



# Ultra-short-term wind speed prediction based on empirical wavelet transform and combined model

Maosen Wang<sup>1</sup> · Zhongda Tian<sup>1</sup>

Received: 15 September 2023 / Accepted: 28 November 2023 / Published online: 15 December 2023  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

Due to the variability, proneness to outside influences, and constantly fluctuating nature of ultra-short-term wind speed sequences, wind speed forecasts in wind farms are not precise. A method to predict ultra-short-term wind speed based on the empirical wavelet transform and a combined model is proposed. Correct ultra-short-term wind speed estimations are crucial for preserving the stability of the electrical power system. Firstly, the empirical wavelet transform technique is used to decompose the ultra-short-term wind speed time sequences into multiple modal constituents; secondly, sample entropy can be utilized for calculating the complexities of the decomposed data elements, and those that can be rebuilt with a similar level of sophistication are rebuilt; after this, the sophistication of the rebuilt components is analyzed, and the improved sparrow search algorithm optimized extreme learning machine and gated recurrent neural network are used as prediction models for various sections; then, an improved sparrow search algorithm to optimize the weight coefficients of each prediction model. Finally, the final prediction results are obtained by multiplying the prediction results of each method by the resulting optimal weight coefficients cumulatively. The research objects are three sets of measured ultra-short-term data on wind speeds with analysis intervals of 15 min and 5 min. Following experimental validation, the suggested method outperforms previous single and combination prediction methods in terms of accuracy when making predictions.

**Keywords** Ultra-short-term · Prediction · Empirical wavelet transform · Sample entropy · Extreme learning machine · Improved sparrow search algorithm · Recurrent neural network

## Introduction

### Background

Since the consumption of non-renewable energy sources like coal, oil, and natural gas has an enormous adverse effect on the ecosystem and climate in today's fast-paced society and economy, developing energy sources that are renewable is a top priority for all nations (Kebede et al. 2022). Over the past two decades, the power of wind has been regarded as one of the cleanest sources of energy from renewable sources, and interest in its expansion is growing (Hao et al. 2020). The electricity grid operates more dependably when

wind and solar energy are incorporated into it (Jiang et al. 2018). Grid integration of wind energy is vital for the regulation of electricity networks since wind energy is non-stationary, variable, and inconsistent (Li et al. 2020a). Correct wind speed forecasting can lessen the strain grid-connected wind power places on the electrical infrastructure.

In terms of time scales, wind speed prediction is commonly divided into four categories: ultra-short-term prediction, shorter-term prediction, medium-term prediction, and prediction for the long term (Moreno et al. 2020). In the process of wind power generation, wind speed prediction on different time scales can be selected as needed for better wind power operation management. The ultra-short-term forecasts have a time unit of minutes, and based on the forecast results, the electricity network can be dispatched directly in real time, which helps to scientifically evaluate the decisions of wind farms. Short-term forecasts, with hourly time scales, have a slightly larger span than ultra-short-term forecasts, are susceptible to natural factors such as weather or temperature, and are generally used to improve grid dispatching

---

Communicated by H. Babaie

---

✉ Zhongda Tian  
tianzhongda@126.com

<sup>1</sup> College of Artificial Intelligence, Shenyang University of Technology, Shenyang 110870, China

capabilities. Medium-term forecasts have a larger time span of days or months than the first two forecasts, and are usually used to provide references for quality maintenance of wind power equipment and rational arrangement of power storage. Compared with other time-scale wind speed forecasts, ultra-short-term speed of the wind estimates require a smaller time span and higher accuracy requirements and can be directly used for real-time dispatching of power systems based on the obtained results, so forecasts of ultra-short-term speed of the wind have significant practical utility.

Currently, there have been numerous articles published both domestically and internationally on the forecasting of ultra-short-term speeds of wind. These publications may be broadly categorized into psychological estimation ways, analysis of time series methods, intelligent prediction approaches, deep learning prediction methods, and combined prediction approaches. Multiple physical parameters are typically used as input information for the physical forecasting approach, such as temperature and humidity, meteorological information, barometric pressure, surface roughness, terrain features, etc. This information is generally obtained through numerical weather prediction (NWP) for longer period wind speed prediction (James et al. 2018). The physical method requires an accurate geophysical model and a large arithmetic power and is inadaptable for ultra-short-term speed of the wind forecast. The principle of analysis of time series approaches is to use the previous moments of the historical data as input to find its internal patterns for prediction. The models mainly include autoregressive models (AR) (Tena García et al. 2019), moving average models (MA) (Wang 2009), autoregressive sliding average models (ARMA) (Jónsdóttir et al. 2018), differential autoregressive sliding average models (ARIMA) (Tyass et al. 2022), and seasonal differential autoregressive sliding average models (SARIMA) (Liu et al. 2021). The time series method requires a relatively high theoretical level for the modeler and is more complex for the modeling itself; the implementation of this method is more constrained solely for the portion of the ultra-short-term speed of the wind forecast where the fluctuation is more uniform because the model itself demands a high degree of smoothness in the initial time series. Intelligent prediction method is to construct a mapping function between independent and dependent variables by learning historical data, and use this relationship for unknown data for prediction, such as support vector machine (SVM) (Li et al. 2020b), least squares support vector machine (LSSVM) (Tian 2020), random forest (RF) (Vassallo et al. 2020), and Gaussian process (GP) (Wang et al. 2020), etc. The advantage of SVM and LSSVM is that they have better prediction results for small data volumes. The drawback is that selecting the kernel characteristics and punitive factors is difficult and necessitates multiple iterations of parameter tuning. RF is quite resilient to noise in

the data and non-significant components without tweaking the structural elements. GP offers excellent generalization properties for non-linear and small sample information, but its disadvantage is that it requires complete sample and feature information. The benefits of models utilizing neural networks in extracting features are gradually being comprehended and implemented in the field of wind speed estimation as machine learning and deep learning advance, such as BP neural network (Liu et al. 2019), extreme learning machine (ELM) (Tan et al. 2020), long short-term memory neural network (LSTM) (Sun et al. 2020), threshold recurrent neural network (GRU) (Gao et al. 2022), convolutional neural network (CNN) (Xu and Wei 2022) and radial basis function neural networks (RBF) (Cui et al. 2019), etc. Through the weights between neurons, the network of neurons prediction method expresses the correlation between the variables that are both independent and dependent, and its parallel structure can handle the problem of real-time well with good fault tolerance. The model can be used with huge samples of information, however it has issues including slow learning and local optimal behavior. Zucatelli et al. (Zucatelli et al. 2019a) study was used to investigate the most effective artificial neural network configuration for predicting wind speed, which is effective in determining the optimal location of the wind turbine and allowing accurate wind speed prediction at different wind measurement heights. Zucatelli et al. (Zucatelli et al. 2019b) study applied different neural network configurations at different locations and heights, performed quantitative analyses and evaluated the statistical results in order to select the configuration that best predicts the real data. The MLP was shown to be the best for short-term wind speed prediction. It has been determined via substantial research that it is challenging to apply a single model for short-term speed of the wind predictions and that the outcome predictions of a single model varied significantly for information on wind speeds from different wind turbines at different and different time periods. Increasing numbers of investigators are concentrating on integrated prediction models due to the single model's constrained predictive power. Combining the benefits of each approach, combined prediction models increase the precision of forecasts and facilitate the selection process. Because these models perform proficiently in the estimation of ultra-short-term wind speeds, they are now widely employed. (Ma et al. 2017). For example, Zucatelli et al. (Zucatelli et al. 2021) proposed a supervised deep learning and wavelet decomposition based wind power prediction method for wind speed prediction at different heights of wind farms. It is shown that the method is satisfactory for predicting wind power and wind power slopes in tropical and subtropical regions of South America. Zucatelli et al. (Zucatelli et al. 2020) proposed a method for predicting wind speed at different heights and locations based on artificial neural network and

wavelet decomposition techniques. The study proves that the best statistical analyses are RNN discrete and Meyer wavelet and the proposed hybrid model (Neural Network Wavelet Packet Decomposition) is capable of predicting wind speed effectively. Tian (Tian 2021) suggested an ultra-short-term speed of the wind modal decomposition forecast approach depending on an adaptive variational modal disintegration and a weighted mixture framework. The empirical research demonstrated that the current approach meets the requirements for multi-step forecasting of ultra-short-term speed of the wind and can increase prediction precision and rapidity of computation. Yang et al. (Yang et al. 2022) established a method to estimate wind speed based on improved singular spectrum analysis (ISSA) data decomposition. The findings of the study demonstrate the effectiveness of the established prediction methodology.

As can be seen from the current state of research above, although progress has been made in each of the prediction models, there are still some problems. Due to the fact that ultra-short-term speed of the wind is variable, susceptible to external factors and fluctuates frequently, it is difficult to use a model to ensure its prediction accuracy, the prediction results of a single model for wind speed data at different time intervals are very different, and the selection of the same model for wind speed data of different complexity will result in a certain waste of resources. Therefore, paper selects the empirical wavelet transform and combined prediction method for modeling.

## Contributions of the paper

In this paper, a joint prediction method based on empirical wavelet transform and combined model is proposed for the problem of grid integration of wind energy, taking into account the variable, susceptible to external factors and fluctuates frequently characteristics of the target sequence. The new model proposed in this paper combines the signal decomposition technique, the emerging intelligent population optimisation algorithm, and the linear and nonlinear prediction methods. In order to avoid wasting resources, this paper chooses to assign different prediction models to the components of the EWT decomposition (an improved SSA-optimised ELM model is assigned to the components with periodic characteristics, and the BiLSTM-GRU model is used for the components with significant non-linearities). In order to avoid the blindness of the traditional method in which the predicted values of each component are directly accumulated, the improved SSA is introduced to optimise the weights of each prediction model as follows.

1. Data disaggregation. The data are decomposed using empirical wavelet transform (EWT) to obtain multiple more stable components.

2. Self-similarity computation and reconstruction. Sample entropy (SE) is used to compute the complexity of the deconstructed EWT components, which is then reassembled in accordance with their complexity.
3. Selection of prediction models for each component. The improved SSA optimized ELM and gated recurrent neural network(GRNN) is selected as prediction models of different components based on the magnitude of the complexity calculated from the sample entropy.
4. Improved intelligent algorithms for optimising parameters. A new approach is put forth to enhance the standard sparrow search (SSA) algorithm. The weight and threshold of extreme learning machine (ELM) are modified nonlinearly according to the number of iterations.
5. Calculation of weight coefficients for each component. The improved SSA is selected to calculate the weight coefficients of the above prediction methods. The optimal weight coefficients acquired by the clever optimization algorithm in accordance with the predictions of each method are multiplied to get the final prediction outcomes.

## Structure of the paper

The framework of this essay is as follows: the background of study and current state of ultra-short-term wind speed anticipation are presented in the first section; the second part describes the methods of EWT, SE, ELM, BiLSTM, GRU, and improved SSA; the third section outlines the data sets that were used and the precise steps of the presented prediction approach; the fourth section uses case studies and results comparison to demonstrate the usefulness of the suggested method; the fifth section summarizes the findings and suggests further research.

## Methodology

### EWT algorithm

Gilles invented the EWT algorithm in 2013 (Gilles 2013). Empirical mode decomposition (EMD) and wavelet transform are used in this technique, combining EMD's adaptive properties with wavelet method's theoretically sound, straightforward, and quick calculation.

By creating adaptive wavelets, the empirical wavelet transform aims at capturing the various single components. The following are the stages of the approach:

For a given signal ( $t$ ), the Fourier transform is first applied to obtain a spectrum  $X(\omega)$ , normalized to the range  $2\pi$ , and according to the Shannon criterion, only the signal properties on  $[0, \pi]$  are discussed during the analysis. Therefore, the  $[0, \pi]$  range is where the Fourier spectrum's supporting

period is identified. The maximum values  $M = \{M_j\}_{j=1,2,3,\dots,N}$  are searched in this range and their values are calculated their corresponding frequencies. where  $N$  represents how many maxima there are. Partition it into  $N$  consecutive intervals and set the boundary of each interval  $v_i$  as:

$$v_i = \frac{\omega_i + \omega_{i+1}}{2} \quad (1)$$

where, there are two frequencies,  $\omega_i$  and  $\omega_{i+1}, v = \{v_i\}_{i=1,2,3,\dots,N}$ .

Create a collection of wavelet filter banks utilizing the concept of Littlewood-Paley and Meyer wavelets, each of which has one low-through filter and  $N-1$  band-through filters. The following two equations showcase the empirical wavelet function  $\varpi_i(\omega)$  and the empirical scale function  $K_1(\omega)$  for  $n > 0$ .

$$\varpi_i(\omega) = \begin{cases} 1, (1 + \xi) \cdot v_i < |\omega| < (1 - \xi) \cdot v_{i+1} \\ \cos\left(\frac{\pi}{2} \cdot \sigma(\xi, v_{i+1})\right), (1 - \xi) \cdot v_{i+1} \leq |\omega| \leq (1 + \xi) \cdot v_{i+1} \\ \sin\left(\frac{\pi}{2} \cdot \sigma(\xi, v_{i+1})\right), (1 - \xi) \cdot v_i \leq |\omega| \leq (1 + \xi) \cdot v_i \\ 0, \text{otherwise} \end{cases} \quad (2)$$

$$K_1(\omega) = \begin{cases} 1, |\omega| \leq (1 - \xi) \cdot v_1 \\ \cos\left(\frac{\pi}{2} \cdot \sigma(\xi, v_1)\right), (1 - \xi) \cdot v_1 < |\omega| \leq (1 + \xi) \cdot v_1 \\ 0, \text{otherwise} \end{cases} \quad (3)$$

where,  $\sigma(\xi, v_1) = \sigma((1/2\xi \cdot v_1) \cdot (|\omega| - (1 - \xi) \cdot v_1))$ , the argument  $\xi$  ensures the difference between two successive transitions,  $\sigma(x) = x^4(35 - 84x + 70x^2 - 20x^3)$ . The inner result of the analytical signal  $f(t)$  and the empirical scale function  $K_1(\omega)$  is expressed as follows for the approximation coefficients  $\theta_f(1, t)$ :

$$\theta_f(1, t) = \langle f, K_1 \rangle = \int f(\tau) \cdot \overline{K_1(\tau - t)} d\tau = F^{-1}[f(\omega) \cdot \overline{K_1(\omega)}] \quad (4)$$

Similarly, the internal product of the initial information  $f(t)$  and the empirical wavelet function  $\varpi_i(\omega)$  stands for the detail coefficients  $\theta_f(i, t)$ :

$$\theta_f(i, t) = \langle f, \varpi_i \rangle = \int f(\tau) \cdot \overline{\varpi_i(\tau - t)} d\tau = F^{-1}[f(\omega) \cdot \overline{\varpi_i(\omega)}] \quad (5)$$

where, the i-th filter's detail coefficient at the t-th time point is symbolized by the symbol  $\varpi_f(i, t)$ .

From this, the original signal can be reconstructed by the following equation The expression is:

$$\begin{aligned} f(t) &= \theta_f(0, t) \cdot K_1(t) + \sum_{n=1}^N \theta_f(n, t) \cdot \varpi_n(t) \\ &= (\theta_f(0, \omega) \cdot K_1(\omega) + \sum_{n=1}^W \theta_f(n, \omega) \cdot \varpi_n(\omega)) \end{aligned} \quad (6)$$

where, the Fourier transformation's forms of  $\theta_f(0, t), \theta_f(n, t)$ , respectively, are  $\theta_f(0, \omega), \theta_f(n, \omega)$ .

Following EWT processing, the signal  $f(t)$  is divided into single-component AM-AF sections, ranging in frequency from low to high:

$$\begin{aligned} f_0(t) &= \theta_f(0, t) \cdot K_1(t) \\ f_1(t) &= \theta_f(n, t) \cdot K_n(t) \end{aligned} \quad (7)$$

The Fourier spectrum is divided to distinguish the various spectrum regions corresponding to the approaches. The Fourier magnitudes must be evaluated to identify their local maxima, which are then ordered in descending order ( $0, \pi$  are excluded) in order to find  $N+1$  borders. Consider a scenario in which the algorithm identifies  $M$  local maxima  $M_j (j = 1, \dots, M)$  and  $M_j$  satisfying:  $M_j > M_M + \alpha \cdot (M_1 - M_M)$  (not the right-hand side of the equation is referred to as the barrier), where  $\alpha$  stands for the relative amplitude relation. Two scenarios could occur:

- 1)  $M \geq N$ , after taking the first  $N-1$  maxima;
- 2)  $M < N$ , if there are fewer approaches in the signal than in the ideal scenario, all maximum values that were identified are retained, and  $N$  is set back to the suitable amount.

The borders of each segment are defined as the local maxima that are greater than the threshold value, and  $N$  is the total amount of these boundaries. By adding 0 and  $\pi$  to this set of boundaries,  $N+1$  boundaries are produced.

## SE algorithm

The SE indicator gauges a time series' sophistication. Inversely, the level of sophistication of the sequence increases with increasing sample entropy value (Ding et al. 2021). In this paper, we use SE to measure the sophistication of each subsequence after decomposition to provide a basis for the reconstruction of the sequence. The specific steps of sample entropy are as follows:

**Step 1.** For a given sequence  $S = \{s(m), m = 1, 2, 3, \dots, M\}$ , contains  $M$  data points.

**Step 2.** Given the embedding dimension  $n$ , the reconfiguration vector expression for the initial sequence  $S$  is:

$$S_i^n = \{s_i^n(i), s_i^n(i+1), \dots, s_i^n(i+n-1)\} \quad (8)$$

where,  $i = 1, 2, 3, \dots, M-n+1$ .

**Step 3.** The biggest absolute difference between the respective components of both vectors  $S_i^n$  and  $S_j^n$  is used to define the distance between them, and its equation is:

$$d[S_i^n, S_j^n] = \max_{k=1,2,\dots,n} (|S_i^n(k) - S_j^n(k)|) \quad (9)$$

**Step 4.** The allowable error  $r$  ( $r > 0$ ) is selected, and its ratio to the number of vectors  $M-n$  is calculated based on the number of vectors  $S_i^n$  and  $S_j^n$  whose distance  $d[S_i^n, S_j^n]$  is less than  $r$ , denoted as  $C_i$ , i. e.:

$$C_i^n(r) = \frac{C_i}{M-n} \quad (10)$$

**Step 5.** For all the  $C_i^n(r)$  accumulate and find the mean value, i.e.:

$$B^n(r) = \frac{1}{M-n+1} \sum_{i=1}^{M-n+1} C_i^n(r) \quad (11)$$

**Step 6.** Add  $n$  to 1 and continue with the above steps;

**Step 7.** With a restricted amount of initial signals, SE is expressed as

$$\text{Sample Entropy}(m, r, M) = -\ln \frac{B^{n+1}(r)}{B^n(r)} \quad (12)$$

## ELM algorithm

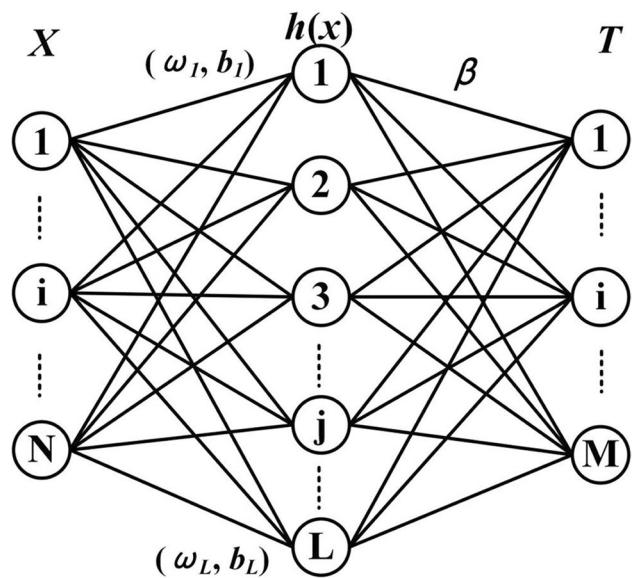
ELM is a neural network technique for single hidden layer feedforward neural networks (SLFN) (Hua et al. 2022). The most crucial characteristic is that the implicit nodes' input weights and biases are produced at random within a predetermined range.

The output layer weights  $\beta$  are the most important objective of training. ELM is frequently employed to solve problems including regression, categorization, clustering, the learning of features, and other matters resulting from its merits of excellent learning efficiency and outstanding generalization capacity. The input layers, the implied layers, and the output layers of the three-layer categorized neural network that makes up the ELM's network topology in Fig. 1 are all fully concatenated to one another.

According to this precise procedure:

**Step 1.** The trained set, which is made up of the beginning  $N$  sequence of samples  $x_1, x_2, \dots, x_N$ , is made up of the sequences  $(x_1, t_1), (x_2, t_2), (x_3, t_3), \dots, (x_N, t_N)$ , with  $x_i = [x_i, x_{i+1}, \dots, x_{i+n-1}]$  being chosen as the data input and  $t_i = x_{i+n}$  being selected as its output, where  $k = N - n > L$ .

**Step 2.** Count the assumed implied level neurons  $L$ , calculate the function of stimulation  $f(x)$ , and create at arbitrary the input weights  $\omega_j$  and the implied level thresholds  $\beta_j$ ,  $1 \leq j \leq L$ . It is possible to state the relationship between ELM input and output as follows:



**Fig. 1** The network structure of ELM

$$t_i = \sum_{j=1}^L \beta_j f(\omega_j \cdot x_i + b_j), i = 1, 2, \dots, k \quad (13)$$

The above Eq. (13) can be rewritten as:

$$T_k = H_k \beta_k \quad (14)$$

where, the neuron matrix is represented by  $H_k$ , the output weight matrix by  $\beta_k$ , and the output layer matrix by  $T_k$ .

**Step 3.** The neuron matrix  $H_k$  be Calculated.

**Step 4.** Apply Eq. (15) to the output weight  $\beta_k$  estimations.

$$\beta_k = (H_k^T H_k)^{-1} H_k^T T_k \quad (15)$$

## GRU algorithm

The GRU neural network was proposed by Cho Kyunghyun in 2014 and the Long Short Term Memory neural network algorithm model, which both belong to the class of recurrent neural network techniques, is the model's forerunner (Wang and Gui 2022). The algorithm has many features, the most prominent of which is the circumvention of the gradient disappearance problem in the RNN algorithm. The LSTM algorithm is presented by means of gates as a way to promote the validity of the input data, and subsequently completes the adjustment of the neuron unit state, including forgetting gates, output gates, and input gates. The GRU neural network further adjusts the concept of gates to improve the design ideas of input gates and forgetting gates, using simply the reset gate and update gate structures, which are utilized to govern the ignoring of information from the

previous instant and the substitution of information from the previous moment into the current moment, which saves working memory and improves the training speed compared to LSTM. Figure 2 displays the model fundamental diagram.

The GRU neural network's neurons are mathematically represented as follows:

$$U_t = \psi(P_U \cdot [I_{t-1}, x_t]) \quad (16)$$

$$R_t = \psi(P_R \cdot [I_{t-1}, x_t]) \quad (17)$$

$$I_t' = \tanh(P \cdot [R_t^* I_{t-1}, x_t]) \quad (18)$$

$$I_t = (1 - U_t)^* I_{t-1} + U_t^* I_t' \quad (19)$$

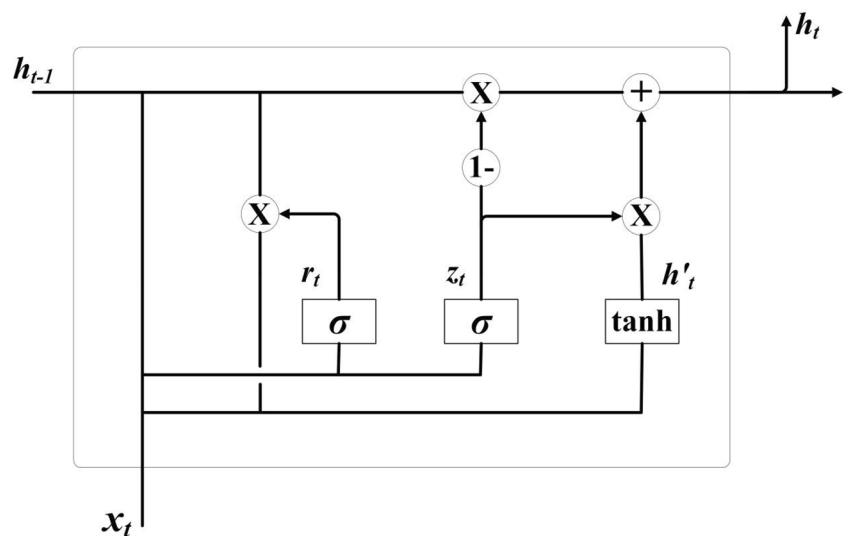
where, the subscript  $t$  represents the moment  $t$ , the update gate is referred to by  $U$ , and the reset gate is referred to by

$R$ . The weight matrix is represented by  $P$ , and the output value is represented by the output value is indicated by the sign  $I_*$ , and the sigmoid function is suggested by the sign  $\psi$ .

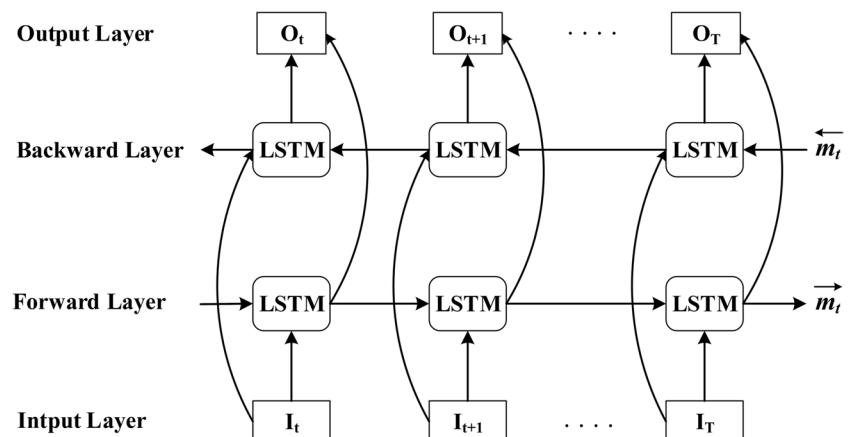
### BILSTM algorithm

The LSTM network is a RNN that solves the loss of information over long distances for long sequences and can memorize valuable information (Sun et al. 2020). The BiLSTM neural network framework model is divided into 2 independent LSTMs, and the sets of inputs are supplied to the two LSTM neural networks in advance and reverse sequence for obtaining features, which fully takes into account the relationship between the front and back of the time series information, and for data with high correlation and with strong periodic changes, the temporal characteristics of the signal can be extracted effectively (Zhao et al. 2022). The structure is shown in Fig. 3.

**Fig. 2** The network structure of GRU



**Fig. 3** The fundamental of BiLSTM



The specific formula of BiLSTM is shown below:

$$\overrightarrow{m}_t = f(\overrightarrow{\omega}_i \cdot I_t + \overrightarrow{m}_{t-1} \cdot \overrightarrow{\omega}_m) \quad (20)$$

$$\overleftarrow{m}_t = f(\overleftarrow{\omega}_i \cdot I_t + \overleftarrow{m}_{t-1} \cdot \overleftarrow{\omega}_m) \quad (21)$$

$$O_t = f(\overrightarrow{\omega}_o \cdot \overrightarrow{m}_t + \overleftarrow{\omega}_o \cdot \overleftarrow{m}_t) \quad (22)$$

where,  $f$  is the activation function,  $\omega$  is the weight and threshold term,  $I_t$  is the input term,  $\overrightarrow{m}_t$ ,  $\overleftarrow{m}_t$  are the outputs indicating the forward and inverse transmission layers, and  $O_t$  is the final output term.

## SSA algorithm

The SSA intelligent optimization technique was introduced by Jiankai Xue et al. in 2020. (Xue and Shen 2020<sup>1</sup>) The total sparrow population is divided into three main parts: discoverers, joiners, and distress responders. The discoverer is in charge of feeding the populace and informing everyone else about where to go foraging and how to get there. To get food, joiners use discoverers. Both discoverers and joiners can be distress responders, constantly adjusting their position for a safer location when attacked by predators.

**A. Initializing the population** Let the sparrow population consist of  $n$  sparrows, and if the dimension of the issue of the optimization variable is  $d$ : variable is  $d$ , then the total sparrow population  $X$  can be expressed as:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,d} \end{bmatrix} \quad (23)$$

where, the total amount of sparrows is indicated by  $n$ , and the dimension of the problem that needs to be solved is indicated by  $d$ .

**B. Discoverer location update** The discoverer is in charge of locating food for the entire community and giving the joiners information on where to obtain food, with a position update formula of:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \times e^{\left(\frac{-i}{K_1 \times \theta}\right)}, K_2 < K_3 \\ X_{i,j}^t + q \times L, K_2 \geq K_3 \end{cases} \quad (24)$$

where, the variables  $t$  are indicative of the current iteration count;  $i$  for the  $i$ -th sparrow,  $i = 1, 2, \dots, n$ ;  $j$  for the dimension,  $j = 1, 2, \dots, d$ ;  $\theta$  for the maximum iteration count; and  $X_{i,j}^{t+1}$  for the location data of the  $i$ -th sparrow in the  $j$ -th dimension;  $K_1$  is an arbitrary number among 0 and 1 that

follows a homogeneous distribution;  $K_2$  is also an unplanned number between 0 and 1 that follows a homogeneous distribution, indicating the warning value;  $K_3$  indicates the safety value and is an arbitrary number between 0.5 and 1 that respects a homogeneous distribution;  $q$  is an arbitrary number that respects a homogeneous distribution; A  $1 \times d$  matrix with all of its numbers being 1 is represented by the letter  $L$ . When  $K_2 < K_3$ , it demonstrates that the sparrow population is relatively safe from predators and the finders can search for a wider range of food. When  $K_2 \geq K_3$ , it indicates that portion of the population's sparrows discovered the predator invasion and alerted their fellow sparrows. As a result, the population of sparrows will swiftly fly to a safe location to forage.

**C. Joiner location update** All sparrows in the population, with the exception of the discoverer, are joiners. The joiners will quickly depart their current place and battle for food when the discoverer finds greater food. The position update formula is:

$$X_{i,j}^{t+1} = \begin{cases} q \cdot e^{\left(\frac{x_w - X_{i,j}^t}{l^2}\right)}, i > n/2 \\ X_o^{t+1} + \left|X_{i,j}^t - X_o^{t+1}\right| \cdot B^+ \cdot L, \text{ else} \end{cases} \quad (25)$$

where,  $i$  stands for the  $i$ -th joiner;  $n$  represents the overall amount of sparrows;  $j$  denotes the joiner's dimension;  $X_o^t$  is the ideal position for the finder at that iteration; and  $X_w$  symbolizes the worst sparrow; a matrix having entries of 1 or -1 with dimensions  $d \times 1$  and  $B^+ = B(B^T B)^{-1}$  is denoted by the letter  $B$ . When  $i > n/2$ , the worst positioned of the sparrows needs to fly away from its location to look for food elsewhere.

**D. Distress responder location update** Distress responders are vulnerable to predator attacks at the periphery of the population and will constantly adjust their position when they perceive danger coming. The equation for updating the position of distress responders is:

$$X_{i,j}^{t+1} = \begin{cases} X_b^t + \lambda \left| X_{i,j}^t - X_b^t \right|, h_i > h_g \\ X_{i,j}^t + \theta \left( \frac{|X_{i,j}^t - X_w^t|}{(h_i - h_w) + \sigma} \right), h_i = h_g \end{cases} \quad (26)$$

where,  $\lambda$  is a Gaussian-distributed arbitrary number that complies with a mean and variance of 0 and 1 as the step control parameter;  $X_b$  is the current global ideal location. The path of the sparrow's maneuvering and the step control parameter are represented by the random number  $\theta$ , which ranges from -1 to 1.  $h_w$  is the current global minimum fitness value;  $h_i$  symbolizes the present  $i$ -th sparrow's fitness amount, while  $h_g$  signifies the present world maximum

fitness value;  $\sigma$  is a very small constant that prevents the denominator from being zero. When  $h_i > h_g$ , the sparrow occupies the outermost of all sparrow positions and is readily found and eaten by predators; when  $h_i = h_g$ , the sparrow in the center location will sense the danger approaching and has to shift nearer to other sparrows to reduce the risk of being predated.

### Improved SSA algorithm

A fresh intelligent swarm optimization approach is called SSA. In comparison to other optimization methods like the bald eagle search algorithm (BES) or the PSO, SSA offers the benefits of quick convergence, quick iteration, high extreme value discovery accuracy, and high solution efficiency. The normal SSA, however, is constrained by the local extremum problem (Tian and Chen 2021). An improved SSA is thus offered according to the aforementioned issues.

**A. Piecewise chaotic mapping initializes the population** Since SSA generates the initial population randomly, it has uneven distribution results and greatly affects the final merit search results. Piecewise chaotic mapping is used for initializing the population as a solution to this issue, and it can significantly enhance the population's initial distribution. The specific formula is:

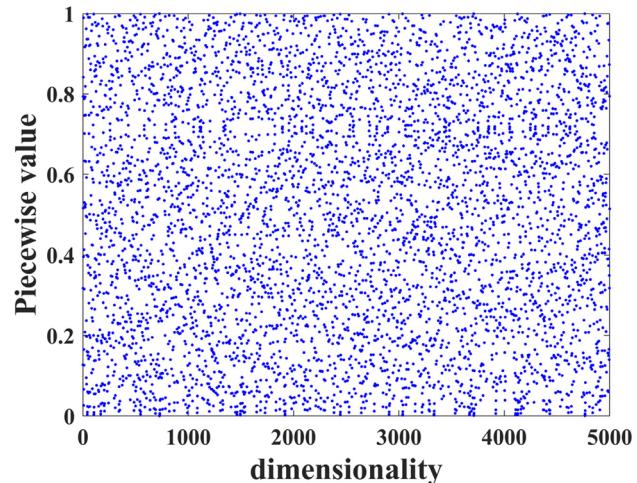
$$x(j+1) = \begin{cases} \frac{x(j)}{e}, & 0 \leq x(j) < e \\ \frac{x(j)-e}{-(e-0.5)}, & e \leq x(j) < 0.5 \\ \frac{-(e-1)-x(t)}{-(e-0.5)}, & 0.5 \leq x(j) < -(e-1) \\ \frac{1-x(t)}{e}, & -(e-1) \leq x(j) < 1 \end{cases} \quad (27)$$

where,  $e$  represents a constant of 0.4;  $j$  symbolizes each sparrow's  $j$ -th dimension. Its Piecewise chaotic mapping is shown in Fig. 4.

Knowing f from Fig. 4 that Piecewise chaotic mapping can distribute the sparrow population evenly in each location of the predefined area, which enhances the population diversity and improves the merit-seeking ability of SSA.

**B. Improved discoverer location updates** Standard SSA has a greater likelihood of falling into the optimal solution that is local in the process of finding the best solution. To improve the algorithm's ability to search globally, the finder's position, the internal improvement of its exponential function, and the removal of random numbers, the improved formula:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(-15 \cdot \left(\frac{i}{2^t}\right)^3\right), & K_2 < K_3 \\ X_{i,j}^t + Q \cdot L, & K_2 \geq K_3 \end{cases} \quad (28)$$



**Fig. 4** Piecewise chaos mapping diagram

The improved update of the discoverer can provide the discoverer with excellent global search capacity in its preliminary iteration, which can enhance the capacity of the original SSA to jump outside the local optimal location and find the global optimal compared to the uniform distribution.

**C. Add optimal value search** The standard SSA is more likely to jump into local optimum, search around optimal value after each iteration, and if a better value is found it can be replaced, and the improved formula is:

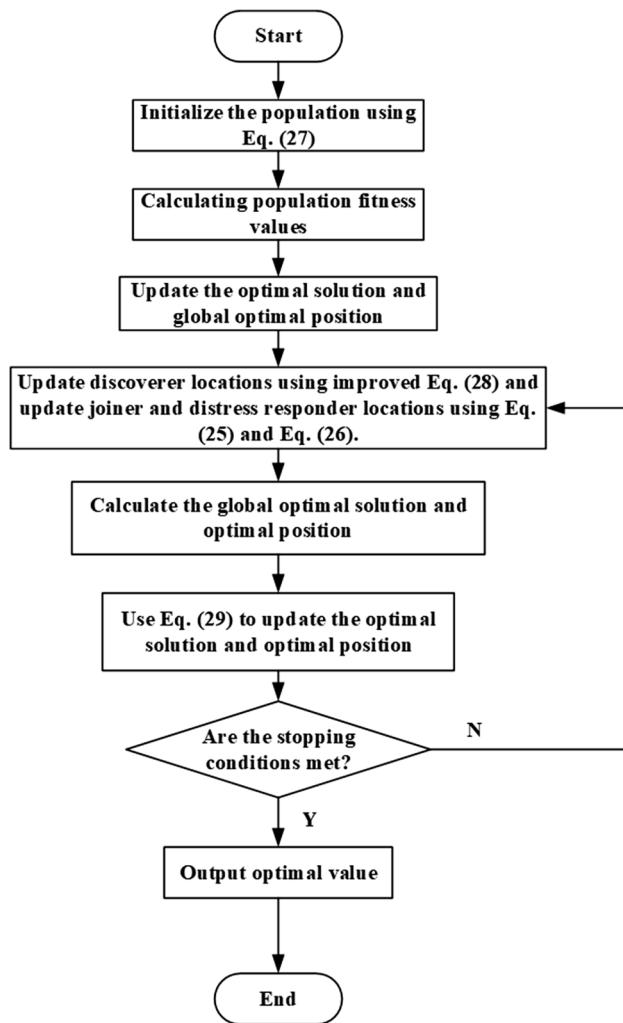
$$X_g^{t+1} = \begin{cases} X_g^t \cdot \left(1 + 0.08 \cdot \sin\left(\frac{t}{T}\right)\right), & F_i > F_g \\ X_g^t, & F_i = F_g \end{cases} \quad (29)$$

where, the present global ideal position is at  $X_g$ , and  $t$  is the amount of iterations.

When  $F_i > F_g$ , a more optimal position is found and the position is updated; when  $F_i = F_g$ , the position remains unchanged. In order for SSA to point to the global optimum, adding the ideal position search ensures a strong preliminary global search performance and enhances the late local search effectiveness. Figure 5 depicts the flow diagram of the improved sparrow search method.

### Performance analysis of improved SSA algorithm

With the goal of verifying the effectiveness of the improved SSA, four benchmark functions in CEC2017 were chosen as test functions: the unimodal function Shifted and Rotated Bent Cigar Function, the multimodal function Shifted and Rotated Lunacek Bi\_Rastrigin Function, mixed function Hybrid Function 1 ( $N=3$ ) and the combined function



**Fig. 5** Flow diagram of the improved sparrow search algorithm

Composition Function 2 ( $N=3$ ), whose expressions are given in Eq. (30), Eq. (31), Eq. (32), and Eq. (33).

$$\begin{aligned} F_1(x) &= f(M(x - o_1)) + F_1^* \\ f(x) &= x_1^2 + 10^6 \sum_{i=2}^D x_i^2 \end{aligned} \quad (30)$$

$$\begin{aligned} F_2(x) &= f\left(M\left(\frac{600(x-0)}{100}\right)\right) + F_2^* \\ f(x) &= \min\left(\sum_{i=1}^D (\hat{x}_i - \mu_0)^2, dD + s \sum_{i=1}^D (x_i - \mu)^2\right) + 10\left(D - \sum_{i=1}^D \cos(2\pi\hat{x}_i)\right) \\ \mu_0 &= 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20-8.2}}, d = 1 \\ y &= \frac{10(x-0)}{100}, \hat{x}_i = 2\text{sign}(x_i^*)y_i + \mu_0, \text{for } i = 1, 2, \dots, D \\ z &= \Lambda^{100}(\hat{x} - \mu_0) \end{aligned} \quad (31)$$

$$\begin{aligned} F_3 &= (x)g_1(M_1z_1) + g_2(M_2z_2) + \dots + g_N(M_Nz_N) + F_3^*(x) \\ z_N &= \left[y_{S_{\sum_{i=0}^{N-1} n_i+1}}, y_{S_{\sum_{i=0}^{N-1} n_i+1}}, \dots, y_{S_D}\right] \\ y &= x - 0_i, S = \text{randperm}(1 : D) \\ n_i &= p_i^*D \end{aligned} \quad (32)$$

where,  $N=3$ ,  $p=[0.2, 0.4, 0.4]$ ,  $g_1$ : Zakharov function,  $g_2$ : Rosenbrock function,  $g_3$ : Rastrigin function.

$$\begin{aligned} F_4(x) &= \sum_{i=1}^N \left\{ \omega_i^* [\lambda_i g_i(x) + \text{bias}_i] \right\} + F_4^* \\ w_i &= \frac{1}{\sqrt{\sum_{j=1}^D (x_j - o_{ij})^2}} \exp\left(-\frac{\sum_{j=1}^D (x_j - o_{ij})^2}{2D\sigma_i^2}\right) \\ \omega_i &= w_i / \sum_{i=1}^N w_i \end{aligned} \quad (33)$$

where,  $N=3, \sigma=[10, 20, 30], \lambda=[1, 10, 1], \text{bias}=[0, 100, 200]$ ,  $g_1$ : Rastrigin function,  $g_2$ : Griewank function,  $g_3$ : Modified Schwefel function.

In this paper, the improved sparrow search algorithm (Improved SSA), rat swarm search algorithm (RSO), bald eagle search algorithm (BES) and sparrow search algorithm (SSA) are used for comparison. The dimensions, domain of variable and global optimum values of the four test functions are given in Table 1. The parameters settings for the aforementioned methods are shown in Table 2.

To demonstrate the stability of the method and lower the inaccuracy of the experimental optimization outcomes, each algorithm was executed independently 35 times for the two test functions, taking into account the likelihood of the results acquired from a single calculation. The average, standard deviation and best fitness of each run were recorded, and Table 3 exhibitions the outcomes of the average of the 35 runs, which shows that the improved sparrow optimization algorithm has the best results for the four test functions.

To show the convergence speed of the improved SSA, the convergence curves of the 4 optimization algorithms for the 4 test functions (a for Shifted and Rotated Bent Cigar Function, b for Shifted and Rotated Lunacek Bi\_Rastrigin, c for Hybrid Function 1 ( $N=3$ ) and d for Composition Function 2 ( $N=3$ )) are shown in Fig. 6. Overall, the improved SSA has a faster convergence rate than the other methods.

### Improved SSA algorithm to optimize ELM

The weights  $\omega$  and bias  $b$  of the ELM layer of input are used as the optimization objectives of the improved SSA. In this article, the fitness function of the anticipated and true values is MSE, and its expression is as follows:

$$F = \frac{1}{D} \sum_{j=1}^D (L_j - \hat{L}_j)^2 \quad (34)$$

where,  $L_j$  is the actual value,  $\hat{L}_j$  is the forecast value, and  $D$  is the total amount of ultra-short-term speed of the wind data.

The following is a description of the optimization process:

**Step 1.** Initialization parameters for the improved SSA: number of populations  $n$ , proportion of discoverers  $P$ , proportion of distress responders  $D$ , maximum number of iterations  $T$ , and distress threshold  $K_2$ .

**Table 1** The four test functions' parameters

Function	Dimensions	Domain of variable	Global optimum values
Shifted and Rotated Bent Cigar	30	[-100 100]	0
Shifted and Rotated Lunacek Bi_Rastrigin	30	[-100 100]	0
Hybrid Function 1 (N=3)	30	[-100 100]	0
Composition Function 2 (N=3)	30	[-100 100]	0

**Table 2** The four algorithms' parameters

Method	Parameters
Improved SSA	Number of iterations: 200; Population size: 50; weight of joiners: 0.6; weight of distress responders: 0.1; $K_2=0.8$ ; $e=0.4$
SSA	Number of iterations: 200; Population size: 50; weight of joiners: 0.6; weight of distress responders: 0.1; $K_2=0.8$
BES	Number of iterations: 200; Population size: 50; $\alpha=2$ ; $a=10$ ; $R=1.5$
RSO	Number of iterations: 200; Population size: 50

**Table 3** Comparison results of improved SSA and other three calculations

Function	Algorithm	Average value	Standard deviation	Best fitness
Shifted and Rotated Bent Cigar	RSO	4.24E+10	1.53E+10	3.21E+10
	BES	5.73E+10	6.68E+09	4.99E+10
	SSA	4.45E+09	7.11E+09	5.53E+08
	Improved SSA	<b>3.01E+09</b>	<b>7.06E+09</b>	<b>1.04E+08</b>
Shifted and Rotated Lunacek Bi_Rastrigin	RSO	792.13	27.62	770.01
	BES	795.78	17.35	792.12
	SSA	794.16	12.68	792.58
	Improved SSA	<b>740.29</b>	<b>11.67</b>	<b>726.93</b>
Hybrid Function 1 (N=3)	RSO	1.05E+10	4.36E+09	5.84E+09
	BES	2.19E+10	2.61E+09	1.87E+10
	SSA	1.51E+10	1.71E+09	1.47E+10
	Improved SSA	<b>5.05E+08</b>	<b>3.78E+08</b>	<b>1.39E+08</b>
Composition Function 2 (N=3)	RSO	5577.46	515.31	5200.02
	BES	3008.27	614.56	2630.49
	SSA	2802.79	235.44	2677.12
	Improved SSA	<b>2582.02</b>	<b>184.05</b>	<b>2454.13</b>

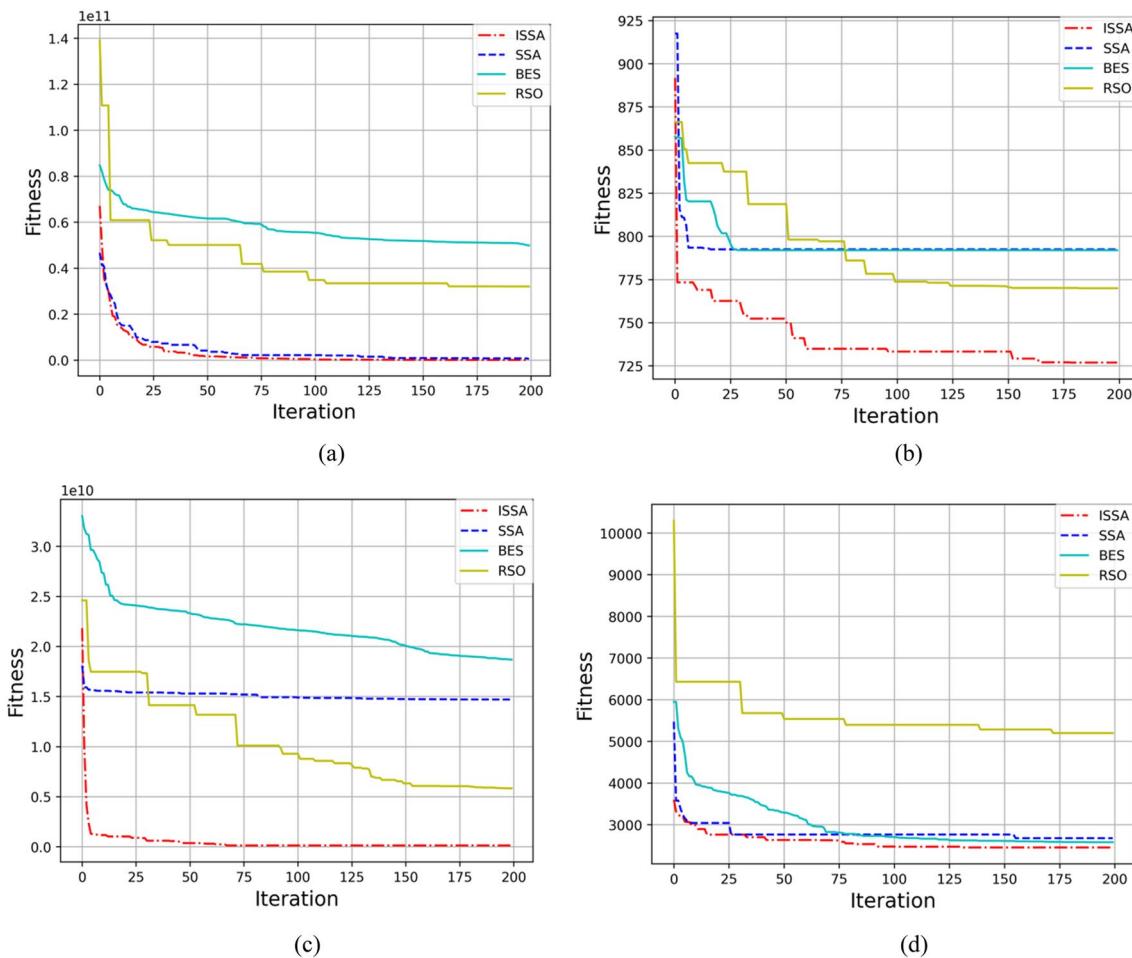
**Step 2.** The optimized object is chosen as the input weight  $\omega$  and bias  $b$  of ELM, which is then regarded as a sparrow individual. Specify the variation region of  $\omega$  and  $b$ . Initialize the improved SSA algorithm. The iteration starts with the number of iterations  $t=1$ .

**Step 3.** Based on the current sparrow position to get  $\omega$  and  $b$  as parameters of ELM, the predicted value of ELM was calculated by Eq. (13). The sparrow population fitness value was calculated by Eq. (34) and ranked, and the position of the sparrow with the smallest and largest fitness value was recorded. The proportion of the top P

was used as discoverer and the subsequent ones were used as joiners and the positions were updated by Eq. (28) and Eq. (25).

**Step 4.** The randomly selected safety value for the distress responder is updated by Eq. (26) for the location. And the optimal solution and the optimal position are updated using Eq. (29).

**Step 5.** Check to see if the algorithm's end condition is satisfied; if not, return to **Step 3** with  $t=t+1$ . The method terminates and the optimal values of  $\omega$  and  $b$  are output if it is satisfied.



**Fig. 6** Average fitness values of the four test functions

## Dataset description and prediction methods

### Dataset description

The Manjing wind farm, located in Shangyi County, Zhangjiakou City, Hebei Province, China, provided the data utilized for this article. Three ultra-short-term speed of the wind datasets were assigned as the data used for testing the model that was created. The first dataset using a sampling interval of 5 min is referred to as dataset a, and the second dataset alongside a sampling interval of 15 min is referred to as dataset b. Both datasets a and b contain 1000 samples. The third dataset with 2000 samples at 5-min sampling intervals is called dataset c. The specific data are shown in Fig. 7. For each dataset, the first 80% of the samples are generally selected as the training data for the model, and the last 20% of the samples are used as the test data.

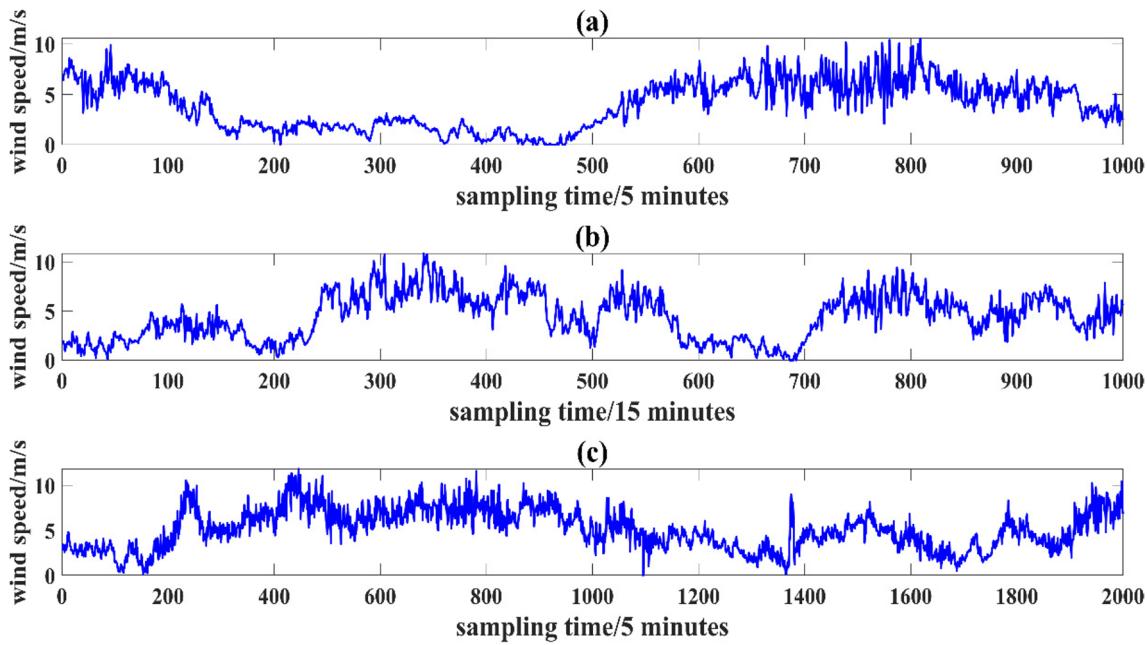
### Data pre-processing

It is required to normalize the data for the purpose of speeding up the operation of the framework because the chosen data contains a range of gaps. In this article, we decided to map the data values to a normal distribution with an average of 0 and a standard deviation of 1 using the Z-score method. The transformation formula is:

$$\tilde{a} = \frac{a - \eta}{\rho} \quad (35)$$

where, the sample data average is represented by  $\eta$ , and the standard deviation by  $\rho$ .

As a result of the features of ultra-short-term speed of the wind, which is variable, easily influenced by external factors, and fluctuates frequently, the For the speed sequence of the ultra-short-term wind after EWT decomposition, it



**Fig. 7** Ultra short-term wind speed datasets

is a difficult task to choose a suitable prediction model for them. incorporates a large amount of complex information. The empirical wavelet transform has high resolution, can accurately and independently decompose each component under weak noise without over-decomposition and avoiding frequency mixing, has good multi-resolution characteristics, adaptive and band separation characteristics, and can express each component of the data more completely, which efficiently enhances the clarity and accuracy of decomposition. Since several modal components can be obtained by decomposing the ultra-short-term wind speed data, the empirical wavelet transform is chosen. The decomposition diagram is shown in Fig. 8. ((A) is the decomposed component of dataset a, (B) is the decomposed component of dataset b and (C) is the decomposed component of dataset c.)

For the speed sequence of the ultra-short-term wind after EWT decomposition, it is a difficult task to choose a suitable prediction model for them. Since sample entropy is often used in time series complexity analysis and fault pathology diagnosis, it can measure the time series complexity. Therefore, this research selects the SE to reconstruct the components and determines the forecasting model of every component after reconstruction. Every component's SE is solved. The  $m$  value is assumed to be 2, and the  $r$  value is assumed to be 0.2 times the modal series' standard deviation. Table 4 presents the outcomes.

It can be seen from the table that several components have very similar complexity, indicating that they contain similar features within them that can be reconstructed. Select 0.05 as the threshold value, and the difference between the two

components less than the threshold value is reconstructed, and the sample entropy size after reconstruction is shown in Table 5.

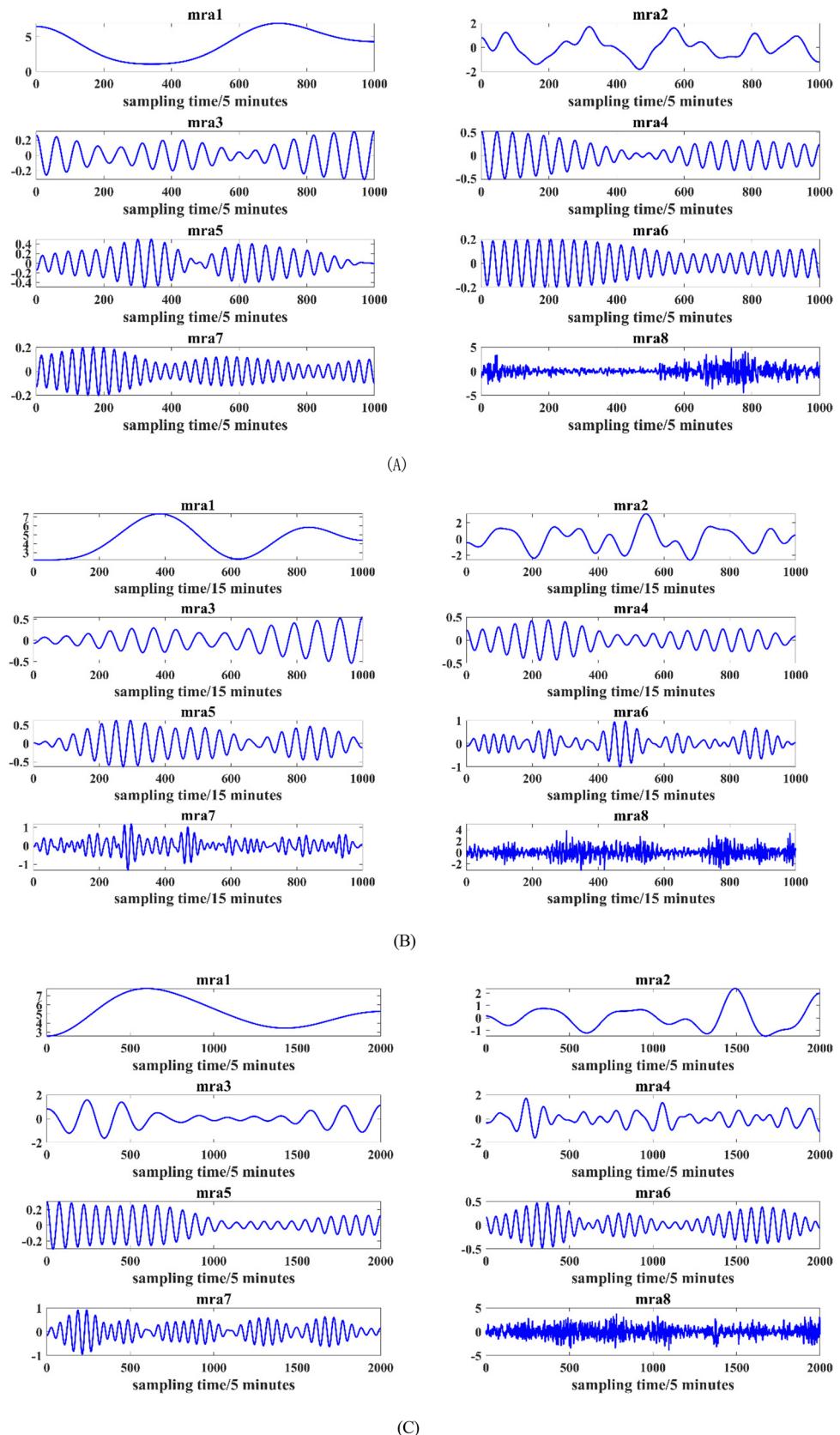
Table 5 demonstrates that the dataset's sample entropy values are spread between 0 and 2. It can be seen from Table 5 and Fig. 8 that for dataset a, mra1, mra2, mra3 and mra4 have low sample entropy values, indicating that their relative complexity is low i.e., they have strong self-similarity and not particularly obvious non-linear characteristics. mra5 has high sample entropy values, indicating that it has very weak self-similarity i.e. high complexity and obvious randomness and non-linear characteristics. According to the complexity and nonlinearity of every component, the components with sample entropy values below 0.5 are selected for prediction by the SSA-optimized extreme learning machine, and those with sample entropy values above 0.5 are selected for prediction by BILSTM-GRU with strong nonlinear regression capability.

### Specific processes in a forecasting approach

The precise steps of the forecasting approach suggested in this article, together with the introduction mentioned above, are as follows.

**Step 1.** Data preprocessing. The initial speed sequence of the ultra-short-term wind is assumed to be of length  $m$ , which is normalized. The original data is divided into  $N$  mra components using the EWT approach.

**Fig. 8** EWT decomposition diagram



(C)

**Table 4** Sample entropy of every component of data sets a, b and c

Dataset	Component	Sample entropy	Dataset	Component	Sample entropy	Dataset	Component	Sample entropy
Dataset a	mra1	0.0179	Dataset b	mra1	0.0257	Dataset c	mra1	0.00996
	mra2	0.1254		mra2	0.1359		mra2	0.03527
	mra3	0.4351		mra3	0.3667		mra3	0.07796
	mra4	0.5017		mra4	0.4531		mra4	0.17809
	mra5	0.5012		mra5	0.5051		mra5	0.28797
	mra6	0.4971		mra6	0.4938		mra6	0.39307
	mra7	0.5221		mra7	0.5757		mra7	0.44642
	mra8	1.3085		mra8	1.8086		mra8	1.77449

**Table 5** Sample entropy of each component after reconstruction of data sets a, b and c

Dataset	Component	Sample entropy	Dataset	Component	Sample entropy	Dataset	Component	Sample entropy
Dataset a	mra1	0.0179	Dataset b	mra1	0.0257	Dataset c	mra1	0.03618
	mra2	0.1254		mra2	0.1359		mra2	0.17809
	mra3	0.4351		mra3	0.4139		mra3	0.28797
	mra4	0.4946		mra4	0.4946		mra4	0.68104
	mra5	1.3084		mra5	0.5757		mra5	1.77449
				mra6	1.8089			

**Step 2.** The SE algorithm calculates the complexity for every of its components. The similarity between the elements is compared and the  $G$  mra components are obtained by reconstructing for those with close similarity.

**Step 3.** According to the different sample entropy values of the reconstructed components, the improved SSA algorithm to optimize ELM and BiLSTM-GRU are selected as the prediction models for the corresponding components.

**Step 4.** The corresponding components are brought into the selected prediction approach to test its predictive capacity. Retrain if it doesn't fulfill the requirements. Otherwise, output the model-related parameters for **step 5**.

**Step 5.** Initialize the algorithm's internal parameters at random, select  $\omega_1, \omega_2, \dots, \omega_G$  as the enhanced SSA algorithm's initial value, and select MSE as its fitness function using the following expressions:

$$\text{fitness} = \frac{1}{M} \sum_{j=1}^M (\kappa_j - \hat{\kappa}_j)^2 \quad (36)$$

where,  $M$  represents the length of the initial speed dataset of the ultrashort-term wind,  $\kappa_j$  is the actual value of the ultrashort-term speed of the wind, and  $\hat{\kappa}_j$  is the forecast value. The formula for  $\kappa_j$  is given in Eq. (37).

$$\hat{\kappa}_j = \sum_{i=1}^G \omega_i \cdot y_i(j) \quad (37)$$

where,  $\omega_i$  indicates every forecast algorithm's weight,  $y_i(j)$  remains the model prediction value, and  $G$  serves as the amount of mra elements.

**Step 6.** The improved SSA algorithm is used for multiple iterations of each model's weight  $\omega_1, \omega_2, \dots, \omega_G$  to discover the perfect solution, and when the fitness function reaches infinity, the output corresponding to the sparrow position is the optimal solution.

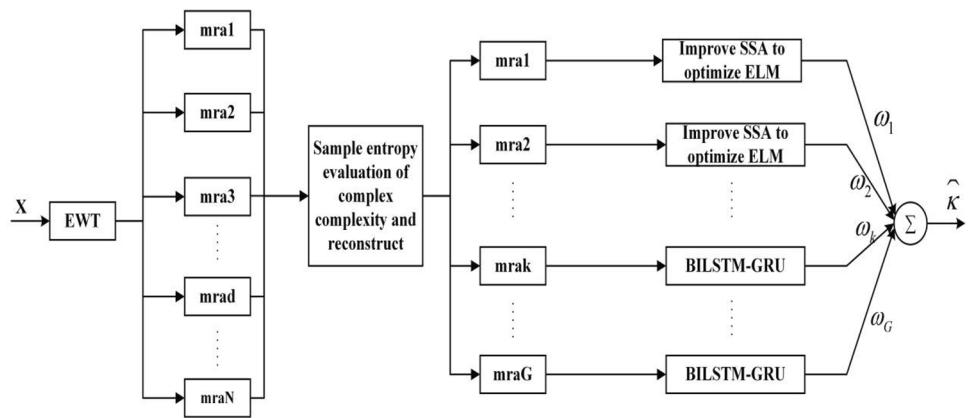
**Step 7.** The parameters and weights of each prediction model obtained in the above steps are utilized with the test set data to evaluate the suggested approach. In summary, the flowchart of the prediction approach proposed in this article is displayed in Fig. 9.

## Analysis of experimental results

### Assessment criteria

Five assessment criteria: mean absolute error (MAE), mean square error (MSE), mean absolute percentile error (MAPE), root mean square error (RMSE), and coefficient of determination ( $R^2$ ), were selected to assess the viability

**Fig. 9** Flowchart of the proposed forecasting approach



of the proposed prediction approach (Li et al. 2023; Wang et al. 2022).

#### 1. Mean absolute error (MAE)

$$MAE = \frac{1}{M} \sum_{j=0}^M |\kappa(j) - \hat{\kappa}(j)| \quad (38)$$

#### 2. Mean square error (MSE)

$$MSE = \frac{1}{M} \sum_{j=0}^M (\kappa(j) - \hat{\kappa}(j))^2 \quad (39)$$

#### 3. Mean absolute percentile error (MAPE)

$$MAPE = \frac{1}{M} \sum_{j=1}^M |\kappa(j) - \hat{\kappa}(j)| \cdot 100/\kappa(j) \quad (40)$$

#### 4. Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{M} \sum_{j=0}^M (\kappa(j) - \hat{\kappa}(j))^2} \quad (41)$$

#### 5. Coefficient of determination (R.<sup>2</sup>)

$$R^2 = 1 - \frac{\sum_{j=1}^M (\kappa(j) - \hat{\kappa}(j))^2}{\sum_{j=1}^M (\kappa(j) - \bar{\kappa})^2} \quad (42)$$

where,  $M$  represents the sample size; the variables  $\kappa(j)$  and  $\hat{\kappa}(j)$  stand for the true and forecast values of the ultra-short-term speed of the wind;  $\bar{\kappa}$  refers to the mean value of the ultra-short-term speed of the wind.

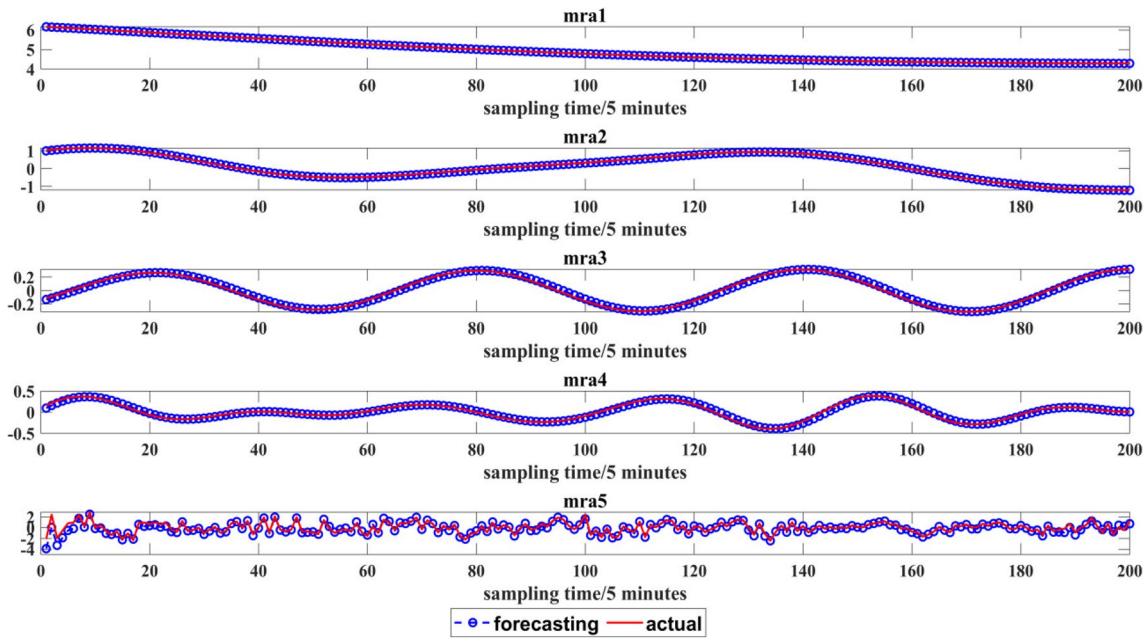
## Experimental results

In this prediction method, for the improved SSA, the highest amount of iterations is chosen as 200, the population size is chosen as 50, the proportion of discoverers P is chosen as 0.6, and the remaining are joiners. The proportion of distress responders D is 0.1. The distress threshold  $K_2$  is 0.8. The range of variation of ELM weights and thresholds to be optimized is  $U \in [-5, 5]$ , the total amount of intermediate layer neurons in ELM is 20, and the location dimension of each sparrow is 40. For the improved SSA to optimize each prediction model weight, the weights that need to be tuned have a variable range of  $U \in [-2, 2]$ . For the three datasets disassembled by EWT in Sect. "Mean absolute percentile error (MAPE)", the position dimension of each sparrow in data set a and c is 5, and the position dimension of each sparrow in data set b is 6. After several validations, the weights of the prediction models corresponding to the components in the two data sets after optimization are presented in Table 6.

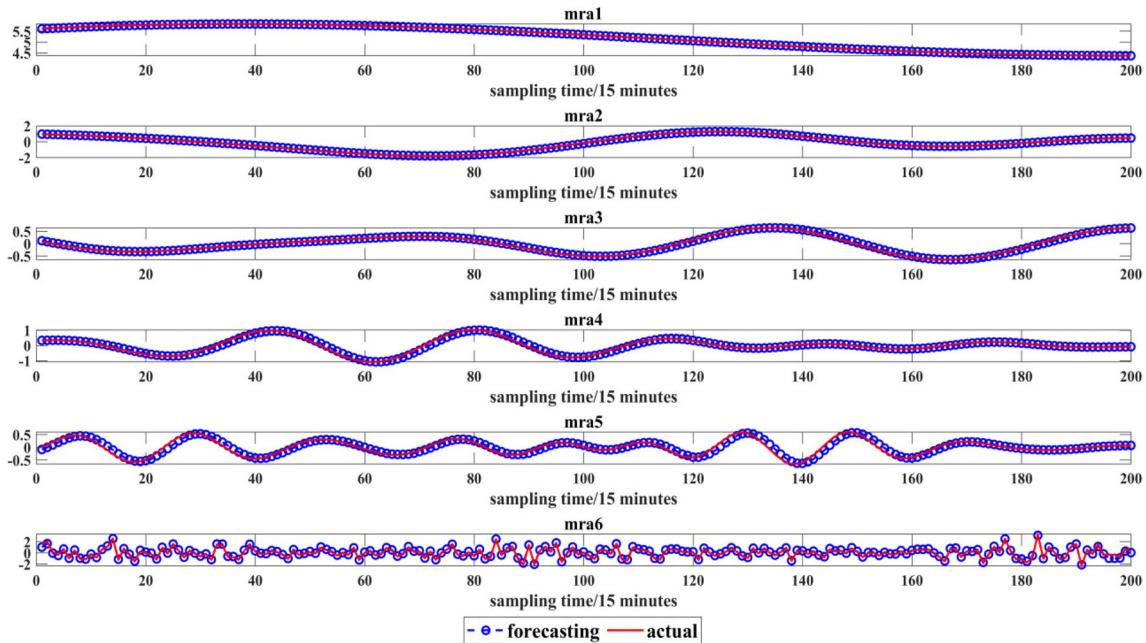
After obtaining the parameters and required weights of each model based on the training set training, the proposed method is validated by the test set data. The forecasting outcomes for every mra component of test set a

**Table 6** Optimization outcomes of the weight coefficients of each mra component

Dataset	Weight	Value	Dataset	Weight	Value	Dataset	Weight	Value
Dataset a	$\omega_1$	1.0271	Dataset b	$\omega_1$	0.9830	Dataset c	$\omega_1$	0.9192
	$\omega_2$	1.1864		$\omega_2$	1.0235		$\omega_2$	1.0172
	$\omega_3$	0.6626		$\omega_3$	0.8258		$\omega_3$	1.2062
	$\omega_4$	1.0042		$\omega_4$	0.8973		$\omega_4$	0.9179
	$\omega_5$	0.9341		$\omega_5$	0.9304		$\omega_5$	1.0928
				$\omega_6$	1.0162			



**Fig. 10** Prediction results for each mra component of test set a

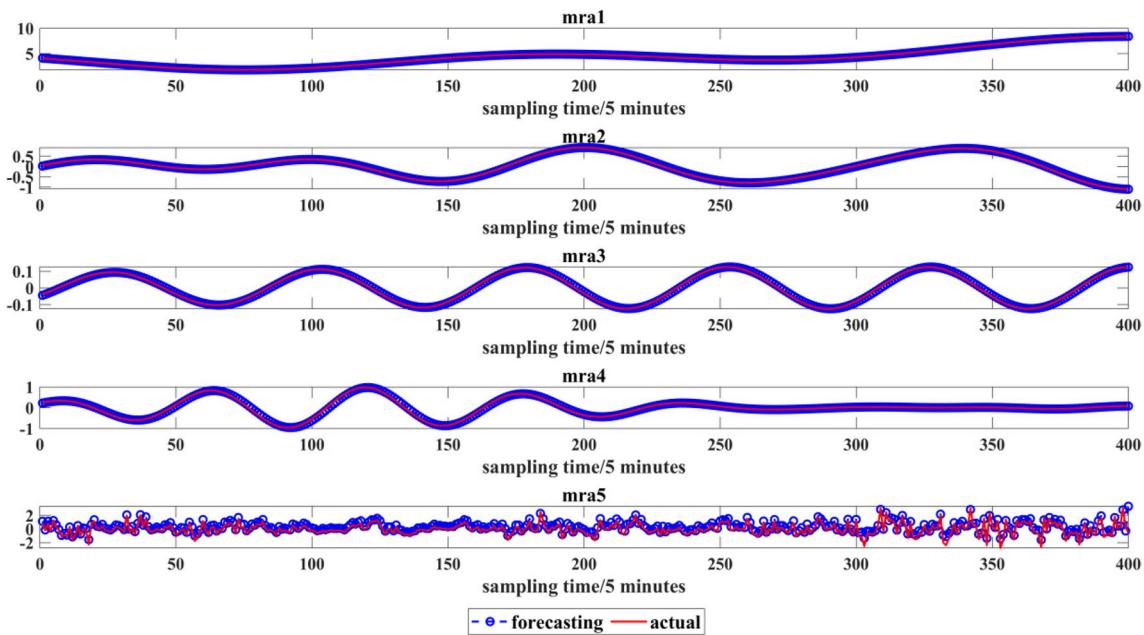


**Fig. 11** Prediction results for each mra component of test set b

after decomposition by EWT and reconstruction by sample entropy are illustrated in Fig. 10. The prediction outcomes for every mra element of test set b after decomposition by EWT and reconstruction by sample entropy are indicated in Fig. 11. The forecasting outcomes for every mra component of test set c after decomposition by EWT and reconstruction by sample entropy are illustrated in Fig. 12. As can be

seen in the picture, the prediction calibration results for each component of the chosen prediction model are good.

After predicting each mra component, the prediction results are multiplied by the corresponding weights and accumulated to produce the ultimate forecast value. The CNN-LSTM structure connects the CNN and LSTM layers in a cascade, and the local features extracted by the CNN



**Fig. 12** Prediction results for each mra component of test set c

are used as inputs to the LSTM, which enables the model to take advantage of both the local perception ability of the CNN and the long-term dependence modelling ability of the LSTM, but because the model contains parameters from both the CNN and the LSTM, the total number of parameters in the model is larger, and it may require more computational resources during training and inference. LSTM-Attention introduces an attention mechanism into the LSTM structure to allow the model to selectively focus on different parts of the input sequence, which helps the model to concentrate on and extract key information, but it is not sensitive to the sequence order, has higher computational complexity, and is sensitive to the length of the input sequence. BiLSTM-GRU is a deep learning network architecture that combines BiLSTM and GRU. The design of this architecture is inspired by the research and practice of different recurrent neural network (RNN) variants. GRU saves working memory and improves training speed compared to LSTM. Combining BiLSTM and GRU can take advantage of the bi-directionality of BiLSTM and the lightweight nature of GRU. This structure has the advantage of being able to better capture global contextual information when dealing with sequential data and has a relatively small number of parameters, making it more feasible when computational resources are constrained. In order to assess the efficacy of the approaches described in this article, eight prediction methods, BiLSTM, BiGRU, CNN-LSTM, LSTM-Attention, transformer, BiLSTM-GRU, EWT-ELM, and without weight coefficient optimization, were selected for comparison. Table 7 displays the necessary parameters for different prediction methods. The

prediction outcomes of these prediction methods for dataset a are illustrated in Fig. 13, the prediction outcomes of these prediction methods for dataset b are demonstrated in Fig. 14, and the prediction outcomes of these prediction methods for dataset c are demonstrated in Fig. 15.

It is clear from Figs. 13, 14 and 15 that the forecast method proposed in this research closely matches the actual ultra-short-term speed of the wind values. The values provided by the prediction method in this paper's predictions can more accurately reflect the actual speed of the wind.

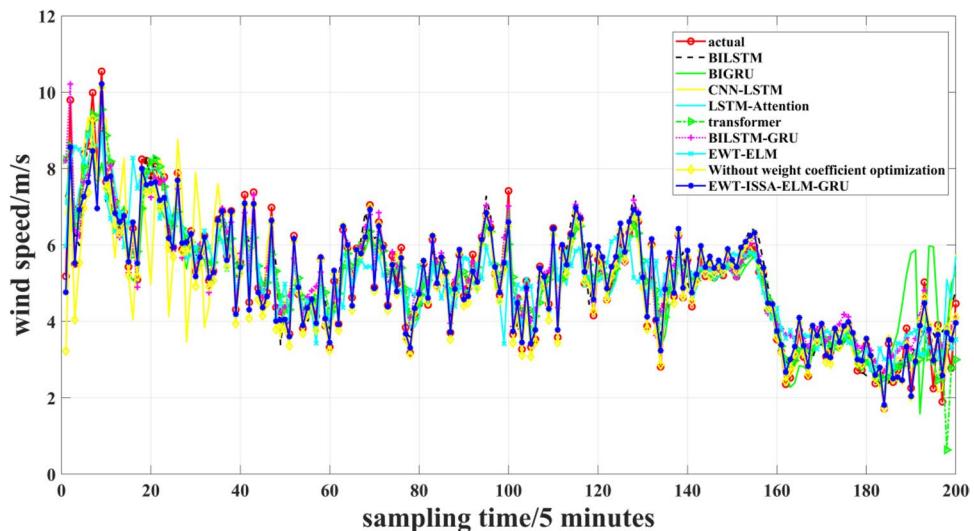
Tables 8, 9 and 10 give a comparison of the five performance metrics of MAE, MAPE, MSE, RMSE, and  $R^2$  for these forecast approaches for the ultra-short-term speed of the wind data sets a, b and c. As seen from the two tables, the MAE, MAPE, RMSE, and MSE of the proposed forecasting approach are the smallest compared to the others, demonstrating that the proposed method can minimize the mistakes to a small level. Meanwhile, the recommended prediction approach's  $R^2$  is, more comparable to 1 than those of the other techniques. The approach does regression forecasting more effectively the closer  $R^2$ 's value is to 1. As a result, the proposed forecasting method has higher prediction accuracy than the other methods that were evaluated.

In order to compare the time requirements of the prediction models, Table 11 shows a comparison of the average computational time requirements for training and prediction of these prediction models. As can be seen from Table 11, the training time of the prediction method in this paper is longer than that of BiLSTM, BiGRU, LSTM-Attention, and CNN-LSTM, and is close to the training time of the other

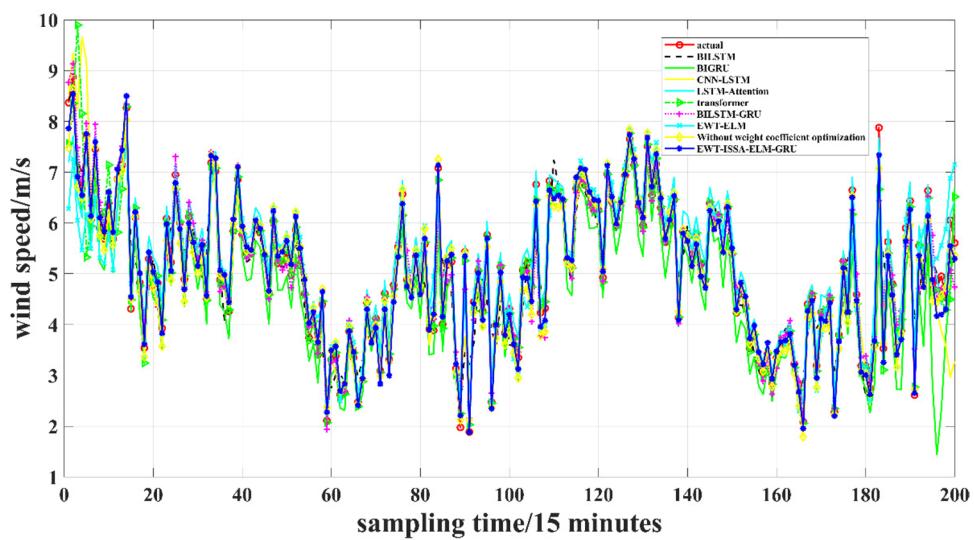
**Table 7** Parameters of forecasting methods for dataset a and dataset b

Prediction method	Dataset a parameters	Dataset b parameters
BILSTM	BILSTM: 100 neurons in 1 hidden layer; the number of output layers is 1; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400	BILSTM: 100 neurons in 1 hidden layer; the number of output layers is 1; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400
BILSTM-GRU	BILSTM: 100 neurons in 1 hidden layer; GRU: 200 neurons in 1 hidden layer; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400	BILSTM: 100 neurons in 1 hidden layer; GRU: 200 neurons in 1 hidden layer; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400
EWT-ELM	ELM: 1 hidden layer with 36 neurons	ELM: 1 hidden layer with 36 neurons
Without weight coefficient optimization	ELM: 1 hidden layer with 20 neurons; BILSTM: 1 hidden layer with 50 neurons; GRU: 1 hidden layer with 100 neurons; learning rate of 0.002, regularization parameter of 0.001, and iteration count of 300	ELM: 1 hidden layer with 20 neurons; BILSTM: 1 hidden layer with 50 neurons; GRU: 1 hidden layer with 100 neurons; learning rate of 0.002, regularization parameter of 0.001, and iteration count of 300
BIGRU	BIGRU: 100 neurons in 1 hidden layer; the number of output layers is 1; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400	BIGRU: 100 neurons in 1 hidden layer; the number of output layers is 1; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400
CNN-LSTM	CNN: 2 convolutional kernels [3,1] (Kebede et al. 2022; Jiang et al. 2018); number of channels is 16; LSTM: 50 neurons in 1 hidden layer; the learning rate is 0.008; the regularization parameter is 0.01; the number of iterations is 400	CNN: 2 convolutional kernels [3,1] (Kebede et al. 2022; Jiang et al. 2018); number of channels is 16; LSTM: 50 neurons in 1 hidden layer; the learning rate is 0.008; the regularization parameter is 0.01; the number of iterations is 400
LSTM-Attention transformer	LSTM: 100 neurons in 1 hidden layer; Attention:se; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400	LSTM: 100 neurons in 1 hidden layer; Attention:se; the learning rate is 0.001; the regularization parameter is 0.001; the number of iterations is 400
	200 neurons in 1 hidden layer; 3 encoders; the learning rate is 0.005; the number of iterations is 80	200 neurons in 1 hidden layer; 3 encoders; the learning rate is 0.005; the number of iterations is 80

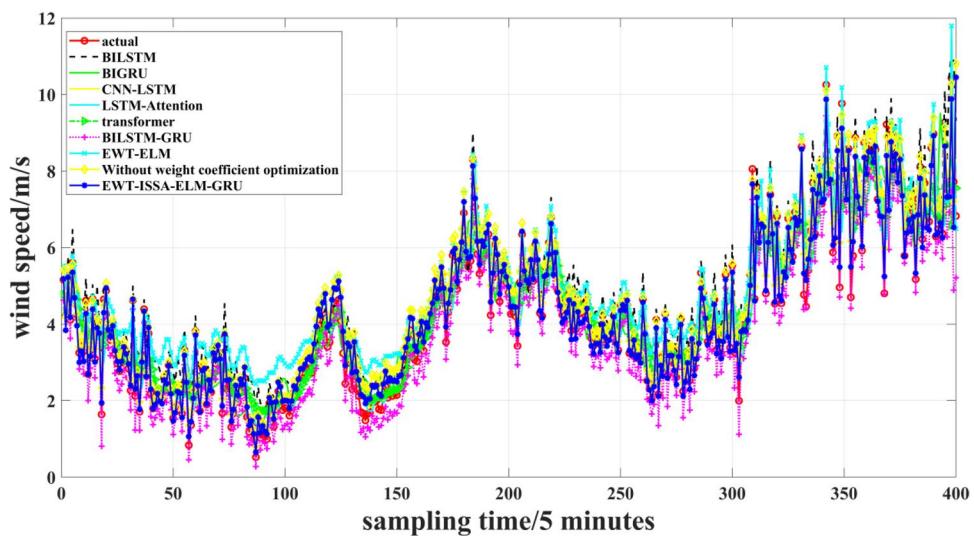
**Fig. 13** Prediction outcomes of the forecasting method for dataset a



**Fig. 14** Prediction outcomes of the forecasting method for dataset b



**Fig. 15** Prediction outcomes of the forecasting method for dataset c



**Table 8** Comparison outcomes of assessment criteria's for dataset a

Prediction method	MAE(m/s)	MSE(m/s)	RMSE(m/s)	MAPE(%)	R <sup>2</sup>
BIGRU	0.5481	0.5609	0.8489	12.815	0.7492
CNN-LSTM	0.6328	0.7795	0.8829	12.646	0.6972
LSTM-Attention	0.4164	0.3379	0.5813	9.483	0.8669
transformer	0.5259	0.4847	0.6947	11.758	0.7091
BILSTM	0.4782	0.3848	0.6203	10.738	0.8465
BILSTM-GRU	0.3592	0.2548	0.5049	8.947	0.8983
EWT-ELM	0.6442	0.6874	0.8291	14.102	0.7257
Without weight coefficient optimization	0.2506	0.1647	0.4059	5.036	0.9343
EWT-ISSA-ELM-BILSTM-GRU	<b>0.2341</b>	<b>0.1178</b>	<b>0.3492</b>	<b>5.006</b>	<b>0.9537</b>

**Table 9** Comparison outcomes of assessment criteria's for dataset b

Prediction method	MAE(m/s)	MSE(m/s)	RMSE(m/s)	MAPE(%)	R <sup>2</sup>
BIGRU	0.4675	0.3919	0.6260	9.586	0.7991
CNN-LSTM	0.3565	0.4498	0.6707	7.328	0.7693
LSTM-Attention	0.3119	0.1174	0.3426	6.705	0.9397
transformer	0.3659	0.4208	0.6567	7.692	0.7567
BILSTM	0.4118	0.3082	0.5552	9.272	0.8419
BILSTM-GRU	0.2134	0.1061	0.3257	4.549	0.9056
EWT-ELM	0.3433	0.2273	0.4767	7.006	0.8834
Without weight coefficient optimization	0.2018	0.0612	0.2473	4.261	0.9686
EWT-ISSA-ELM-BILSTM-GRU	<b>0.1602</b>	<b>0.0383</b>	<b>0.1958</b>	<b>3.362</b>	<b>0.9804</b>

**Table 10** Comparison outcomes of assessment criteria's for dataset c

Prediction method	MAE(m/s)	MSE(m/s)	RMSE(m/s)	MAPE(%)	R <sup>2</sup>
BIGRU	0.6079	0.6648	0.8134	16.556	0.8374
CNN-LSTM	0.6821	0.8502	0.9221	18.772	0.7921
LSTM-Attention	0.6728	0.8469	0.9203	18.157	0.7929
transformer	0.5608	0.9981	0.7734	16.184	0.8537
BILSTM	0.5477	0.4552	0.6746	18.859	0.8867
BILSTM-GRU	0.4891	0.3078	0.5548	13.922	0.9247
EWT-ELM	0.7545	0.7464	0.9639	19.285	0.8174
Without weight coefficient optimization	0.3565	0.2178	0.4667	10.442	0.9467
EWT-ISSA-ELM-BILSTM-GRU	<b>0.2097</b>	<b>0.0956</b>	<b>0.3091</b>	<b>6.738</b>	<b>0.9766</b>

**Table 11** Comparison outcomes of calculation of demand

Prediction method	dataset a		dataset b		dataset c	
	Training time(s)	Pridiction time(ms)	Training time(s)	Pridiction time(ms)	Training time(s)	Pridiction time(ms)
BIGRU	89	121	124	141	214	390
CNN-LSTM	142	111	136	103	173	151
LSTM-Attention	116	108	118	92	203	260
transformer	167	382	162	393	196	507
BILSTM	94	74	97	106	137	205
BILSTM-GRU	148	113	192	326	201	474
EWT-ELM	154	173	184	171	192	171
Without weight coefficient optimization	212	271	225	252	273	660
EWT-ISSA-ELM-BILSTM-GRU	<b>215</b>	<b>273</b>	<b>228</b>	<b>254</b>	<b>276</b>	<b>661</b>

four prediction models. Although the time complexity of the prediction model proposed in this paper increases, it improves the accuracy of the prediction, and compared with the sampling time of the target sequence, the increase in the computation time is still much smaller than the sampling time, which can meet the practical applications.

## Conclusion

In this paper, on the basis of EWT and a combined prediction model, an ultra-short-term speed of the wind forecast method is proposed. Firstly, the original ultra-short-term wind speed data are disassembled by EWT, and the signals of different frequency bands are extracted separately to obtain multiple smooth mra components. Different mra components have different characteristics, and choosing an appropriate model helps guarantee the accuracy of every component's prediction and can reduce the difficulty of modeling. The SE is computed to assess every mra component's complex rate, and those with very similar complexity are reconstructed, which can accelerate the forecast speed of the built model and reduce unnecessary time spent. The complex rate of every mra element after reconstruction is analyzed by using SE, and the improved SSA-optimized ELM and RILSTM-GRU are utilized as the models for forecasting for different components. The ultimate forecast values are established by multiplying the prediction values of the corresponding models of every element by the respective weight coefficients. Where the improved SSA benefits the weight coefficients for every sub-prediction approach. The final prediction values can be made with reduced errors thanks to the optimized weight coefficients. The method of this paper is validated by using the actual collected ultra-short-term wind speed data for 5- and 15-min periods. The comparison between the simulation experiments and other models shows that the method improves the precision of predictions and that the established model for forecasting is effective.

The proposed forecast method requires longer computational time compared to a single model. Moreover, the proposed prediction method only uses historical data and does not consider the effects of external factors such as temperature and topography. In order to further cut down on time consumption and enhance the predicting effect, future work on this article will attempt to combine the above-mentioned elements and develop a multi-input forecasting approach. To further verify its efficacy, the suggested forecasting method is further applied to time series of network traffic, traffic flow, and stock prices.

**Authors contributions** Maosen Wang: Software, Validation, and Writing.

Zhongda Tian: Conceptualization, Methodology, Software, Validation, Writing, and Funding acquisition.

**Funding** This paper is supported by the Natural Science Foundation of Liaoning Province of China (No. 2020-MS-210) and the Science Research Project of Liaoning Education Department (No. LJKZ0143).

**Data availability** The source data used to support the findings of this study is available from the corresponding author upon request.

## Declarations

**Ethical approval** Not applicable.

**Consent to participate** Not applicable.

**Consent to publish** The manuscript was vetted and approved for publication by all authors.

**Competing interests** The authors declare no competing interests.

## References

- Cui Y, Yan S, Zhang H et al (2019) Ultra-short-term prediction of wind power based on chaos theory and ABC optimized RBF neural network[C]. In: IEEE 3rd International Electrical and Energy Conference (CIEEC) IEEE 2019:1422–1427
- Ding J, Chen G, Huang Y et al (2021) Short-term wind speed prediction based on CEEMDAN-SE-improved PIO-GRNN model[J]. Measurement and Control 54(1–2):73–87
- Gao L, Zhao L, Kong F et al (2022) Research method of ultra-short-term wind power prediction based on PSO-GRU prediction[C]. In: 2022 8th Annual International Conference on Network and Information Systems for Computers (ICNISC). IEEE 2022:967–972
- Gilles J (2013) Empirical wavelet transform[J]. IEEE Trans Signal Process 61(16):3999–4010
- Hao SQ, Kuan ATH, Rudd CD (2020) A circular economy approach to green energy: Wind turbine, waste, and material recovery. Sci Total Environ 702:135054
- Hua L, Zhang C, Peng T et al (2022) Integrated framework of extreme learning machine (ELM) based on improved atom search optimization for short-term wind speed prediction[J]. Energy Convers Manage 252:115102
- James EP, Benjamin SG, Marquis M (2018) Offshore wind speed estimates from a high-resolution rapidly updating numerical weather prediction model forecast dataset[J]. Wind Energy 21(4):264–284
- Jiang Y, Chen M, Wen B (2018) Interval optimization of the day-ahead clearing schedule considering the real-time imbalance power with wind power integration[J]. International Transactions on Electrical Energy Systems 28(10):e2610
- Jónsdóttir G M, Hayes B, Milano F (2018) Continuous-time ARMA models for data-based wind speed models[C]. In: 2018 Power Systems Computation Conference (PSCC). IEEE 2018:1–7
- Kebede AA, Kalogianannis T, Van Mierlo J et al (2022) A comprehensive review of stationary energy storage devices for large scale renewable energy sources grid integration[J]. Renew Sustain Energy Rev 159:112213

- Li L, Li Y, Zhou B et al (2020a) An adaptive time-resolution method for ultra-short-term wind power prediction[J]. *Int J Electr Power Energy Syst* 118:105814
- Li LL, Zhao X, Tseng ML et al (2020b) Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm[J]. *J Clean Prod* 242:118447
- Li K, Yan X, Han Y (2023) Multi-mechanism swarm optimization for multi-UAV task assignment and path planning in transmission line inspection under multi-wind field[J]. *Appl Soft Comput* 111033
- Liu X, Lin Z, Feng Z (2021) Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM[J]. *Energy* 227:120492
- Liu Z, Yang J, Liu Y et al (2019) Ultra short term prediction of wind power based on improved BP neural network[J]. *Ship Engineering* 41(1):282–287
- Ma X, Jin Y, Dong Q (2017) A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting[J]. *Appl Soft Comput* 54:296–312
- Moreno SR, da Silva RG, Mariani VC et al (2020) Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network[J]. *Energy Convers Manage* 213:112869
- Sun Y, Wang P, Zhai S et al (2020) Ultra short-term probability prediction of wind power based on LSTM network and condition normal distribution[J]. *Wind Energy* 23(1):63–76
- Tan L, Han J, Zhang H (2020) Ultra-short-term wind power prediction by salp swarm algorithm-based optimizing extreme learning machine[J]. *IEEE Access* 8:44470–44484
- Tena García JL, Cadenas Calderón E, González Ávalos G et al (2019) Forecast of daily output energy of wind turbine using sARIMA and nonlinear autoregressive models[J]. *Adv Mech Eng* 11(2):1687814018813464
- Tian Z (2020) Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM[J]. *Eng Appl Artif Intell* 91:103573
- Tian Z (2021) Modes decomposition forecasting approach for ultra-short-term wind speed[J]. *Appl Soft Comput* 105(11):107303
- Tian Z, Chen H (2021) A novel decomposition-ensemble prediction model for ultra-short-term wind speed[J]. *Energy Convers Manage* 248:114775
- Tyass I, Bellat A, Raihani A et al (2022) Wind speed prediction based on seasonal ARIMA model[C]. In: E3S Web of Conferences. EDP Sciences, vol 336
- Vassallo D, Krishnamurthy R, Sherman T et al (2020) Analysis of random forest modeling strategies for multi-step wind speed forecasting[J]. *Energies* 13(20):5488
- Wang YL (2009) Moving average calculation method of wind speed of the ground automatic meteorological observation systems[J]. *Journal of Meteorological Research and Application* 30(02):75–77
- Wang Y, Gui R (2022) A hybrid model for GRU ultra-short-term wind speed prediction based on tsfresh and sparse PCA[J]. *Energies* 15(20):7567
- Wang J, Wang S, Zeng B, Lu H (2022) A novel ensemble probabilistic forecasting system for uncertainty in wind speed. *Appl Energy* 313:118796
- Wang L, Li K, Ji Z et al (2020) An ultra-short-term prediction method for wind speed series based on Gaussian process median regression[C]. In: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE 2020:495–499
- Xu X, Wei Y (2022) An ultra-short-term wind speed prediction model using LSTM and CNN[J]. *Multimedia Tools and Applications* 81(8):10819–10837
- Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering* 8(1):22–34
- Yang Q, Deng C, Chang X (2022) Ultra-short-term/short-term wind speed prediction based on improved singular spectrum analysis[J]. *Renewable Energy* 184:36–44
- Zhao Z, Nan H, Liu Z et al (2022) Multi-step interval prediction of ultra-short-term wind power based on CEEMDAN-FIG and CNN-BiLSTM[J]. *Environ Sci Pollut Res* 29(38):58097–58109
- Zucatelli PJ, Nascimento EGS, Aylas GYR et al (2019a) Short-term wind speed forecasting in Uruguay using computational intelligence[J]. *Heliyon* 5(5):e01664
- Zucatelli PJ, Vitória U, Santo E et al (2019b) Short-range wind speed predictions in subtropical region using artificial intelligence[J]. *J Systemics Cybern. Informatics* 17(4):1–8
- Zucatelli PJ, Nascimento EGS, Santos AÁB et al (2020) Nowcasting prediction of wind speed using computational intelligence and wavelet in Brazil[J]. *Int J Comput Methods Eng Sci Mech* 21(6):343–369
- Zucatelli PJ, Nascimento EGS, Santos AÁB et al (2021) An investigation on deep learning and wavelet transform to nowcast wind power and wind power ramp: A case study in Brazil and Uruguay[J]. *Energy* 230:120842

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.