A Short-term Wind Speed Prediction Method Based on the ICPO-CNN-BiGRU-Attention Model

Lingzhi Wang*, *Member, IAENG*, Cheng Li, Wanjun Li, Chenyang Li

***Abstract*—Due to the randomness and fluctuation of wind speed, the accuracy of wind speed prediction through a single model is relatively low. In response to this issue, this paper integrated the advantages of the Improved Crested Porcupine Optimizer (ICPO), Convolutional Neural Network (CNN), Bi-directional Gated Recurrent Unit (BiGRU), and the Attention mechanism, and constructed the ICPO-CNN-BiGRU-Attention prediction model. Firstly, aiming at the disadvantage that traditional CPO algorithm is easy to fall into local optimization, and the good point set strategy is used to initialize the population, so that the population distribution is uniform. Tangent flight operator and Cauchy mutation strategy are used to optimize the position of the Crested porcupine, balance the ability of local development and global search, and further improve the convergence performance of the algorithm. Then, this model used the ICPO algorithm to optimize the learning rate, the convolution kernel size, the number of BiGRU neurons, and the key value of Attention mechanism in the combined CNN-BiGRU-Attention model, and the optimal hyper-parameter combination was given to the combined model for model training and wind speed prediction. Based on the wind speed data from a wind farm in the United States, the proposed model's prediction performance was compared with that of CNN, BiGRU, CNN-BiGRU, CNN-BiGRU-Attention, and CPO-CNN-BiGRU-Attention, and the error evaluation indexes of each model were calculated. The experimental results indicate that the proposed model in this paper is superior to others, with higher prediction accuracy and better stability, which verifies the feasibility and superiority of the ICPO-CNN-BiGRU-Attention model in wind speed prediction, and has certain practical significance and application value.**

***Index Terms*—Wind speed prediction, Crested porcupine optimizer, Combined model, Good point set initialization, Tangent flight strategy, Cauchy mutation strategy**

Manuscript received September 13, 2024; revised March 22, 2025.

This work was supported in part by the National Natural Science Foundation of China (52177194, 62073259), Xi 'an Aeronautical Vocational and Technical College power distribution terminal network load storage interactive technology innovation team project (KJTD21-002), and Research on control strategy of isolated island AC microgrid with limited communication (23XHZK-06).

Lingzhi Wang is a professor at Xi'an University of Posts and Telecommunications, Xi'an 710121, China (corresponding author to provide email: wlzmary@126.com).

Cheng Li is a postgraduate student of the School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (email: 2056866020@qq.com).

Wanjun Li is a professor at Xi'an Aeronautical Polytechnic Institute and the Innovation Team of Xi'an Aeronautical Polytechnic Institute Distribution Area Terminal Source Network Load Storage Interactive Technology, Xi'an 710089, China (email: 895367064@qq.com)

Chenyang Li is a postgraduate student of the School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (email: xupt_lcy@163.com).

I. INTRODUCTION

WITH the extraction and consumption of fossil energy sources, including coal, oil, and natural gas [1], the emission of greenhouse gases such as carbon dioxide continues to rise, aggravating the greenhouse effect of the earth. Thus, the problem of fossil energy shortage and environmental pollution has become increasingly prominent [2]. In the process of facing these challenges, the development and application of new energy have become increasingly important [3].

Wind energy is a renewable energy source that is characterized by being low-carbon and clean [4]. It not only has great development potential, but also can reduce the dependence on fossil energy and minimize greenhouse gas emissions, thereby mitigating environmental pollution [5]. However, owing to the fluctuating characteristics and instability of wind speed [6], wind speed has a certain volatility. The integration of wind power into the grid significantly affects the stability of the power system's operation [7]. Therefore, accurate forecasting of wind speed is of great significance to boost the efficiency of wind energy utilization, and reducing greenhouse gas emissions. It holds significant importance in achieving sustainable energy development and environmental protection [8].

At present, domestic and foreign experts have done a lot of study on short-term wind speed prediction. One of the commonly used prediction methods is machine learning, which includes Convolutional Neural Networks (CNN) [9], Gated Recurrent Unit (GRU) [10], Back Propagation (BP) [11], etc. However, the single model for predicting wind speed has certain limitations, and numerous researchers have proposed combined models.

Combined models integrate multiple single prediction models to enhance the prediction accuracy [12]. Chen et al. [13] built a short-term wind speed combined prediction model relying on a two - step decomposition method. Robust empirical model decomposition (REMD) and wavelet packet decomposition (WPD) were used to decompose the wind speed series, and then the decomposed data were utilized to train and forecast the LSTM-ARIMA model. The experimental outcomes demonstrate that the hybrid forecasting model achieves greater accuracy compared to individual models. He et al. [14] constructed the combined ARIMA-LS-SVM model, and corrected the prediction error of the ARIMA model by introducing the least square support vector machine (LS-SVM). The evaluation index of the combined model has been significantly improved compared with the ARIMA model. Wang et al. [15] created a

CNN-LSTM-ARIMA ultra-short-term wind speed prediction model, and the experimental findings verifies the combined model's superior predictive accuracy.

In the process of wind speed prediction, the prediction model's hyperparameters are typically adjusted manually based on experience, making it challenging to ensure the effectiveness of predictions. To enhance the model's performance, an optimization algorithm was employed to refine the model's hyperparameters [16]. Wang et al. [17] adopted an enhanced sparrow search algorithm to perform the hyperparameter optimization of the CNN - BiGRU model, and adaptively searched for the best parameter combination. The findings reveal that the ISA-CNN-BiGRU model achieves superior prediction accuracy compared to other models. Zhu et al. [18] used the Firefly improved Sparrow search algorithm (FA-SSA) to optimize the parameters of the CNN-BiLSTM-Attention runoff prediction combined model, and the optimized model has higher prediction accuracy. Cheng et al. [19] proposed a gray wolf optimization algorithm to optimize the CNN-BiLSTM ultra-short-term wind power prediction model, and the experimental outcomes validated the efficacy and advantages of the proposed model.

In this study, to minimize the error and enhance the accuracy of the wind speed forecasting model, the CNN-BiGRU-Attention model was constructed by combining the CNN network for feature extraction of wind speed data, the BiGRU network for capturing bidirectional dependency of wind speed series data and the Attention mechanism for optimizing the output weights of network models. Meanwhile, the combined model's hyperparameters were optimized by the Improved Crested Porcupine Optimizer (ICPO). The specific improvement measures are as follows:

(1) The good point set strategy is employed to initialize the population, ensuring a more uniform initial distribution.

(2) The tangent flight strategy was applied to enhance the position update formula of Crested Porcupine during the initial defense phase.

(3) The Cauchy mutation strategy was utilized to refine the position update formula for Crested Porcupine's fourth defense stage.

Then, the ICPO-CNN-BiGRU-Attention model is constructed for wind speed prediction. Experimental validation was performed using the wind speed data from a wind farm in the United States across different wind turbines. The prediction performance of the ICPO-CNN-BiGRU-Attention model was assessed by comparing it with the CNN, BiGRU, CNN-BiGRU, CNN-BiGRU-Attention, and CPO-CNN- BiGRU-Attention models. Meanwhile, the error evaluation indexes for each model were calculated to further demonstrate the proposed method's effectiveness and advantages.

II. Theoretical Basis

A. Convolutional Neural Network (CNN)

CNN is a deep learning model, which mainly includes input layer, convolutional layer, pooling layer, fully connected layer and output layer [20], and its structure is shown in Fig. 1. The input layer is used to accept the input of the original data; convolutional layer is the cores of CNN, and extracts input data's features; the pooling layer is employed to reduce the input and extract the most essential features [21]; the fully connected layer transforms the features extracted by the convolutional and pooling layers into the final output categories. Output layer is the network's prediction of the input data.

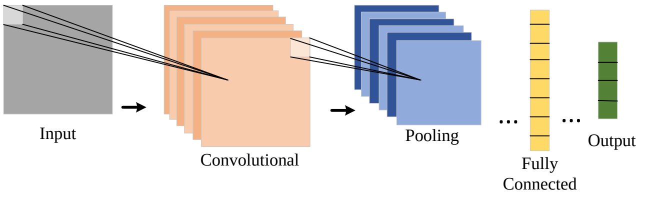


Fig. 1. The CNN structure

CNN is categorized into one-dimensional (1D CNN), two-dimensional (2D CNN), and three-dimensional (3D CNN) convolutions based on its structural design [22]. 1D CNN is utilized for processing time series data, 2D CNN is employed in image processing and text recognition, and 3D CNN primarily handles data with both time and spatial dimensions. The wind speed data analyzed in this paper belong to time series data. Thus, 1D CNN is selected to extract the features of the original wind speed data, and eliminate the latent interdependencies among various features, extraneous noise, and instability components. Then, the wind speed data processed by CNN is fed into the BiGRU for prediction.

B. Bi-directional Gated Recurrent Unit (BiGRU)

BiGRU is a special variant of recurrent neural networks (RNN) [23] with a combination of two one-way GRU networks, and is commonly applied for time series data processing. The structure of GRU includes an update gate and a reset gate, as shown in Fig. 2. The function of the reset gate r_t is to selectively forget the irrelevant information of the previous moment, and minimize the impact of irrelevant information on critical features. The update gate z_t is utilized to assess the size of the current time step in retaining the state information from the previous time step, thereby strengthening the correlation between temporal characteristics [24].

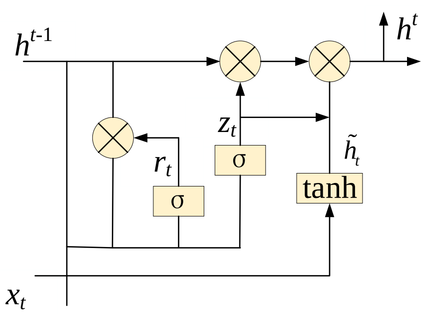


Fig. 2. The GRU structure

r_t and z_t are defined by Equations (1) and (2):

$$r_t=\sigma(w_r x_t+u_r h_{t-1}) \tag{1}$$

$$z_t=\sigma(w_z x_t+u_z h_{t-1}) \tag{2}$$

where, x_t is the input at the current time; w_r and u_r are the weight matrixes of reset gate; w_z and u_z are the weight matrixes of the update gate; h_{t-1} is the status of the previous time; h_t is the output at the current time; σ is the sigmoid activation function.

The current output h_t of the GRU is calculated by Equations (3) and (4):

$$h_t = z_t\, h_{t-1} + (1 - z_t)\, \tilde{h}_t \tag{3}$$

$$\tilde{h}_t = \tanh\,(w_z x_t + u_r r_t\, h_{t-1}) \tag{4}$$

where, tanh is the hyperbolic tangent activation function, and $\tilde{h}_t$ is the hidden state of reset gate.

In contrast to basic one-way GRU, BiGRU is capable of simultaneously utilizing information flows in both forward and reverse directions for analyzing time series. Structurally, BiGRU is a bidirectional recurrent neural network formed by integrating two GRUs that propagate in opposite directions [25]. The structure of BiGRU is illustrated in Fig. 3.

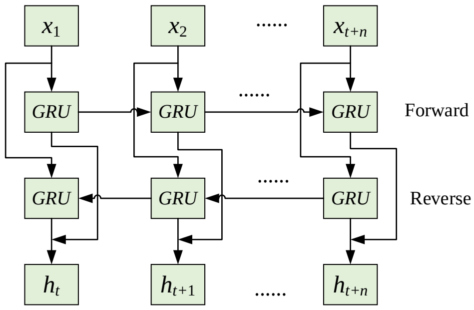


Fig. 3. The BiGRU structure

C. Attention Mechanism

The Attention mechanism, widely used in artificial intelligence, mimics the way humans allocate focus when processing information [26], and its structure is shown in Fig. 4. Through the attention mechanism, the model can effectively utilize temporal information, and concentrate on the nearest time point, thus improving the model's forecasting precision.

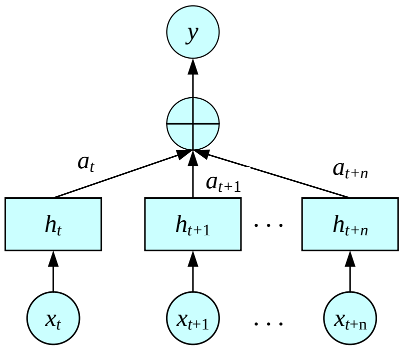


Fig. 4. The Attention structure

where, x is the input of BiGRU; h is the output of the BiGRU; a is the different weights assigned to h by the Attention mechanism; y is the output of the Attention mechanism.

D. Crested Porcupine Optimizer

The Crested Porcupine Optimizer (CPO) is an innovative intelligent optimization algorithm proposed in 2024 [27]. It draws inspiration from the four defense strategies of the Crested Porcupine (CP) against predators: sight, sound, smell, and physical attack. The whole algorithm completes the exploration and development based on the above strategies. During the optimization process, sound and sight defense mechanisms help the algorithm explore uncharted territory for a global search, known as the exploration phase. Smell and physical attack mechanisms are used to develop known information and conduct a local search, called the development phase.

(1) Population initialization

The CPO algorithm generates the initial population using Equation (5):

$$X_i = Lb + rand \times (Ub - Lb),\quad i = 1, 2, ..., N \tag{5}$$

where, Ub and Lb are the upper and lower bounds of the population, respectively; $rand$ is a random number between [0, 1]; N is the population number.

(2) Cyclic Population Reduction

The Cyclic Population Reduction (CPR) technique is a unique mechanism in the CPO algorithm, which enhances convergence speed while preserving population diversity. The CPO implemented CPR to ensure that only threatened CPs would activate the defense mechanism, while not including all CPs in the population. Therefore, those threatened CPs are removed from the population during the optimization process in this strategy. Subsequently, they are re-introduced into the population to enhance diversity and prevent entrapment in local minima. The CPR technique is simulated by Equation (6):

$$N = N_{\min} + (N - N_{\min}) \times \left(1 - \frac{rem(t, T_{\max}/T)}{T_{\max}/T}\right) \tag{6}$$

where, N_{min} is the ratio of the selected partial CP; rem is a complementary function; t is the t-th iteration; T_{max} is the maximum number of iterations; T is a cyclic variable that controls the number of times the performed CPR technique.

(3) The first defense strategy: sight

When CPs are faced with danger, they intimidate predators by flapping their feathers. Subsequently, the predators make decisions according to their proximity distance to CPs. This behavior is represented by Equation (7).

$$X_i^{t+1} = X_i^t + randn \times |\, 2 \times rand \times X_{CP}^t - y_i^t \,| \tag{7}$$

where, X^t is the individual position at the t-th iteration; $randn$ is utilized to generate random variables that satisfy the normal distribution (when $|randn| < 1$ or > -1, predators will move closer to CPs, otherwise away from CPs); X_{CP} is the optimal solution in the population at the t-th iteration; y^t is the location of the predator, and calculated by Equation (8):

$$y_i^t = (X_i^t + X_r^t)/2 \tag{8}$$

where, X_r is the position of CP randomly selected in the population, and r is the random number between [1, N].

(4) The second defense strategy: sound

When the first defense strategy does not drive the predator away, the CPs execute the second defense strategy. CPs threaten and repel predators by making noise, and this behavior is simulated by Equation (9):

$$X_i^{t+1} = (1 - U) \times X_i^t + U \times (y_i^t + rand \times (X_{r1}^t - X_{r2}^t)) \tag{9}$$

where, r_1 and r_2 are different random numbers between [1, N], respectively. U is a randomly generated binary vector, and determines whether a predator approaches CPs. When U=0, predators are away from the threatened CPs; When U=1, the predator continues to move towards CPs. Otherwise, the distance between the predator and CP stays unchanged.

(5) The third defense strategy: smell

When both the first and second defense policies fail, CPs implement the third defense strategy. By releasing an unpleasant odor to deter predators, this behavior is simulated by Equation (10):

$$\begin{aligned} X_i^{t+1} &= (1-U)\times X_i^t + U\times(X_{r1}^t + S_i^t \\ &\times(X_{r2}^t - X_{r3}^t) - rand\times\delta\times\gamma_t\times S_i^t) \end{aligned} \quad (10)$$

where, r_3 is a random number between [1, N] that differs from r_1 and r_2; S is the odor diffusion factor; δ controls the search direction; γ_t is a defense factor. S, δ, and γ_t are calculated by Equations (11)-(13).

$$S_i^t = \exp\left(\frac{f(X_i^t)}{\sum_{k=1}^{N} f(X_k^t)+\varepsilon}\right) \quad (11)$$

$$\delta = \begin{cases} 1, & if\ \ rand \le 0.5 \\ -1, & else \end{cases} \quad (12)$$

$$\gamma_t = 2\times rand\times(1-(t/T_{\max}))^{(1/T_{\max})} \quad (13)$$

where, $f(\cdot)$ is the objective function value of the i-th individual at the t-th iteration; ε is a small value to avoid being divided by zero.

(6) The fourth defense strategy: physical attack

All three existing defense strategies are ineffective, indicating that the predator is very close to CPs, and CPs execute the fourth defense strategy. CPs attack the predator until it is incapacitated, and this behavior is simulated by Equation (14):

$$\begin{aligned} X_i^{t+1} &= X_{CP}^t + (\alpha\cdot(1-rand)+rand) \\ &\times(\delta\times X_{CP}^t - X_i^t) - rand\times\delta\times\gamma_t\times F_i^t \end{aligned} \quad (14)$$

where, α is the convergence speed factor. F_i is the average force of CP affecting the i-th predator, and calculated by Equations (15)-(18):

$$F_i^t = rand\times m_i\times(V_i^{t+1}-V_i^t) \quad (15)$$

$$m_i = f(X_i^t)/(\exp(\sum_{k=1}^{N} f(X_k^t))+\varepsilon) \quad (16)$$

$$V_i^t = X_i^t \quad (17)$$

$$V_i^{t+1} = X_r^t \quad (18)$$

where, m_i is the mass of the i-th individual; V^t is the initial speed of the individual at the t-th iteration; V^{t+1} is the final speed of the individual at t+1-th iteration.

(7) Transition between exploration and development phases

In CPO, the first and second defense strategies are utilized for location updates in the exploration phase, and the third and fourth defense strategies are used in the development phase. The conversion of the two-stage location update is simulated by Equation (19):

$$X_i^{t+1} = \begin{cases} \begin{cases} Eq.(3), & rand_1 < rand_2 \\ Eq.(5), & rand_1 \ge rand_2 \end{cases}, & rand_3 < rand_4 \\ \begin{cases} Eq.(6), & rand_5 < Tf \\ Eq.(10), & rand_5 \ge Tf \end{cases}, & rand_3 \ge rand_4 \end{cases} \quad (19)$$

where, $rand_1$-$rand_5$ are different random numbers between [0, 1]; Tf is a constant value between [0, 1], and is used to alternate between two defense strategies during the development phase.

E. Improved Crested Porcupine Optimizer

As a recently developed algorithm, CPO shows robust optimization performance and fast convergence. However, it faces an imbalance between global exploration and local refinement, hindering the attainment of the global optimal solution. To address these limitations, this research introduces an enhanced CPO algorithm.

(1) Good point set initialization

Since the CPO population initialization method is randomly generated, it may lead to the initial population distribution being too concentrated, which increases the risk of the algorithm converging to local optima. Random initialization may not cover effectively all regions of the solution space, resulting in an incomplete search. To solve this problem, this paper introduces a good point set to initialize the CP (CPs) population.

Suppose Gs is a s-dimensional Euclidean geometric space, then $r \in$ Gs.

$$P_n(q) = \left\{\left(\left\{r_1^{(n)}\cdot q\right\},\left\{r_2^{(n)}\cdot q\right\}\ldots\left\{r_s^{(n)}\cdot q\right\}\right), 1\le q\le n\right\} \quad (20)$$

where, deviation φ(n) satisfies $\varphi(n) = C(r,\varepsilon)n^{-1+\varepsilon}$. C ($r$, ε) is a constant, only related to r and ε(ε>0). Then $P_n(q)$ is called a good point set and r is called a good point. The value of r is:

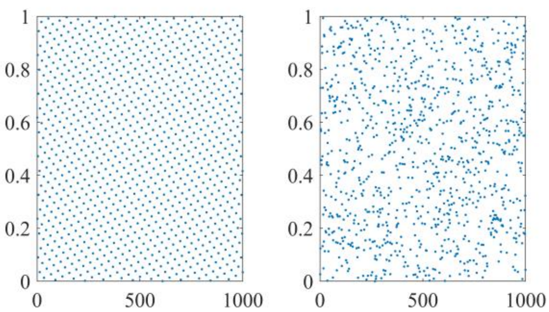
$$r = \left\{2\cos\left(\frac{2\pi q}{p}\right)\right\}, 1\le q\le s \quad (21)$$

where, p is the smallest prime number satisfying (p-3) /2⩾s.

Therefore, based on the good point set theory, the new initialization equation is:

$$X_i = Lb + \{P_n(q)\}(Ub-Lb),\ \ i=1,2,...,N \quad (22)$$

Fig. 5 shows 1000 crested porcupine individuals generated in the range [0,1] using random generation and good point set theory, respectively. In Fig. 5, the CP individuals are evenly distributed in the entire search space after initialization with the good point set theory, and the quality and stability of the initial population are improved, which helps the algorithm escape local optima and achieve the global optimal solution.



(a) Good point set (b) Random

Fig. 5. Initialization of CPs.

(2) Tangent flight strategy

In the first defense strategy phase, the distance between the predator and the CPs will either decrease or increase, which will make the algorithm's local search and global search unbalanced. So, this study introduced the tangent flight strategy to improve the CPO, its mathematical model can be described as follows:

$$f = \tan(v\frac{\pi}{2}) \quad (23)$$

$$v = randn(1,d) \quad (24)$$

where, v is a random number with a uniform distribution in the range of [0,1]; d refers to the dimension of the function.

The position update formula after introducing the tangent flight strategy in the first defense stage is modified as:

$$X_i^{t+1} = X_i^t + randn \times |\, 2 \times rand \times X_{CP}^t - y_i^t \,| \cdot \tan(v\frac{\pi}{2}) \quad (25)$$

Tangent flight balances local refinement and global exploration effectively. Tangent flight effectively balances local refinement and global exploration. The tangent flight operator acts as a scaling factor to regulate the distance between the predator and CPs, which improves the CPO algorithm's convergence and prevents it from being trapped in local optima.

(3) Cauchy mutation strategy

The predator is very close to the CPs when the algorithm reaches the fourth defense strategy stage. This results in the limitation of the search space of the predator, which makes the algorithm to easily become trapped in the local optimum and unable to conduct the global search. To solve this shortcoming, the Cauchy mutation operator is incorporated to enhance population diversity and assist the algorithm in escaping local optima during later stages. The probability density function of the Cauchy distribution is:

$$f(x) = \frac{1}{\pi(1+x^2)}, x \in (-\infty, +\infty) \quad (26)$$

Fig. 6 illustrates the curves corresponding to the Gaussian and Cauchy distributions.

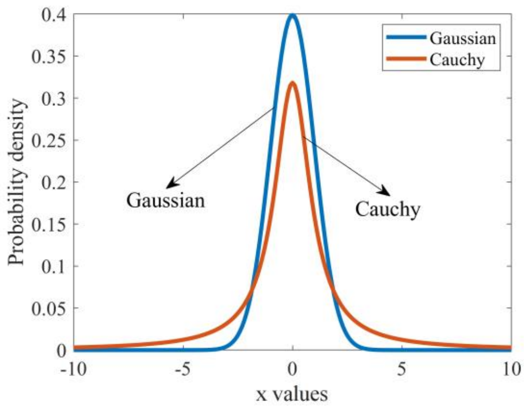


Fig. 6. Probability density function curve of Gaussian distribution and Cauchy distribution.

The Cauchy distribution is similar to the standard Gaussian distribution in that it is a continuous probability distribution. However, the value of the Cauchy distribution at the origin is smaller, and the rate of approaching zero at both ends is slower, so it can produce larger perturbations than the Gaussian distribution.

Therefore, Cauchy mutation was used to perturb the position of the CP population in the fourth defense strategy, to expand the search scale of the CPO algorithm and improve the ability of the algorithm to jump out of the local optimal. The updated formula of the fourth defense strategy after improvement is shown in Equation (27):

$$\begin{aligned} X_i^{t+1} = {} & X_{CP}^t + (\alpha \cdot (1 - rand) + rand) \\ & \times (Cauchy \times X_{CP}^t + X_{CP}^t) - rand \times \delta \times \gamma_t \times F_i^t \end{aligned} \quad (27)$$

F. Algorithm performance testing

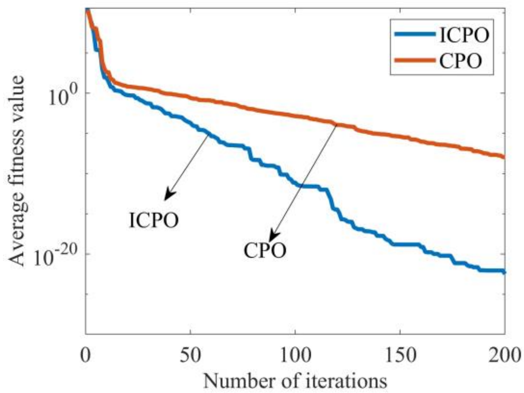
To evaluate the performance of the proposed ICPO algorithm, validation experiments were conducted using various test functions. The experiments included two unimodal and two multimodal functions, as detailed in Table I.

In Table I, f_1 and f_2 represent unimodal functions, while f_3 and f_4 indicate multi-modal functions. For the initialization of each algorithm, the population size is 30, the maximum number of iterations reaches 200, and the spatial dimension is 30. Fig. 7 shows the average convergence of the two algorithms when running independently for 30 times in different test functions.

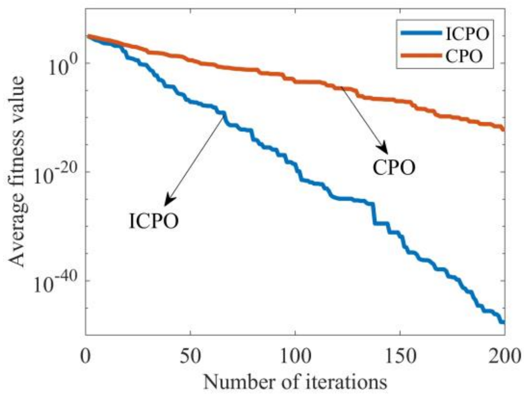
From Fig. 7, the ICPO algorithm has a smaller fitness value and faster convergence speed than the CPO algorithm regardless of unimodal functions or multi-modal functions. This confirms the optimization effect and stability characteristics of the ICPO algorithm.

TABLE I
Test functions

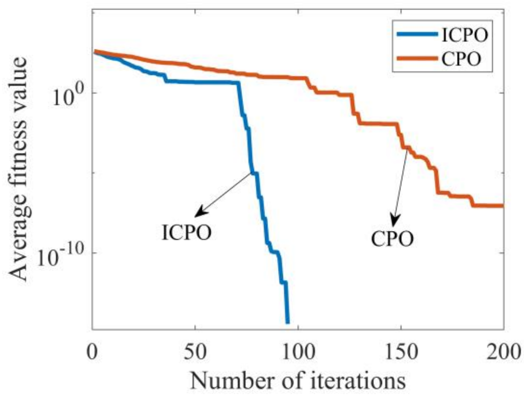
Function	Boundary	Optimal Solution
$f_1(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
$f_2(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	[-100,100]	0
$f_3(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[-5.12,5.12]	0
$f_4(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0



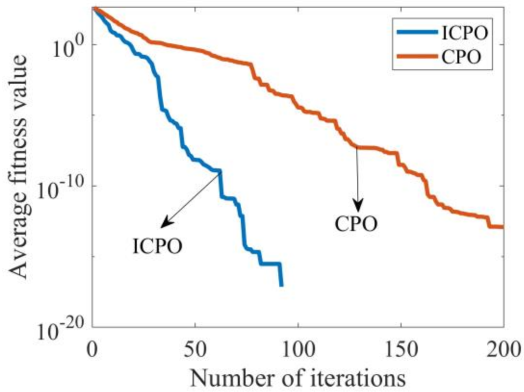
(a) f_1



(b) f_2



(c) f_3



(d) f_4

Fig. 7. Convergence curve of the test function

To comprehensively evaluate ICPO's performance, statistical metrics (best values, standard deviation, and average) were calculated for four test functions after running both algorithms independently 30 times under identical conditions. The results are presented in Table II.

Table II
The statistics of four test functions based on two algorithms

Algorithm Function	 Index	CPO	ICPO
f_1	best	1.62e-16	**2.06e-43**
	std	8.82e-08	**8.09e-27**
	average	3.81e-08	**2.29e-27**
f_2	best	8.85e-28	**8.49e-85**
	std	1.04e-09	**8.73e-53**
	average	1.95e-10	**1.59e-53**
f_3	best	0	**0**
	std	18.7263	**0**
	average	3.4189	**0**
f_4	best	0	**0**
	std	3.13e-12	**0**
	average	5.79e-13	**0**

For the ICPO algorithm, the minimum and average values of the four test functions possess the least standard deviation and are in closest proximity to the theoretical optimal solutions. This offers complete verification that the optimization performance and stability of ICPO algorithm are superior.

Besides, to prove the stability of ICPO algorithm more clearly, the box diagram was drawn in Fig. 8.

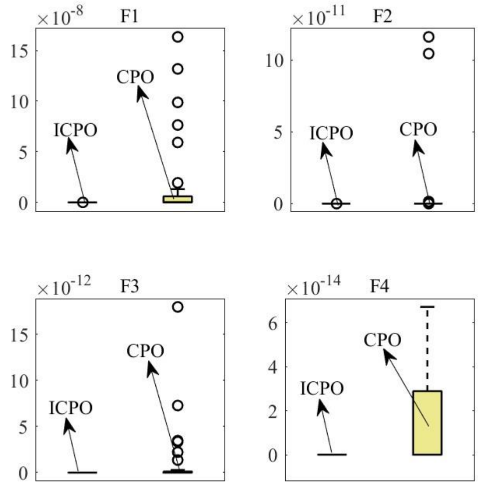


Fig. 8. Boxplot of two algorithms

In Fig. 8, the ICPO algorithm exhibits a narrower box height, lower median, and fewer outliers across all test functions, highlighting its superior performance and strong stability.

III. ICPO-CNN-BiGRU-Attention Prediction Model

With the aim of enhancing the exactness in wind speed prediction, a short-term forecasting model using ICPO-CNN-BiGRU-Attention is introduced.

First, the CNN-BiGRU-Attention model is built by combining CNN, BiGRU, and Attention, and its structure is shown in Fig. 9.

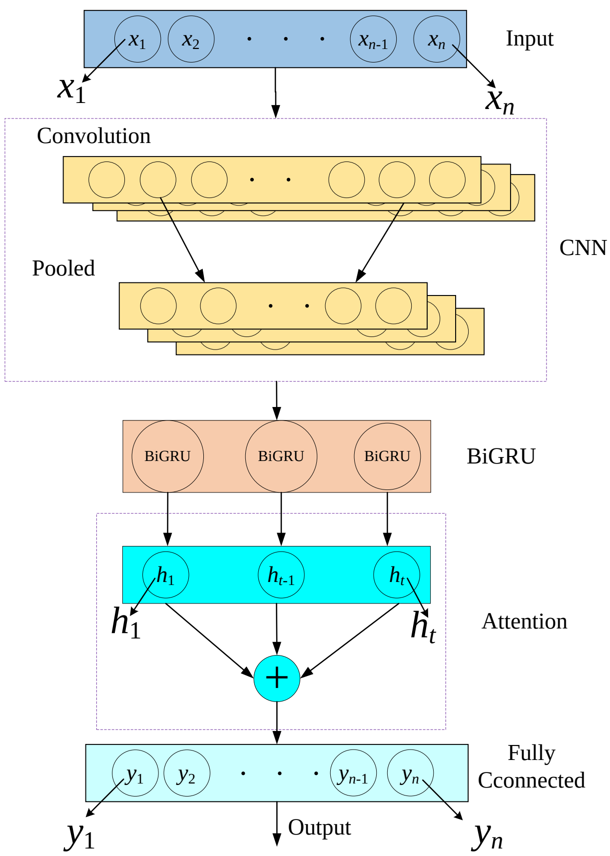


Fig. 9. The CNN-BiGRU-Attention structure diagram

The processed wind speed data are input into the model, and the convolution layer is utilized to carry out convolution calculation and feature extraction for the multi-dimensional data. The maximum pooling layer serves the purpose of extracting main features and ignoring irrelevant features, and reducing data complexity. The pooled one-dimensional feature data are sent to the BiGRU layer, fully extracting the time series features of the data, and then the time series training is carried out. The Attention mechanism is utilized for distributing diverse weights to the feature vectors processed by BiGRU, and the output wind speed is calculated through the fully connected layer. The structure of the CNN-BiGRU-Attention model includes the following units:

Step 1: Input layer. The input vector dimension is 800×10, and the wind speed data are processed into 10 columns by the rolling prediction method.

Step 2: CNN layer. This layer consists of a convolution layer and a pooling layer. The size of the convolution kernel is 3, the step size is 1, and the activation function adopts the ReLU function [28]. To retain more wind speed data information, the pooling layer utilizes a maximum pooling approach with a pooling size of 3 and a step size is 1 [29]. The data undergoes feature extraction via the convolutional and pooling layers before being passed into the BiGRU network.

Step 3: BiGRU layer. Forward and reverse BiGRU layers are employed to analyze the features extracted by the CNN layer.

Step 4: Attention layer. The Attention layer receives the hidden state h_t from BiGRU as input.

Step 5: Output layer. The fully connected layer is utilized to link with the Attention layer, the number of neurons is 25, and the Sigmoid function is adopted as the activation function.

Subsequently, the ICPO algorithm is utilized to optimize the values of four hyperparameters in the CNN-BiGRU-Attention network, reduce the risk of overfitting and enhance the model's prediction accuracy. The convolutional kernel number in the CNN convolutional layer, the neuron number in the BiGRU hidden layer, the learning rate, and the key value of the Attention mechanism were set as the optimization parameters. Fig. 10 shows the optimization and prediction flow of the ICPO-CNN-BiGRU-Attention.

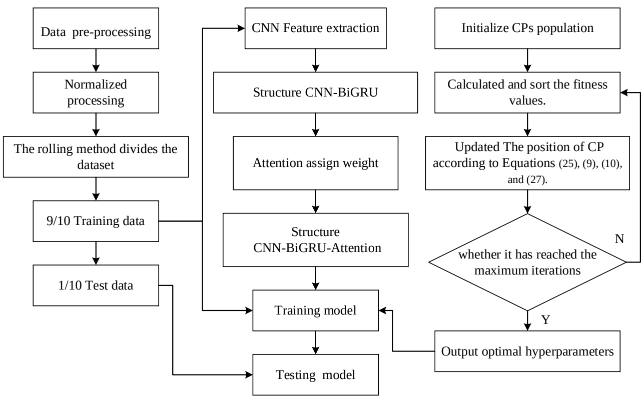


Fig. 10. The ICPO-CNN-BiGRU-Attention model flow chart

The optimization and prediction process of the combined model is as follows:

Step 1: The maximum iterations, population size, and other parameters of the ICPO algorithm were initialized, and the mean square error was selected as the fitness function.

Step 2: Initialize the ICPO-CNN-BiGRU-Attention parameters, and set the parameter optimization interval. The upper and lower limits of the four hyperparameters are as follows: [2, 10], [10, 50], [0.001, 0,01], [2, 50].

Step 3: The initial position of the CP individual was randomly generated, and calculated the fitness values. Sort and find the best and worst fitness values, representing the corresponding individual positions.

Step 4: The position of CP was updated according to Equations (25), (9), (10), and (27), and the fitness value of CP was recalculated. Meanwhile, the individual optimal position and global optimal position were updated [30].

Step 5: The algorithm judges if the maximum iterations have been reached. Once this limit is attained, the algorithm halts and outputs the optimal CP positions along with the best - performing model parameters. Otherwise, return to **step 4** and the iteration will continue.

Step 6: The ICPO-CNN-BiGRU-Attention model is trained with optimal network parameters, and then proceed short-term wind speed prediction.

IV. Example Analysis

A. Data collection and pre-processing

The experiment adopts the historical operation data of wind turbines in a wind farm in the United States, the dataset consists of a set of data every 5 minutes. The wind speed data of wind turbines A-02, A-09, and A-13 were selected as the sample data. 810 wind speed data points are chosen from 00:00 on September 1 to 19:25 on September 3 for each wind turbine.

Meanwhile, the data are processed using a rolling prediction method with n=10, where data points $x_1, x_2, \ldots, x_n$ predict x_{n+1}, and $x_2, x_3, \ldots, x_{n+1}$ forecast x_{n+2}. After processing, each dataset comprises 800 data points. The first 720 points were used as training samples, while the remaining 80 were used as test sets to assess predictive performance. The three processed data sets are represented as dataset A-02(dA-02), dataset A-09(dA-09), and dataset A-13(dA-13).

Given the substantial volume of input data for the model, this may affect the convergence performance and learning rate of the neural network. Therefore, before training and testing neural networks, the data need to be normalized, and can enhance the network's ability to extract data correlations and improve training effectiveness and accuracy. The normalization formula is shown in (28):

$$y_k = \frac{x_k - x_{\min}}{x_{\max} - x_{\min}} \tag{28}$$

where, x_k is the original data before normalization; y_k is the data after normalization; x_{max} and x_{min} are the maximum and the minimum values respectively in the dataset before normalization [31].

B. Error evaluation index

The Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-square (R^2) were used as evaluation indicators. Equations (29)-(33) show the computation formulas:

$$\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left(x_i - \hat{x}_i\right)^2 \tag{29}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(x_i - \hat{x}_i\right)^2} \tag{30}$$

$$\mathrm{MAE}=\frac{1}{M}\sum_{i=1}^{M}\left|x_i - \hat{x}_i\right| \tag{31}$$

$$\mathrm{MAPE}=\frac{1}{M}\sum_{i=1}^{M}\left|\frac{x_i - \hat{x}_i}{x_i}\right|\times 100\% \tag{32}$$

$$\mathrm{R}^2 = 1-\left(\sum_{i=1}^{M}\left(x_i - \hat{x}_i\right)^2 / \sum_{i=1}^{M}\left(x_i - \hat{x}_i\right)^2\right) \tag{33}$$

where, M indicates the number of samples, x_i represents the measured wind speed data, $\hat{x}_i$ denotes the predicted wind speed data from the six models, and $\bar{x}_i$ is the average of the measured wind speed data. Among, the smaller the values of MSE, RMSE, MAE the MAPE, the better the prediction effect of the models. The closer the R^2 value is to 1, the greater the stability for the models.

C. Model parameter settings

Adam optimization algorithm was adopted to optimize the network parameters, and ICPO algorithm parameters were set according to the literature [27]. The parameters were set as shown in Table III.

TABLE III
Parameter setting

Argument	Parameter setting	Argument	Parameter setting
GRU training times	1000	T	2
Learning rate	0.001	α	0.2
Regularization parameter	0.001	N	8
convolution kernel	3	T_{max}	10
pooling size	3	N_{min}	6
Tf	0.8		

D. Experimental results and analysis

To assess the predictive performance of the ICPO-CNN-BiGRU-Attention model, six models (CNN, BiGRU, CNN-BiGRU, CNN-BiGRU-Attention, CPO-CNN-BiGRU-Attention, and ICPO-CNN-BiGRU-Attention) were experimentally verified using three datasets from the different wind turbine, and the prediction results were shown in Figs. 11-13. Meanwhile, the error evaluation indexes of six models based on different datasets are calculated and listed in Tables IV-VI.

For the prediction results of dA-02 in Fig. 11, the proposed model outperforms other prediction models at numerous sample points, and those points in the 60-80 range are particularly closer to the actual measured values. Single models CNN and BiGRU have the largest prediction error, while combined models always perform better than single models.

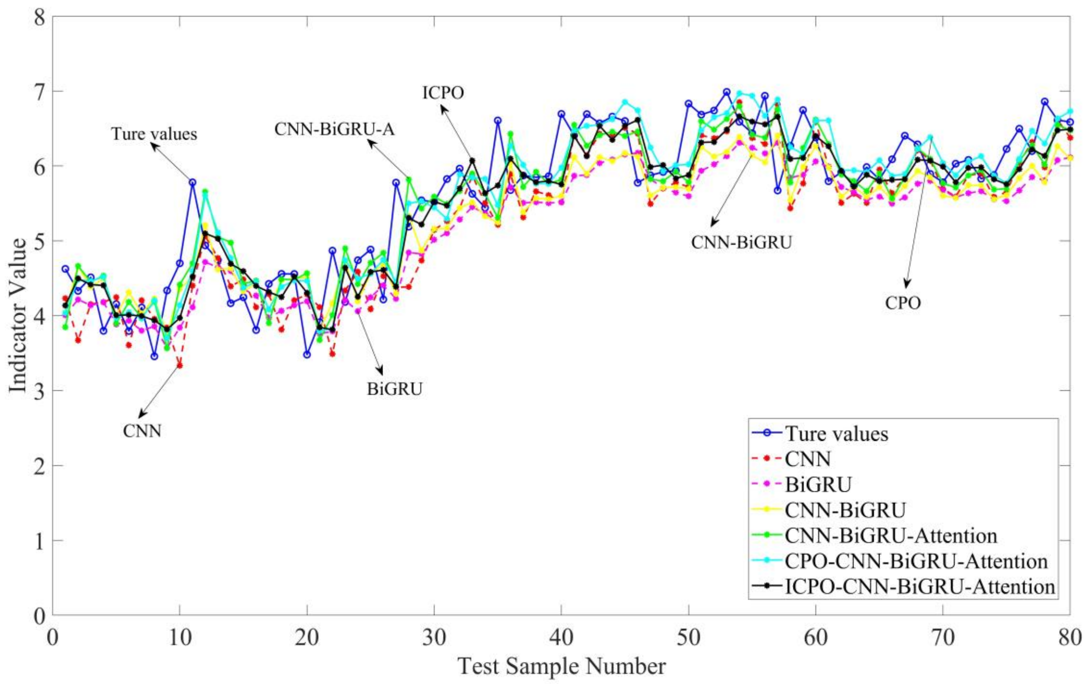


Fig. 11. Prediction comparison of six models (dA-02)

TABLE IV
Error evaluation indexes (dA-02)

Model	MSE(m/s)	RMSE(m/s)	MAE(m/s)	MAPE	R^2
CNN	0.32864	0.57327	0.44042	8.9865%	62.1057%
BiGRU	0.36665	0.60552	0.49931	9.9931%	45.6348%
CNN-BiGRU	0.31458	0.56087	0.46113	8.9135%	68.4147%
CNN-BiGRU-Attention	0.26379	0.51361	0.39398	7.6487%	71.5631%
CPO-CNN-BiGRU-Attention	0.23994	0.48984	0.36898	7.0739%	73.9673%
ICPO-CNN-BiGRU-Attention	0.21545	0.46416	0.34577	6.7482%	76.6246%

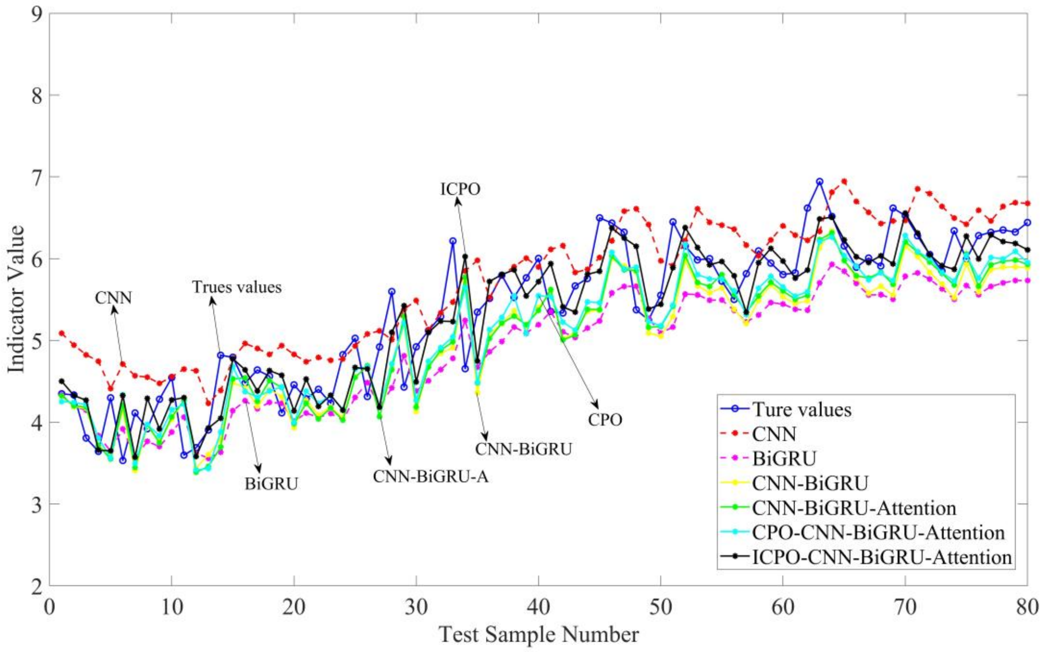


Fig. 12. Prediction comparison of six models (dA-09)

TABLE V
Error evaluation indexes (dA-09)

Model	MSE(m/s)	RMSE(m/s)	MAE(m/s)	MAPE	R^2
CNN	0.32841	0.57307	0.47216	8.4198%	47.3617%
BiGRU	0.39111	0.62539	0.53688	11.0813%	24.0449%
CNN-BiGRU	0.31103	0.5577	0.46552	9.5663%	66.2559%
CNN-BiGRU-Attention	0.28816	0.53681	0.44142	9.1204%	67.8157%
CPO-CNN-BiGRU-Attention	0.24408	0.49404	0.39663	8.1449%	69.201%
ICPO-CNN-BiGRU-Attention	0.17389	0.417	0.30886	6.1511%	78.3561%

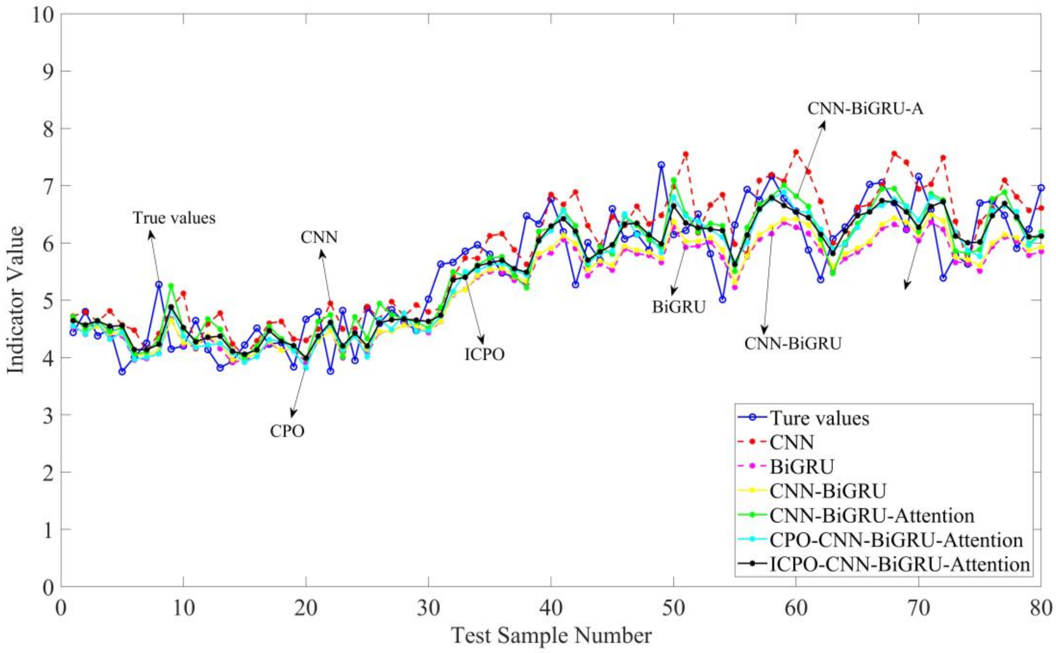


Fig. 13. Prediction comparison of six models (dA-13)

TABLE VI
Error evaluation indexes (dA-13)

Model	MSE(m/s)	RMSE(m/s)	MAE(m/s)	MAPE	R^2
CNN	0.45157	0.67199	0.51464	8.758%	59.8299%
BiGRU	0.38349	0.61926	0.49369	9.4922%	41.4462%
CNN-BiGRU	0.34983	0.59146	0.46988	8.9496%	68.4131%
CNN-BiGRU-Attention	0.3234	0.56868	0.45491	8.3591%	68.9194%
CPO-CNN-BiGRU-Attention	0.30112	0.54884	0.43881	8.2619%	71.2465%
ICPO-CNN-BiGRU-Attention	0.26646	0.5162	0.40855	7.6009%	74.3915%

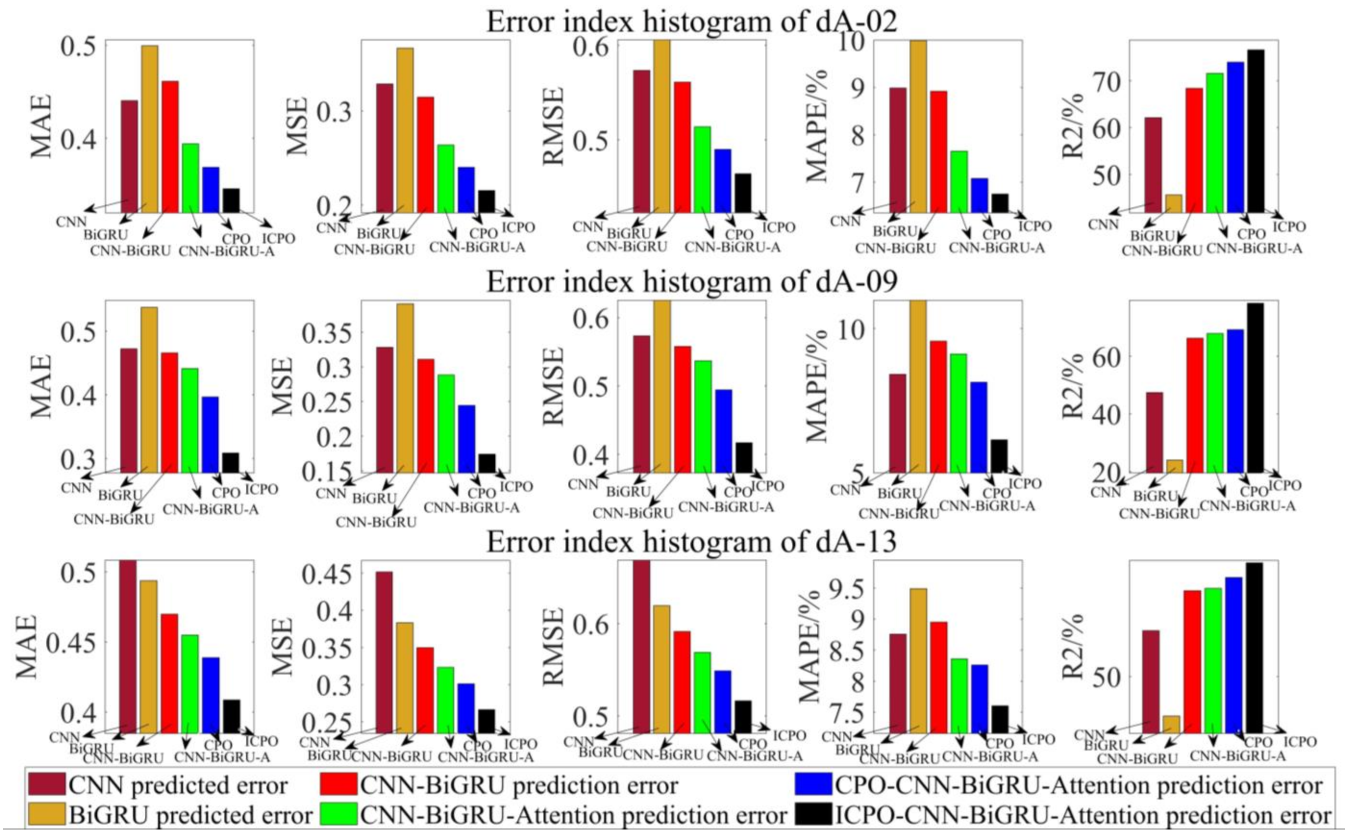


Fig. 14. Error index histogram of six models

In Table IV, the five error evaluation indexes of the ICPO-CNN-BiGRU-Attention model were compared with CNN, BiGRU, CNN-BiGRU, CNN- BiGRU-Attention, and CNN-BiGRU-Attention. MSE decreased by 34.44%, 41.2%, 31.5%, 18.3%, and 10.2%, respectively; RMSE declined by 19.0%, 23.3%, 17.2%, 9.6%, and 5.2%, respectively; MAE reduced by 21.5%, 30.8%, 25.0%, 12.25%, and 6.3%, respectively; MAPE diminished by 24.9%, 32.5%, 24.3%, 11.8%, and 4.6%; R^2 increased by 23.8%, 67.9%, 12.0%, 7.1%, and 3.6%, respectively. Therefore, the ICPO-CNN-BiGRU-Attention model manifests the most outstanding prediction capabilities.

As shown in Fig. 12, among the prediction outcomes yielded by the six models, the ICPO-CNN-BiGRU-Attention model indicated the best performance, while the single model had the largest prediction error.

In Table V, as opposed to the other five models, the five error evaluation indexes of the proposed model have been improved to a certain extent. MSE decreased by 47.1%, 55.5%, 44.1%, 39.7%, and 28.7%, respectively; RMSE declined by 27.2%, 33.3%, 25.2%, 22.3, and 18.5%, respectively; MAE reduced by 34.6%, 42.5%, 33.6%, 30.0%, and 22.1%, respectively; MAPE diminished by 26.9%, 44.5%, 35.7%, 32.5%, and 24.5%; R^2 increased by 65.4%, 225.8%, 18.3%, 15.5%, and 13.2%, respectively.

Observing the prediction results in Fig. 13, except for a few sample points, the predicted values generated by the model proposed in this paper are much nearer to the measured values compared with those of other models. Based on the computation presented in Table VI, the error evaluation index for the ICPO-CNN-BiGRU-Attention model demonstrates superior performance. So, it can be concluded that the findings align with those of dA-02 and dA-09.

For a clearer comparison of the six prediction models, their performance was assessed using MSE, RMSE, MAE, MAPE, and R^2. The results are shown in Tables IV-VI. To visualize the error metrics more effectively, histograms were used to display the error evaluation indicators for each model across the three datasets, as illustrated in Fig. 14.

Fig. 14 clearly describes the change trend of the five error evaluation indexes. Evidently, when considering different datasets, the model presented herein has the best calculation results of the error evaluation indexes and the optimal prediction performance.

In summary, the proposed model has the best prediction effect on dA-09, which is obviously better than that of the other two wind turbines. However, from the comparison of error evaluation indicators on the data of different wind turbines, the ICPO-CNN-BiGRU-Attention model always maintains the optimizer prediction performance among the six prediction models.

IV. Conclusion

Due to the randomness and variability of wind speed, achieving accurate wind speed predictions is challenging. To tackle this issue, this study introduces a short-term wind speed prediction method using a combined CNN-BiGRU-Attention model optimized by the ICPO algorithm, validated with datasets from various wind turbines.

The conclusions of this paper mainly include:

(1) This paper integrates the strengths of CNN, BiGRU, and Attention mechanisms to construct a CNN-BiGRU-Attention model, addressing the issue of low prediction accuracy for wind speed using a single model.

(2) The good point set initialization strategy, tangent flight operator, and Cauchy mutation strategy are employed to enhance the traditional CPO algorithm, with test functions confirming that the ICPO algorithm achieves faster convergence and higher accuracy.

(3) The ICPO algorithm was applied to optimize the four hyperparameters of the CNN-BiGRU-Attention model, addressing the challenges of multiple parameters and difficult selection in the combined model, and constructing the ICPO-CNN-BiGRU-Attention wind speed prediction model. This significantly enhanced the efficiency and accuracy of the model.

(4) The ICPO-CNN-BiGRU-Attention model was tested using datasets from various wind turbines to validate its universality for short-term wind speed prediction. The results demonstrate that the proposed model outperforms other comparative models.

In conclusion, the proposed ICPO-CNN-BiGRU-Attention model effectively performs short-term wind speed forecasting and shows superior predictive performance across datasets from different wind turbines. This research provides valuable insights for the power sector to enhance the dispatch of wind energy resources and holds significant practical implications.

REFERENCES

[1] H. Y. Long, Y. L. He, W. Xiang, et al. “Research on Short-term Wind Speed Prediction Based on Adaptive Hybrid Neural Network with Error Correction”. *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1290-1304, 2023.

[2] G. G. Chen, Z. J. Jiang, H. Tan, et. al. “Sliding-mode design with disturbance observer optimized by improved HBA algorithm for frequency regulation of multi-area power system”. *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 994–1013, 2022.

[3] Y, S. Huang, J. W. Li, Y. S. Li, et al. “An Improved Hybrid CNN-LSTM-Attention Model with Kepler Optimization Algorithm for Wind Speed Prediction”. *Engineering Letters*, vol. 32, no. 10, pp. 1957-1965, 2024.

[4] Y. P. Cheng, J. H. Li, S. G. Cai, et al. “Improved Maximum Power Point Tracking Control of Power Signal Feedback Method for Permanent Magnet Synchronous Generator Considering Loss”. *Electric Power*, vol. 56, no. 10, pp. 62-70, 2023.

[5] Z. Zhang, L. Lin, S. Gao, et al. “Wind speed prediction in China with fully-convolutional deep neural network”. *Renewable and Sustainable Energy Reviews*, vol. 201, pp.114623, 2024.

[6] S. C. Zhang, C. S. Zhu, X. T. Guo, “A novel combined model based on hybrid data decomposition, MSWOA and ENN for short-term wind speed forecasting”. *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 1000–1008, 2023.

[7] G. H. Yang, X. Qi, R. Jia, et al. “Short-Term Wind Power Forecast Based on CNN&LSTM-GRU Model Integrated with CEEMD-SE Algorithm”. *Electric Power*, vol. 57, no. 02, pp. 55-61, 2024.

[8] H. Shu, W. Song, Z. Song, et al. “Multistep short- term wind speed prediction with rank pooling and fast Fourier transformation”. *Wind Energy*, vol. 27, no. 7, pp. 667-694, 2024.

[9] K. W, Chen, L. Z. Wang, H. Y. Wu, et al. “Background-Aware Correlation Filter for Object Tracking with Deep CNN Features”. *Engineering Letters*, vol. 32, no. 7, pp. 1353-1363, 2024.

[10] F. X. Zhu, X. Yang, J. S. Yang, et al. “Design of Digital FIR Filter Based on Long Short-Term Memory Neural Network”. *Engineering Letters*, vol. 32, no. 10, pp. 1993-2001, 2024.

[11] X. C. Li, X. Zhang, J. Liu, et al. “Prediction of Transmission Line Icing Thickness Applying AMPSO-BP Neural Network Model”. *Electric Power Construction*, vol. 42, no. 09, pp. 140-146, 2021.

[12] J. P. Shang, W. H. Li, L. Xi, et al. “Agricultural product price prediction based on EMD-PSO-ARIMA model”. *Hubei Agricultural Sciences*, vol. 63, no. 8, pp. 1-6, 2024.

[13] H. F. Chen, H. Chen, H. Wang, et al. “Short-term wind speed prediction by combining two-step decomposition and ARIMA-LSTM”. *Acta Energiae Solaris Sinica*, vol. 45, no. 02, pp. 164-171, 2024.

[14] J. He, X. F. Wang. “Short-term wind speed prediction based on ARIMA and LS-SVM composite model”. *Mechanical & Electrical Engineering Technology*, vol. 52, no. 08, pp. 30-34, 2023.

[15] S. M. Wang, S. T. Zhang, J. Y. Lou. “Ultra-Short-Term Wind Speed Forecasting Research Based on CNN-LSTM-ARIMA”. *Advances in New and Renewable Energy*, 2024, https://link.cnki. net/urlid/44.1698. TK.20240109.1025.002.

[16] Y. F. Lu, W, L. Kong, Y. Tian, et al. “Short-term traffic flow prediction based on SSA-CNN-BiLSTM combination model”. *Transport Research*, vol. 10, no. 01, pp. 18-27, 2024.

[17] R. Wang, X. C. Xu, J. Lu. “Short-term wind power prediction: Feature selection and ISSA-CNN-BiGRU approach”. *Advanced Engineering Sciences*, vol. 56, no. 03, pp. 228-239, 2024.

[18] H. Zhu, Y. Z. Hu, M. C. Yin, et al. “Monthly runoff forecast based on CNN- BiLSTM-attention-FA-SSA”. *Yangtze River*, vol. 54, no.12, pp. 96-104, 2024.

[19] J. Cheng, D. Chen, C. Li, et al. “Ultra-short-term wind power prediction based on GWO-CNN-BiLSTM”. *Science Technology and Engineering*, vol. 23, no. 35, pp. 15091-15099, 2023.

[20] Y. L. Shu, H. L. Zhang. “Mine pressure prediction model based on sparrow optimization algorithm and CNN-BiLSTM”. *Shandong Coal Science and Technology*, vol. 42, no. 05, pp. 124-130, 2024.

[21] M. Imani. “Electrical load-temperature CNN for residential load forecasting”. *Energy*, vol. 227, pp. 120480, 2021.

[22] K. S. Liu, J. W. Tu, J. R. Zhang, et al. “Research on intelligent control algorithm of high-rise building based on convolution neural network”. *Journal of Chongqing University*, 2024, https://link.cnki.net/urlid/ 50.1044.N.20240507.1029.002.

[23] Y. S. Ye, Y. L. Xu, Y. Li, et al. “Electric vehicle load forecasting with fusion model based on similar weeks and mode decomposition”. *Power System Technology*, 2024, https://doi.org/10.13335/j.1000-3673. pst.2024.0225.

[24] H. Liu, G. Q. Li, X. J. Zhu, et al. “Exploration and application of deep learning-based wellbore deformation fore-casting model”. *Journal of China Coal Society*, 2024, https://doi.org/10.13225/j.cnki.jccs.2024. 0069.

[25] J. M. Hou, Y. Meng, Z. Li, et al. “Multi-Energy Load Forecasting in Integrated Energy Systems Based on Comprehensive Correlation Index and SA-BiGRU Network”. *Electric Power Construction*, vol. 45, no. 05, pp. 118-130, 2024.

[26] W. J. Liu, H. He, Y. N. Wang, et al. “Short-Term Power Load Forecasting Method Based on CEEMDAN and LSTM-Attention Network”. *Electric Power Construction*, vol. 43, no. 02, pp. 98-108, 2022.

[27] B. M. Abdel, M. Reda, M. Abouhawwash. “Crested Porcupine Optimizer: A new nature-inspired metaheuristic”. *Knowledge-Based Systems*, vol. 284, pp. 111257, 2024.

[28] D. W. Yu, M. Liu. “Non-invasive load decomposition of substation based deep convolutional neural network and support vector machine”. *Electric Power Science and Engineering*, vol. 36, no. 06, pp. 24-30, 2020.

[29] Y. X. Huang. “Research on short-term load forecasting in intelligent distribution network”. *Guiyang: Guizhou University*, 2021.

[30] N. Wang, Y. Q. Zhou, X. Shao. “Wind energy resource assessment of wind farm based on hybrid neural network”. *Transactions of China Electrotechnical Society*, vol. 30, no. 14, pp. 370-376, 2015.

[31] P. M. Dong, H. Tian, Z. R. Guan, et al. “Correction of moisture content detection error of crude oil based on support vector machine”. *Technology & Development of Chemical Industry*, vol. 49, no. Z1, pp. 79-82, 2020.