Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 4, pp. 3114–3125, DOI 10.12694/scpe.v25i4.2974

A NOVEL WIND POWER PREDICTION SCHEME BY COUPLING THE BP NEURAL NETWORK MODEL WITH THE FIREWORKS ALGORITHM

YONGGANG LI[*], YAOTONG SU[†], LEI XIA[‡], YONGFU LI[§], HONG XIANG[¶], AND QINGLONG LIAO[‖]

Abstract. Wind power has unpredictable, intermittent traits due to meteorological conditions and environmental factors. Large-scale grid integration of wind energy will undoubtedly challenge system stability. This study developed a fireworks algorithm-backpropagation (FWA-BP) neural network model to forecast wind power using wind speed, direction, and power as model inputs. Optimization of the BP network weights and thresholds occurred through the fireworks algorithm. Compared to a standard BP network, the FWA-BP model yielded improved prediction accuracy seen through a lower mean squared error. This implies that the approach introduced in this paper significantly enhances global search capabilities, prediction accuracy, and speed. It contributes to enhancing the reliability of the power system, optimizing resource allocation, and improving wind power scheduling, with substantial potential and economic significance.

Key words: Backpropagation neural network; Power prediction; Wind speed; Wind power; Wind direction; Fireworks algorithm

1. Introduction. In recent years, more research has started to focus on wind energy due to its renewability, cost-effectiveness and environmental friendliness.This renewable energy option stands out for its ease of development and lack of pollution [1, 2, 3, 4]. However, the irregular and fluctuating output of wind power that is correlated with changing weather patterns and environmental variables poses integration challenges when connecting substantial wind energy capacities to the electric grid system. This integration presents difficulties in balancing electricity supply and demand, ensuring system stability, and effectively managing grid operations [5, 6, 7, 8]. Accurate forecasting of wind power output becomes essential for efficient resource distribution and wind power dispatch. For instance, some studies propose new frameworks and utilize optimization algorithms to enhance the short-term wind power generation forecasting accuracy. There are also those employing hybrid optimization algorithms to optimize neural networks for improved prediction accuracy. Furthermore, research in the field of deep learning introduces novel transfer models. Enhanced forecasting capabilities serve to strengthen the power system's resilience while also raising the ability to integrate more renewable wind energy production. Additionally, it assists in optimizing wind farm maintenance schedules and alleviates the need for extensive energy storage, thereby resulting in significant economic advantages [9, 10, 11, 12, 13, 14].

Ongoing research in the field seeks to enhance our understanding of wind power dynamics and improve forecasting methodologies. Advanced forecasting techniques, such as machine learning algorithms and predictive modelling, are being explored to address the challenges associated with the variability of wind power. These endeavors aim to further unlock the potential of wind energy, making it an even more reliable and integral component of the global energy landscape. In recent decades, scholars researching wind power generation have proposed a plethora of methods for forecasting wind power output in wind farms. Prior research has proposed using Adaboost-PSO-ELM model to address uncertainties and fluctuations in wind power prediction.

[*]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (lyg@cqupt.edu.cn, Corresponding Author).
[†]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.
[‡]State Grid Chongqing Electric Power Company, Chongqing 401121, China.
[§]State Grid Chongqing Electric Power Company, Chongqing 401121, China.
[¶]State Grid Chongqing Electric Power Company, Chongqing 401121, China.
[‖]State Grid Chongqing Electric Power Company, Chongqing 401121, China.

This integrative approach aims to improve generalization capabilities beyond existing models. Validation on turbine data from Turkey demonstrated the Adaboost-PSO-ELM model provides more accurate and robust wind power forecasting [15]. In related work examining the inherent variability of wind energy, the parametric sine function superposition recurrent neural network is introduced in the prediction process for iterative tuning. This algorithm can effectively extract multiple features from intermittent wind power data, such as time series data of wind speed, wind direction, atmospheric pressure, and more. This capability has been confirmed through its robust predictive performance [16] .A recent study proposed a Differential Evolution-Backpropagation (DE-BP) algorithm for wind forecasting to address backpropagation limitations including local optima and slow speeds. Validation results show the DE-BP approach boosts accuracy around 5% over traditional BP models for power prediction. Additionally, prediction time decreased 23.1% compared to genetic algorithm-BP alternatives [17].There is also research proposing an improved BP neural network prediction method, namely an iterative genetic optimization BP neural network power prediction model, with the input being the power from one hour ahead and other influencing factors. Tests have proven that this model can effectively meet the power system's relevant short-term power prediction requirements for wind farms [18].

This study employs a Backpropagation (BP) neural network by integrating historical data on real power production, wind speed, and direction. While BP networks have shown promise in wind power forecasting, their performance relies heavily on model parameter initialization and optimization. Additionally, few studies explore the integration of advanced optimization algorithms like fireworks algorithms to improve model training. To address these gaps, this study puts forward a fireworks-enhanced backpropagation (FWA-BP) approach. Following simulations with actual data from a specific wind farm using Matlab, the results show that the Fireworks Algorithm optimization led to a notable decrease in errors when applying the backpropagation neural network model for wind power prediction. This enhancement leads to increased accuracy and speed in forecasting, addressing the limitations of a standalone BP neural network and ultimately achieving superior performance.

2. Wind Power Prediction Model. Wind power generation has numerous advantages that other renewable energy sources lack, but its drawbacks are also quite evident. The ongoing rise in installed wind power capacity exacerbates the technical constraints associated with wind power. Wind power production varies significantly between high and low wind speeds. These sudden changes can exert a notable influence on the primary power grid, and if they surpass the grid's capacity thresholds, they may result in a grid failure. Therefore, accurate prediction of wind power output is of utmost importance.

2.1. Analysis and Management of Wind Power Influencing Factors. We will analyze this from five aspects.

1. The correlation between wind energy output and wind velocity. The equation for the wind turbine's harnessed wind power is:

$$W = \frac{1}{2}\rho V^3 \pi r^2 C_p \tag{2.1}$$

In this equation, ρ represents air density, V is the wind speed, r is the rotor's swept area radius, and C_p is the wind turbine power coefficient [19].

Wind power generation varies markedly from traditional sources, mainly owing to its dependence on wind speed as a vital element. In Figure 2.1, the graphic depicts the non-linear relationship between wind energy generation and wind velocity.

In real-world wind farm power production, the link between wind velocity and turbine output is nonlinear. This can be characterized by the following segmented function:

$$W(v_w) = \begin{cases} 0, & v_w \le v_i \\ (X + Yv_w + Zv_w^2)p_u, & v_i < v_w \le v_u \\ W_u, & v_u < v_w \le v_o \\ 0, & v_o < v_w \end{cases} \tag{2.2}$$

In the equation, v_i, v_o, and v_u represent the starting, shutdown, and designated wind velocities, in that

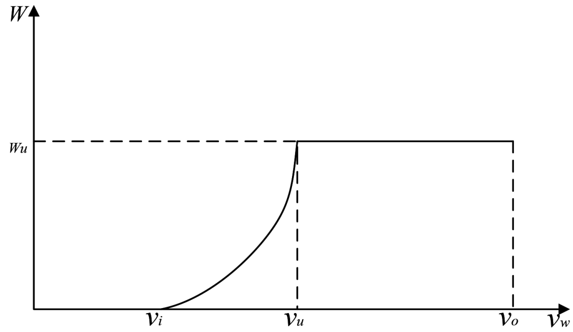

Fig. 2.1: Wind Power Output Characteristics Curve.

order. Meanwhile, p_u signifies the rated power output, X, Y,and Z are parameters outlining the wind energy curve.

The parameters of wind turbine power characteristic curve obtained from observed data are as follows:

$$X = \frac{1}{(v_i - v_u)^2}\left[v_i(v_i + v_u) - 4(v_i \times v_u)(\frac{v_i + v_u}{2v_u})^3\right] \tag{2.3}$$

$$Y = \frac{1}{(v_i - v_u)^2}\left[4(v_i + v_u)(\frac{v_i + v_u}{2v_u})^3 - (3v_i + v_u)\right] \tag{2.4}$$

$$Z = \frac{1}{(v_i - v_u)^2}\left[2 - 4(\frac{v_i + v_u}{2v_u})\right] \tag{2.5}$$

The quation 2.2 shows that in different intervals, the wind power and speed of wind farm show a variety of functional relationships.

Additionally, use V_m to represent the measured value, and the normalized wind speed V_n is::

$$V_n = \frac{V_m - V_{\min}}{V_{\max} - V_{\min}} \tag{2.6}$$

2. The correlation between wind direction and wind power. Wind turbines use a yaw mechanism, guided by wind direction and speed, to capture more wind energy. The yaw mechanism typically has a time delay, so wind turbines cannot immediately align with the wind, resulting in varying power output at the same wind speed. Indeed, wind direction has a vital role in influencing wind turbine power output.

Wind direction is normalized as:

$$\theta_n = \frac{\theta_m - \theta_{\min}}{\theta_{\max} - \theta_{\min}} \tag{2.7}$$

Where θ_n represents the normalized wind direction angle, θ_m is the measured wind direction angle, $\theta_{\min}$ and $\theta_{\max}$ correspond to the minimum and maximum wind direction angles, respectively.

3. The correlation between air pressure and wind power. Wind power output exhibits correlation with atmospheric pressure; therefore, atmospheric pressure data should be incorporated when formulating wind power output forecasting models.

Pressure is normalized as:

$$P_n = \frac{P_m - P_{\min}}{P_{\max} - P_{\min}} \tag{2.8}$$

where P_n represents the normalized atmospheric pressure, P_m is the measured atmospheric pressure, $P_{\min}$ and $P_{\max}$ correspond to the minimum and maximum atmospheric pressures, respectively.

At the same time, pressure is related to temperature and atmospheric density, and the formula for calculation is as stated below:

$$\rho = \rho_0 \frac{T_0}{T} \frac{P}{P_0} \tag{2.9}$$

Where ρ_0, T_0, P_0 and represent standard air density, temperature, and atmospheric pressure, and represent local temperature and pressure.

4. The relationship between temperature and wind power. When the temperature is too low, the turbine blades may freeze and not function properly, leading to a decrease in wind power output. Conversely, when the temperature is too high, certain components of the wind turbines may not withstand the high temperatures, which can also affect wind power output. Additionally, temperature is related to pressure and air density.

Temperature is normalized as:

$$T_n = \frac{T_m - T_{\min}}{T_{\max} - T_{\min}} \tag{2.10}$$

5. The correlation between humidity and wind power. The relative humidity of the air and air density are inversely related. Relative humidity affects density, as indicated by equation 2.9, which shows that density affects temperature and atmospheric pressure. Therefore, local air humidity also influences wind power output.

Humidity is normalized as:

$$S_n = \frac{S_m - S_{\min}}{S_{\max} - S_{\min}} \tag{2.11}$$

Where S_n represents the normalized humidity, S_m is the measured humidity, $S_{\min}$ and $S_{\max}$ correspond to the mini-mum and maximum humidity, respectively.

2.2. BP Neural Network Predicting Wind Power. The three-layer structure of BP neural network is shown in Figure 2.2, n, m, and l represent the number of nodes in each layer [20]. The input variables $x_i(i = 1, 2, ..., n)$ enter the input layer. They are organized into a matrix, where rows represent the number of samples, and columns represent the feature dimensions of each sample. Each row corresponds to a sample, containing three types of features. The hidden layer outputs are denoted $h_j(j = 1, 2, ..., l)$, and the final outputs from the network are $y_k(k = 1, 2, ..., m)$. The transfer function mapping from input to hidden layer is $\varphi(x)$, the input-hidden weight matrix is ω_{ij}, and the hidden-output matrix is ω_{ik}. Matrices a_j and b_k provide thresholds for the hidden and output layers.

The input to the j-th hidden node is calculated as:

$$n_{ij} = \sum_{i=1}^{n} \omega_{ij} x_i - a_j \tag{2.12}$$

The hidden node j takes input:

$$hj = \phi(n_{ij}) = \phi\left(\sum_{i=1}^{n} \omega_{ij} x_i - a_j\right) \tag{2.13}$$

For output node k, the input is:

$$n_{jk} = \sum_{j=1}^{l} \omega_{jk} h_j - b_k \tag{2.14}$$

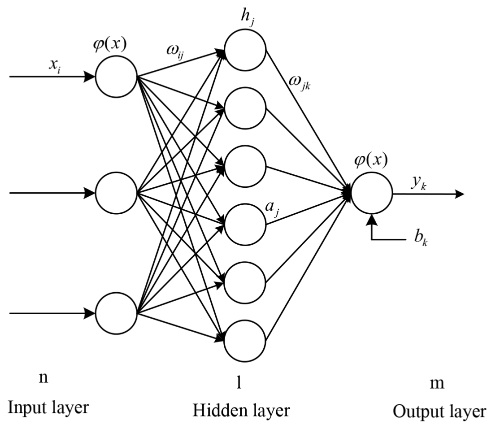

Fig. 2.2: Structure diagram of the BP neural network model.

Node k in the output layer produces output:

$$e_k = \phi\left(n_{jk}\right) = \phi\left(\sum_{j=1}^{l} \omega_{jk} h_j - b_k\right) \tag{2.15}$$

The parameters of influencing factors such as wind speed and wind direction serve as input features in the network model, acting as neurons in the input layer. Specifically, each neuron in the input layer corresponds to an input feature parameter, as follows:

1. Wind speed v serves as an input feature, corresponding to a neuron in the input layer. This neuron receives the raw wind speed data v as input.

2. Wind direction θ serves as another input feature, also corresponding to a neuron in the input layer. This neuron receives the raw wind direction data θ as input.

The values of these two input nodes are multiplied by weights associated with the respective connections and then passed to the hidden layer. The hidden layer and subsequent layers furtherand subsequent layers process this information, ultimately outputting the predicted wind power. Through multiple iterations of training, the model automatically adjusts the weight coefficients The model automatically adjusts the weight coefficients through multiple iterations of training to achieve the optimal fitting effect between input feature parameters and output values.

The BP neural network demonstrates a robust capability for non-linear mapping and can approximate non-linear functions through the utilization of two iterative processes: signal forward propagation and error backpropagation. This allows the network to satisfy specific conditions for error reduction in its output [21, 22].

2.3. The BP Neural Network Model's Drawbacks. This model also has some limitations, mainly in the following aspects:

1. The convergence rate during network can be relatively slow, primarily due to the influence of two key factors: the learning rate and the activation function's derivative. The choice of the learning rate directly impacts the magnitude of network parameter updates, while the activation function's derivative adjusts weights during the backpropagation process, further affecting convergence performance. As the network aims to address complex nonlinear problems, the "zigzag phenomenon" often occurs in the objective function, impeding the convergence speed [23].

2. Entering local optimal solutions is a common susceptibility in certain scenarios. In a BP neural network, when the error function equals 0, indicating no gradient, the weights and thresholds cannot be updated. If we assume that the nonlinear problem being solved is a standard convex function, there would be a unique extremum, and its local optimum would be the global optimum. Through the network, optimal solutions can

(a) High-quality fireworks exploding.

(b) Low-quality fireworks exploding.

Fig. 3.1: The process of fireworks exploding with varying quality.

be identified, and the most effective converging weights and thresholds can be determined. However, in most cases, nonlinear problems have multiple extremum points within different intervals of their functions. In such cases, once the BP neural network reaches a local optimum, it stops updating, making it unable to continue towards finding the global optimum [24].

3. Selecting the optimal network architecture can be difficult, as the needs vary depending on the problem. While BP neural networks have proven effective in diverse applications, carefully configuring the structure based on the unique characteristics of each challenge is important. Overall architectural choices must be informed by both the general best practices of the method and the distinctive qualities of the particular predictive task.

3. Wind Power Prediction Model. By self-explosion to generate sparks and subsequently exploring the surrounding area, the Fireworks Algorithm possesses additional advantages due to this search mechanism. In comparison to traditional genetic algorithms and particle swarm optimization, it can better uncover hidden information in deep layers, offers higher randomness, and demonstrates greater versatility. In the fireworks algorithm, the initial fireworks and exploded fireworks positions are considered as a set of candidate solutions. The solutions are evaluated using an objective function that determines their quality. A fitness function then works to retain the higher performing solutions. This process is iteratively repeated. Eventually, the obtained firework positions converge towards the optimal solution, with the best output result achieved in the final iteration. High-quality fireworks typically generate a large number of sparks when the explosion radius is relatively small, while low-quality fireworks, due to a larger explosion radius, produce more scattered sparks. Figure 3.1 provides a simple comparison of the number of sparks and explosion range between high-quality and low-quality fireworks.

The Fireworks Algorithm demonstrates features related to the diversity of the population, notably observed in the variety of fireworks, explosion amplitudes, and the quantity of sparks. Additionally, it reflects diversity in Gaussian mutation and displacement operations. Diversity broadens the scope of global search, preventing it from getting stuck in local optima while not adversely affecting the algorithm's convergence capability. Some studies have elucidated its unique advantages, such as its search mechanism and efficiency in multi-objective optimization. Other research has focused on improvements to the Fireworks Algorithm and the selection of parameters for its applications [25, 26, 27, 28]. Considering its application in predicting wind power generation using BP neural networks, the Fireworks Algorithm can address many shortcomings of traditional BP neural network predictions.

3.1. Data Processing. The simulation analysis was conducted using data from a wind turbine in Turkey. The experimental data consists of wind speed, wind direction, and wind power output data at regular intervals. Divide the dataset into two segments, one designated for model training and the other for testing. Normalize the input vector to the range [0,1] using the provided formula [29]:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.1}$$

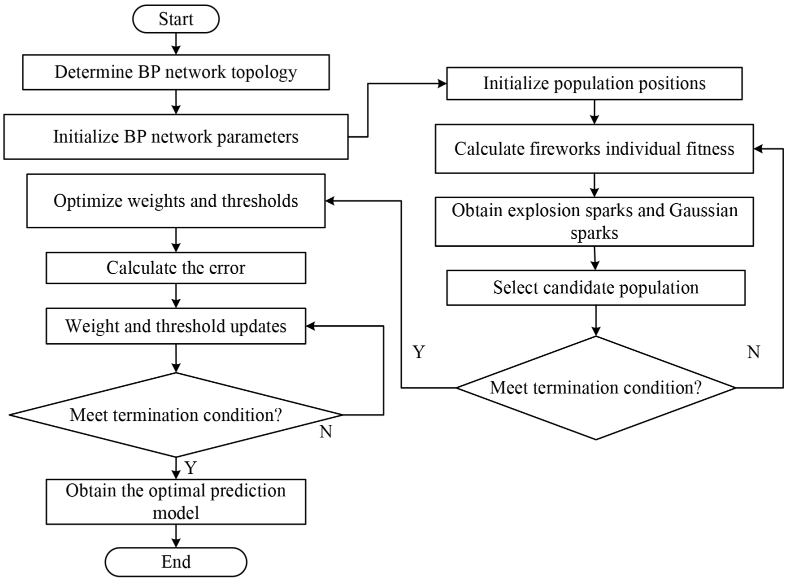

Fig. 3.2: Flow of Wind Power Prediction Using FWA-BP Neural Network.

Here, x is the actual data value, while $x_{\max}$ and $x_{\min}$ refer to the maximum and minimum values in the input vector. x^* is the normalized result.

3.2. The construction of the wind power forecasting model for FWA-BP. Use actual power, wind speed and direction as inputs to the neural network. Employ the fireworks algorithm to search optimal network weights and thresholds.

Here are the specific steps:

Step 1: Establish a BP neural network;

Step 2: Initialize firework population positions and define upper and lower limits for number of explosions;

Step 3: Calculate the fitness values for every individual in the fireworks population by utilizing the formulas to determine the count of fireworks generated in each explosion, represented as N_i and the explosion radius L_i.

$$N_i = c \times \frac{k_{\max} - f(x_i) + \beta}{\sum_{i=1}^{N}(k_{\max} - f(x_i) + \beta)} \tag{3.2}$$

$$L_i = d \times \frac{f(x_i) - k_{\min} + \beta}{\sum_{i=1}^{N} f(x_i) - k_{\min} + \beta} \tag{3.3}$$

where $k_{\max} = \max(f(x_i))$ $(i = 1, 2, \ldots, N)$ and $k_{\min} = \min(f(x_i))$ $(i = 1, 2, \ldots, N)$ represent the worst and best fitness values among all firework individuals in the current population, respectively. The function $f(x)$ represents the fitness value of an individual x. In this paper, the fitness function value corresponds to the mean squared error. c restricts total sparks, d is the explosion radius upper bound. β prevents division by zero;

Step 4: Perform the explosion difference operation to generate sparks for the fireworks individuals, as well as Gaussian-mutated sparks after Gaussian mutation. The formulas for calculating the position offsets are as follow:

$$ex_{ik} = x_{ik} + h \tag{3.4}$$

$$cx_{ik} = x_{ik} \times r \tag{3.5}$$

$$h = A_i \times rand(1, -1) \tag{3.6}$$

where A_i is the explosion radius for firework i. h gives position offset, ex_{ik} is a spark from the i-th firework's subsequent explosion. x_{ik} is firework i's k-th dimension value. cx_{ik} Gaussian-mutates x_{ik} based on distribution $r \sim N(1, 1)$;

Step 5: The subsequent generation of fireworks population comprises candidate fireworks. From the populations of fireworks, explosion sparks, and sparks produced through Gaussian mutation, select the one with the lowest fitness value. Additionally, $N - 1$ fireworks are selected through fitness-proportional selection;

Step 6: Determine if the termination condition has been reached. If so, move to the next step. Otherwise, return to the previous step and repeat the process;

Step 7: Retrieve optimal weights and thresholds, perform error checking, repeatedly validate and optimize the model, and thresholds to make wind power predictions using the model;

Step 8: Optimize the updating of network connection weights and thresholds using the Fireworks Algorithm. The following formulas provide the expressions for weight threshold optimization, based on which (2.12) - (2.15) are updated:

$$\omega_{ij} = \omega_{ij} + \eta H_j \left(1 - H_j\right) x_i \sum_{k=1}^{n} \omega_{jk} e_k \tag{3.7}$$

$$\omega_{jk} = \omega_{jk} + \eta H_j e_k \tag{3.8}$$

$$a_j = a_j + \eta H_j (1 - H_j) x_i \sum_{k=1}^{n} \omega_{jk} e_k \tag{3.9}$$

$$b_k = b_k + \eta e_k \tag{3.10}$$

where η represents the learning rate;

Step 9: Evaluate whether the termination condition has been fulfilled. If so, end the current iteration. otherwise, if the specified error performance criteria are not met, and the iteration limit has not been reached, continue with step 8.

Figure 3.2 illustrates the computational process for wind power forecasting using the optimized FWA-BP network.

4. Case Study Analysis. To assess prediction accuracy, this paper simulated wind farm power output over a period. The dataset contained wind speed, direction and power, serving as input for both BP and FWA-BP models. The first 150 data rows were utilized for training, while rows 151-185 were used for testing.

Based on an empirical formula for the number of hidden layers:

$$l = \sqrt{n + m} + \alpha \tag{4.1}$$

where n and m represent the number of nodes in the input and output layers, respectively, and α is a constant ranging between 1 and 10.

We used the MATLAB neural network toolbox to design, train and test the BP neural network. The version of MATLAB used was R2018a, and the related modules of the neural network toolbox used were version 11.1. The computer configuration we used for the simulation was an AMD Ryzen 7 5800H with Radeon Graphics, 3.20 GHz, and 16 GB of memory. The graphics card was an RTX 3060, running on the Windows 11 operating system. Based on the empirical formula in equation 4.1, the range for the number of hidden layer neurons was calculated to be 3 to 12. After approximately 10 simulation trials and adjustments, the number of neurons was finalized at 6.

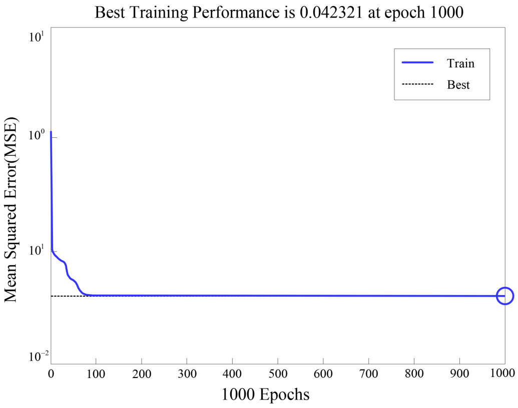

Fig. 4.1: BP neural network training error curve.

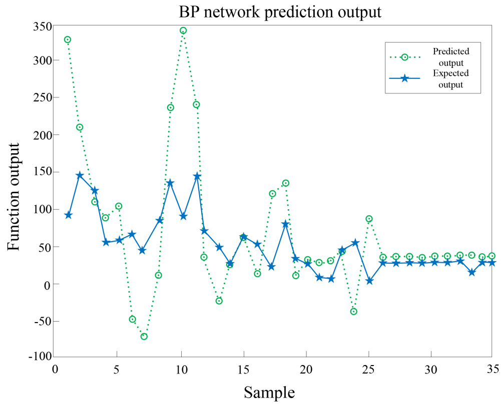

Fig. 4.2: BP neural network test data prediction results.

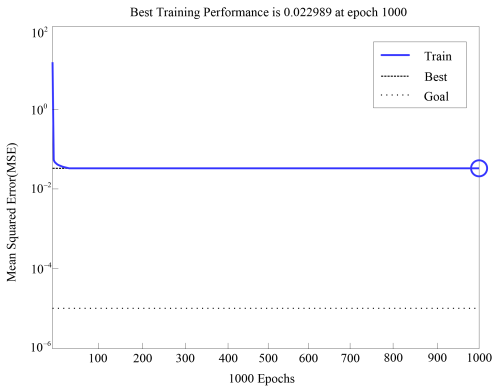

Fig. 4.3: FWA-BP neural network training error curve.

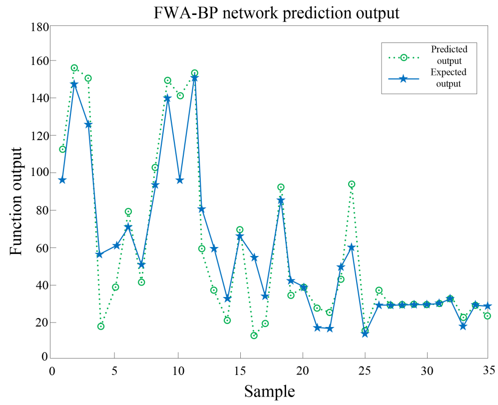

Fig. 4.4: FWA-BP algorithm test data prediction results.

Figure 4.1 and 4.2 show the training mean squared error, prediction speed, and comparison between predicted and actual outputs by using the BP neural network for prediction.

Figure 4.1 shows the BP neural network's mean squared error reached 0.042321 after 100 training iterations, with no further decrease. This is 0.042221 off the target value.Figure 4.2 shows the predicted and actual wind power results, with the dashed line representing the predicted values and the solid line depicting the observed/real values. While most of the predicted points closely match the actual values, there are slight discrepancies. Notably, about 11 prediction points show significant deviations. This is due to the inherent limitation of the BP neural network structure. As a simple three-layer feedforward network, its ability to simulate nonlinear mappings is limited, resulting in poor approximation performance for complex real-world systems. Additionally, factors such as the number of training epochs and the setting of the learning rate contribute to the inability of the BP network to achieve lower training loss and testing error. Therefore, the next step involves improving the BP network using the fireworks algorithm for prediction. For this, the following parameters were set: population size $N = 30$, explosion radius and spark number adjustment constants ($d = 5$, $c = 30$), upper and lower limits for the number of fireworks explosions ($L_m = 2$, $B_m = 0.8$), Gaussian-mutated sparks $G = 5$, and a maximum iteration count $T = 1000$.

Figure 4.3 shows the learning error curve for the FWA-BP algorithm during the training process. After

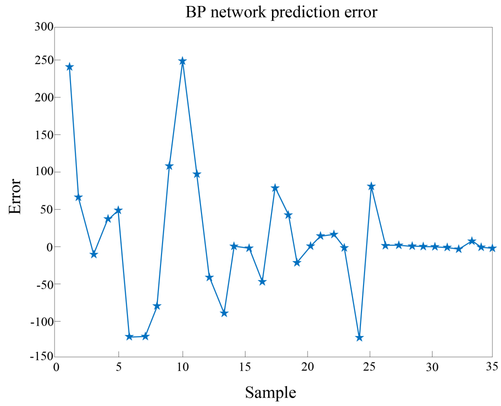

Fig. 4.5: BP neural network prediction error.

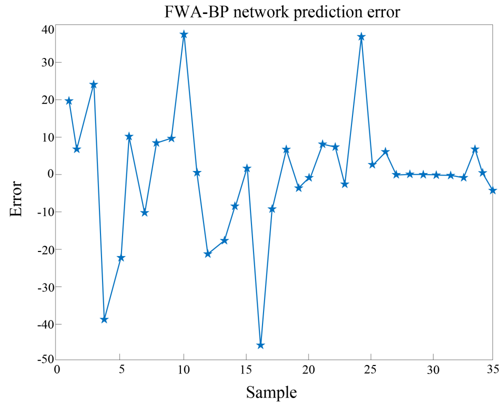

Fig. 4.6: FWA-BP algorithm prediction error.

approximately 50 training iterations, the mean squared error had already reached 0.22989, which is twice as fast as the BP algorithm and represents a reduction of 0.187569 in the mean squared error. In Figure 4.4, the dashed and solid lines closely match, indicating that the predicted and actual results are in close agreement. Only about 4 prediction points show significant errors. This is because the Fireworks Algorithm accomplishes a global randomized optimization search for the network's topology, number of nodes, and connection weights. It finds a better network structure, allowing for a more scientific and rational determination of training parameters, such as the learning rate, number of training epochs, and other hyperparameters. Additionally, it optimizes the weights ω and thresholds α in the BP neural network. A comparison shows that the FWA-BP algorithm notably enhanced predictive accuracy and meaningfully decreased errors.

Figure 4.5 illustrates the prediction errors for the BP network. The vertical axis spans -150 to 250, a 370 unit difference between the maximum and minimum prediction errors. This suggests that the BP algorithm exhibits significant fluctuations in prediction errors. Figure 4.6 illustrates the prediction errors for the FWA-BP network. Similarly, the error range falls within -50 to 40, which is 24% smaller than the BP network. The upper and lower error limits for the FWA-BP model is 110, a reduction of 260 compared to the BP network's prediction errors. This shows that the FWA-BP algorithm significantly outperforms the BP neural network in terms of fitting.

Figures 4.7 and 4.8 depict the linear regression fitting results for both algorithms. In Figure 4.7, the solid line and dashed line have substantial differences, with many data points lying far from the line. In contrast, in Figure 4.8, the solid line is notably closer to the dashed line, and most data points are clustered around it. This shows that FWA-BP algorithm has better fitting performance and verifies the superiority of FWA-BP algorithm.

5. Conclusion. Predicting wind energy production is crucial for properly allocating resources and scheduling wind power. More accurate predictions improve power system dependability and deliver major economic benefits. Since backpropagation (BP) neural networks for power forecasting can get trapped in local optima and are slow to converge, this study optimized them with a fireworks algorithm for estimating wind farm output. Historical data from an actual wind farm was leveraged to test predictions from both regular and optimized BP networks. Key findings show that the fireworks-enhanced BP neural network algorithm for wind power forecasting, reduces the error by 0.187569 compared to a standalone BP neural network, enhances training convergence speed, and provides predictions that are closer to the actual power output. This demonstrates a significant improvement in performance after optimizing the BP network. Moreover, the algorithm exhibits improved stability and reduced volatility.

While this study furnishes robust tools and methods for wind power forecasting, substantial potential remains for further research. Further investigation into improving the model's interpretability would be valuable for better understanding prediction results. Additionally, accounting for uncertainties can help provide more

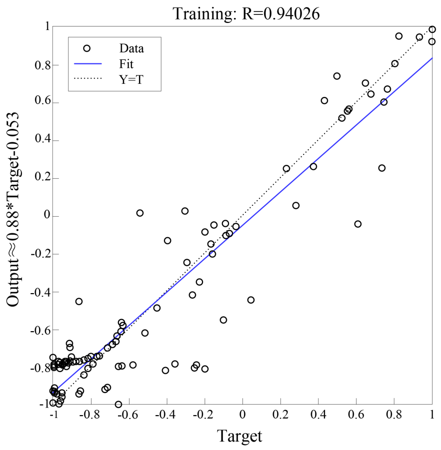

Fig. 4.7: BP algorithm's linear regression fitting performance.

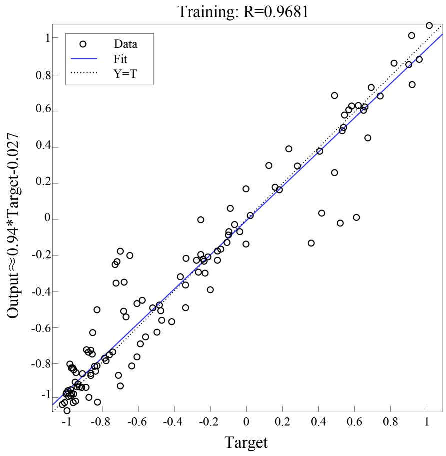

Fig. 4.8: FWA-BP algorithm's linear regression fitting results.

accurate estimations of the certainty in wind power output projections. It is crucial to apply the research outcomes to the administration and operation of actual wind farms to validate the algorithm's usefulness in real-world production settings and address any practical challenges.

Acnowledgements. This research was funded by State Grid Chongqing Technology Project (2023Yudian Technology No. 32).

REFERENCES

[1] Xiaokang Peng, Zicheng Liu and Dong Jiang, *A review of multiphase energy conversion in wind power generation*, Renew. Sust. Energ. Rev. 147: 111172 (2021).

[2] Archer, C. L., and Jacobson, M. Z, *Evaluation of global wind power*, J. Geophys. Res. 110: D12 (2005).

[3] Sukanta Roga, Shawli Bardhan, Yogesh Kumar and Sudhir K. Dubey, *Recent technology and challenges of wind energy generation: A review, Sustain*, Energy Technol. 52: 102239 (2022).

[4] Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang and Qianyi Liu, *A review of wind speed and wind power forecasting with deep neural networks*, Appl. Energy. 304: 117766 (2021).

[5] Farah Shahid, Aneela Zameer and Muhammad Muneeb, *A novel genetic LSTM model for wind power forecast*, Energy. 223: 120069 (2021).

[6] I. Akhtar, S. Kirmani and M. Jameel, *Reliability Assessment of Power System Considering the Impact of Renewable Energy Sources Integration Into Grid With Advanced Intelligent Strategies*, IEEE Access. 9: 32485-32497 (2021).

[7] Ilhami Colak, Seref Sagiroglu and Mehmet Yesilbudak, *Data mining and wind power prediction: A literature review*, Renew. Energ. 46: 241-247 (2012).

[8] Yang Li, Ruinong Wang, Yuanzheng Li, Meng Zhang and Chao Long, *Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach*, Appl. Energy. 329: 120291 (2023).

[9] O. Abedinia, M. Lotfi, M. Bagheri, B. Sobhani, M. Shafie-khah and J. P. S. Catalão, *Improved EMD-Based Complex Prediction Model for Wind Power Forecasting*, IEEE Transactions on Sustainable Energy. 11: 2790-2802 (2020).

[10] Barbosa de Alencar, D.; De Mattos Affonso, C.; Limão de Oliveira, R.C.; Moya Rodríguez, J.L.; Leite, J.C. and Reston Filho, J.C, *Different Models for Forecasting Wind Power Generation: Case Study*, Energies. 10: 1976 (2017).

[11] Oh, J.; Park, J.; Ok, C.; Ha, C. and Jun, H.-B, *A Study on the Wind Power Forecasting Model UsingTransfer Learning Approach*, Electronics. 11: 4125 (2022).

[12] Ellahi, M.; Usman, M.R.; Arif, W.; Usman, H.F.; Khan, W.A.; Satrya, G.B.; Daniel, K. and Shabbir, N, *Forecasting of Wind Speed and Power through FFNN and CFNN Using HPSOBA and MHPSO-BAACs Techniques*, Electronics. 11: 4193 (2022).

[13] M. A. Hossain, E. Gray, J. Lu, M. R. Islam, M. S. Alam, et al, *Optimized Forecasting Model to Improve the Accuracy of Very Short-Term Wind Power Prediction*, IEEE Transactions on Industrial Informatics. 19: 10145-10159 (2023).

[14] Aoife M. Foley, Paul G. Leahy, Antonino Marvuglia and Eamon J. McKeogh, *Current methods and advances in forecasting of wind power generation*, Renew. Energ. 37: 1-8 (2012).

[15] G. An, Z. Jiang, X. Cao, Y. Liang, Y. Zhao, et al, *Short-Term Wind Power Prediction Based On Particle Swarm Optimization-Extreme Learning Machine Model Combined With Adaboost Algorithm*, IEEE Access. 9: 94040-94052 (2021).

[16] Xin Liu, Jun Zhou and Huimin Qian, *Short-term wind power forecasting by stacked recurrent neural networks with parametric sine activation function*, Electr. Power Syst. Res. 192: 107011(2021).

[17] Li N, Wang Y, Ma W, Xiao Z and An Z, *A Wind Power Prediction Method Based on DE-BP Neural Network*, Front. Energy Res. 10: 844111 (2022).

[18] Hu, Dongmei and Zhang, Zhaoyun and Zhou, Hao, *Research on wind power Prediction based on BP neural Network*, 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). pp: 1-5 (2022).

[19] A. Žertek, G. Verbič and M. Pantoš, *Optimised control approach for frequency-control contribution of variable speed wind turbines*, IET Renew. Power Gener. 6: 17-23 (2012).

[20] Zhen-hai Guo, Jie Wu, Hai-yan Lu and Jian-zhou Wang, *A case study on a hybrid wind speed forecasting method using BP neural network*, Knowledge-Based Systems. 24: 1048-1056 (2011).

[21] Gong, ShuGong, Ren-XiHuang and Dai-Zheng, *Prediction of wind power by chaos and BP artificial neural networks approach based on genetic algorithm*, J ELECTR ENG TECHNOL. 10: 41-46 (2015).

[22] Zheng Wang, Bo Wang, Chun Liu and Wei-sheng Wang, *Improved BP neural network algorithm to wind power forecast*, JoE. 2017: 940-943 (2017).

[23] Li Xiaofeng, Xiang Suying, Zhu Pengfei and Wu Min, *Establishing a dynamic self-adaptation learning algorithm of the BP neural network and its applications*, INT J BIFURCAT CHAOS. 25: 1540030 (2015).

[24] Yin Fei, Mao Huajie, Hua Lin, Wei Guo and Maosheng Shu, *Back propagation neural network modeling for warpage prediction and optimization of plastic products during injection molding*, Materials & design. 32: 1844-1850 (2011).

[25] Vijay Kumar, Jitender Kumar Chhabra and Dinesh Kumar, *Optimal Choice of Parameters for Fireworks Algorithm*, Procedia Computer Science. 70: 334-340 (2015).

[26] Li J, Tan Y, *A Comprehensive Review of the Fireworks Algorithm*, CSUR. 52: 1-28 (2019).

[27] Tan, Y., Yu, C., Zheng, S., and Ding, K., *Introduction to Fireworks Algorithm*, IJSIR. 4: 39-70 (2013).

[28] Li, Y.; Tan, Y, *Hierarchical Collaborated Fireworks Algorithm*, Electronics. 11: 948 (2022).

[29] Wang, J., Fang, K., Pang, W., and Sun, J, *Wind Power Interval Prediction Based on Improved PSO and BP Neural Network*, J. Electr. 12: 989-995 (2017).

Edited by: Jingsha He
Special issue on: Efficient Scalable Computing based on IoT and Cloud Computing
Received: Dec 4, 2023
Accepted: Jan 18, 2024