

Research Article

Improved Chicken Swarm Algorithms Based on Chaos Theory and Its Application in Wind Power Interval Prediction

Bing Wang , Wei Li , Xianhui Chen , and Haohao Chen 

College of Energy and Electrical Engineering, Hohai University, Nanjing 211100, China

Correspondence should be addressed to Bing Wang; icekingking@hhu.edu.cn

Received 3 December 2018; Accepted 13 January 2019; Published 30 January 2019

Academic Editor: Francesco Lolli

Copyright © 2019 Bing Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Probabilistic interval prediction can be used to quantitatively analyse the uncertainty of wind energy. In this paper, a wind power interval prediction model based on chaotic chicken swarm optimization and extreme learning machine (CCSO-ELM) is proposed. Traditional optimization has limitations of low population diversity and a tendency to easily fall into local minima. To address these limitations, chaos theory is adopted in the chicken swarm optimization (CSO), which improves its performance and efficiency. In addition, the traditional cost function does not reflect the deviation degree of off-interval points; hence, an evaluation index considering the relative deviation of off-interval points is proposed in this paper. Finally, the new cost function is taken as the fitness function, the output layer weight of ELM is optimized using CCSO, and the lower upper bound estimation (LUBE) is adopted to output the prediction interval directly. The simulation result shows that the proposed method can effectively reduce the average bandwidth, improve the quality of interval prediction, and guarantee the interval coverage.

1. Introduction

Recently, with the development of power systems and the popularization of renewable energy, distributed renewable energy has been growing rapidly, and the system complexity and uncertainty levels have increased significantly as well. As a clean and widely used resource, wind has become one of the most popular renewable sources. However, the intermittent and randomness of wind brings great challenges to power generation, transmission, and distribution. Therefore, it is necessary to predict wind power accurately to improve power quality [1, 2].

The widely used prediction models of wind power are mainly divided into physical models and statistical models. The former needs to analyse and model the internal structure of a draught fan, and the process is more complex [3, 4]. The latter requires a large amount of historical data, and models such as the artificial neural network (ANN) [5], the support vector machine (SVM) [6], and the AR integrated moving average (ARIMA) [7] are applied to analyse the nonlinear relationship between wind power and its impact factors.

However, these models are mainly focused on deterministic point prediction, and prediction error always exists and cannot be eliminated because of the randomness and non-stationarity of wind [8, 9]. Therefore, from the perspective of decision-making, the use of point prediction will cause some negative impact on the stability and reliability of the power grid.

Compared to point prediction, probabilistic interval prediction can provide more quantitative information for the uncertainty of wind power [10]. The reliability and clarity of the prediction interval can be evaluated by the confidence coefficient and average bandwidth, respectively [11]. These aspects are instructive for the decision-making of the power grid [12]. Traditional interval construction methods are often placed after the deterministic prediction model with specific prior assumptions. Quantile regression [13], Bootstrap [14], mean-variance estimation [15, 16], delta methods [17], and Bayesian [18] are usually used to obtain the prediction interval. These methods require a large amount of computation and have great complexity. In [19], an interval prediction model based on the lower upper bound estimation

(LUBE) is proposed. The dual-output neural network is applied to the output prediction interval directly, which makes the model relatively simple and efficient. In [20–22], the objective function is improved for the LUBE model, which limits the deviation of the points outside the interval and effectively reduces the average bandwidth and improves the interval quality. However, traditional neural networks (NNs) employed in the LUBE method have the problem of overtraining and high computational burden [20]. Compared to the traditional NNs, the extreme learning machine (ELM) has the advantage of simpler and faster learning speed, better generalization performance, and not having to adjust the weights of the input layer [23]. Still much further optimization of weight and threshold of ELM are need.

Chicken swarm optimization (CSO) is a new bioinspired algorithm for optimization problems [24], the chickens' diverse movements can be conducive for the algorithm to strike a good balance between the randomness and determinacy for finding the optima. Recently, CSO has aroused great concern. For example, Wu analysed the convergence of CSO [25] and the algorithm is demonstrated to meet two convergence criteria, which ensures the global convergence. However, weak local search ability of CSO limits its optimization performance; hence CSO can be improved by combining the strong search ability of other algorithms. Chaotic systems are famous for inherent ergodicity, irregularity, and pseudorandomness; these basic traits make it perform better than random operators in local searching [26]. Chaotic systems search extensively for solutions and can find a desirable solution within a practical time. In [27], the phenomenon of chaos is embedded at different stages of PSO in order to make the search process more efficient. In this paper, the local search ability of chaos is introduced into the CSO's position updating rules, and chaotic chicken swarm optimization (CCSO) is proposed and then applied to solve the wind power interval prediction.

Based on the above discussion, the detailed procedures of this study are as follows: first, the basic characteristics of the CSO and the chaos theory are considered, and the performance of CSO is improved by the chaos theory; then CCSO is proposed. Second, the performance of CCSO is tested by benchmark functions, and the results show that it performs better than other optimizations. Third, ELM is applied to learn historical data, furthermore, in order to reduce the possibility of falling into local optimum because of premature convergence; CCSO is adopted to optimize the weight and threshold of ELM [28] and then LUBE method is adopted to directly output the prediction interval of wind power. Finally, the feasibility of the model is verified by simulation.

In conclusion, the main contributions of this paper include the following three aspects. First, an improved CSO based on chaos theory is proposed in this paper, and its performance is verified to be better than before. Second, a new objective function considering deviation of the points outside the interval is proposed; then the overall quality of the interval prediction is improved. Third, a hybrid wind power interval prediction model based on ELM optimized by CCSO

is proposed, which maintains narrower average bandwidth and higher quality of interval.

The organization of this paper is as follows. In Section 2, the original CSO, chaotic optimization strategy, ELM, and LUBE method are introduced. In Section 3, improving CSO based on chaos theory is proposed, and its performance is tested; then a new objective function of interval prediction is proposed. In Section 4, the simulation results are explained. Finally, Section 5 provides the conclusions.

2. Literature Review

2.1. Original Chicken Swarm Optimization. CSO is a new bioinspired algorithm proposed by Meng in 2014 [23]. It optimizes problems by simulating the hierarchal order and the behaviors of the chicken swarm. This paper makes the following assumptions:

- (1) The whole swarm is composed of several groups, and each group is composed of a dominant rooster and several hens and chicks.
- (2) Each rooster leads its group. The chicken with the best fitness would be selected as the rooster in the group, the chickens with worst fitness would be selected as chicks, and the other chickens would be hens. Furthermore, the number of hens is greater than the number of chicks in each group.
- (3) The relationship between the hens and the rooster is randomly established, as well as the relationship between the mother hens and the chicks. These relationships remain unchanged within G iterations ($G \leq M$, M is total number of iterations) and update every G iterations to maintain the diversity of population and reduce the possibility of local convergence.
- (4) In the group, each hen forages around the rooster, each chick forages around its mother hen, and each rooster randomly steals food from other roosters.
- (5) Assume that $rNum$, $hNum$, $cNum$, and $mNum$, respectively, indicate the number of roosters, hens, chicks, and mother hens in the whole chicken swarm. N is the population size, D is the individual dimension, and $x_i^j(t)$ ($i \in [1, \dots, N]$, $j \in [1, \dots, D]$) represents the individual position in the t_{th} iteration.

In [23], the position of the rooster can be updated as follows:

$$x_i^j(t+1) = x_i^j(t) * [1 + N(0, \sigma^2)] \quad (1)$$

$$\sigma^2 = \begin{cases} 1 & f_i \leq f_k \\ \exp\left[\frac{(f_k - f_i)}{|f_i + \varepsilon|}\right] & otherwise \end{cases} \quad (2)$$

$$k \in [1, rNum], k \neq i$$

where $N(0, \sigma^2)$ is a normal distribution with mean 0 and variance σ^2 , ε denotes a minimal constant to prevent zero division errors, k denotes the rooster of the other groups, and

f denotes the corresponding fitness of the individual. The position of the hen can be updated as follows:

$$x_i^j(t+1) = x_i^j(t) + S_1 * \text{rand} * [x_{r_1}^j(t) - x_i^j(t)] + S_2 * \text{rand} * [x_{r_2}^j(t) - x_i^j(t)] \quad (3)$$

$$S_1 = \exp \left\{ \frac{(f_i - f_{r_1})}{[\text{abs}(f_i) + \varepsilon]} \right\} \quad (4)$$

$$S_2 = \exp(f_{r_2} - f_i) \quad (5)$$

where rand denotes the random numbers uniformly distributed within $[0, 1]$, r_1 is an index of the rooster which is the group-mate of the hen i_{th} , r_2 is an index of the chicken (rooster or hen) which is randomly selected from the swarm, and $r_1 \neq r_2$. The position of the chick can be updated as follows:

$$x_i^j(t+1) = x_i^j(t) + F_L * [x_m^j(t) - x_i^j(t)] \quad (6)$$

where m indicates the mother hen of the i_{th} chick, and F_L is a parameter that means the chick would follow its mother to forage for food.

2.2. Chaos Optimization Strategy. Chaotic variables of randomness, regularity, and ergodicity exist, and a logistic map is widely used. The logistic map is defined by

$$x_{n+1} = \mu * x_n * (1 - x_n) \quad (7)$$

where $\mu \in [0, 4]$, $x_n \in [0, 1]$, $n = 1, 2, 3 \dots$. When $\mu = 4$, the logistic map is in chaos region, and there are 4 fixed points in the system: 0, 0.25, 0.75, and 1. When these fixed points are removed, the chaotic variables of the logistic map show ergodicity in the interval $[0, 1]$ as long as n is large enough.

A Tent map is a piecewise linear map, which has better uniform distribution and higher iterative speed [29]. Its iterative formula is defined by

$$x_{n+1} = \begin{cases} 2\lambda * x_n & 0 \leq x_n \leq \frac{1}{2} \\ 2\lambda * (1 - x_n) & \frac{1}{2} \leq x_n \leq 1 \end{cases} \quad (8)$$

where $\lambda \in [0, 1]$, $x_n \in [0, 1]$, $n = 1, 2, 3 \dots$. When $\lambda = 1$, the Tent map is in chaos region, and there are 4 fixed points in the system: 0, 0.25, 0.5, and 0.75. In addition, there is a periodic cycle in 4 other points: 0.2, 0.4, 0.6, and 0.8. When these fixed points and periodic points are removed, the chaotic variables of the Tent map show ergodicity in the interval $[0, 1]$.

Figure 1 shows the distribution of two map sequences after 5000 iterations with the initial point ($x_0 = 0.3451$, $y_0 = 0.3452$). As we can see from Figure 1, the logistic map is relatively concentrated in the two interval distributions of $[0, 0.1]$ and $[0.9, 1]$. As a whole, its uniformity is not as good as that of the Tent map. This affects the efficiency of optimization.

2.3. Extreme Learning Machine. ELM is a single hidden layer feedforward neural network. Different from traditional backpropagation (BP) networks, input layer weights and thresholds of hidden layer neurons are randomly generated in ELM and are not adjusted after fixing, nor during the training process. The generalized matrix theory is applied in ELM to solve the output layer weight [30], which has the advantages of fast learning speed and outstanding generalization performance.

Assume that n , l , and m , respectively, indicate the number of neurons in the input layer, the hidden layer, and the output layer. For Q samples, the model output T can be expressed as follows:

$$T = [t_1, t_2, \dots, t_Q]_{m \times Q} \quad (9)$$

where

$$t_j = \begin{bmatrix} \sum_{i=1}^l \beta_{i1} g(w_i x_j + b_i) \\ \sum_{i=1}^l \beta_{i2} g(w_i x_j + b_i) \\ \vdots \\ \sum_{i=1}^l \beta_{im} g(w_i x_j + b_i) \end{bmatrix} \quad (j = 1, 2, \dots, Q) \quad (10)$$

where w_i is the input weight vector, b_i indicates the threshold of hidden layer neurons, β_{ij} is the weight between the hidden layer and the output layer, x_j is the input vector of ELM, and $g(x)$ is the activation function of the hidden layer neurons. The specific steps of ELM are described as follows.

(1) Determine the number of hidden layer neurons, and randomly set w_i and b_i .

(2) An infinitely differentiable function is selected as the activation function of the neurons in the hidden layer, and then the output matrix of the hidden layer is calculated.

(3) Calculate the output layer weight matrix H , $\hat{\beta} = H^+ T'$ (H^+ is the generalized inverse matrix of H).

2.4. LUBE Method. Traditional probabilistic interval prediction is based on the assumption of the probability distribution of wind power data or prediction errors, such as delta and Bayesian methods, as their calculation process is complicated which affects the reliability and robustness of the prediction. The LUBE method is a nonparametric statistical method that constructs the prediction interval directly [12]. Dual-output neural network is applied in LUBE to output the lower and upper bounds, without considering the distribution hypothesis. The LUBE network structure based on ELM is shown in Figure 2.

U and L are the upper and lower bounds, respectively, for the prediction of wind power. To optimize the objective function in the ELM training process, in this paper, CCSO is applied to train the network. The diversity of the chicken swarm algorithm is combined with the local search ability of the Tent map, which improves the overall optimization

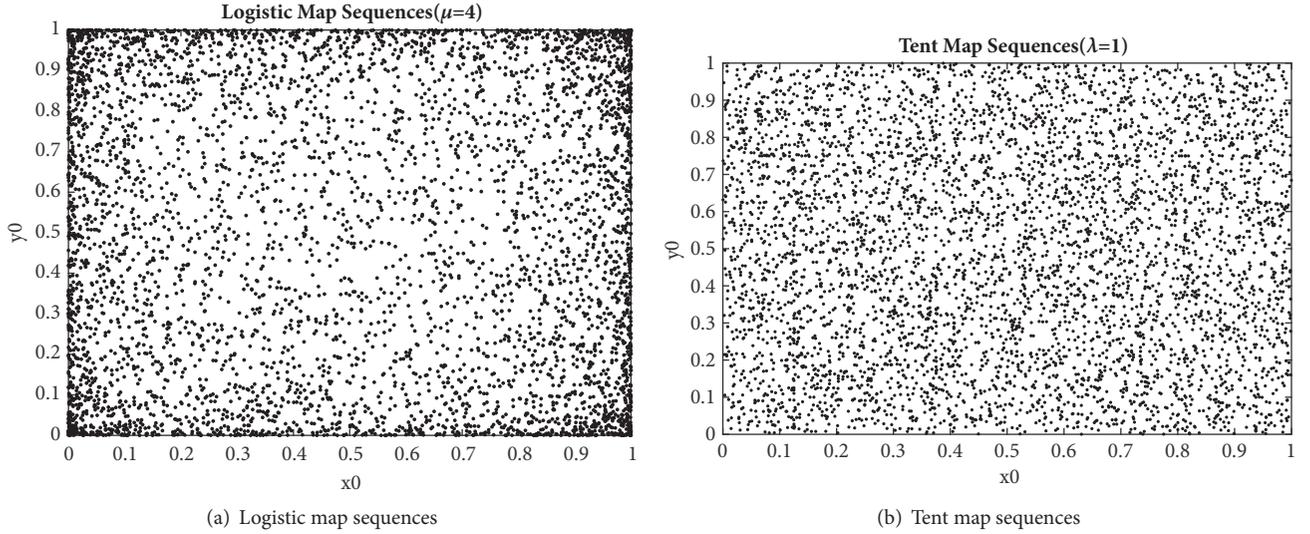


FIGURE 1: The 2-dimensional distribution of two chaotic systems.

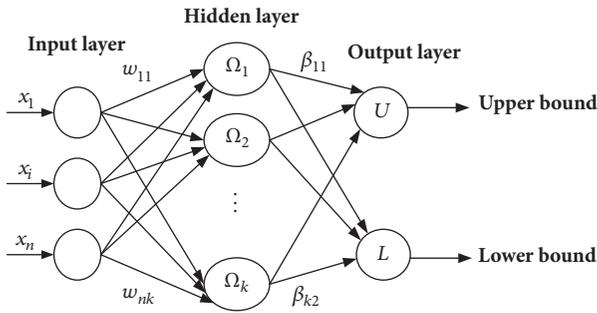


FIGURE 2: Interval prediction structure based on ELM.

performance of CCSO. A set of output layer weights β are found in the solution space to minimize F ; then U and L are taken as the outputs directly.

3. Proposed Methods

3.1. Chaotic Chicken Swarm Optimization. In this paper, the Tent map is applied to a local search based on the best individual of the chicken swarm, and the randomly selected rooster is replaced by the selected individual. Chaotic chicken swarm optimization is proposed in the end. The key points are described as follows.

(1) *Self-Adaptation Search Space.* Chaotic search is more effective in a small space, but it takes a long time to search in a large space, which affects the efficiency of the algorithm. In this paper, chaotic search space is adaptively adjusted according to the progress of the algorithm.

$$\begin{aligned} X_{\min}(d) &= bestX(d) - |bestX(d)| * \alpha \\ X_{\max}(d) &= bestX(d) + |bestX(d)| * \alpha \end{aligned} \quad (11)$$

where $X_{\min}(d)$ is the lower bound of the search space for the d_{th} dimension, $X_{\max}(d)$ is the upper bound of the search space for the d_{th} dimension, $bestX(d)$ is the d_{th} dimension of individuals with the best fitness in the chicken swarm, and $\alpha \in (0, 0.5)$ is the chaotic search factor, which is applied to adaptively adjust the search space.

(2) *Self-Adaptation Chaotic Mutation Probability.* In the early search period, chaotic mutation leads to a decrease in convergence rate. In this paper, the chaotic mutation probability is adaptively adjusted to have a smaller value in the early period and a larger value in the later period.

$$p_t = 1 - \frac{1}{1 + \log t} \quad (12)$$

where p_t is chaotic mutation probability in the t_{th} iteration. The logarithmic function is applied to increase the mutation probability during the iterative process, which improves the efficiency of the algorithm as well.

The CCSO is described as follows.

Step 1. Determine the parameters: the chicken swarm size N , the number of iterations M , the individual dimensions d , the updated frequency of the swarm G , the number of the roosters, the hens, the chicks and the mother hens which are, respectively, P_1, P_2, P_3 , and P_4 , the following coefficient F_L , the maximum step of chaotic search c_{\max} , and the chaotic search factor α .

Step 2. Initialize the swarm. The initial swarm is randomly generated between the upper and lower bounds, and the iterations are initialized.

Step 3. Update the chicken swarm according to formulas (1) ~ (6).

Step 4. Select the individual with the best fitness $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ in the current swarm. According to chaotic

mutation probability p_t , chaotic sequences are generated based on p_g . Then, map p_{gi} ($i = 1, 2, \dots, d$) to Tent's definition domain $[0, 1]$:

$$z_i = \frac{p_{gi} - X_{\min}(i)}{X_{\max}(i) - X_{\min}(i)}, \quad (i = 1, 2, \dots, d) \quad (13)$$

In addition, chaotic variable sequences $z_i^{(m)}$ ($m = 1, 2, \dots, c_{\max}$) are generated according to (8) and then restored by (12).

$$p_{gi}^{(m)} = X_{\min}(i) + [X_{\max}(i) - X_{\min}(i)] z_i^{(m)} \quad (14)$$

Compare the fitness of the new individual $\hat{p}_g^* = (p_{g1}, p_{g2}, \dots, p_{gi}^{(m)}, \dots, p_{gd})$ and the original individual and then retain the better one. Each dimension of p_g is searched c_{\max} times and the best individual is preserved. Assume that the final individual obtained from the chaotic search is \hat{p}_g .

Step 5. Randomly select a rooster in the swarm, replace it with \hat{p}_g , and then update the global optimal individual.

Step 6. If the maximum number of iterations is reached or the error requirements are met, stop; otherwise, return to Step 3.

3.2. Benchmark Function Tests. In this paper, 6 multipeak functions [31] are introduced to test the performance of CCSO compared to that of chicken swarm optimization (CSO), particle swarm optimization (PSO), and chaotic particle swarm optimization (CPSO).

Multipeak functions selected in this paper are shown in Table 1. CCSO, CSO, PSO, and CPSO were used to optimize the benchmark functions; the calculation was repeated 30 times and averaged. The parameters of CCSO are as follows: $M = 1000$, $N = 100$, $d = 30$, $G = 10$, $p_1 = 0.15$, $p_2 = 0.7$, $p_3 = 1 - p_1 - p_2$, $p_4 = 0.5$, $c_{\max} = 30$, $\alpha = 0.4$, and $F_L = \text{rand}(0.4, 0.9)$. Table 2 shows the optimization results of each function.

It can be seen from Table 2 that CCSO has higher accuracy as compared to PSO, CSO, and CPSO. The Tent map in CCSO guarantees the diversity of the swarm while avoiding premature convergence in the algorithm, which greatly decreases the possibility of falling into the local minimum. In addition, the self-adaptation of the chaotic search space and mutation probability improves the efficiency and robustness of the algorithm, and the Benchmark functions validate the optimization performance of CCSO, but its practicability needs to be further studied. Therefore, the application of CCSO in wind power interval prediction will be introduced.

3.3. Objective Function of Interval Prediction. The construction of the prediction interval consists of estimating the upper and lower bounds of the interval under a certain confidence level, which indicates that the evaluation needs to consider the accuracy and quality of the interval [14, 32, 33]. In order to evaluate the accuracy of the interval, the prediction interval

coverage probability (PICP) was introduced, which is defined as follows:

$$PICP = \frac{1}{N} \sum_{i=1}^N c_i \quad (15)$$

where N is the number of predicted points, and if the actual target is within the prediction interval, $c_i = 1$, otherwise $c_i = 0$. $PICP \in [0, 1]$, bigger PICP indicates higher accuracy, and smaller indicates the reverse.

On the other hand, to evaluate the quality of the prediction interval, PI normalized average width (PINAW) is introduced, which is defined as follows:

$$PINAW = \frac{1}{NR} \sum_{i=1}^N (U_i - L_i) \quad (16)$$

where U_i and L_i , respectively, indicate the upper and lower bound of the interval, and R is the range of target values, which is used to normalize the average bandwidth. $PINAW \in [0, 1]$, bigger PINAW indicates higher quality, and smaller indicates the reverse. PICP and PINAW can only assess the effect of the prediction interval unilaterally. Therefore, to transform complex multiobjective problems into single-objective problems, a comprehensive index consisting of both PICP and PINAW known as coverage width criterion (CWC) has been developed [34], which is expressed as follows.

$$CWC = PINAW [1 + \gamma(PICP) e^{-\eta(PICP-\mu)}] \quad (17)$$

where

$$\gamma(PICP) = \begin{cases} 0, & PICP \geq \mu \\ 1, & PICP < \mu \end{cases} \quad (18)$$

where μ is a predetermined confidence degree and η is the penalty coefficient when PICP is less than μ . CWC takes coverage rate and average bandwidth into account, but there is no evaluation of the deviation of the point out of the interval. Furthermore, known information is not fully utilized in the training process. Therefore, a third evaluation index is designed in this paper: prediction interval relative deviation (PIRD).

$$PIRD = \frac{1}{N} \sum_{i=1}^N \left| \frac{(U_i + L_i)/2 - y_i}{(U_i + L_i)/2} \right| \quad (19)$$

where y_i is the actual target value. The improved objective function adds PIRD to the original CWC; that is,

$$\begin{aligned} \text{Minimize: } F(\beta) \\ = PINAW [1 + \gamma(PICP) e^{-\eta_1(PICP-\mu)}] + \eta_2 PIRD \end{aligned} \quad (20)$$

where η_1 replaces η in CWC and η_2 is the penalty coefficient of PIRD. The penalty cost is high when $PICP < \mu$, which is unfavorable to the optimization of the objective function. When $PICP > \mu$, the penalty cost of PIRD will be considered to meet the minimum target. Both coverage rate and average bandwidth are considered by minimizing the new cost function, and the deviation of the outer points is effectively decreased. The known training information is fully utilized to acquire a narrower interval.

TABLE 1: Benchmark functions.

Benchmark Function	Bounds	Dimension	Optimum	
Sphere	$f_1(x) = \sum_{i=1}^N x_i^2$	$[-100, 100]$	30	0
Rosenbrock	$f_2(x) = \sum_{i=1}^N [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-2.048, 2.048]$	30	0
Rastrigin	$f_3(x) = \sum_{i=1}^N [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	30	0
Griewank	$f_4(x) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	30	0
Schaffer	$f_5(x) = \frac{\sin^2 \sqrt{\sum_{i=1}^N x_i^2} - 0.5}{\left[1 + 0.001 \left(\sum_{i=1}^N x_i^2\right)\right]^2} + 0.5$	$[-100, 100]$	30	0
Schwefel	$f_6(x) = \sum_{i=1}^N x_i + \prod_{i=1}^N x_i $	$[-50, 50]$	30	0

TABLE 2: Comparison of algorithm optimization performance.

Benchmark Function	PSO		CSO		CPSO		CCSO	
	Mean	Time/s	Mean	Time/s	Mean	Time/s	Mean	Time/s
Sphere	8.26e-27	2.0341	2.49e-56	4.2767	8.20e-33	4.5064	2.21e-71	5.3914
Rosenbrock	27.3155	2.4687	26.8989	4.3950	15.0153	4.7216	0.0615	5.6236
Rastrigin	10.9445	1.7184	0	4.0715	2.49e-3	4.3475	0	5.4463
Griewank	1.18e-4	2.1252	0	4.5120	0	5.0941	0	5.5376
Schaffer	0.0372	2.1998	0.0782	4.4537	0.0372	4.8122	0.0097	5.3727
Schwefel	0.5633	1.5425	5.86e-41	3.8612	1.61e-3	4.6713	1.43e-92	4.9231

3.4. Model Implementation. Based on the ELM and CCSO network model, the LUBE method is applied to construct the prediction interval. Details of the main steps are discussed below.

- (1) Data preprocessing: the training data and test data are normalized to $[-1, 1]$ to avoid errors caused by different dimensional data.
- (2) Constructing training sample set: determine the input vector $X = [x_1, \dots, x_4, x_5, x_6, x_7]$, where $x_1 \sim x_4$ are wind power data for 4 moments before the forecast point. x_5, x_6 , and x_7 relatively indicate wind speed, sine of the wind direction, and temperature of the forecast point. The original training sample (X_i, t_i) is slightly fluctuated; that is,

$$\begin{aligned} U_i &= t_i (1 + 20\% * \text{rand}) \\ L_i &= t_i (1 - 20\% * \text{rand}) \end{aligned} \quad (21)$$

where rand is a random number uniformly distributed between $[0, 1]$. U_i and L_i obtained are used as the outputs of the training data.

- (3) Parameter initialization: randomly generate input weights and thresholds of ELM, initialize the parameters of the CCSO according to Step (1) in Section 2.3, and randomly generate the initial swarm.
- (4) Constructing interval prediction mode: a double-output extreme learning machine prediction model is constructed according to the LUBE method, the output weight β is taken as an individual of the

chicken swarm, and the objective function F is taken as the fitness of the individual; the lower the fitness, the better the individual.

- (5) Iterative optimization: update the chicken swarm and optimize the objective function according to Section 3.1 Step 3~Step 6. The training is terminated when the maximum iterations are attained or the fitness of the best individual can no longer be decreased. Then, the best individual β and corresponding fitness $F(\beta)$ are taken as the outputs.
- (6) Interval calculation: the individual β obtained in (5) is used as the weight of the output layer, and the prediction interval is directly calculated by ELM. Then PPIC, PINAW, and PIRD of the interval are calculated.

4. Case Studies

In this section, two parts are introduced to verify the proposed model. The first part shows the performance of CCSO applied in interval prediction compared to other optimization algorithms. In the second part, wind power interval prediction results are analyzed under a confidence level of 90%. The wind data from an existing wind farm in China is taken as an example. In this study, the first 85% is taken as the training sample set, and the last 15% is taken as the test sample set.

4.1. Comparison and Analysis of Optimization Performance. In this paper, the PSO, CSO, CPSO, and CCSO are used to optimize the training process of ELM. Then, the performance

TABLE 3: Algorithm parameter setting.

Algorithm	Parameters
PSO	$c_1 = 1.22, c_2 = 1.49, \omega_{\max} = 1, \omega_{\min} = 0.7$
CSO	$p_1 = 0.15, p_2 = 0.7, p_3 = 0.15, p_4 = 0.5, G = 10, FL = \text{rand}(0.4, 0.9)$
CPSO	$c_1 = 1.22, c_2 = 1.49, \omega_{\max} = 1, \omega_{\min} = 0.7, c_{\max} = 30$
CCSO	$p_1 = 0.15, p_2 = 0.7, p_3 = 0.15, p_4 = 0.5, G = 10, FL = \text{rand}(0.4, 0.9), c_{\max} = 30, \alpha = 0.4$

of the 4 algorithms is compared and analyzed. The maximum iterations the algorithm can have are 300, the swarm size is 100, the confidence level is 90%, η_1 and η_2 in the objective function $F(\beta)$ are 50 and 0.3, respectively, and the other parameters are shown in Table 3. MATLAB 2014a is used to code and run the simulation. The PICP, PINAW, PIRD, and the convergence curve of the fitness function in each algorithm are shown in Figures 3–7.

Because of the penalty, the PICP of the 4 algorithms are always above 90%. If the constraints are not met, the fitness function will rise rapidly. In the early period of PSO, PINAW decreased rapidly, but it changed very little after the 40th iteration. This was because all the individuals in the swarm learned from the global best individual, lacked population diversity in the later period, and could not jump out after falling into the local extremum, that is, could not achieve premature convergence. PINAW in CSO has been decreased by 30% in the 10th generation, which proves that subgroup division method of CSO can greatly increase the diversity of the population. However, due to the lack of strong local search ability in the later period, the decline speed of PINAW slows significantly until it no longer declines. CPSO is an improved method of chaotic search combined with particle swarm optimization, in which the decline trend of PINAW in CPSO is improved obviously, and PINAW drops to about 28.5% after the 150th iteration. Subgroup division method and chaotic theory are combined in CCSO, and the population diversity and local search ability have been improved. Furthermore, self-adaptation search space and self-adaptation chaotic mutation probability are used to enhance the optimization efficiency and decrease the possibility of falling into local optimum greatly. PINAW has dropped to 27.7% at the 50th iteration. Compared to other algorithms, CCSO shows more outstanding optimization ability.

4.2. Results and Discussions. Four kinds of trained models were simulated and verified by the test sample set. The prediction results of the wind power interval of each model are shown in Figures 8–11.

Under a confidence level of 90%, the prediction interval of CCSO is narrower than that of the other three algorithms, and only a few points fall outside the interval. Furthermore, the deviation of these peripheral points is greatly decreased by PIRD. Therefore, the negative effects of points outside the interval will be minimized.

It can be seen from Table 4 that different optimization algorithms differ greatly in the prediction results. The PICP of each algorithm is higher than 90% because of the penalty and PICP of CCSO is slightly lower than that of PSO; meanwhile,

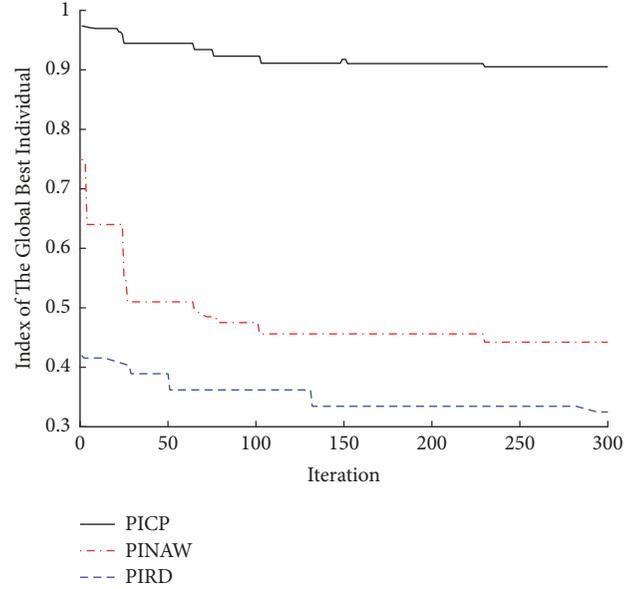


FIGURE 3: Convergence curve of PSO.

TABLE 4: Evaluation indexes of 4 models.

Algorithm	Index			
	PICP %	PINAW %	PIRD %	$F(\beta)$ %
PSO	91.59	38.71	35.94	49.49
CSO	90.86	31.25	27.16	39.40
CPSO	90.51	28.34	21.71	38.85
CCSO	91.13	25.54	16.25	30.42

PINAW and PIRD are greatly decreased. Compared to PSO, CSO, and CPSO, the PINAW is reduced by 12.77%, 5.31%, and 2.8%, respectively, and PIRD is reduced by 19.69%, 10.91%, and 5.46%, respectively, in CCSO. Overall, the proposed method effectively improves the clarity of wind power interval prediction and guarantees reliability, which is of certain reference value for the operation of decision makers.

5. Conclusion

Accurate prediction of wind power directly affects the stable operation of the power system. Traditional point prediction can only give the wind power value at the corresponding moment, and there is no quantitative analysis of the uncertainty of the error. However, interval prediction is an effective method to quantitatively predict the uncertainty of wind power, and the interval can be evaluated with both clarity and reliability. In this paper, an improved chicken swarm

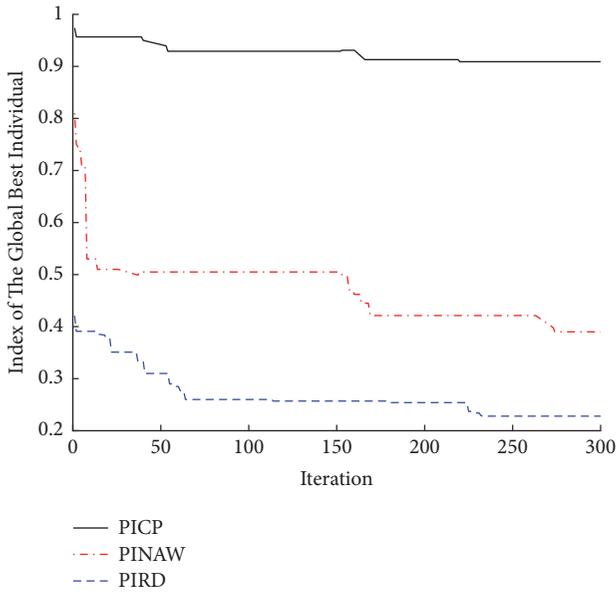


FIGURE 4: Convergence curve of CSO.

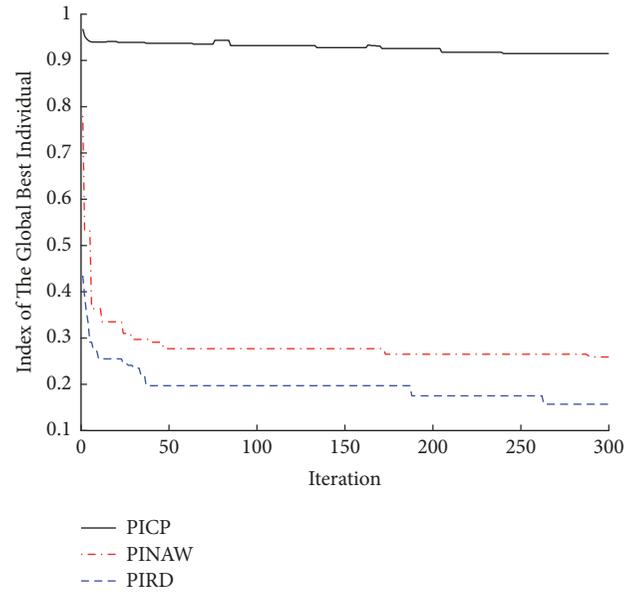


FIGURE 6: Convergence curve of CCSO.

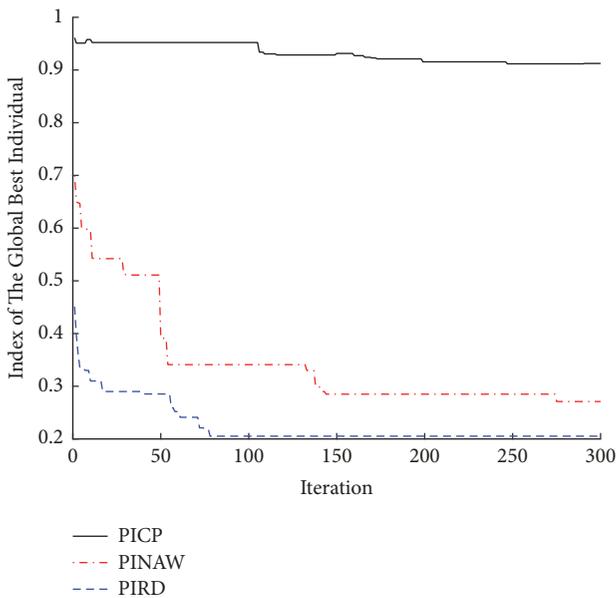


FIGURE 5: Convergence curve of CPSO.

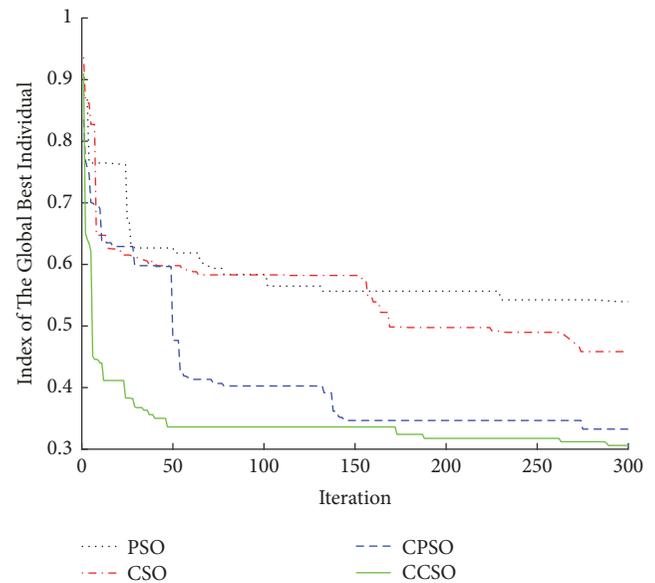


FIGURE 7: Convergence curve of fitness function.

optimization based on chaos theory is proposed, and an interval prediction model of wind power based on CCSO optimized ELM is constructed. To address the problem of large deviation in the traditional objective function CWC, a new objective function considering deviation PIRD is proposed. The benchmark function tests and case studies show the following.

- (1) CCSO proposed in this paper outperforms the PSO, CSO, and CPSO in terms of both optimization accuracy and robustness; the reason is that CCSO inherits the population diversity of CSO and the local search ability of chaotic system.

- (2) PIRD proposed in this paper takes into account the influence of the points outside the interval of the prediction results, effectively decreases the relative deviation, and improves the overall quality of the interval prediction.

- (3) Compared to other prediction models, CCSO-ELM has faster convergence speed, narrower PINAW, and lower PIRD; in a word, the proposed CCSO-ELM model constructs higher quality prediction intervals.

The high quality of training samples can effectively improve the prediction accuracy, and advanced feature selection methods can construct an optimal set of inputs for the CCSO-ELM models. Therefore, the future research will aim at advanced feature selection methods.

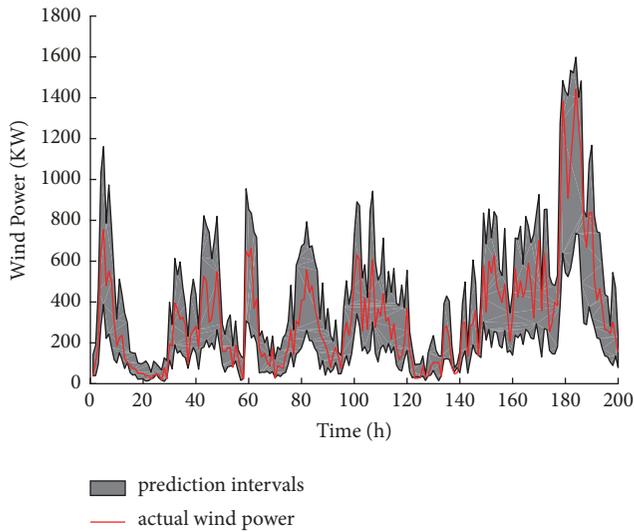


FIGURE 8: Prediction interval of PSO.

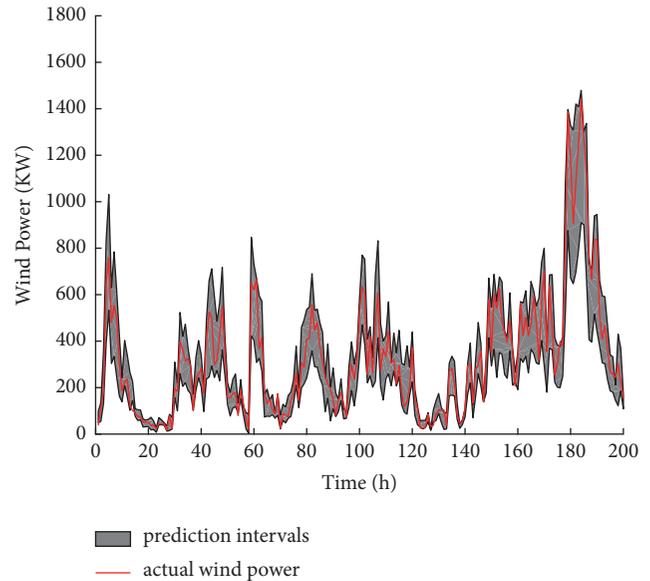


FIGURE 11: Prediction interval of CCSO.

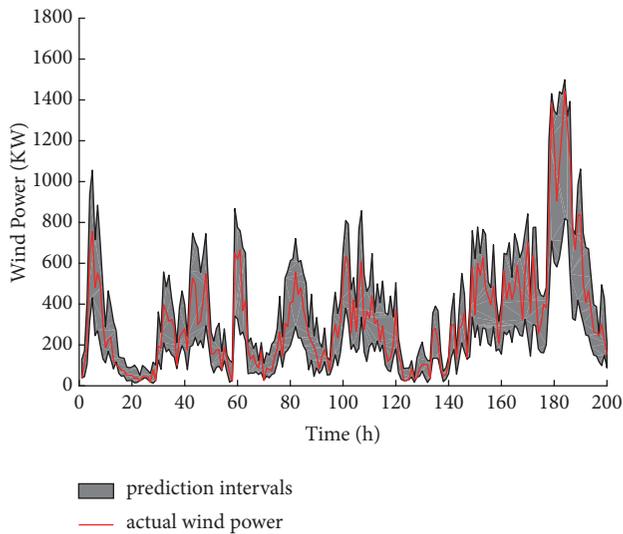


FIGURE 9: Prediction interval of CSO.

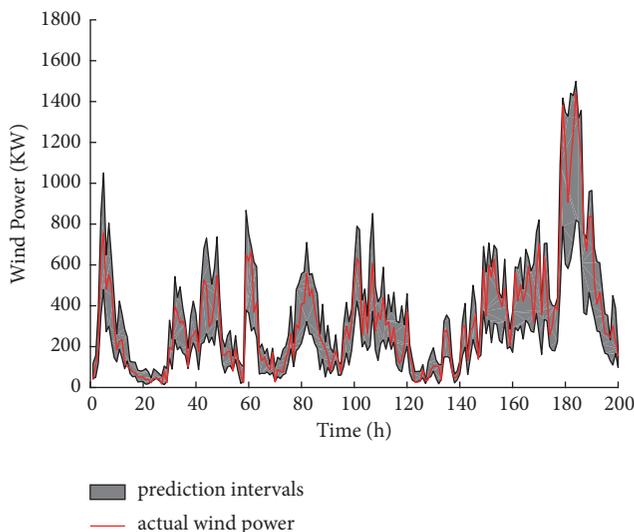


FIGURE 10: Prediction interval of CPSO.

Data Availability

The wind power data used to support the findings of this study have not been made available because of secrecy agreement.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project is supported by National Natural Science Foundation of China (51777058).

References

- [1] M. J. Ghadi, S. H. Gilani, H. Afrakhte, and A. Baghratian, "A novel heuristic method for wind farm power prediction: A case study," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 962–970, 2014.
- [2] J. Wang, Q. Zhou, H. Jiang, and R. Hou, "Short-term wind speed forecasting using support vector regression optimized by cuckoo optimization algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 619178, 13 pages, 2015.
- [3] P. Jiang, S. Qin, J. Wu, and B. Sun, "Time series analysis and forecasting for wind speeds using support vector regression coupled with artificial intelligent algorithms," *Mathematical Problems in Engineering*, vol. 2015, Article ID 939305, 14 pages, 2015.
- [4] T. Sousa, H. Morais, Z. Vale, P. Faria, and J. Soares, "Intelligent energy resource management considering vehicle-to-grid: a simulated annealing approach," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 535–542, 2012.
- [5] E. E. Elatter, "Prediction of wind power based on evolutionary optimised local general regression neural network," *Generation Transmission & Distribution Iet*, vol. 8, no. 5, pp. 916–923, 2014.

- [6] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Transactions on Industry Applications*, vol. 48, no. 3, pp. 1064–1069, 2015.
- [7] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, "ARIMA-based time series model of stochastic wind power generation," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 667–676, 2010.
- [8] M. Lange, "On the uncertainty of wind power predictions - Analysis of the forecast accuracy and statistical distribution of errors," *Journal of Solar Energy Engineering*, vol. 127, no. 2, pp. 177–184, 2005.
- [9] C. W. Potter and M. Negnevitsky, "Very short-term wind forecasting for Tasmanian power generation," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 965–972, 2006.
- [10] Y. Zhang, J. X. Wang, and X. F. Wang, "Review on probabilistic forecasting of wind power generation," *Renewable & Sustainable Energy Reviews*, vol. 32, pp. 255–270, 2014.
- [11] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1845–1856, 2010.
- [12] D. Sáez, F. Ávila, D. Olivares, C. Cañizares, and L. Marín, "Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 548–556, 2015.
- [13] J. B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy*, vol. 7, no. 1, pp. 47–54, 2004.
- [14] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.
- [15] A. Khosravi and S. Nahavandi, "An optimized mean variance estimation method for uncertainty quantification of wind power forecasts," *International Journal of Electrical Power & Energy Systems*, vol. 61, pp. 446–454, 2014.
- [16] W. Yao, Z. Zeng, and C. Lian, "Generating probabilistic predictions using mean-variance estimation and echo state network," *Neurocomputing*, vol. 219, pp. 536–547, 2017.
- [17] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval construction and optimization for adaptive neurofuzzy inference systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 983–988, 2011.
- [18] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. Van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537–547, 2011.
- [19] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 3, pp. 337–346, 2011.
- [20] G. Zhang, Y. Wu, K. P. Wong, Z. Xu, Z. Y. Dong, and H. H.-C. Iu, "An advanced approach for construction of optimal wind power prediction intervals," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2706–2715, 2015.
- [21] A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "A new fuzzy-based combined prediction interval for wind power forecasting," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 18–26, 2016.
- [22] Z. Shi, H. Liang, and V. Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 3, pp. 1177–1187, 2018.
- [23] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [24] X. B. Meng, Y. Liu, X. Z. Gao et al., "A new bio-inspired algorithm: chicken swarm optimization," in *Proceedings of the 5th International Conference on Swarm Intelligence*, pp. 74–85, Hefei, China, 2014.
- [25] D. Wu, S. Xu, and F. Kong, "Convergence analysis and improvement of the chicken swarm optimization algorithm," *IEEE Access*, vol. 4, pp. 9400–9412, 2016.
- [26] J. A. Koupaei, S. Hosseini, and F. M. Ghaini, "A new optimization algorithm based on chaotic maps and golden section search method," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 201–214, 2016.
- [27] P. Chauhan, M. Pant, and K. Deep, "Parameter optimization of multi-pass turning using chaotic PSO," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 2, pp. 319–337, 2015.
- [28] C. Segura, A. Hernandez-Aguirre, F. Luna, and E. Alba, "Improving diversity in evolutionary algorithms: new best solutions for frequency assignment," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 539–553, 2017.
- [29] N. Dong, X. Fang, and A.-g. Wu, "A novel chaotic particle swarm optimization algorithm for parking space guidance," *Mathematical Problems in Engineering*, vol. 2016, Article ID 5126808, 14 pages, 2016.
- [30] G. B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [31] X. S. Yang, *Nature-Inspired Optimization Algorithm*, Elsevier, 2014.
- [32] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 9, pp. 1341–1356, 2011.
- [33] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Optimal prediction intervals of wind power generation," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1166–1174, 2014.
- [34] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303–315, 2014.



Hindawi

Submit your manuscripts at
www.hindawi.com

