in series.

**Table 3**

Comparison of accuracy between proposed models and non-EEMD benchmarks.

	Persistence	ARIMR	BPNN	ELM	RNN	GRU	Proposed model
MAE (m/s)	0.1559	0.1372	0.1280	0.1274	0.1275	0.1202	0.0570
RMSE (m/s)	0.2952	0.2849	0.2978	0.2915	0.3038	0.2722	0.1337
MAPE (%)	2.9086	2.5929	2.7176	2.4340	2.3301	2.3099	1.0622

**Table 4**

Comparison of percentage performance of the proposed model (EEMD-GA-LSTM) over non-EEMD benchmarks.

	Persistence	ARIMR	BPNN	ELM	RNN	GRU
$P_{MAE}(\%)$	63.43	58.45	55.46	55.25	55.29	52.57
$P_{RMSE}(\%)$	54.70	53.07	55.10	54.13	55.99	50.88
$P_{MAPE}(\%)$	63.48	59.03	60.91	56.35	54.41	54.01

(such as the local minimums in Fig. 10(b)), reflecting a general superiority in terms of the different evaluation matrix as shown in Tables 5 and 6. An intuitive comparison is also made through chart graphs in Figs. 11 and 12, showing a noticeable improvement that the overall RMSE for the benchmarks decreased from nearly 0.3 to about 0.2 (Figs. 8 and 11 red bars), MAE changed from all over 0.1 to averagely less than 0.1 (Figs. 8 and 11 blue bars) and MAPE dropped from approximately 2.5–2.0 (Fig. 12). From another aspect, through the comparison between Table 3 and Table 5, the prediction performance is greatly raised by nearly 30% after implementing the EEMD method, proving the feasibility of employing EEMD method for accuracy enhancement.

The above evidences have successfully confirmed that the EEMD procedure can provide a useful feature set and promote the prediction accuracy. Nevertheless, the proposed model still outperform those EEMD-benchmarks by over 30 percent rates. To investigate the reason, an evaluation of the feature selection is discussed in next section.

#### 4.3.2. Evaluation of the selected feature

Particularly, in this section, three LSTM models will be further compared as an evaluation of feature selection: pure LSTM model, EEMD-LSTM model with all features included and the proposed EEMD-GA-LSTM model.

Fig. 13 shows the comparison of the representative training processes of 200 epochs on three feature sets using identical LSTM structure, where the y-label (i.e. the loss function) is the MAE between prediction and actual data after Min-Max normalization. As it can be referred, the final training loss was decreasing stage by stage when EEMD and GA module were successively added, dropping from 0.00555 for pure LSTM to 0.00333 for EEMD-LSTM and finally to 0.00252 when GA wrapper is involved.

Fig. 14 shows the prediction results of three LSTM models on the test

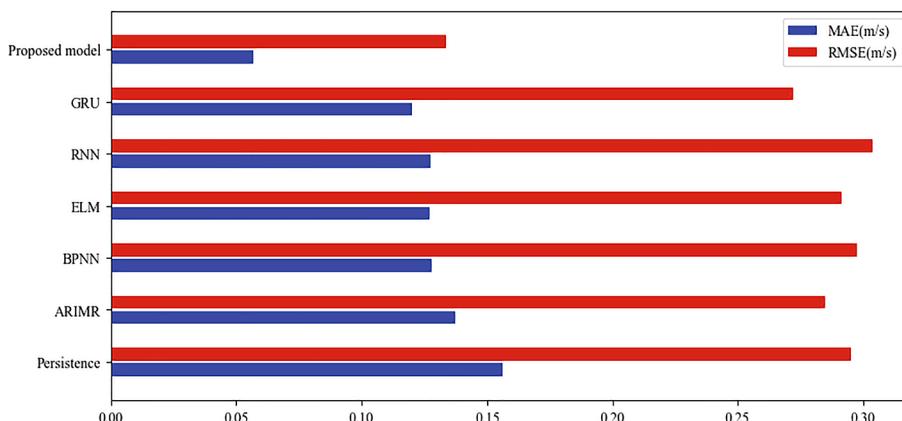


Fig. 8. Bar graph comparison of MAE and RMSE values of present model with non-EEMD benchmarks.

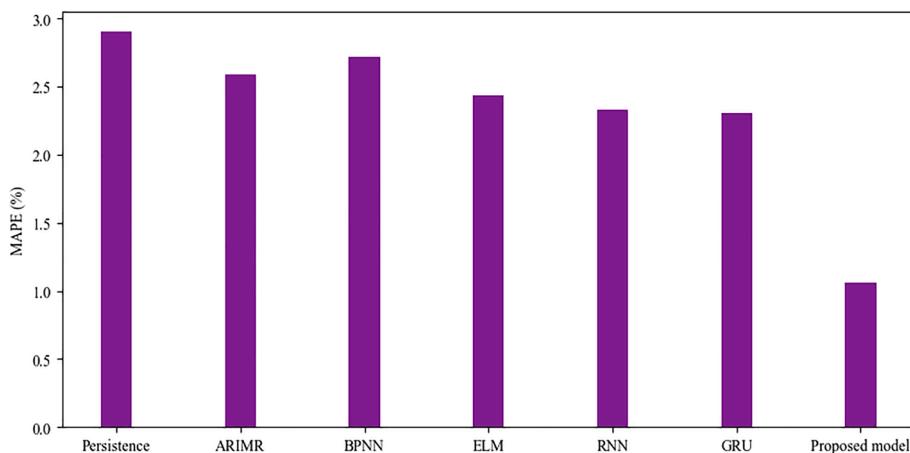
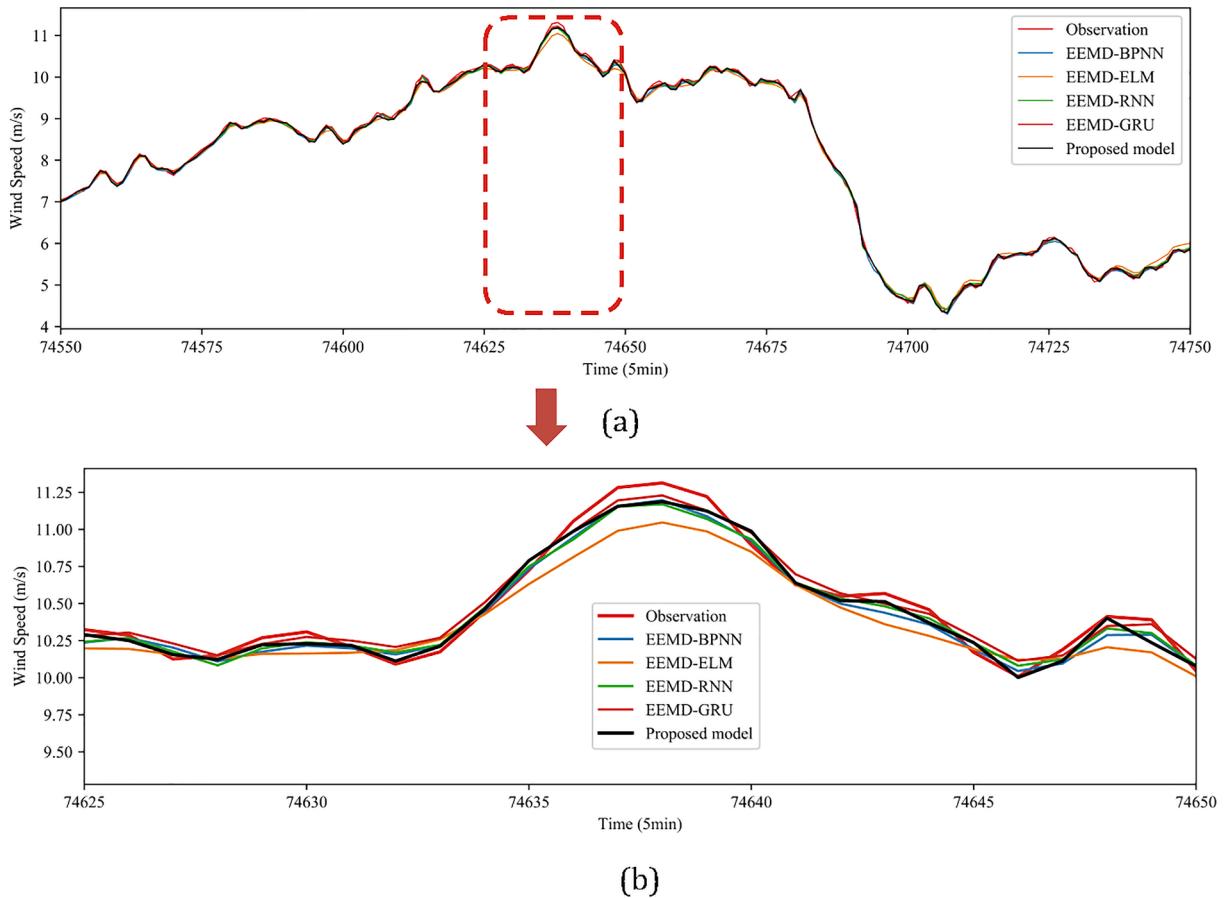


Fig. 9. Bar graph comparison of MPE values of present model with non-EEMD benchmarks.



**Fig. 10.** Comparison forecasted values with EEMD benchmarks: (a). 200-min portion extracted from test set; (b). closer 25-min zoom-in series.

**Table 5**  
Comparison of accuracy between proposed models and EEMD benchmarks.

	EEMD-BPNN	EEMD-ELM	EEMD-RNN	EEMD-GRU	Proposed model
MAE (m/s)	0.0883	0.1118	0.0826	0.0895	0.0570
RMSE (m/s)	0.2015	0.2012	0.2003	0.1919	0.1337
MAPE (%)	2.2520	2.2537	1.5284	1.8666	1.0622

**Table 6**  
Comparison of percentage performance of the proposed model (EEMD-GA-LSTM) over EEMD benchmarks.

	EEMD-BPNN	EEMD-ELM	EEMD-RNN	EEMD-GRU
$P_{MAE}(\%)$	35.44	49.01	30.99	36.31
$P_{RMSE}(\%)$	33.64	33.54	33.25	30.32
$P_{MAPE}(\%)$	52.83	52.86	30.50	43.09

set. As it can be clearly pointed, compared with the other two, there is relatively large error between the prediction of pure LSTM (blue line) and the observation (red line). On the other hand, from Fig. 14(b), the curve of the proposed model, EEMD-GA-LSTM curve (black) is obviously more sticking on the actual data than the EEM-LSTM curve (yellow). More statistical comparison is shown in Table 7, Figs. 15 and 16. Though the error of pure LSTM model is generally better than naive non-EEMD benchmarks in Table 3, it is also doubled from the proposed model. Similarly, for EEMD-LSTM model, in spite that the performance is generally improved compared with other EEMD-benchmarks in Table 5,

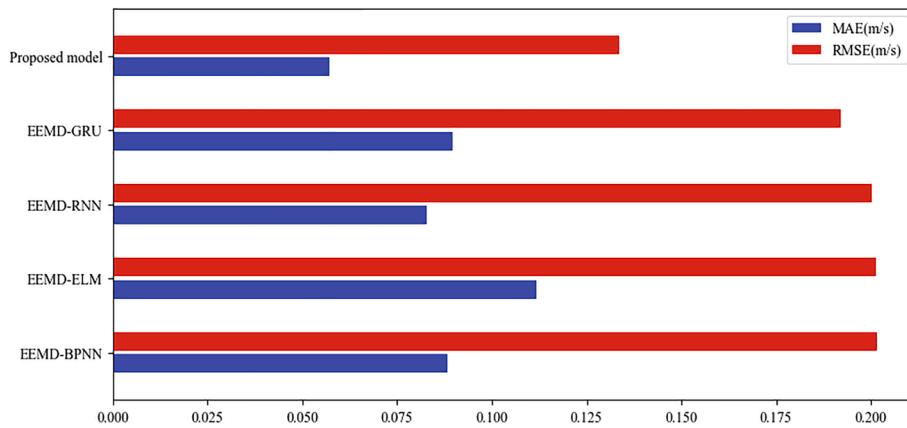
there is also at least 20% difference away from the current outcome.

In real applications, the short-term wind speed prediction from the proposed framework can be realized through a distributed system (Fig. 17), where one system is used to train the off-line models and the other is used for on-line predictions. For the largely scaled wind history, the influence of single upcoming data on EEMD spectrum can almost be negligible. Therefore, the on-line system can use the pre-trained models to make the timely prediction within a period of time. Meanwhile, receiving more and more records, the offline system can simultaneously update the model. As the number of data is remarkably increasing, the model will be transmit back into the online system for further applications.

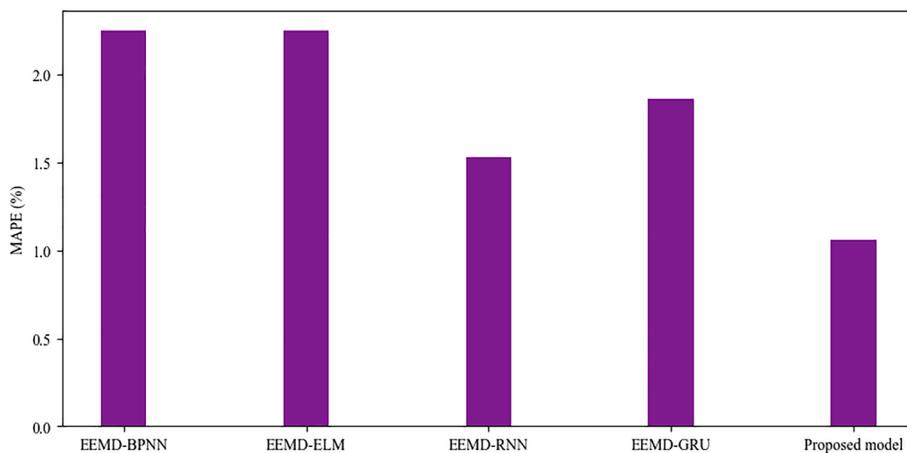
## 5. Conclusion

In this paper, a hybrid machine-learning framework is proposed as a way to achieve short-term wind speed prediction under an extensive dataset of wind history. The model here suggested is composed of three main algorithms: EEMD, GA and LSTM. First, considering a large scaled wind history and adopting EEMD as the signal decomposition technique, it was possible to obtain a potential feature set comprising a bunch of inerratic sub-datasets, known as the intrinsic mode functions. Then, through an iterative process using the coupled GA-LSTM algorithm, the heuristically selected feature set and the well-tuned machine-learning model are synchronously established. While running the proposed framework, the workflow is fully automatic and no prior functions are necessary. This easiness in the process makes it suitable for end-to-end requirements in applications.

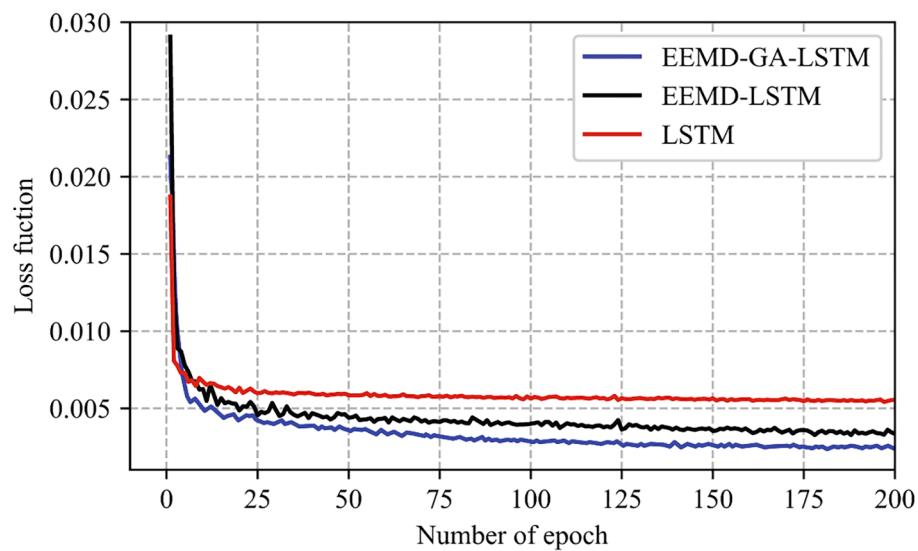
Further on, the feasibility and effectiveness of the model were comprehensively validated through an analytic study. Based on various



**Fig. 11.** Bar graph comparison of MAE and RMSE values of present model with EEMD benchmarks.



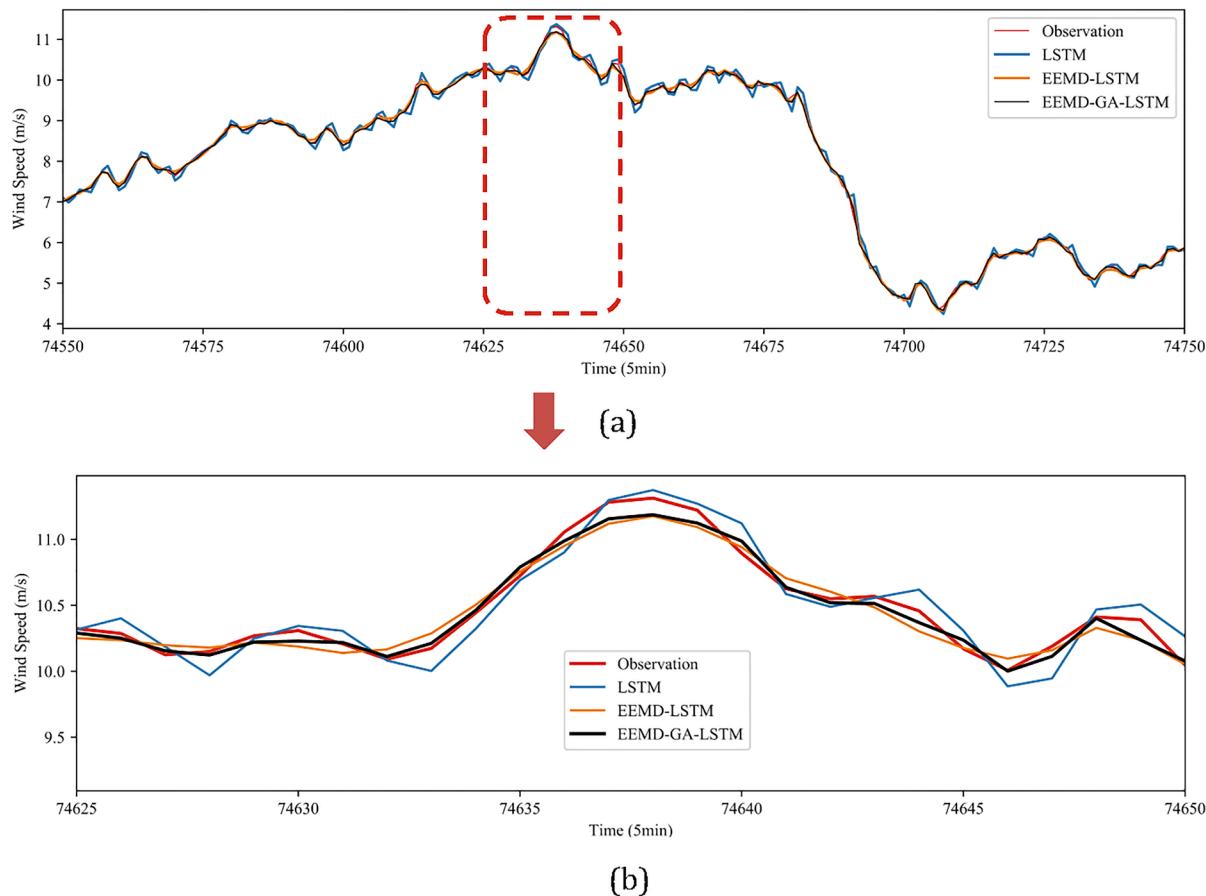
**Fig. 12.** Bar graph comparison of MPE values of present model with EEMD benchmarks.



**Fig. 13.** The comparison of the training processes on LSTM, EEMD-LSTM and EEMD-GA-LSTM.

evaluation standards, including MAE, RMSE and MAPE, the proposed method shows its impressive superiority beyond the mainstream models. To begin with, compared with non-EEMD models, the prediction accuracy of the current model is considerably improved by 56.25%, on average. Afterwards, in comparison with EEMD method connecting

with other learning prototypes, there is also a noticeable enhancement of averagely 38.48% on prediction accuracy. In addition, the results from the same learning model with different feature sets are considered, and the proposed method is still believed to be more powerful from two points of view: First, thanks to the GA wrapper for feature selection, the



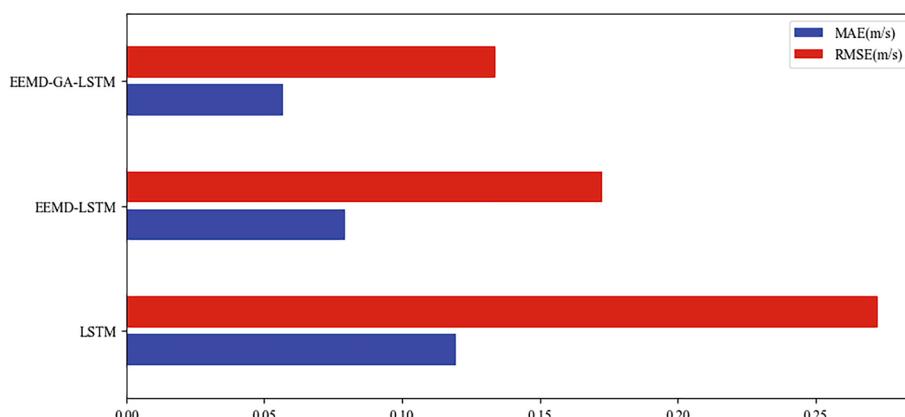
**Fig. 14.** Comparison of predicted values among LSTM models: (a). 200-min portion extracted from test set; (b). closer 25-min zoom-in series.

**Table 7**  
Comparison of accuracy between LSTM models.

Criteria	Models		
	LSTM	EEMD-LSTM	EEMD-GA-LSTM (Proposed model)
MAE (m/s)	0.1197	0.0793	0.0570
RMSE (m/s)	0.2727	0.1723	0.1337
MAPE (%)	2.2364	1.6207	1.0622
$P_{MAE}(\%)$	52.38	28.12	/
$P_{RMSE}(\%)$	50.97	22.40	/
$P_{MAPE}(\%)$	52.50	34.46	/

input dimension of the model is largely reduced to less than one third of the length of the original feature set, making it become more concise and robust to data perturbation. Moreover, in terms of the forecasting precision, it is also greatly optimized compared with the all-feature and non-feature models under all kinds of criteria of assessments, reaching 51.95% and 28.33% of improvements, respectively. Hence, all three parts of the algorithm contribute to a higher performance of the model and its effectiveness for short-term wind speed prediction is therefore confirmed.

In the future, there is still a gap to further improve and enrich the current study regarding some aspects: First of all, it is possible to investigate and compare different searching methods as the feature wrapper for the proposed framework, such as particle swarm



**Fig. 15.** Bar graph comparison of MAE and RMSE values of LSTM models.

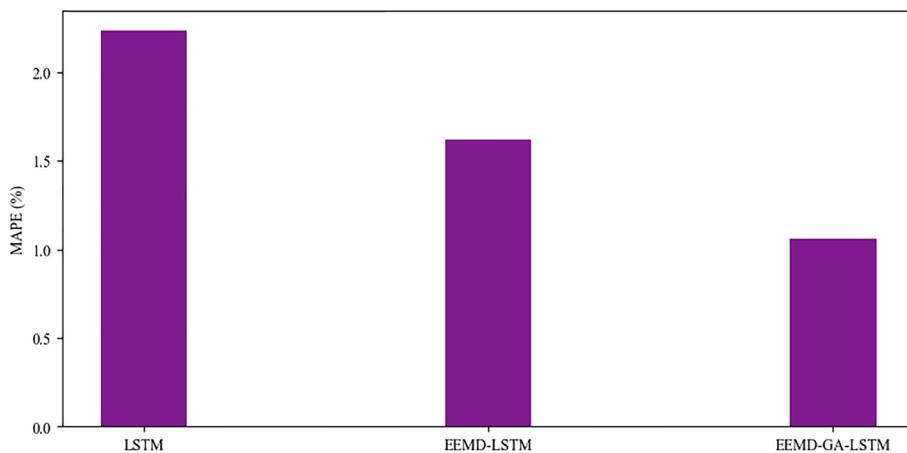


Fig. 16. Bar graph comparison of MAPE values of LSTM models.

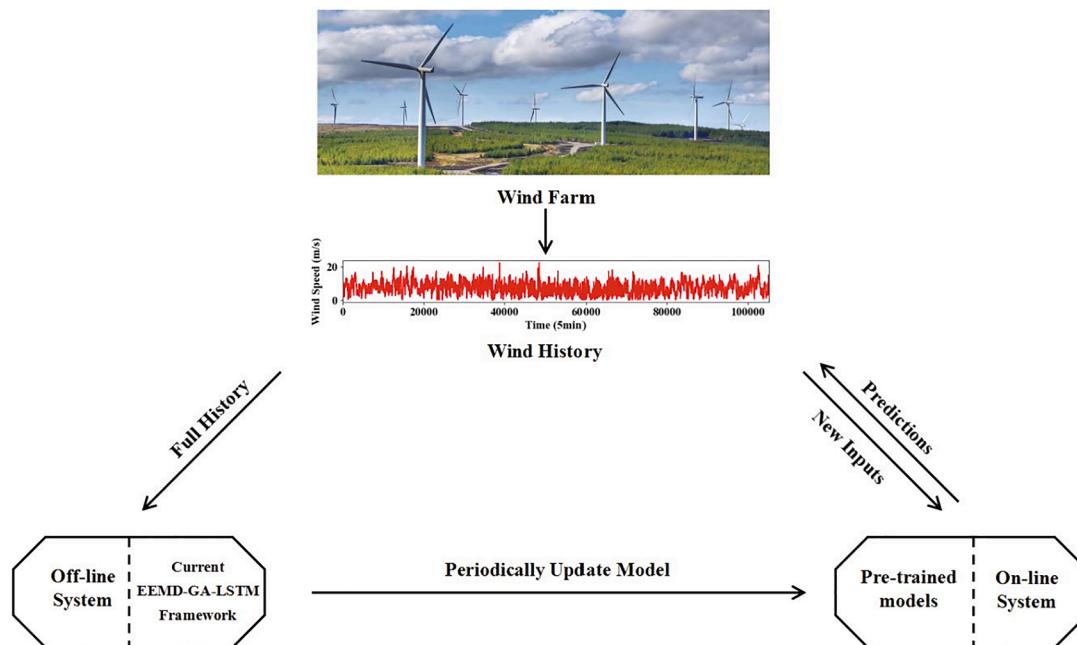


Fig. 17. Flowchart of the distributed system for the current framework in real application.

optimization (PSO) method and evolutionary algorithm (EA). Secondly, there is a breach to add other potential attributes with physical definitions (e.g. humidity, temperature and air pressure) into the feature set of the current framework and to discuss their influence on model performance. Last but not least, it is meaningful to consider other large-scaled wind datasets from different wind farm regions and with different resolutions, aiming to validate the proposed model in a more global way.

#### CRediT authorship contribution statement

**Yaoran Chen:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Resources, Validation, Visualization, Writing - original draft, Writing - review & editing. **Zhikun Dong:** Methodology. **Yan Wang:** Resources, Data curation. **Jie Su:** Writing - review & editing. **Zhaolong Han:** Project administration, Funding acquisition, Supervision, Writing - review & editing. **Dai Zhou:** Project administration, Funding acquisition, Writing - review & editing. **Kai Zhang:** Writing - review & editing. **Yongsheng Zhao:** Writing - review & editing. **Yan Bao:** Project administration, Funding acquisition, Supervision, Writing - review & editing.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

The financial supports from the National Natural Science Foundation of China (Nos. 51879160, 51809170, 11772193 and 51679139), Innovation Program of Shanghai Municipal Education Commission (No.: 2019-01-07-00-02-E00066), and Shanghai Natural Science Foundation (No 18ZR1418000) are gratefully acknowledged. This research is also sponsored in part by Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (No. ZXDF010037), Program for Intergovernmental International S&T Cooperation Projects of Shanghai Municipality (No. 18290710600), Program for International Cooperation of Shanghai Science and Technology (No. 18160744000) and “Shuguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal

Education Commission (No. 19SG10).

## References

- [1] Guliyev F. Trump's "America first" energy policy, contingency and the reconfiguration of the global energy order. *Energy Policy* 2020;140:111435.
- [2] Lacal-Arantegui R. Globalization in the wind energy industry: contribution and economic impact of European companies. *Renewable Energy* 2019;134:612–28.
- [3] Viviescas C, Lima L, Diuana FA, Vasquez E, Ludovique C, Silva G, et al. Contribution of Variable Renewable Energy to increase energy security in Latin America: complementarity and climate change impacts on wind solar resources. *Renewable Sustainable Energy Rev* 2019;113:109232.
- [4] Song D, Yang Y, Zheng S, Deng X, Yang J, Su M, et al. New perspective on maximum wind energy extraction of variable-speed wind turbines using previewed wind speeds. *Energy Convers Manage* 2020;206:112496.
- [5] Kibona TC. Application of WRF mesoscale model for prediction of wind energy resources in Tanzania. *Sci Afr* 2020;7:e00302.
- [6] Miguel AP, Carlos OC, Felipe CF, Gonzalo MM. Wind power forecasting for a real onshore wind farm on complex terrain using WRF high resolution simulations. *Renewable Energy* 2019;135:674–86.
- [7] Min Y, Bin W, Liang-Li Z, Xi C. Wind speed forecasting based on EEMD and ARIMA. In: Chinese Automation Congress (CAC); 2015. <https://doi.org/10.1109/cac.2015.7382700>.
- [8] Khodayar M, Wang J, Manthouri M. Interval deep generative neural network for wind speed forecasting. *IEEE Trans Smart Grid* 2019;10(4):3974–89.
- [9] Moreno SR, Silva RGD, Mariani VC, Coelho LDS. Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network. *Energy Convers Manage* 2020;213:112869.
- [10] Lowery C, O'Malley M. Impact of wind forecast error statistics upon unit commitment. *IEEE Trans Sustainable Energy* 2012;3(4):760–8.
- [11] Zhang Z, Ye L, Qin H, Liu Y, Wang C, Yu X, et al. Wind speed prediction method using Shared Weight Long Short-Term Memory Network and Gaussian Process Regression. *Appl Energy* 2019;247:270–84.
- [12] Qin Y, Li K, Liang Z, Lee B, Zhang F, Gu Y. Hybrid forecasting model based on long short term memory network and deep learning neural network for wind signal. *Appl Energy* 2019;236:262–72.
- [13] Yu C, Li Y, Bao Y, Tang H, Zhai G. A novel framework for wind speed prediction based on recurrent neural networks and support vector machine. *Energy Convers Manage* 2018;178:137–45.
- [14] Vinothkumar T, Deeba K. Hybrid wind speed prediction model based on recurrent long short-term memory neural network and support vector machine models. *Soft Comput* 2019. <https://doi.org/10.1007/s00500-019-04292-w>.
- [15] Cai H, Jia X, Feng J, Li W, Hsu YM, Lee J. Gaussian process regression for numerical wind speed prediction enhancement. *Renewable Energy* 2020;146(2):2112–23.
- [16] Cai H, Jia X, Feng J, Yang Q, Hsu YM, Chen Y, et al. A combined filtering strategy for short term and long term wind speed prediction with improved accuracy. *Renewable Energy* 2019;136:1082–90.
- [17] Qiu Y, Li Q, Pan Y, Huang L, Sun C, Yang H, et al. Modeling the adaptive uncertainty sets of robust optimization based on long short-term memory network and modified fuzzy information granulation. *IEEE Access* 2020;8:56072–80.
- [18] He Q, Wang J, Lu H. A hybrid system for short-term wind speed forecasting. *Appl Energy* 2018;226(15):756–71.
- [19] Moreno SR, Leandro DSC. Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy inference system. *Renewable Energy* 2017. S0960148117311904.
- [20] Zhao X, Wei H, Li C, Zhang K. A hybrid nonlinear forecasting strategy for short-term wind speed. *Energies* 2020;13.
- [21] Tan L, Han J, Zhang H. Ultra-short-term wind power prediction by salp swarm algorithm-based optimizing extreme learning machine. *IEEE Access* 2020;8:44470–84.
- [22] Han Y, Tong X. Multi-step short-term wind power prediction based on three-level decomposition and im-proved grey wolf optimization. *IEEE Access* 2020;99:1.
- [23] Tian Z, Li S, Wang Y. A prediction approach using ensemble empirical mode decomposition-permutation entropy and regularized extreme learning machine for short-term wind speed. *Wind Energy* 2020;23(2).
- [24] Liu H, Mi X, Li Y. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Convers Manage* 2018;159:54–64.
- [25] Chen G, Li L, Zhang Z, Li S. Short-term wind speed forecasting with principle-subordinate predictor based on conv-lstm and improved bpnn. *IEEE Access* 2020;99:1.
- [26] Zhang Y, Pan G, Chen B, Han J, Zhao Y, Zhang C. Short-term wind speed prediction model based on GA-ANN improved by VMD. *Renewable Energy* 2020;156:1373–88.
- [27] Wang HZ, Li GQ, Wang GB, Peng JC, Jiang H, Liu YT. Deep learning based ensemble approach for probabilistic wind power forecasting. *Appl Energy* 2017;188(15):56–70.
- [28] Shivam K, Tzou JC, Wu SC. Multi-step short-term wind speed prediction using a residual dilated causal convolutional network with nonlinear attention. *Energies* 2020;13(7):1772.
- [29] Sandhu KS, Nair AR. A comparative study of ARIMA and RNN for short term wind speed forecasting. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE; 2019. p. 1–7.
- [30] Wang J, Qin S, Zhou Q, Jiang H. Medium-term wind speeds forecasting utilizing hybrid models for three different sites in Xinjiang, China. *Renewable Energy* 2015;76:91–101.
- [31] Li C, Tang G, Xue X, Saeed A, Hu X. Short-term wind speed interval prediction based on ensemble GRU model. *IEEE Trans on Sustainable Energy* 2019;1(1):99.
- [32] Li C, Tang G, Xue X, Saeed A, Hu X. Short-term wind speed interval prediction based on ensemble gru model. *IEEE Trans Sustainable Energy* 2019;99:1.
- [33] Tascikaraoglu A, Uzunoglu M. A review of combined approaches for prediction of short-term wind speed and power. *Renewable Sustainable Energy Rev* 2014;34:243–54.
- [34] Tanaka I, Ohmori H. Method selection in different regions for short-term wind speed prediction in Japan. 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). 2015.
- [35] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9(8):1735–80.
- [36] Zhang D, Lindholm G, Ratnaweera H. Use long short-term memory to enhance internet of things for combined sewer overflow monitoring. *J Hydrol* 2018;556:409–18.
- [37] Melo GAD, Sugimoto DN, Tasinoff PM, Santos AHM, Cunha AM, Dias LAV. A new approach to river flow forecasting: lstm and gru multivariate models. *IEEE Lat Am Trans* 2020;17(12):1978–86.
- [38] Jianjun H, Chuanjin Y, Yongle L, Huoyue X. Ultra-short term wind prediction with wavelet transform, deep belief network and ensemble learning. *Energy Convers Manage* 2020;205:112418.
- [39] Santhosh M, Venkaiah C, Kumar DMV. Ensemble empirical mode decomposition based adaptive wavelet neural network method for wind speed prediction. *Energy Convers Manage* 2019;168:482–93.
- [40] Huimin Z, Meng S, Wu D, Xinhua Y. A new feature extraction method based on eemd and multi-scale fuzzy entropy for motor bearing. *Entropy* 2016;19(1):14.
- [41] Wu Z, Huang NE. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Adv Adapt Data Anal* 2009;1(01):1–41.
- [42] Huang Y, Yang L, Liu S, Wang G. Multi-step wind forecasting based on ensemble empirical mode decomposition, long short term memory network and error correction strategy. *Energies* 2019;12(10):1822.
- [43] Huang Y, Liu S, Yang L. Wind speed forecasting method using EEMD and the combination forecasting method based on gpr and LSTM. *Sustainability* 2018;10.
- [44] Qin Q, Lai X, Zou J. Direct multistep wind speed forecasting using LSTM neural network combining EEMD and fuzzy entropy. *Appl Sci* 2019;9(1).
- [45] Draxl C, Clifton A, Hodge BM, McCaa J. The wind integration national dataset (wind) toolkit. *Appl Energy* 2015;151:355–66.
- [46] Gullu A, Pal S. Deep learning with Keras. Packt Publishing Ltd.; 2017.
- [47] Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, et al. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc A* 1998;454(1971):903–95.
- [48] Wu Z, Huang NE. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Adv Adapt Data Anal* 2009;01(01):1–41.
- [49] Biffen RH. Mendel's laws of inheritance and wheat breeding. *J Agric Sci* 1905;1(1):4–48.