

# ***Short-term Wind Power Forecasting Model Based on BP Neural Network Algorithm Optimized by Particle Swarm Optimization***

**Bowen Zhang**

*Northeast Electric Power University, Jilin, China  
2022301011035@neepu.edu.cn*

**Abstract.** Under the impetus of the "double - carbon" goal, wind power, as one of the main forms of new - energy power generation, has been growing in significance. However, the unpredictability of wind power output has presented difficulties for the secure and stable operation of the power system as well as real - time scheduling plans. Regarding the issues that the prediction of wind power output based on the traditional BP neural network has a slow convergence rate and is prone to getting trapped in local optima, this paper puts forward a hybrid wind - power prediction model (PSO - BP), where the BP neural network is enhanced by the particle - swarm optimization (PSO) algorithm. This approach optimizes the initial weights and thresholds of the BP neural network via the global search of the PSO algorithm. As a result, it enhances the model's convergence capacity and, to some degree, circumvents the problem of local optimality. Public wind - power datasets are utilized in the experiments. The PSO - BP and traditional BP models are trained and tested multiple times under the same circumstances. The outcomes indicate that, in comparison with the traditional BP model, the PSO - BP model, while ensuring the convergence speed, mitigates the local optimization issue of the neural network. It significantly improves the reliability and precision of the model's prediction results. Moreover, it offers robust technical backing for the short - term prediction of wind power, which is conducive to improving the power - system consumption plan and ensuring its safe operation.

**Keywords:** Wind power forecasting, BP neural network, Particle Swarm Optimization

## **1. Introduction**

Guided by the "dual - carbon" objective, China is actively advancing the building of a new power system. In recent years, China's rich wind resources, predominantly in the northwest and southeast coastal regions[1], have been vigorously exploited. By the end of 2024, China's cumulative national power - generation installed capacity is approximately 3.35 billion kilowatts, with the installed capacity of wind power being around 520 million kilowatts, accounting for roughly 15.52% and gradually rising to the third position nationwide[2]. Simultaneously, the IEA anticipates that by 2030, the combined share of global wind power and photovoltaic power generation in global power generation will reach 30%[3]. Due to the uncertainty of wind power and the challenges of large -

scale storage, precise and effective wind - power forecasting holds great importance for power - system consumption planning and the safe and stable operation of the power system[4].

Based on the prediction outcomes, current deterministic prediction methods for wind power[5] can be classified into physical prediction and statistical prediction. Among these, statistical prediction is a more prevalent approach in the current research domain, mainly encompassing differential autoregressive moving average (ARIMA)[6], Bayesian optimization[7], and neural network[8]. Specifically, ARIMA is frequently employed in real - time prediction. However, a low - order model has poor accuracy, while a high - order one is complex and computationally costly. Bayesian methods are adept at handling data uncertainty but are highly reliant on prior information, making their application challenging. Neural networks rely on the capacity for nonlinear mapping to enhance prediction accuracy but place high demands on hyper - parameter selection, training resources, and methods.

The PSO - BP (Particle Swarm Optimization - Back Propagation) algorithm employed in this paper is based on particle - swarm optimization and an error - back - propagation multi - layer feed - forward neural network, and it is a deterministic prediction model. At first, it makes use of the particle - swarm optimization algorithm's capacity to quickly find the optimal solution in a high - dimensional space. This helps in determining good iterative initial values for the inertia weights and thresholds of the neural network.

Afterward, the BP neural network is trained to boost its wind - power prediction capabilities. To a certain degree, this speeds up the model's training process and overcomes the problem that the BP neural network is likely to fall into a local optimum. Experiments show that the model achieves a higher wind - power prediction accuracy with a smaller number of iterations. As a result, it provides a dependable foundation for wind - power output, ensuring the stable operation of the power system.

## 2. Methodology

Artificial intelligence techniques possess significant benefits when it comes to wind power prediction. Conventional prediction approaches usually struggle to handle the intricate non - linear traits of wind power. In contrast, artificial intelligence algorithms like the BP neural network are capable of autonomously discerning the complex mapping connection between input data and output power. By being trained using a substantial quantity of historical data, the BP neural network has the ability to construct an accurate prediction model.

### 2.1. BP

The BP neural network is a type of multi - layer feed - forward neural network, composed of an input layer, a hidden layer (also known as the implicit layer), and an output layer. These layers are interconnected via weights. Its learning procedure involves two main processes: the forward propagation of the signal and the backpropagation of the error.

During the forward propagation, the input signal is processed successively, starting from the input layer, passing through the hidden layer, and finally reaching the output layer. Let's assume that the input layer has  $n$  nodes, the hidden layer has  $m$  nodes, and the output layer has  $l$  nodes. The weight connecting node  $i$  in the input layer to node  $j$  in the hidden layer is denoted as  $w_{ij}$ , while the weight from node  $j$  in the hidden layer to node  $k$  in the output layer is represented by  $v_{jk}$ .

The input to the implicit layer node  $j$  is:

$$net_j = \sum_{i=1}^n w_{ij}x_i$$

After processing by the activation function  $f(\cdot)$ , the output of the hidden layer node  $j$  is:

$$y_j = f(net_j)$$

Commonly used activation functions such as the Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}}$$

The input to the output layer node  $k$  is:

$$net_k = \sum_{j=1}^m v_{jk} y_j$$

The output to the output layer node  $k$  is:

$$\widehat{y}_k = f(net_k)$$

When the actual output of the output layer,  $\widehat{y}_k$ , does not match the desired output,  $y_k^*$ , the error back propagation stage is entered. Define the error function as:

$$E = \frac{1}{2} \sum_{k=1}^l (\widehat{y}_k - y_k^*)^2$$

The error signal is propagated backward along the original connection path through the back-propagation algorithm, and the weights of each layer are adjusted according to the error gradient descent method to minimize the error function  $E$ . The weight adjustment formula is:

$$\Delta w_{ij} = \eta \frac{\partial E}{\partial w_{ij}}$$

$$\Delta v_{jk} = \eta \frac{\partial E}{\partial v_{jk}}$$

Where  $\eta$  is the learning rate, which controls the step size of the weight adjustment. If the learning rate is excessively high, it might lead to the algorithm oscillating throughout the training procedure, thereby making convergence arduous. On the other hand, an overly low learning rate will cause the training speed to be extremely sluggish.

In actual applications, the adaptive learning rate tactic is frequently adopted. This enables the learning rate to be adjusted dynamically in accordance with the variations in error that occur during the training process.

## 2.2. PSO

The particle swarm optimization algorithm is an evolutionary computation method. As a stochastic search algorithm grounded in swarm intelligence, it aims to find the optimal solution of a problem using a mathematical optimization algorithm model that mimics the foraging behavior of a flock of birds or a school of fish. The PSO commences from a random solution, and through iterative processes, it hunts for the optimal solution, assessing the solution's quality via the fitness degree. During the iterative procedure, the algorithm looks for the global optimum by trailing the currently discovered optimum. It is a parallel algorithm that is straightforward to implement, boasts high accuracy, and has a rapid convergence rate.

In the context of the Particle Swarm Optimization (PSO) algorithm, each particle stands for a prospective solution to the given problem. These particles traverse the search space at a particular velocity. The adjustment of the particles' velocity and position occurs in a dynamic manner, relying on the individual extreme value (termed pBest) and the global extreme value (referred to as gBest). If we consider that the search space has  $D$  dimensions, then the position of the  $i^{th}$  particle can be depicted as follows:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

Velocity is expressed as:

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$$

The optimal positions (individual poles) experienced by particle  $i$  during the flight are:

$$P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$$

The optimal positions (global extrema) experienced by the entire particle population are:

$$P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$$

In every iteration, the particle modifies its velocity and position within the solution space (also known as the search space) based on the subsequent equations:

Speed Update Formula:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1j}(t)(p_{ij} - x_{ij}(t)) + c_2 r_{2j}(t)(p_{gj} - x_{ij}(t))$$

Position Update Formula:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

Here,  $t$  indicates the present iteration count. The inertia weight  $\omega$  plays a crucial role in determining how much of the particle's previous velocity it inherits. Meanwhile,  $c_1$  and  $c_2$  are the acceleration constants, commonly known as learning factors.  $c_1$  governs the step - size by which the particle moves towards its own historical best position, and  $c_2$  does the same for the global best position.  $r_{1j}(t)$  and  $r_{2j}(t)$  are random values that are evenly distributed within the range  $[0,1]$ . These random numbers introduce a stochastic element into the particle's movement, helping the algorithm explore different regions of the search space.

### 2.3. Pso-Bp

In the PSO - BP wind power forecasting model, at the beginning, the global search capability of the PSO algorithm is employed to optimize the initial weights and thresholds of the BP neural network. The weights and thresholds of the BP neural network are encoded as the position vectors of particles. The PSO algorithm then iteratively adjusts the positions of these particles, with the aim of minimizing the prediction error as the objective function. Through this procedure, a more advantageous combination of weights and thresholds is discovered.

Subsequently, these optimized weights and thresholds are fed into the BP neural network. Following this, training and prediction are conducted on the pre - treated wind power data. By taking these steps, both the global optimization - finding ability of the PSO algorithm and the nonlinear mapping ability of the BP neural network are fully utilized, thus improving the accuracy and reliability of wind power prediction.

### 3. Case analysis

#### 3.1. Data description

To validate the efficacy of the wind power prediction approach founded on the PSO - BP algorithm, this paper utilizes the public dataset for wind power prediction as the source dataset for the algorithmic analysis.

#### 3.2. Data preprocessing

Because the adopted dataset contains actual system operation errors, issues such as missing or abnormal data exist. To guarantee the training accuracy of the PSO - BP model and prevent abnormal values from negatively affecting the generalization ability of the neural network, the dataset undergoes pre - processing prior to being used for training. First, the normalization method is employed to map all data (both inputs and outputs) to the interval of  $[-1, 1]$ . This serves to eliminate the influence of magnitude and modal values, thus accelerating the model's convergence speed. Second, the processed dataset is split into a training set (the initial 360 sets of time - step data) and a test set (the final 24 sets of time - step data). These are respectively utilized for model training and test comparison.

#### 3.3. Evaluation indicators

For a thorough and unbiased evaluation of the PSO - BP model's performance in wind power prediction, this paper selects the root mean square error (RMSE) and mean absolute error (MAE) as the metrics for model assessment.

In particular, the root - mean - square error is adept at reflecting the degree of deviation between the predicted value and the actual value. It is highly sensitive to outliers and can measure the degree of disparity between them. The smaller the value of the root - mean - square error, the closer the prediction is to the actual value.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

In contrast, the mean absolute error represents the average of the absolute values of the differences between the predicted values and the true values. It is effective in quantifying the absolute magnitude of the prediction error. To some degree, it can mitigate the impact of outliers in individual data points on the overall data scenario. This makes it a robust metric, well - suited for depicting the overall state of the prediction outcomes.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Then, by the complementary characteristics of the two types of model evaluation indexes, RMSE and MAE, the prediction accuracy of the wind power prediction model can be evaluated in a more

comprehensive way.

### 3.4. Parameter selection

Regarding the BP neural network, the hyperbolic tangent function (tansig) is selected as the activation function for the hidden layer, and the linear function (purelin) is used as the activation function for the output layer. To determine the initial values of the remaining hyperparameters, such as the number of nodes, inertia weights, and thresholds, the PSO algorithm conducts a global search for optimization.

As for the PSO optimization algorithm, its parameters are set to the default values of the PSO algorithm function within the Matlab simulation environment. It is important to note that the particle dimension D is set to 31 in accordance with the formula mentioned earlier.

### 3.5. Algorithm comparison

The PSO-BP and traditional BP neural network prediction models are now compared with other related prediction patterns using the same dataset training and controlling the two-dimensional variables of iteration accuracy and the number of iterations with RMSE and MAE assessment metrics.

Considering that PSO is not able to completely solve the problem of BP network which is easy to fall into the local optimum, and at the same time, the test and training results of the algorithm model are not general. Therefore, in this paper, the PSO-BP and BP algorithms are trained and tested in Matlab simulation environment using the same dataset for 20 repetitions, and the test results are recorded and compared with the evaluation indexes.

Table 1: Data comparison results

	MAE average	RMSE average
<b>PSO-BP</b>	0.1686	0.2414
<b>BP</b>	1.9082	3.7266

It can be seen that, compared with the traditional BP neural network, the introduction of PSO algorithm to some extent avoids the problem that BP neural network is easy to fall into the local optimum, so that the model can converge to the global optimal solution faster, thus significantly reducing the value of the RMSE, MAE, and improving the prediction accuracy.

In addition, the number of iterations for PSO optimization and the chance of outliers in the MAE/RMSE, without going into the range of the number of overfitting iterations, predicts a decrease in the probability of outliers as the number of iterations increases.

## 4. Conclusion

This paper applies a PSO - BP optimized neural network algorithm for real - time short - term wind power prediction. The PSO algorithm, characterized by its robust global search ability, rapid iterative convergence speed, and efficient use of computational resources, is harnessed to supply dependable initial hyperparameter values for the BP neural network. This significantly enhances the BP neural network's convergence rate and, to some degree, overcomes its tendency to get trapped in local optima.

Initially, the wind power dataset in this paper undergoes anomaly correction and normalization, followed by the completion of the division into a neural network training set and a test set. Subsequently, the PSO - BP model is trained and tested using this dataset to achieve short - term prediction of wind power output.

Finally, by contrasting the prediction indices of the PSO - BP model and the traditional BP model with the same example, it is demonstrated that the PSO - BP model can substantially reduce iteration time and arithmetic resources while ensuring prediction accuracy. This offers strong technical support for the safe and reliable operation of the power system when wind power is integrated.

## References

- [1] SONG Jing. Research on wind resource distribution and wind power planning in China[D]. North China Electric Power University, 2013.
- [2] The National Energy Administration (NEA) releases the 2024 national power industry statistics
- [3] Renewables 2024 Analysis and forecast to 2030
- [4] Duan Xuewei, Wang Ruiqi, Wang Zhaoxin, et al. Shandong Electric Power Technology, 2015, 42(07):26-32. )
- [5] WU Sushuang, CAI Xiaolu, LI Junxian. Hydropower & New Energy, 2024, 38(03):38-41+45. DOI:10. 13622/j. cnki. cn42-1800/tv. 1671-3354. 2024. 03. 010.
- [6] F. A. Eldali, T. M. Hansen, S. Suryanarayanan and E. K. P. Chong, "Employing ARIMA models to improve wind power forecasts: A case study in ERCOT, " 2016 North American Power Symposium (NAPS), Denver, CO, USA, 2016, pp. 1-6, doi: 10. 1109/NAPS. 2016. 7747861
- [7] YANG Manrou, TIAN Hai. Short-term wind power prediction based on Bayesian optimization XGBoost[J]. Electronic Devices, 2024, 47(05):1389-1395. )
- [8] XIAO Liexi, ZHANG Yu, ZHOU Hui, et al. Ultra-short-term wind power prediction based on IAQA-VMD-LSTM[J]. Journal of Solar Energy, 2023, 44(11):239-246. DOI:10. 19912/j. 0254-0096. tynxb. 2022-1054.