Short-term Wind Speed Prediction with Master-slave Performance Based on CNN-LSTM and Improved POABP

Gonggui Chen, Mengyuan Zhu, Jing Huang, Yi Fu, Xiaochuan Xie and Hongyu Long*

***Abstract*—Clean energy has taken center stage in discussions about how to ensure that humanity has a sustainable future when fossil resources are depleted. Wind energy is also playing a bigger and bigger part in the energy supply as a clean, renewable energy source. Hence, the use of wind energy is significantly impacted by an accurate and appropriate wind speed prediction system. The non-linear wind speed prediction in this research is handled by a master-slave prediction model based on a convolutional neural network and long-short memory (CNN-LSTM) network and an improved Pelican Optimization Algorithm BP neural network (IPOABP). In this model, CNN-LSTM is used to obtain a prediction sequence for each subsequence, producing a smoother feature sequence. A BPNN BP (Back-Propagion Neural Network) is then trained using the reconstructed sequence, and the reconstructed sequence is processed by inverse empirical modal decomposition (EMD) for the prediction sequence. Hence, the use of wind energy is significantly impacted by an accurate and appropriate wind speed prediction system. The non-linear wind speed prediction in this research is handled by a master-slave prediction model based on a convolutional neural network and long-short memory (CNN-LSTM) network and an improved Pelican Optimization Algorithm BP neural network (IPOABP). In this model, CNN-LSTM is used to obtain a prediction sequence for each subsequence, producing a smoother feature sequence. A BPNN BP (Back-Propagion Neural Network) is then trained using the reconstructed sequence, and the reconstructed sequence is processed by inverse empirical modal decomposition (EMD) for the prediction sequence. (RMSE=0.0596, MAPE=0.8923, MAE=0.0481) were obtained as minimum values from experiments on 3 different datasets.**

Manuscript received October 28, 2022; revised May 22, 2023.

Gonggui Chen is a professor at Key Laboratory of Industrial Internet of Things and Networked Control. Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065,China.e-mail:chenggpower@126.com).

Mengyuan Zhu is a graduate student of Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: zhumengyuanowo@163.com).

Jing Huang is a senior engineer of the State Grid Chongqing Electric Power Company, Chongqing 400014, China (e-mail: JingHuang_cq@163.com).

Yi Fu is a senior engineer of the State Grid Chongqing Wuxi Power Supply Company, Chongqing 405800, China (e-mail: YiFu_wx@163.com)。

Xiaochuan Xie is a senior engineer of the State Grid Chongqing Pengshui Power Supply Company, Chongqing 409600, China (e-mail: XiaochuanXie_ps@163.com)

Hongyu Long is a professor level senior engineer at Chongqing Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (corresponding author to provide phone: +8613996108500; e-mail: longhongyu20@163.com).

***Index Terms*—wind speed forecasting, SSA, LSTM, IPOA-BP**

I. Introduction

The 21st century's tremendous economic growth has been matched by an increase in energy usage. Wind power is one of the least expensive ways to produce electricity, therefore today energy is recognized as a key component determining the economic development of nations all over the world [1,2]. China added 71.67 GW of new wind power capacity, bringing its installed capacity to 338.31 GW, or 40.4% of the total installed capacity worldwide [3,4]. However, because wind speed is so unpredictable and volatile, it is crucial for wind farms to have a reliable mechanism for predicting wind speed [5,6].

After decades of updates and iterations, numerous researchers have developed a number of efficient techniques, primarily physical techniques, statistical techniques, in order to increase wind speed predicting accuracy, artificial intelligence approaches and hybrid models were used.

Broadly speaking, there are three types of wind speed forecasts: long-term, medium-term, and short-term [7,8]. The coordination of power dispatch and the use of short-term wind speed forecasts as a tool to maintain the security and stability of the power system are both crucial [9,10]. The most effective data pre-processing technique is suggested in an enhanced fully aggregated empirical modal decomposition using adaptive white noise and a multi-objective dragonfly algorithm for managing the fluctuation and stochasticity of wind speed series data is described in the review [11]. Most existing prediction models disregard the significance of data pre-processing and are sensitive to the constraints of a single model, resulting in poor forecast accuracy. ICCEMDAN is used to divide the original wind speed into many series and then discard high frequencies before reconstructing the windspeed. The new combined model effectively overcomes the constraints of the single model to produce precise and consistent prediction results by considering both the linear and non-linear aspects of the series [12]. The trend component is predicted using CNNGRU, and the detail component is predicted using SVR. In Article [13], the time-series coupling information in the data is mined using a one-dimensional convolutional neural network. The article rarely takes the turbulent component of the wind into account compared to earlier projections of mean wind speed. The literature [14] uses a correlation analysis-based strategy for choosing time-delay features in order to produce superior structural predictors. A large

improvement in prediction accuracy may be shown by contrasting the performance with widely used competitors like ANN, SVR, and ELM. According to this framework, WPD divides the initial wind speed sequence into a number of sub-layers, and CEEMDAN further divides all sub-layers into several IMF (intrinsic modal functions). Finally, three neural network models are used to forecast the deconstructed wind speed data: BP, RBF, and GRNN. A mixture model is proposed in the press using grey relational theory and wind speed profiles [15,16] as a way to benefit from each individual prediction model and enhance wind power forecasting. To estimate the weights of each independent model, different wind speed segments and comparable wind speed frequencies are taken into consideration. The wind speed frequency is considered. This case study demonstrated that mixture forecasting systems have wider utility for very short-term wind output forecasting. (15 minutes in advance) [17,18].

Deep learning has recently attracted the attention and favor of many academics due to its widespread use in a variety of sectors and major benefits. Particularly, in many sectors, LSTM, a type of recurrent neural network (RNN), outperforms conventional techniques. By using an optimized recursive generalized learning system (ORBLS), sample entropy (SE), an extended completely ICEEMDAN, and a broadened temporal convolutional network, Article [19] presents an innovative hybrid model with the purpose of improving the precision of ultra-short-term wind speed prediction. A CNN, an LSTM model, and a new, strong approach to improving WSF are presented in article [20]. The proposed hybrid technique is composed of two primary components: feature encoding, LSTM autoencoder-based downscaling, and convolutional LSTM-based prediction. The LSTM autoencoder minimizes the computational load of the predictive convolutional LSTM technique in the first stage by removing the uncertainty present in the raw wind speed data. Next, in the 2nd phase, the best characteristics are extracted using CNN, and the wind speed is predicted using LSTM. Suggested hybrid model outperforms other models used in one-step to five-step forecasting. A novel spatio-temporal correlation model (STCM) based on CNN-LSTM is suggested in research [21] for use in the forecasting of wind power over an extremely short period of time. Reconstructing the raw meteorological variables at various historical time points at various locations within the target wind farm into the model's input window represents a novel method of data reconstruction. The study's [21] CNN-LSTM-based STCM is relevant to wind farms that may gather meteorological data at various places. In order to predict short-term offshore wind speeds, the article [22] evaluates six different deep learning-based models: CNN, superimposed LSTM, bidirectional LSTM, CNN-LSTM, multilayer perceptron, and two-dimensional convolutional LSTM. Few researches have expanded the challenge to two-dimensional wind planes; previous studies have mostly concentrated on single-location forecasting. This paper uses a mix of a CNN autoencoder and an LSTM to present a unique deep learning model for two-dimensional regional wind speed prediction. A significant estimate of the spatial wind speed distribution in a two-dimensional wind farm is provided by the overall MAE value of the present model, which decreases to 0.35 m/s and is 32.7%, 28.8%, and 18.9% different from predictions made using the persistent, simple ANN, and LSTM models.

This research provides a deep learning method and an improved heuristic optimization to raise the level of accuracy of the modelled predictions, so that wind speed predictions are more accurate and work better. The following are this paper's main contributions: (1) The data are pre-processed and post-processed by integrating SSA, CEEMDAN, and invert-EMD, considering the features of wind speed data. Inverse empirical modal decomposition is used to recreate the prediction subsequences that were derived from the primary predictor (invert-EMD). (2) Deep knowledge and BPNN inspired algorithms based on the optimization of a principle slave prediction model is created. A better BP network is used by the predictor to produce secondary predictions of invert-EMD reconstructed sequences after building a principal model based on CNN-LSTM. (3) An IPOA is suggested to optimize the weight parameters of the BP network in order to prevent the network from succumbing to local optimization. (4) Five distinct models were tested on three datasets of 15-minute mean wind speeds in order to confirm the efficiency of the modules' functions and the success of the overall method. A comparison of the models' predicted outcomes led to some findings.

II. Proposed Model

Two methods for forecasting short-term wind speed are suggested in this paper: a deep learning-based master-slave prediction model and a BP neural network adjusted by heuristic algorithms. The model is made up of the IPOABP, CNN-LSTM data pre-processing, and CNN-LSTM data post-processing. The stages for model prediction are described below.

1) The initial data on primary wind speed needs to be collected in order to denoise the primary wind speed and split the primary wind speed series into subseries. The data pre-processing component is then created by combining SSA and CEEMDAN. The data pre-processing component will be clearly detailed in Chapter 3.

2) Following denoising using SSA, the wind speed sequence is divided into two parts: training and testing. To get the final prediction results, 80% of the denoised wind speed sequence is used as the training set and the remaining 20% as the BPNN test set.

3) Using CEEMDAN approach, training set was separated into multiple sub-series, and each sub-series was then split into training sets 1 and 2.

4) Fuzzy on can be used to determine the time complexity of each component (FE), and intrinsic modal function components (IMF) are then combined to shape a new subsequence, using the spearman collinearity ratio.

5) The CNN-LSTM network is utilized as the master prediction, and thus for each subsequence. Training is performed on training set 1 and testing is performed on training set 2. The test sequences of the training set 2 subsequences are then obtained in CNN-LSTM network.

6) Using inverse EMD as data post-processing, the predicted sequences of each subsequence are reconstructed

and a reconstructed sequence can be obtained by inverse EMD.

7) IPOA-BP is used to perform the next step of prediction on the reconstructed sequences to complete the final prediction results.

8) The hybrid prediction model combining CNN-LSTM and the IPOA-BP network is evaluated on the test set to get the final test results. Six unique models are compared in this study to assess the predicted capability of the proposed model, and section 4 outlines the evaluation criteria for doing so.

III. Methodology

Data pre-processing, master predictor (CNN-LSTM), data post-processing, and slave predictor are the four parts of the model indicated above (IPOA-BPNN). This chapter will describe the proposed model and the algorithmic part in detail.

A. Data per-processing

1) Singular Spectrum Analysis (SSA)

SSA is a somewhat popular technique for analyzing and predicting event Series using non-linear time sequence fishery data. Its base is the singular value decomposition of a matrices built from time periods, which allows the deconstruction of trends, oscillatory components, and noise from a time series.[24].

SSA has a fairly wide application to time series since it does not need the assumption of a parametric model or smoothness requirements[25]. Consists of two parts: decomposition and reconstruction.

a) Decomposition

Equation (1) shows how the initial wind speed sample size is turned into input tensors into a matrix, which is composed of a vector of dimension $y_i=(x_1,\ldots,x_{i+L-1})$.

$$Y=\begin{Bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \cdots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{Bmatrix} \tag{1}$$

L stands for the embed window length $L=N-L+1$.

Calculate the singular values of the matrices YY^T, D^i, and the left-hand odd matrix, where the odd ratio and its vegetation are provided by Eq. (2) and Eq., respectively (3)

$$Y=Y_1+Y_2+\ldots+Y_d \tag{2}$$

$$y_1=\sqrt{\lambda_i}C_iD_i \tag{3}$$

Where, d is the matrix YY^T number of ranks.

b) Reconstruction

Grouping: This phase separates matrix Y into d submatrices in order to discover the most valuable signal components. The m-matrix with singular deviations higher than 0 is the part of the preliminary wave speed series that expresses the long-term tendency; the submatrix of variance formed by these $R=\{r_{1,\ldots,}r_m\}$ component matrices are given as 1, and relevant long-term tendency factor matrix is $Y_R=Y_{r1}+Y_{r2}+\ldots+Y_{rm}$, the sequence for each matrix is indicated by the number Y_{r1} 、 Y_{r2}... Y_{rm}. The cycle concludes with the noise component of the original wind speed.

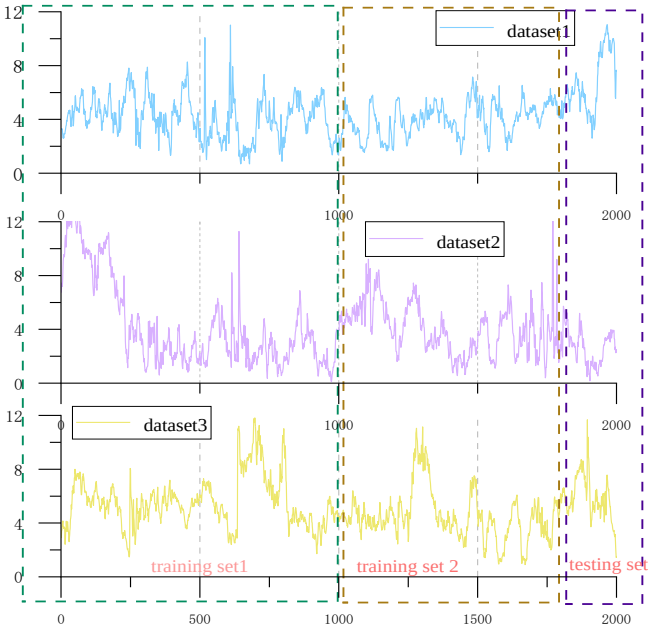

Fig.1 Three sets of original wind speeds

Reconstruction: The noise element will be altered to a spurious time series *Xnoise*, which can be realised by the formulation. At the same time, the matrix *Yp* will be converted to an adequate wind speed timeseries $X_R=\{X_{r1,\ldots,}X_{rm}\}$ by means of diagonal averaging (4). After application of SSA processing, the result of the original wind speed time series may be seen in the equation (5)

$$y_{ri}=\begin{cases} \frac{1}{i}\sum_{m=1}^{i} y^*_{m,k-m+1} & 1\le i<l \\ \frac{1}{l}\sum_{m=1}^{i} y^*_{m,k-m+1} & l\le i\le k \\ \frac{1}{N-i+1}\sum_{m=i-k1}^{N-k+1} y^*_{m,k-m+1} & k<i\le N \end{cases} \tag{4}$$

$$X_N=X_{r1}+X_{r2}+\ldots+X_{rm}+X_{noise} \tag{5}$$

Where, Parameters $l=min(L,K)$, $k=max(L,K)$.

When it comes to the data pre-processing phase, window length and The amount of parity r of the reconfigured sign, in addition to the selection of the two parameters to be determined in the phases described earlier, are all very important factors. The combination of the prior relationship of the empty formulation to the real data structure of this study is depicted in Figure 5; the SSA denoising outcome of the wind speed time sequence is displayed in Fig.2, and the choice of variables and their relative values are indicated in TABLE I below. Figure 5: The combination of the prior empirical formulae with the real data structure of this study.

TABLE I
SSA Parameters And Values

Parameter name	Value
Window length L	8
singular values number r	4
characteristic value r_1	98.16%
characteristic value r_2	0.9596%
characteristic value r_3	0.3632%
characteristic value r_4	0.1514%

2) CEEMDAN

3) The EEMD and CEEMD decomposition methods reduce Elimination of modal blending in EMD breakdown by adding pairs of impaired Gaussian white noise to the ribbon

decomposed signal. Nevertheless, the eigenmode components formed from these two signals always include some residual white noise, which has an effect on the signals' further analysis and processing[26].

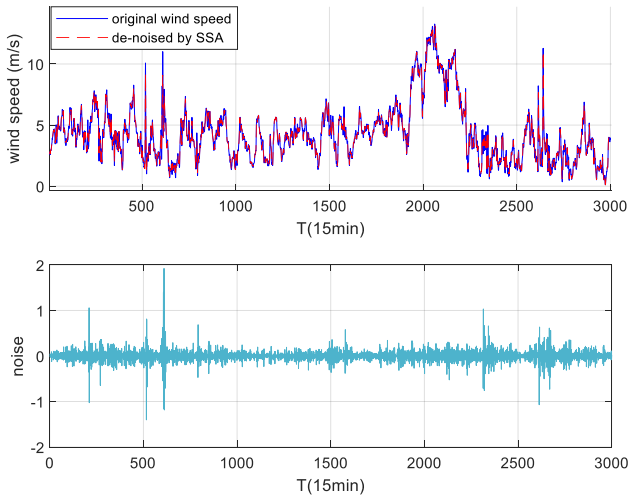

Fig.2 . De-noised windspeed with SSA

To address these issues, TorresME et al. (2011) introduced of Noise CEEMDAN method, an unique signal decomposition approach.[27].

CEEMDAN, in contrast to EEMD, focuses on dissecting time series into IMFs (intrinsic mode functions), and it employs two strategies to decrease residual noise:

a) Adding the EMD decomposition containing the auxiliary noise signal directly to raw signal.

b) The discretization of EEMD and CEEMD is an integral mean of the obtained modal variables after the empirical modal factorization, while the classification of CEEMDAN is an entire mean of the acquired IMF components after the first stage of computing, to yield the final first-order IMF content, and then replicate the above routine for the residuals. The issue of moving white noise from higher to lower frequencies has now been resolved quite satisfactorily.

The first one is to change the adjunctive noise, the second is a change in the break-up course, which is a better solution to the phenomenon of modal mixing that exists in empirical modal decomposition (EEMD)[28]. The specific analysis is described as follows. The principle is shown in Fig.3.

Principle of CEEMDAN algorithm as shown below:

Let $E(\bullet)$ be the eigenmode decomposition i eigenmode component was derived following EMD decomposition and by CEEMDAN decomposition is

(1) Adding Gaussian white noise to the decomposition signal $y(t)$ yields the new signal $y(t)+(-1)^q \varepsilon v^j(t)$, The formula is shown in equation (6). Where， q=1,2.

The first-order component of the new signal's EMD decomposition is obtained eigenmodal component C_1.

$$E(y(t)+(-1)^q \varepsilon v^j(t)) = C_1^j(t) + r^j \tag{6}$$

(2) The first characteristic modular component of the CEEMDAN deck is acquired by equally dividing the N outcome modular components. As shown in equation (7).

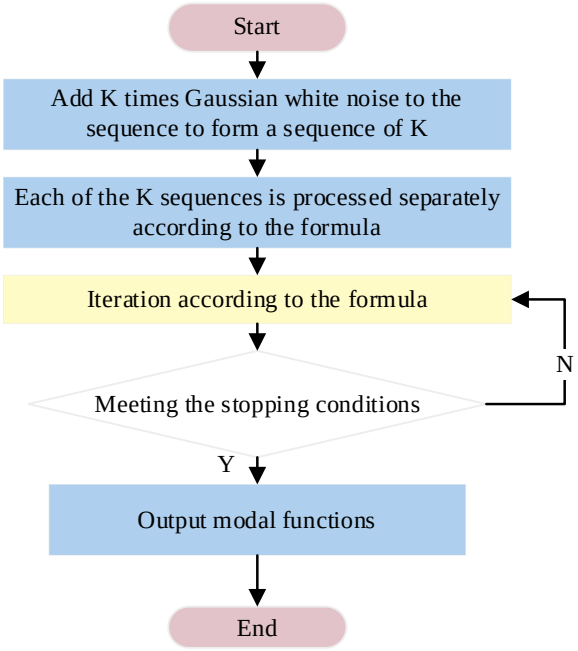

Fig.3 The CEEMDAN implementation process

$$\overline{C_1(t)} = \frac{1}{N}\sum_{j=1}^{N} C_1^j(t) \tag{7}$$

(3) Calculate the residuals using equation (8) after deleting the 1st modal component：

$$r_1(t) = y(t) - \overline{C_1(t)} \tag{8}$$

(4) The first feature mode segment D1 of the EMD factorization can be derived from the novel signal by adding positive and negative matched Gaussian white noise to the new signal. The new signal is then used as a carrier for the CEEMDAN decomposition:

$$\overline{C_2(t)} = \frac{1}{N}\sum_{j=1}^{N} D_1^j(t) \tag{9}$$

(5) After deleting the 2nd modal component, determine residuals.

$$r_2(t) = r_1(t) - \overline{C_2(t)} \tag{10}$$

(6) Repeat the technique outlined above until the remaining signal is a function that is univocal and cannot be further profiled. The algorithm then finishes, K eigenmodal components are obtained, original signal $y(t)$ is decomposed as:

$$y(t) = \sum_{k=1}^{K} \overline{C_k(t)} + r_k(t) \tag{11}$$

4) Fuzzy entropy (FE)

When estimating the sophistication of a time series, one of the criteria assessed is FE. The FE measures the magnitude of the probability of a new pattern being generated, analogous to the physical meanings of AE and SE; the larger the value of the measure, the greater the likelihood of a new format being generated, and thus the greater the series' complexity[29], the structure of FE is shown in Fig.7.

The steps for implementing fuzzy entropy are as follows: Given an time series, [$u(1)$, $u(2)$,…,$u(N)$] time series in N-dimensional. Determine the similarity tolerance limit r and

the phase space dimension m(m<N-2). and reconstruct the phase space as shown in equation (12):

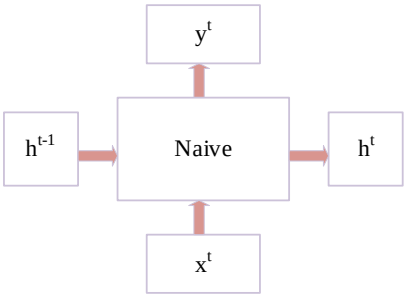

Fig.4 CNN framework diagram

$$X(i)=[u(i),u(i+1),\ldots,u(i+m-1)]-u_0(i) \tag{12}$$

Where, $u_0(i)=\frac{1}{m}\sum_{j=0}^{m-1}u(i+j)$.

The fuzzy affiliation function introduced in this paper, is shown in equation (13).

$$A(x)=\begin{cases}1, x=0\\ e^{[-\ln(2)\left(\frac{x}{r}\right)^2]}, x>0\end{cases} \tag{13}$$

Where, r is the similarity tolerance limit. For i=1,2,...N-m+1, calculate equation（14）:

$$A_{ij}^m=e^{[-\ln(2)(\frac{d_{ij}^m}{r})^2]} \tag{14}$$

Where, $j=1,2,\ldots,N-m+1,\ j\neq i.$

$$d_{ij}^m=d[X(i),X(j)]=$$
$$\max_{j=1,2,\ldots,m}(|u(i+p-1)-u_0(i)-u(j+p-1)-u_0(j)|] \tag{15}$$

d_{ij}^m is the maximum absolute distance between the window vectors $X(i)$ and $X(j)$. The calculation formula is shown in equation (15).

For each i, find the average to obtain:

$$C_t^m(r)=\frac{1}{N-m}\sum_{j=1,j\neq i}^{N-m+1}A_{ij}^m \tag{16}$$

Defining $\Phi^m(r)$ as:

$$\Phi^m(r)=\frac{1}{N-m+1}\sum_{i=1}^{N-m+1}C_i^m(r) \tag{17}$$

The FE of the original time series is:

$$FuzzyEn(m,r)=\lim_{N\to\infty}[\ln\Phi^m(r)-\ln\Phi^{m+1}(r)] \tag{18}$$

For a restricted data set, the fuzzy upper bound is calculated as:

$$FuzzyEn(m,r,N)=[\ln\Phi^m(r)-\ln\Phi^{m+1}(r)] \tag{19}$$

B. CNN-LSTM model

1) Convolutional Neural Networks (CNN)

a) Concepts of Convolutional Neural Networks

Convolutional neural networks are one of the most successful applications of deep learning techniques (CNNs). These include convolutional neural networks in 1D, 2D, and 3D [30]. Three-dimensional CNN are often used for the discrimination of optical images and video data, whereas two-dimensional CNN are often employed for imagery verification text recognition in one-dimensional convolutional neural networks.[31].

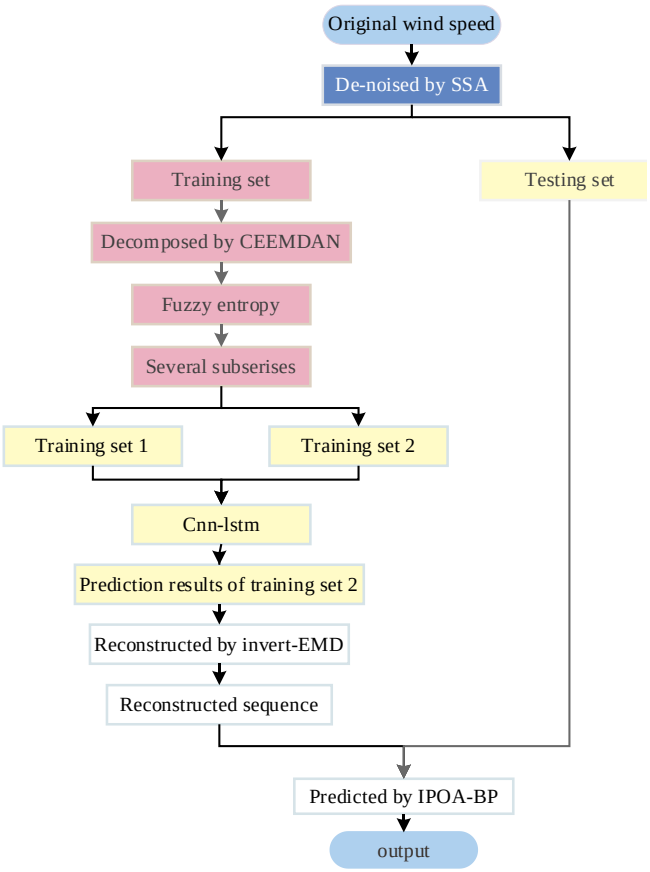

Fig.5 Data transfer process

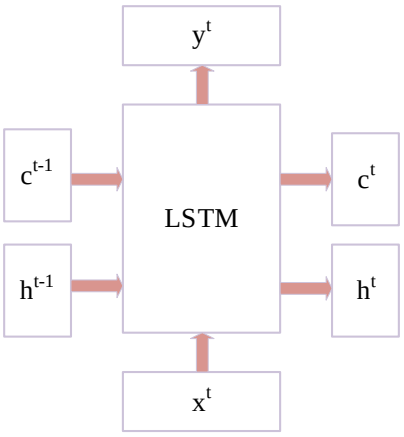

Fig.6 LSTM framework diagram

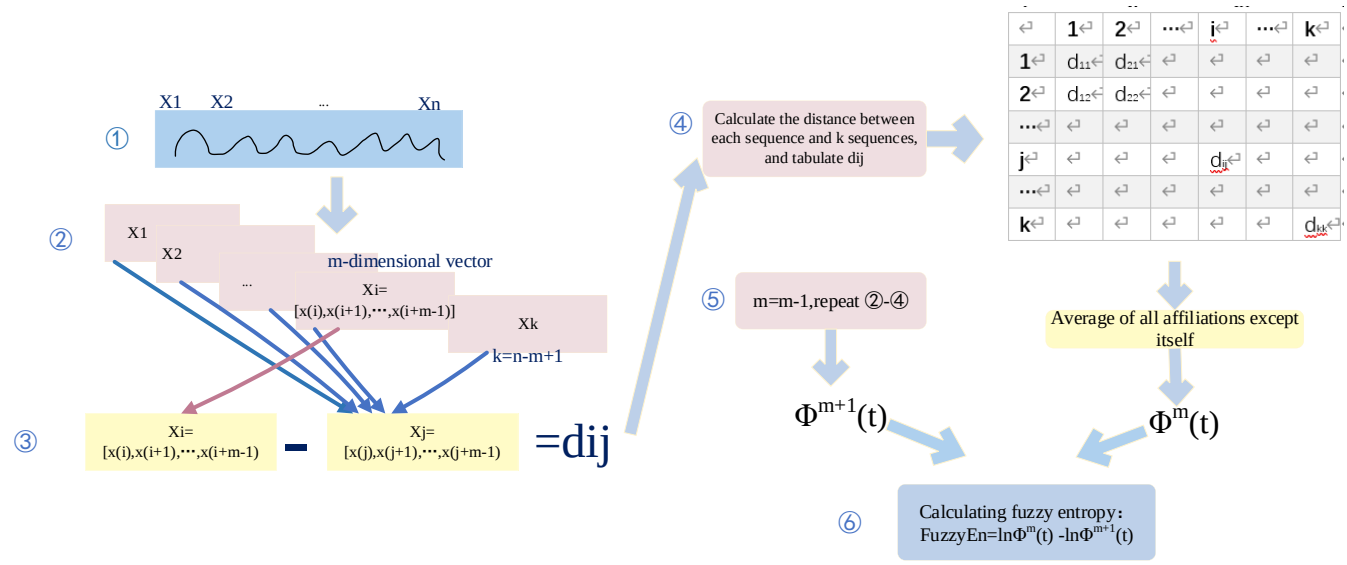

Fig.7 Flowchart of FE implementation

b) Convolutional neural network structure

Typically, CNN have the following layers.

Convolutional layers: The convolutional layers of a CNN are made up of a number of convolutional units, the arguments of each involution unit are revised using back propagation. To extract multiple properties from input data, convolutional procedures are performed. After The first convolutional layer, which may only pick up some lower-level functions, such as edges, lines and corners, a higher layered network may repeatedly extract more intricate characteristics from the low-level input.

Excitation layer: the output of the convolutional layer is mapped in a non-linear fashion utilizing well-known excitation functions like as the Sigmoid, Tanh, and ReLU functions. Rapid convergence and straightforward gradient detection are hallmarks of a typical ReLU excitation function for a CNN.

Pooling layer: The pooling layer, which is located between two convolutional layers and has two functions, is as follows: 1) To reduce the dimensionality of feature vector that is generated by the convolutional layer as well as the number of training parameters; 2) To lower the amount of noise that is sent and to only save the most crucial picture data.

Generally speaking, there are two general types of pooling:

Max Pooling: We will take maximum value in the sliding scale window.

Average Pooling: We will take average of all numbers in the sliding widget

2) long short-term memory (LSTM)

The primary goal of LSTM, a distinct kind of RNN, is to Solving the grade vanishing and gradient exposure problem in long sequential coaching [32]. Simply said, it outperforms traditional RNNs in longer sequences. Figures 4 and 6 demonstrate how the two differ from one another.

In LSTM, where the passed-on changes extremely slowly and the c^t output often represents the value of the passed-on from the prior state, the h^t in RNN is identical to the c^t. Fig.6 depicts the LSTM's internal structure.

a) Structure of the LSTM

Splicing vectors multiplied by a texture of weights to produce the three states z^f, z^i, and z^o, which are then use sigmoid activated features as threshold state conversion to values between 0 and 1. A *tanh* activation function transforms z into a number between -1 and 1. (*tanh* is used here because it is intended to be applied as entry data and not as a gating sign.).

b) Three phases within the LSTM

First phase: The forgetting phase: the forgetting gate. This stage is a period of partial obliviousness to the input from the earlier node. In simple terms, this stands for "forget the not-so-important and keep in mind the vital". Specifically, the z^f (f for forget) is computed as a forgetting gate, and the c^t controls what needs to be kept and what needs to be forgotten from the previous state.

Second phase: input gate stage of a selective memory. This stage 'remembers' just some of the inputs from earlier stages. The choice of input x^t is the major concern. The more significant portions are recorded, while the less significant portions are recorded less. The previously computed z represents the current input. z^i chooses the gating signal to be used (i for information). The outcome of these two phases is combined to produce the c^t that will be transferred to the next state, which is represented by the first equation in the above figure.

The third step is the output gate phase. Which will be utilized as the current state's output will be decided at this step. The z^o is generally used to regulate it. Moreover, the c^o from the earlier phase is deflated (varied by a *tanh* activation function). The output $y(t)$ is often finally acquired by the h^t variation, much like a typical RNN.

3) CNN-LSTM

In this investigation, a CNN-LSTM net was utilized as the principal prediction period. With CNN, the danger of overfitting is effectively decreased, and it excels in feature extraction and generalization.

This research proposes a 1-separated convolutional level (conv1D chunks) to further exploit pre-processed input from

the LSTM, which is used as the primary predictor. LSTM provides one output at each time step, ability to treat 1D time chains and yield results. during several experiments, the activation feature of the LSTM layer was defined as a sigmoid factor, while the activation function of each convolutional layer is identified as *PreLU*. Moreover, zero padding is employed to maintain the same dimensionality between the convolutional layers' input and output[33].

Each subsequence of training set 1 is for train CNN-LSTM, and each subsequence of training set 2 is used to test it. The output of CNN-LSTM will be the prediction outcomes for each subsequence of the training set 2. In section IV, forecast accuracy will be judged based on MAE, RMSE and MAPE between the feed and CNN-LSTM forecast. The predictions will also be rebuilt using invert-EMD.

C. Data post-processing

Inverse EMD is the same as reversing EMD, and the execution flow of inverse EMD is depicted in Fig. 9 displayed. Inverse EMD is used as a Data post-processor, rebuild subseries of CNN-LSTM predictions.

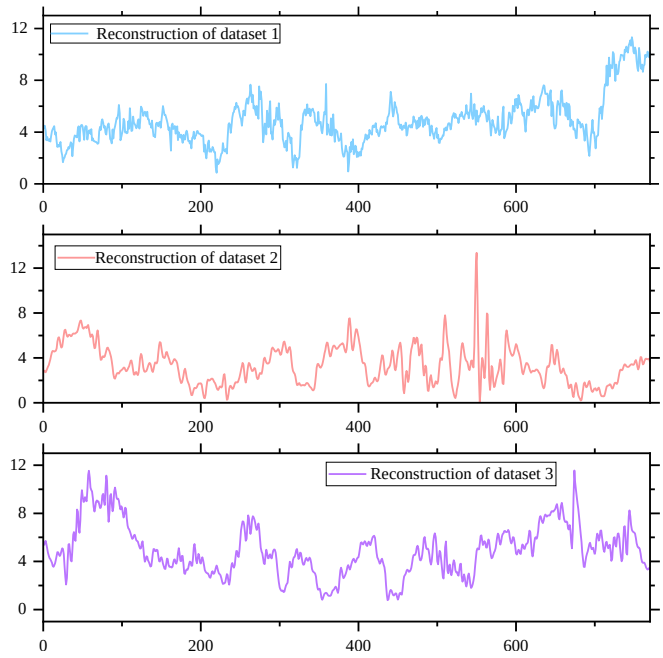

Fig.8 Reconstructed wind speed sequence

Each sequence's share of the set is evaluated according to the magnitude of the frequency threshold to determine whether the sequence meets the reconstruction conditions. As observed in Fig. 8, the training set 2 predicted sequence, also known as the reconstructed sequence in this research, is the outcome of the inverse EMD.

D. The subordinate prediction model

1) Pelican Optimization Algorithm (POA)

The Pelican Optimization Algorithm (POA), developed in 2022 by Pavel Trojovský and Mohammad Dehghani, models pelicans' natural hunting behavior[34].

The pelican is a huge bird that can grab and swallow anything because to its enormous neck pouch and long beak. The bird loves to live moving masse in colonies of hundreds of other pelicans, they often hunt jointly, dropping down from a height of 10 to 20 meters when they see their victims.

The basic idea behind program is that POA mimics the two stages of pelican assault and hunting behavior.

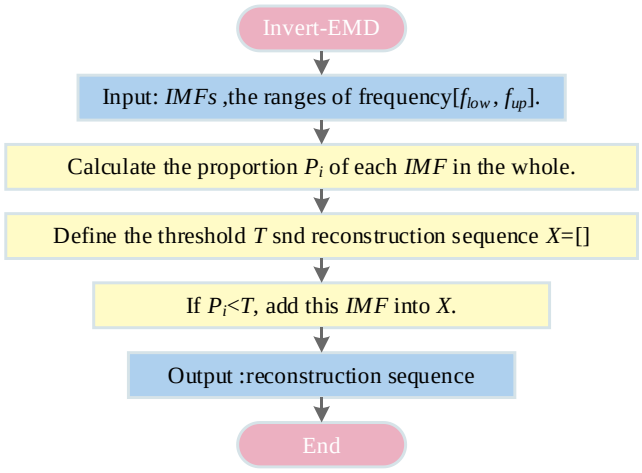

Fig.9 Workflow diagram for invert-EMD

a. Approaching prey (global search phase)

The pelican locates its prey in the first phase and then heads in that direction. Modelling this pelican tactic allows for a sweep of the search universe and takes advantage of the proposed POA's exploratory capabilities to find new areas of the search space. An important aspect of the POA refers to the prey's placement being randomly derived in the search space, which enhances the spatial aspects of the precise search problem resolved. The model in equation (20) below simulates the aforementioned idea as well as the pelican's approach to its prey's position statistically.

$$x_{i,j}^{P_1}=\begin{cases}x_{i,j}+rand\cdot(p_j-I\cdot x_{i,j}), F_p<F_i;\\ x_{i,j}+rand\cdot(x_{i,j}-p_j), \quad else;\end{cases} \tag{20}$$

I is an arbitrary integral number that can be either taken as 1 or 2. When I = 2, the shifts of each entity can be added, causing it to visit a new zone of the search space. Where pj is the hunter's position in dimension j and Fp is its value of the goal factor.

b. Surface flight (local search phase)

The second step took place when the pelican came to the top of the water, unfurling its wings, and lifts the fish into its neck pouch. By using this tactic, the pelican catches more fish in the assault area, and by simulating this behavior, proposed POA may converge to more advantageous locations in hunting region, improving local search capabilities and the capacity to utilize the POA. The behavior of the pelican during the search is mathematically depicted in equation (21) as follows: From a mathematical perspective, the algorithm must checkpoint near pelican's location to converge to a nicer place：

$$x_{i,j}^{P_2}=x_{i,j}+R\cdot(1-\frac{t}{T})\cdot(2\cdot rand-1)\cdot x_{i,j} \tag{21}$$

Where: t is the number of iterations and R=0.2 is a constant; is the maximum number of iterations.

2) Differential evolution

Based on evolutionary concepts like genetic algorithms, Rainer Storn and Kenneth Price suggested differential

evolution (DE) in 1997[35]. It is basically a multi-objective optimization algorithm (MOEAs) for the purpose of finding the overall optimum solution in a multi-dimensional space. The continuous variable is the optimization variable. [36].

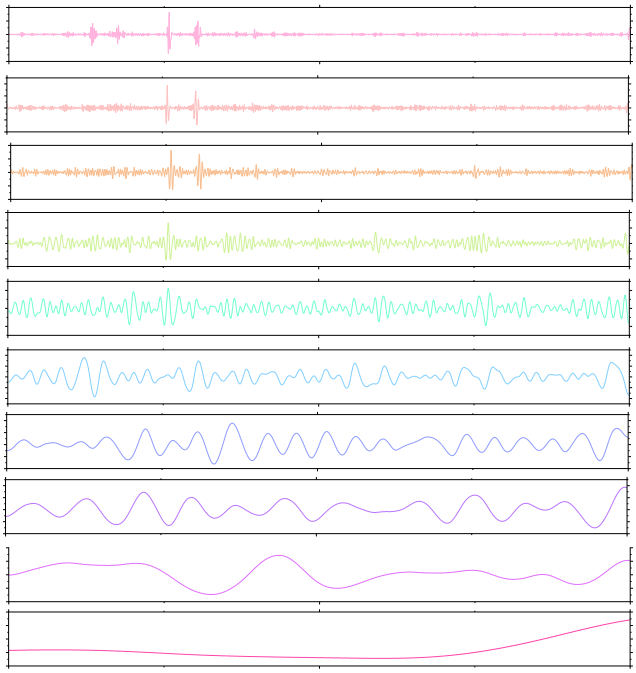

Fig.10. Pre-processed results for dataset 1

The procedure for the method is indicated below:

a) Initialization of the population: In most cases, it is done so in a haphazard manner within a predetermined area in order to locate Maximal value of f(x) at the superior border of the region [*a*, *b*]. After this has been accomplished, The preliminary values are treated purely as a continuous spread of [*a*, *b*].

b) Variational operation: the difference is reflected in this phase by picking any three distinct values for *x1*, *x2*, and *x3* from the existing solution. This is done such that the new solution is written as $x_0=x_1+F*(x_1+x_2)$, where *F* is the variational factor.

c) Intersection operation: exchange of part of the settlement; a crossover element must be assigned.

d) Processing of bounds: after variances and intersections, the join may be placed outside the desired range; you may hence choose to substitute values according to the endpoints of the interval, or you may even select a result that is within the limit of the ranges. If the join is outside of the required interval, then you must handle the boundary in one of two ways.

e) Operation of selection: Evaluate the goodness of a resolution using an assessment filter, for instance by locating the minimum level of the flow and then feeding the closure into the function. The lesser the value, the more likely it is that the quality of the settlement is high and the more likely it is to be checked for IPOA-BP.

To address its shortcomings in global search, the difference optimization algorithm is offered as an enhancement to the Pelican optimization algorithm, A group intelligence-based optimization algorithm informed by pelican tracking strategies. Meanwhile, the initial threshold is the threshold point between the concealed and export layers in the BP neural network, which is extremely sensible to the link weights between the neurons in the input, concealed and export layers. Nevertheless, The starting weapons and triggers of the BP neural network are chosen stochastically, despite the fact that these factors have a considerable effect on the capacity of the product. to make accurate predictions. Optimizing the BP neural network's initial weights and thresholds using IPOA helps make the network less restrictive and the prediction model more accurate. Figure 11 depicts the three-dimensional structure of IPOA-BP.

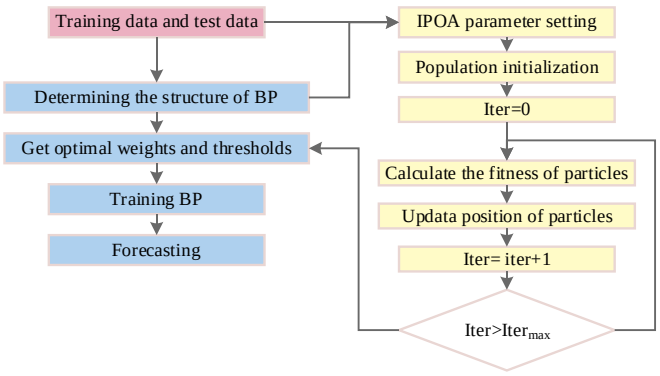

Fig.11 The structure of IPOA-BP

In addition, parameters have a role in determining the prediction accuracy, making it equally important to choose appropriate parameters. The proposed model parameters presented in TABLE II are based on extensive experimentation and earlier empirical formulations.

TABLE II
PARAMETER INITIALIZATION

Parameter name	Value
Number of iterations	100
Population size	30
Evolution times	30
Number of input layer nodes	5
Number of hidden layer nodes	11
Number of output layer nodes	1

A series of tests comparing the benchmark functions before and after the modified Novel algorithms were formulated to test the potency of the IPOA algorithm as suggested in this study. TABLE III displays the actual test functions that were run. Each of the F1-F4 benchmark functions stands for a unique concept: The accuracy of this approach to optimization is primarily checked using two features, F1 and F2, which are separable for unimodal variables (US) and non-separable for unimodal variables (UN). The functions F3 and F4 are polymodal detachable and polymodal non-detachable respectively. [37].

These problems were selected to test the algorithm's capacity to identify global optimization since they are non-linear and feature local optima. The experimental results for the benchmark function are displayed in iterative comparison graphs for each test function in Fig.12–15.

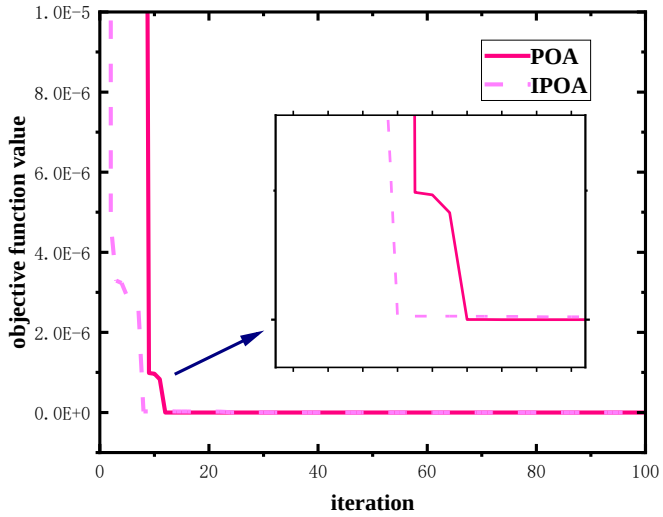

Fig.12 convergence graphs of the benchmark function F1

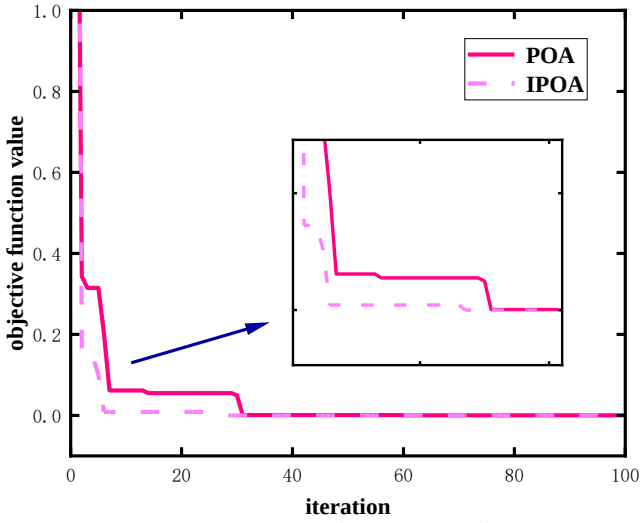

Fig.13 convergence graphs of the benchmark function F2

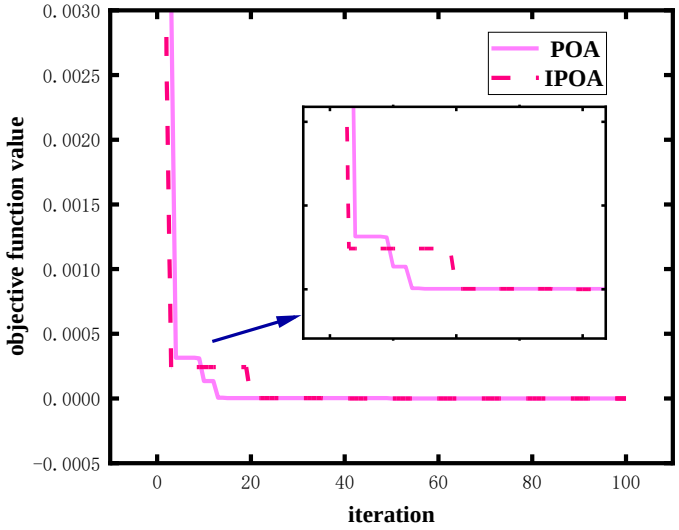

Fig.14 convergence graphs of the benchmark function F3

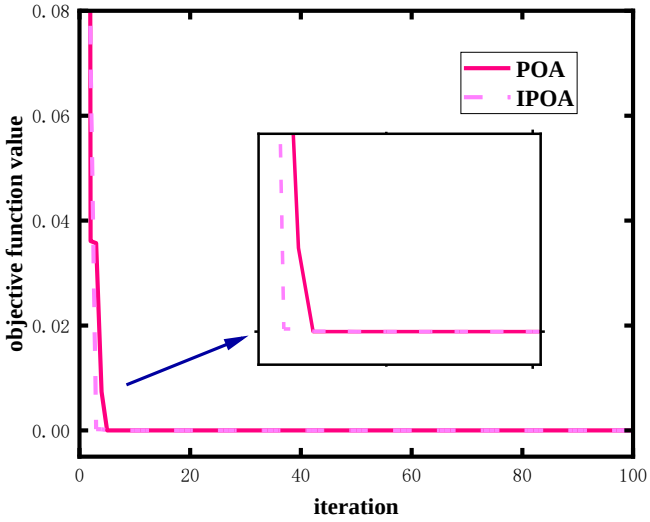

Fig.15 convergence graphs of the benchmark function F4

TABLE IV displays the mean, standard deviation, worst, and best values of the output values of the four benchmark functions employed to statistically examine the performance of search strategy, keeping in mind stochastic character of the algorithm utilized.

It is easy to show that IPOA is superior in both convergence speed and convergence accuracy compared to POA by comparing their respective convergence curves under four distinct benchmark test functions. Moreover, as shown in TABLE IV, the enhanced Pelican optimization method improves upon the original Pelican optimization algorithm in terms of optimization accuracy (by 51.6%) and stability (by 46.3%).

TABLE III
BENCHMARK FUNCTION

Function	Feature	Function expression	Boundary	Dim
F1	US	$f_1(x)=\max\{\|x_i\|,1\le i\le D\}$	$[-100,100]^D$	30
F2	UN	$f_2(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)+(x_i-1)]$	$[-30,30]^D$	30
F3	MS	$f_3(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{D}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{D}u(x_i,5,100,4)$	$[-50,50]^D$	30
F4	MN	$f_4(x)=-20\exp(-0.2\sqrt{D^{-1}\sum_{i=1}^{D}x_i^2})-\exp(D^{-1}\sum_{i=1}^{D}\cos 2\pi x_i)+20+e$	$[-32,32]^D$	30

TABLE IV
PERFORMANCE COMPARISON of ALGORITHMS

Benchmark function simulation results					
Function	Algorithm	*MEAN*	*STDEV*	*BEST*	*WORST*
F1	POA	3.3421E-04	4.4556E-05	2.5548E-47	0.0425
	IPOA	**3.3745E-07**	**7.0254E-11**	**6.2548E-69**	**0.0118**
F2	POA	0.0625	0.0278	2.4815E-04	0.8077
	IPOA	**0.0338**	**0.0145**	**1.6818E-06**	**0.5255**
F3	POA	3.5698E-04	0.0267	4.5692E-04	0.0347
	IPOA	**5.4421E-12**	**0.0197**	**1.3841E-06**	**0.0145**
F4	POA	2.6481E-03	7.1854E-04	8.1917E-16	0.0589
	IPOA	**5.1278E-07**	**6.6157E-07**	**3.5618E-25**	**0.02468**

IV. RESULT IN ANALYSIS

A. Description of wind speed data

The analysis used 15-minute data on average wind speeds gathered from a wind park in Jiangsu Province. The data are monthly measurements of Wind speed captured in real time at the same point in order to simplify comparison and obtain regularity.

The IPOA-BP model was trained using the first 1800 values as the training set for a CNN-LSTM predictor and then validated using the obtained prediction sequences. The last 200 values serve as a test set for the final prediction, and These outputs were used to estimate the usefulness of the suggested model.

B. Data pre-processing results

The input wind speed is preprocessed in the hybrid model using SSA and CEEMDAN, which performs operations including denoising and decomposition. Figure 10 displays the data before to processing. To generate the fuzzy entropy values of each IMF component, 500 sets of Gaussian white noise with a non-standard variance of 0.2 were applied. Overlaying and recombining the components with considerable correlation and above led to the FE-IMF, and the computational scale was lowered to prevent error buildup from over-decomposition.

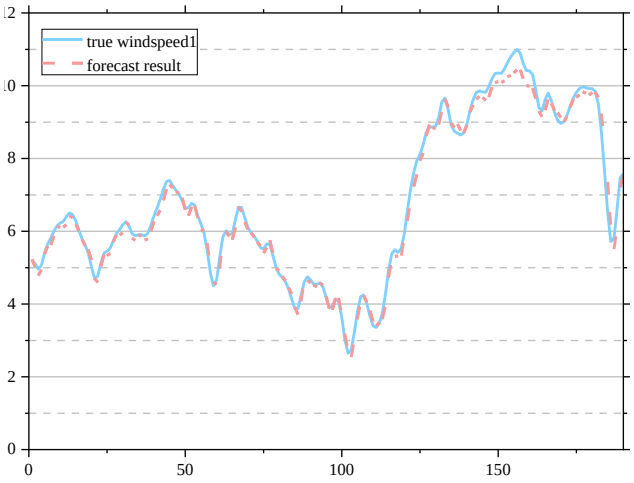

Fig.16. Prediction results of the model in dataset 1

C. Performance evaluation indicators

Researchers have devised and used a wide variety of measures for measuring performance in published works. The same four error criteria—MAE, MSE, RMSE, and MAPE—have been used to evaluate the precision of wind speed forecasts. TABLE V explains what they signify and how to precisely calculate them.

TABLE V
THE PERFORMANCE EVALUATION INDEX

Index	Meaning	Formula
MSE	Mean Square Error	$MSE=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$
MAE	Mean Absolute Error	$MAE=\frac{1}{n}\sum_{i=1}^{n}\lvert y_i-\hat{y}_i\rvert$
RMSE	Root Mean Squared Error	$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$
MAPE	Mean Absolute Percentage Error	$MAPE=\frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{\hat{y}_i-y_i}{y_i}\right\rvert\times 100\%$

Where y_i represents the factual values, $\hat{y}_i$ represents the anticipated values, and n represents the predicted amount of wind speed. Positive and negative prediction errors may be mutually cancelled if MAE and RMSE have the same amplitude. The degree of variability in the underlying data can be estimated by the MSE; the lower the MSE score, the more precise the predictive model. MAPE has high resilience since it the differences between actual and predicted amounts are not only inspected, but also the wireless link between the erroneous and real figures. The model's performance is connected to the metrics, with lower metrics indicating higher performance.

D. Prediction results and comparative analysis

The forecast results of the model put forward in this research are shown in Figures 16-18. This is done before comparing the results to those of other models. T clarify the utility of the models that have been suggested, six extremely typical prediction models, including SSA-ARIMA, SSA-Elman, SSA-CNN-LSTM-POA-BP, SSA-PSO, and LSTM, as well as the proposed models themselves, have been chosen for assessment and analysis in this work. The final prediction result is determined by taking average value obtained from 10 separate executions of each model. This is done because the process of making predictions using neural network prediction models is subject to a certain amount of random interference, which means that the results will not be the same for each operation.

The accuracy of the models' predictions was evaluated using one of four distinct fundamental metrics, and the results of those evaluations are shown in Tables VI-VIII, below.

As shown in the TABLE VI-VIII

a) Neural network models such as SSA-LSTM, SSA-CNN-LSTM, and intelligent algorithm optimized SSA-PSO-BPNN achieve performance metrics that are much superior

than those of SSA-ARIMA and SSA-Elman models. This is in comparison to standard single models. For illustration, the Model has a MAPE of 0.89% in dataset 2, which is much lower than the 3.08% that SSA-ARIMA and Elman both achieve.

For sample, in dataset 1, the MSE, MAE, MAPE, and RMSE of the model in this article were 0.0031, 0.0481, 0.8923%, and 0.0596, respectively, whereas the SSA-PSO-BPNN had MSE, 0.492, 0.9143%, and 0.0618. The four values of performance indicators achieved by the present study's model are below the limits reached by the preceding two deep learning brands. In practice, suggested hybrid model offers a high level of dependability.

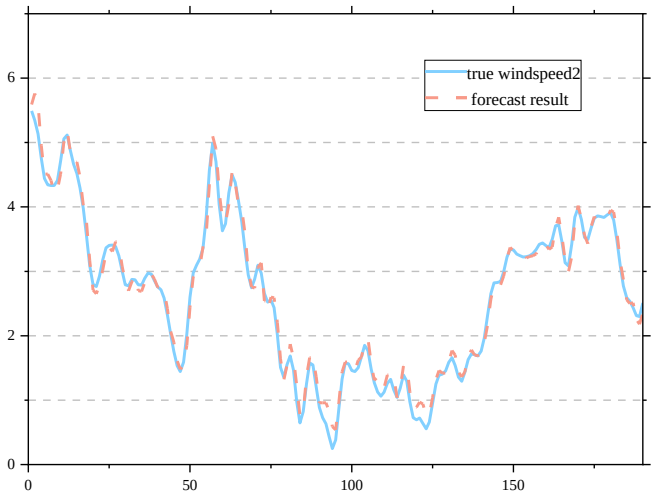

Fig.17. Prediction results of the model in dataset 2

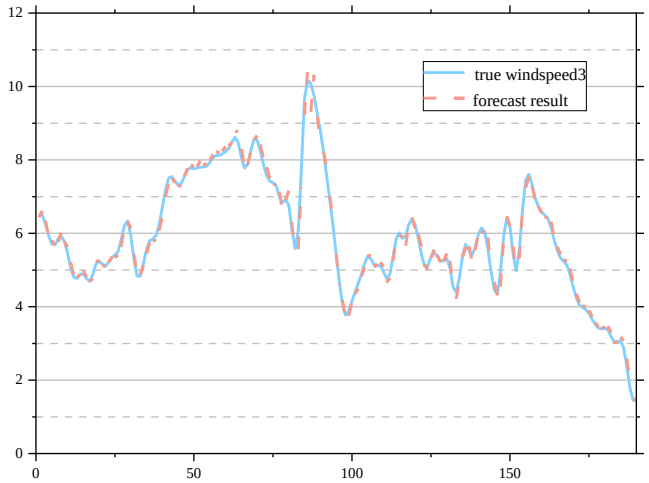

Fig.18. Prediction results of the model in dataset

b) For the three experimental datasets, the hybrid model outperformed the single model in terms of predictive performance. In dataset 3, for example, the MAPE and RMSE of the SSA-LSTM model were 4.45% and 0.316, respectively, whereas the SSA-CNN-LSTM values were 4.098% and 0.281. Moreover, the SSA-CNN-LSTM model's LSTM runtime was around double that of the SSA-CNN-LSTM model.

This suggests that the present hybrid approach, which includes data pre-processing, data breakdown, and intelligent algorithm optimization, is more dependable in terms of boosting prediction accuracy. In this study, SSA-CNN-LSTM model picked as dominant predictor because to its comprehensiveness.

c) The prediction model developed in this research outperforms the SSA-PSO-BPNN optimized using the particle swarm method in terms of accuracy. This occurrence shows that utilizing a profound learning network as the prime predictor may increase the prediction model's performance.

TABLE VI
FORECASTING ERROR CRITERIA OF DIFFERENT MODELS FOR DATASET 1

model	MAE	MAPE	MSE	RMSE
Proposed	**0.0471**	**0.8913**	**0.0035**	**0.0592**
SSA-POABP	0.0491	0.9144	0.0035	0.0617
SSA-PSOBP	0.0497	0.9437	0.0034	0.0604
SSA-CNN-LSTM	0.0494	0.9258	0.0032	0.0622
SSA-LSTM	0.0742	1.7194	0.0085	0.0945
SSA-Elman	0.2048	3.0842	0.0731	0.0608
SSA-ARIMA	0.1661	3.0802	0.0441	0.2100

d) The lower the index value for the wind speed projection model, the greater the model capability. Despite the fact that the metrics of SSA-PSO-BPNN in dataset 3 are poorer than the models in this study, such as MAPEs of 1.135% and 1.892% for the models in this paper and SSA-PSO-BPNN, predictive precision and stable model in this article are better than the models in the comparison. Also, the model's indicator values are lower. This occurrence suggests that the model is capable of meeting the standards for high accuracy in wind speed predictions.

e) In terms of accuracy performance indicators, the model's indicator values derived on the three data sets have a fluctuation range of less than 0.2, shows that the model has a high broadening power and can be adapted to different wind velocity data sets.

V. CONCLUSION

Improving the reliability and consistency of wind speed predictions is the primary goal of this experiment, this research proposes a hybrid model based on SSA, CEEMDAN, fuzzy entropy, invert-EMD, CNN-LSTM, and IPOA to optimize the BP neural network. To increase data quality, the model employs SSA for data pre-processing and noise reduction on raw wind speed data acquisition at 15-minute headways. The experimental findings suggest that SSA may enhance data quality and lower final prediction error after denoising.

TABLE VII
FORECASTING ERROR CRITERIA OF DIFFERENT MODELS FOR DATASET 2

model	MAE	MAPE	MSE	RMSE
Proposed	**0.0787**	**2.0234**	**0.0095**	**0.0957**
SSA-POABP	0.0823	2.1211	0.0108	0.1062
SSA-PSOBP	0.0933	2.2744	0.0218	0.1154
SSA-CNN-LSTM	0.1045	2.45653	0.0375	0.1538
SSA-LSTM	0.0952	2.3521	0.0325	0.1527
SSA-Elman	0.6085	16.250	0.5862	0.7653
SSA-ARIMA	0.2657	6.7858	0.1174	0.3424

CEEMDAN was used to denoise the data and yield IMF ingredients with various qualities. The denoised raw wind speed data were separated into numerous sub-series, with CNN-LSTM serving as the primary predictor for each.

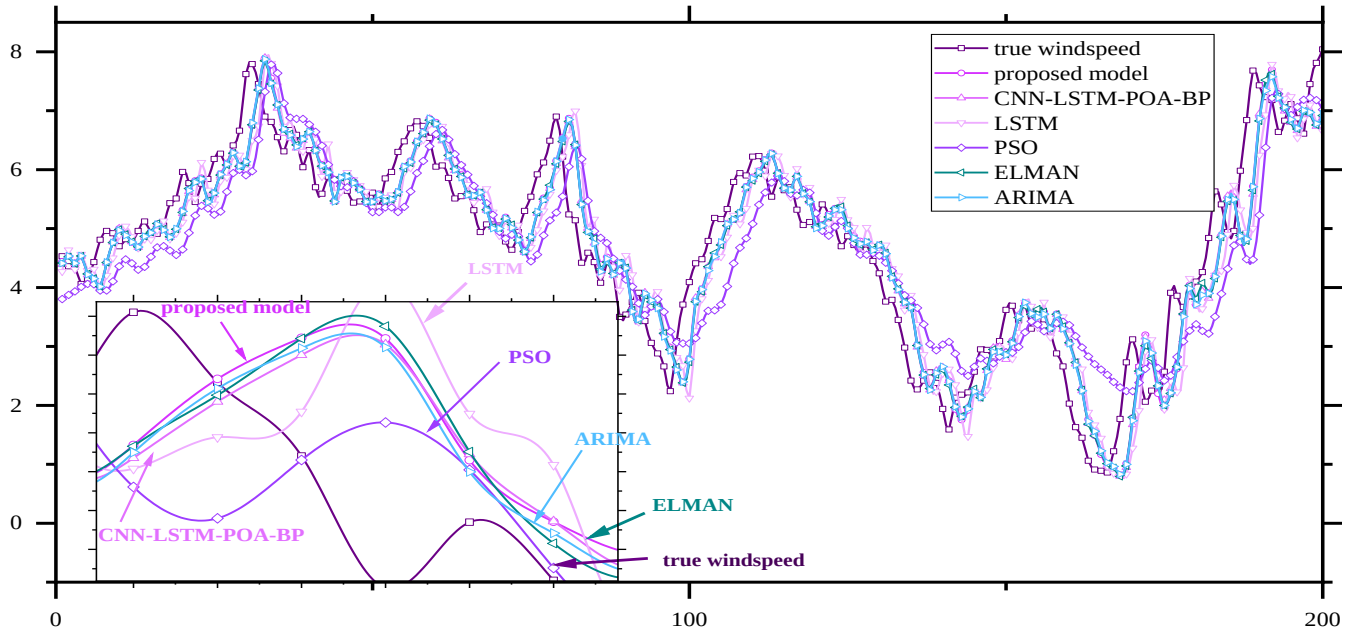

Fig.19. Prediction of dataset1

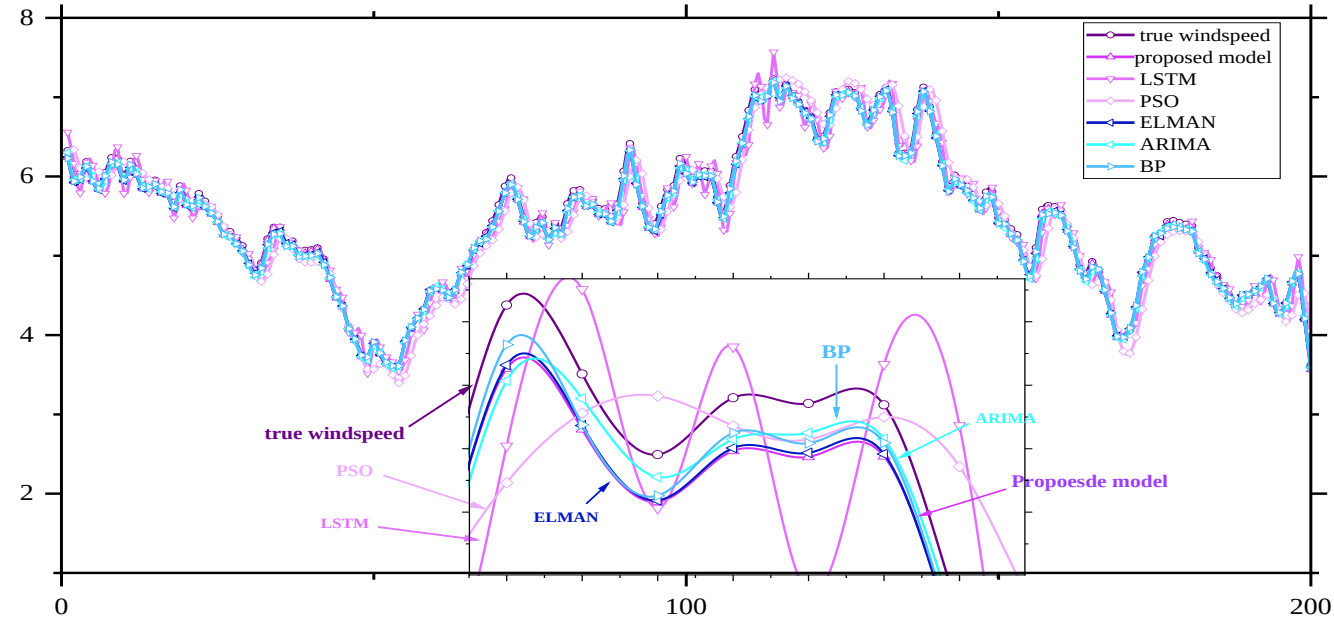

Fig.20. Prediction of dataset2

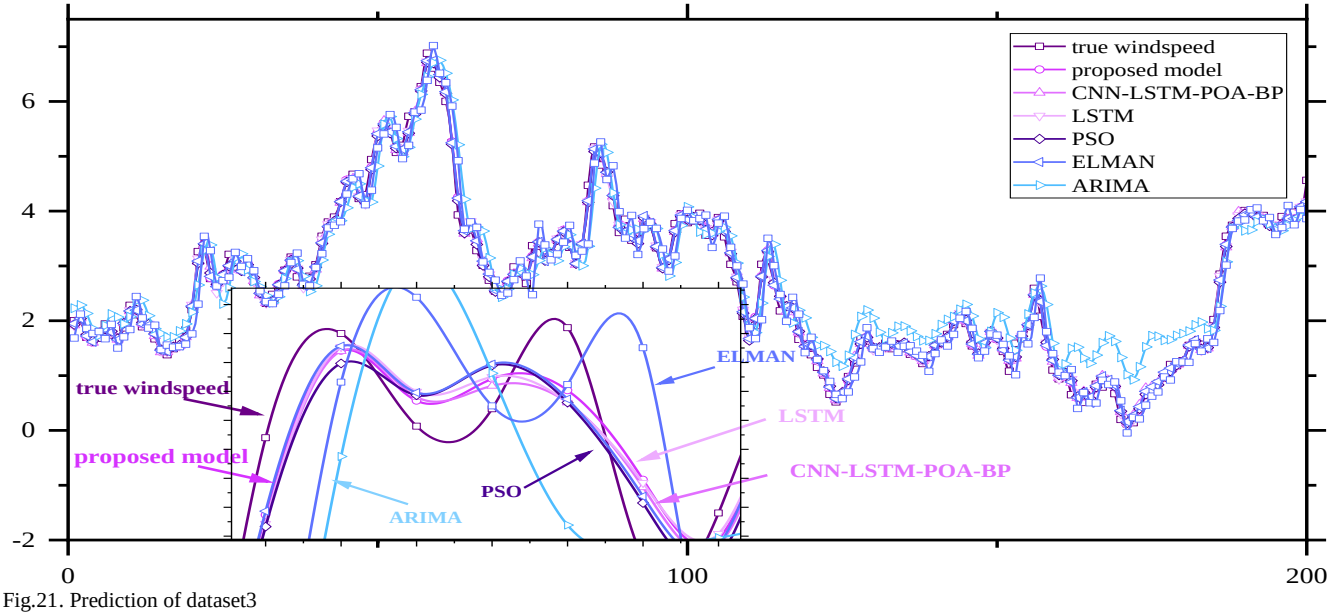

Fig.21. Prediction of dataset3

To rebuild the sub-series, data post-processing was built using invert-EMD. The sub-predictor IPOA-BPNN is utilized to train the reconstructed sequences, and the prediction sequences acquired from the main predictor are used to complete the final wind speed prediction. To create a hybrid predictor, the model incorporates the strengths of deep learning algorithms and classical artificial neural networks, while raw wind speed data is processed using data mining methods to increase data quality.

TABLE VIII
FORECASTING ERROR CRITERIA OF DIFFERENT MODELS FOR DATASET 3

model	MAE	MAPE	MSE	RMSE
Proposed	**0.052**	**1.135**	**0.006**	**0.078**
SSA-POABP	0.059	1.254	0.0073	0.083
SSA-PSOBP	0.106	1.892	0.024	0.184
SSA-CNN-LSTM	0.216	4.098	0.079	0.281
SSA-LSTM	0.234	4.450	0.092	0.303
SSA-Elman	0.714	11.889	0.735	0.857
SSA-ARIMA	0.201	5.136	0.129	0.374

Using three wind speed datasets, five different models were examined, including the suggested models SSA-PSO-BP, SSA-ARIMA, SSA-POABP, SSA-Elman, and SSA-LSTM. The performance histograms in Fig.19-21 show that:(1) the suggested models exhibit good prediction accuracy. The MAPE of the three datasets, for example, is under 2%, while the MSE is lower than 0.06. (2) The principle-dependent forecast policy depicted in this research permitted the model to leverage more data aspects and attain improved prediction accuracy. For illusions, in set 3, model's MAPE and RMSE are 1.135% and 0.078, respectively. The signal CNN-LSTM model's MAPE and RMSE are 4.098% and 0.281, respectively. (3) By removing noise from the row wind speed sequence of the data, the wind curve is smoothed out and anomalies are eliminated to some degree pre-processing. (4) The model has a high level of stability and generalizability. The MSE discrepancy between the three sets is less than 0.03 m/s, the RMSE difference is lower than 0.02 m/s, and the MAPE variation is lower than 1%, based on the preference matrix of the three data sets. (5) The model beat the other five prediction models in terms of stability, directionality, and variability. Finally, the fundamental advantage of the approach suggested in this the findings are that it incorporates the merits of both CNN and LSTM algorithms. Most present approaches are concerned with maximizing the weight parameters of classic neural networks rather than extracting the major aspects of wind speed data.

Our future research will concentrate on multi-feature forecasting. (1) To develop individual prediction models, we will propose a recent revolutionary deep learning technique like GAN and bidirectional LSTM. (2) We will fine-tune the model's structure. (3) Adding CNN-LSTM input eigen volume for projecting wind speed data based on several climatic factors such as temperature, height, humidity and wind direction. (4) To test model's flexibility, we will apply it to different kinds of data.

REFERENCES

[1] G. Memarzadeh and F. Keynia, "A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets," *Energy Conversion and Management*, vol. 213, article. 112824, 2020.

[2] M. Elsaraiti and A. Merabet, "Application of Long-Short-Term-Memory recurrent neural networks to forecast wind speed," *Applied Sciences-Basel*, vol. 11, no. 5, article. 2387, 2021.

[3] J. Shi, Z. Ding, W. Lee and Y. Yang, *et al.*, "Hybrid forecasting model for very short term wind power forecasting based on grey relational analysis and wind speed distribution features," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 521-526, 2014.

[4] Z. Liu, R. Hara and H. Kita, "Hybrid forecasting system based on data area division and deep learning neural network for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 238, article. 114136, 2021.

[5] S. Salcedo-Sanz, A. M. Perez-Bellido, E. G. Ortiz-Garcia and A. Portilla-Figueras, *et al.*, "Hybridizing the fifth generation mesoscale model with artificial neural networks for short-term wind speed prediction," *Renewable Energy*, vol. 34, no. 6, pp. 1451-1457, 2009.

[6] J. Matevosyan and L. Soder, "Minimization of imbalance cost trading wind power on the short-term power market," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1396-1404, 2006.

[7] M. Gendeel, Y. Zhang and A. Han, "Performance comparison of model with VMD for short-term wind speed forecasting," *IET Renewable Power Generation*, vol. 12, no. 12, pp. 1424-1430, 2018.

[8] Aasim, S. N. Singh and A. Mohapatra, "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting," *Renewable Energy*, vol. 136, pp. 758-768, 2019.

[9] X. Nan, Q. Li, D. Qiu and Y. Zhao, *et al.*, "Short-term wind speed syntheses correcting forecasting model and its application," *International Journal of Electrical Power & Energy Systems*, vol. 49, pp. 264-268, 2013.

[10] Y. Huang, S. Liu and L. Yang, "Wind speed forecasting method using EEMD and the combination forecasting method based on GPR and LSTM," *Sustainability*, vol. 10, no. 10, article. 3693, 2018.

[11] Z. K. Liu, P. Jiang, L. F. Zhang and X. S. Niu, "A combined forecasting model for time series: Application to short-term wind speed forecasting," *Applied Energy*, vol. 259, article. 114137, 2020.

[12] J. L. Garcia, E. C. Calderon, G. G. Avalos and E. R. Heras, *et al.*, "Forecast of daily output energy of wind turbine using sARIMA and nonlinear autoregressive models," *Advances in Mechanical Engineering*, vol. 11, no. 2, 2019.

[13] H. Liu, X. W. Mi, Y. F. Li and Z. Duan, *et al.*, "Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression," *Renewable Energy*, vol. 143, pp. 842-854, 2019.

[14] X. Y. Zhao, N. Jiang, J. F. Liu and D. R. Yu, *et al.*, "Short-term average wind speed and turbulent standard deviation forecasts based on one-dimensional convolutional neural network and the integrate method for probabilistic framework," *Energy Conversion and Management*, vol. 203, pp. 1-15, 2020.

[15] Z. D. Tian and H. Chen, "A novel decomposition-ensemble prediction model for ultra-short-term wind speed," *Energy Conversion and Management*, vol. 248, article. 114775, 2021.

[16] S. Kani and M. M. Ardehali, "Very short-term wind speed prediction: A new artificial neural network-Markov chain model," *Energy Conversion and Management*, vol. 52, no. 1, pp. 738-745, 2011.

[17] Z. D. Tian, "Modes decomposition forecasting approach for ultra-short-term wind speed," *Applied Soft Computing*, vol. 105, article. 107303, 2021.

[18] Z. D. Tian, "Preliminary research of chaotic characteristics and prediction of Short-Term wind speed time series," *International Journal of Bifurcation and Chaos*, vol. 30, no. 12, article. 2050176, 2020.

[19] Z. D. Tian, "Modes decomposition forecasting approach for ultra-short-term wind speed," *Applied Soft Computing*, vol. 105, article. 107303, 2021.

[20] X. J. Liu, H. Zhang, X. B. Kong and K. Y. Lee, "Wind speed forecasting using deep neural network with feature selection," *Neurocomputing*, vol. 397, pp. 393-403, 2020.

[21] Q. Y. Wu, F. Guan, C. Lv and Y. Z. Huang, "Ultra-short-term multi-step wind power forecasting based on CNN-LSTM," *IET Renewable Power Generation*, vol. 15, no. 5, pp. 1019-1029, 2021.

[22] Y. R. Chen, Y. Wang, Z. K. Dong and J. Su, *et al.*, "2-D regional short-term wind speed forecast based on CNN-LSTM deep learning model," *Energy Conversion and Management*, vol. 244, article. 114451, 2021.

[23] H. Liu, X. W. Mi and Y. F. Li, "Smart deep learning-based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short-term memory network," *Energy Conversion and Management*, vol. 166, pp. 120-131, 2018.

[24] Q. Yang, C. Deng and X. Chang, "Ultra-short-term / short-term wind speed prediction based on improved singular spectrum analysis," *Renewable Energy*, vol. 184, pp. 36-44, 2022.
[25] N. Wei, L. Yin, C. Li and W. Wang, *et al.*, "Short-term load forecasting using detrend singular spectrum fluctuation analysis," *Energy*, vol. 256, p. 124722, 2022.
[26] Y. Lin, Z. Lin, Y. Liao and Y. Li, *et al.*, "Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM," *Expert Systems with Applications*, vol. 206, p. 117736, 2022.
[27] Z. Liu, P. Jiang, L. Zhang and X. Niu, "A combined forecasting model for time series: Application to short-term wind speed forecasting," *Applied Energy*, vol. 259, p. 114137, 2020.
[28] M. Santhosh, C. Venkaiah and D. M. V. Kumar, "Short-term wind speed forecasting approach using Ensemble Empirical Mode Decomposition and Deep Boltzmann Machine," *Sustainable Energy, Grids and Networks*, vol. 19, p. 100242, 2019.
[29] Y. Gao, R. Bao, Z. Pan and G. Ma, *et al.*, "Mechanical equipment health management method based on improved intuitionistic fuzzy entropy and case reasoning technology," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105372, 2022.
[30] S. Harbola and V. Coors, "One dimensional convolutional neural network architectures for wind prediction," *Energy Conversion and Management*, vol. 195, pp. 70-75, 2019.
[31] H. Liu, X. Mi, Y. Li and Z. Duan, *et al.*, "Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression," *Renewable Energy*, vol. 143, pp. 842-854, 2019.
[32] H. Liu, X. Mi, Y. Li and Z. Duan, *et al.*, "Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression," *Renewable Energy*, vol. 143, pp. 842-854, 2019.
[33] A. Agga, A. Abbou, M. Labbadi and Y. El Houm, "Short-term self-consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models," *Renewable Energy*, vol. 177, pp. 101-112, 2021.
[34] P. Trojovsky and M. Dehghani, "Pelican optimization algorithm: A novel Nature-Inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, article. 855, 2022.
[35] P. Jiang, H. Yang and J. Heng, "A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting," *Applied Energy*, vol. 235, pp. 786-801, 2019.
[36] Y. Hu and L. Chen, "A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm," *Energy Conversion and Management*, vol. 173, pp. 123-142, 2018.
[37] Mousami Turuk, R Sreemathy, Sadhvika Kadiyala, Sakshi Kotecha, and Vaishnavi Kulkarni, "CNN Based Deep Learning Approach for Automatic Malaria Parasite Detection," IAENG International Journal of Computer Science, vol. 49, no.3, pp745-753, 2022