Research Article

Short-Term Wind Speed Forecast Based on B-Spline Neural Network Optimized by PSO

Zhongqiang Wu, Wenjing Jia, Liru Zhao, and Changan Wu

*Key Lab of Industrial Computer Control Engineering of Hebei Province, College of Electric Engineering,
Yanshan University, Qinhuangdao 066004, China*

Correspondence should be addressed to Zhongqiang Wu; mewzq@163.com

Received 21 December 2014; Revised 24 March 2015; Accepted 7 April 2015

Academic Editor: Julien Bruchon

Copyright © 2015 Zhongqiang Wu 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.

Considering the randomness and volatility of wind, a method based on B-spline neural network optimized by particle swarm optimization is proposed to predict the short-term wind speed. The B-spline neural network can change the division of input space and the definition of basis function flexibly. For any input, only a few outputs of hidden layers are nonzero, the outputs are simple, and the convergence speed is fast, but it is easy to fall into local minimum. The traditional method to divide the input space is thoughtless and it will influence the final prediction accuracy. Particle swarm optimization is adopted to solve the problem by optimizing the nodes. Simulated results show that it has higher prediction accuracy than traditional B-spline neural network and BP neural network.

1. Introduction

Wind power is a kind of clean, free, and renewable natural resource. In wind power generation systems, the randomness and volatility of wind will affect the quality of power and the reliability of power system. Improving the prediction accuracy of short-term wind speed is of great significance for the real-time scheduling of power, reliability improving of power supply, and cost lowering of the wind power generation [1–3].

The common methods of wind speed forecasting include time series method, support vector machine (SVM) method, and wavelet analysis method. Time series method is adopted in [4, 5], making full use of the sample data and the prediction model is simple which can reduce the calculation time, but it does not take the correlation among the sample data into account, and the estimation of model orders is quite complicated. In [6, 7] SVM is adopted, SVM can make indivisible vector in low dimension space map into divided function in high dimension space by choosing appropriate kernel function, the generalizing ability is improved and the local minimization problem is solved, and it has higher prediction accuracy than time series method. The multiresolution

analysis character of wavelet transform and the generalizing ability of SVM are combined to realize the short-term wind speed forecasting in [8]. The method can change the nonlinear phenomenon very well and improve the prediction accuracy, but the design process is rather complex.

In view of the various defects of prediction methods above in practical application, a prediction method based on B-spline neural network optimized by particle swarm optimization (PSO-BSNN) is proposed. Firstly, the wind speed data are used to calculate correlation integral function [9, 10] and prove that the discrete time series is chaotic. By phase space reconstruction [11, 12], the high dimension chaotic attractors are recovered. BSNN can change the division of input space and the definition of basis function flexibly [13, 14]. For any input, only a few outputs of the hidden layers are nonzero, so the outputs are simple and convergence speed is fast, but traditional method which divides the input space into linear space is thoughtless and will influence the final prediction accuracy. PSO is an intelligent search method. It has strong global search ability and is also easy to be realized [15–20]. It is used to optimize the nodes of BSNN. Simulated results show that PSO-BSNN has higher prediction accuracy than BSNN and BPNN.

FIGURE 1: Wind speed time series.

2. Phase-Space Reconstruction

A series of continuous sample data are collected from a field of wind farm in Colorado State. Set the sample interval of wind speed as 10 min; there will be 6 sample data in each hour. Let the 6 data be averaged; then the mean value of each hour is got. The discrete sample data of 10 days are recorded as $\bar{x} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]$ ($n = 240$). For better usage of nonlinear nature of neural network transfer function, the sample data are normalized into the interval $[0, 1]$, and the normalization formula can be expressed as follows:

$$x_i = \frac{\bar{x}_i - \bar{x}_{\min}}{\bar{x}_{\max} - \bar{x}_{\min}}, \quad (0 \leq i \leq n), \quad (1)$$

where $\bar{x}_i$ is sample data, $\bar{x}_{\min}$ is minimum of all sample data, $\bar{x}_{\max}$ is the maximum of all sample data, x_i is normalization data in the interval $[0, 1]$, and n is the length of time sequence. The normalized wind speed v is displayed in Figure 1.

2.1. Takens Theorem. Takens has proved that an appropriate embedding dimension m can be found which satisfies $m \geq 2d + 1$, and d is the dimension of dynamic system. The principle of the dynamic system can be recovered by phase-space reconstruction. If τ is the delay time and m is embedding dimension, then $\mathbf{x} = [x_1, x_2, \dots, x_n]$ can be described as $\{\mathbf{X}_i\}$, after phase-space reconstruction. $\mathbf{X}_i$ is phase point of the reconstructed time series, which can be described as follows:

$$\mathbf{X}_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(m-1)\tau}] \in R^m, \quad (2)$$

$i = 1, 2, \dots, M,$

where $M = n - (m - 1)\tau$ is the number of phase point.

2.2. Parameter Selection. The selection of embedding dimension m and delay time τ is very important for the time series $\mathbf{x}$ to be reconstructed, which will influence the final prediction accuracy. The saturation correlation dimension method is given to calculate embedding dimension m , and the correlation integral function can be stated as follows:

$$C(r) = \frac{1}{M^2} \sum_{i,j=1}^M \theta(r - a_{i,j}), \quad r > 0, \quad (3)$$

FIGURE 2: The relation curve between $\ln r$ and $\ln C(r)$. FIGURE 3: The relation curve between e and m .

where r is searching radius, $a_{i,j} = \|\mathbf{X}_i - \mathbf{X}_j\|$, and $\theta(\cdot)$ is Heaviside unit function:

$$\theta(r - a_{i,j}) = \begin{cases} 0, & r - a_{i,j} \leq 0, \\ 1, & r - a_{i,j} > 0. \end{cases} \quad (4)$$

For a proper scope of r , the attractor dimension is $e = \ln C(r) / \ln r$; the relation curve between $\ln r$ and $\ln C(r)$ is obtained as in Figure 2. The relation curve between m and e is obtained as in Figure 3.

It can be seen from Figure 3 that when $m = 5$, the attractor dimension e reaches the maximum, so $m = 5$ is the saturation correlation dimension. The logarithm value of correlation integral function is less than zero; namely, $C(r) \in [0, 1]$ and then the change trend of $C(r)$ is decay exponentially, which indicates that the time series $\mathbf{x}$ is chaotic rather than noise series.

Mutual information method is used to determine the delay time τ . The information analysis of time series $\mathbf{X}_i$ can be defined as the entropy function:

$$H(\mathbf{X}_i) = - \sum_{l=0}^{m-1} P(x_{i+l\tau}) \log P(x_{i+l\tau}), \quad (5)$$

where $H(\cdot)$ is entropy function and $P(\cdot)$ is probability density function.

FIGURE 4: The relation curve between τ and $I(\tau)$.

If $\mathbf{X}_i$ and $\mathbf{X}_j$ are discrete random variables, its joint entropy function can be expressed as follows:

$$H(\mathbf{X}_i, \mathbf{X}_j) = - \sum_{l=0}^{m-1} \sum_{\mu=0}^{m-1} P(x_{i+l\tau}, x_{j+\mu\tau}) \log P(x_{i+l\tau}, x_{j+\mu\tau}). \quad (6)$$

Mutual information function is

$$I(\tau) = H(\mathbf{X}_i) + H(\mathbf{X}_j) - H(\mathbf{X}_i, \mathbf{X}_j). \quad (7)$$

The relation curve between τ and $I(\tau)$ is displayed in Figure 4.

Mutual information function $I(\tau)$ signifies the correlation between x_t and $x_{t+\tau}$. If $I(\tau) = 0$, they are completely irrelevant, and when $I(\tau)$ reaches the minimum for the first time, the value τ is the delay time required. It can be seen from Figure 4, and $\tau = 4$.

3. B-Spline Neural Network

BSNN is designed based on spline interpolation theory and it can change the division of input space and definition of the basis function flexibly. For any input, only a few outputs of the hidden layers are nonzero so that the outputs are simple and convergence speed is faster. It can be used to realize the nonlinear function approximation with high precision.

3.1. Basis Function of BSNN. According to (2), BSNN prediction model is structured as shown in Figure 5, where $\mathbf{X} = [x_0, \dots, x_l, \dots, x_{m-1}] \in R^m$ is the input vector and y is the output of the network and corresponds to the mean value of wind speed in the next hour.

(1) *The Division of Input Space.* The input space has to be divided before the basis function being defined. $\mathbf{U}_l$ is a finite interval for x_l :

$$\mathbf{U}_l = \{x_l : x_l^{\min} \leq x_l \leq x_l^{\max}\}, \quad (0 \leq l \leq m-1). \quad (8)$$

Then interval $\mathbf{U}_l$ can be divided as follows:

$$x_l^{\min} < \lambda_{l,1} \leq \dots \leq \lambda_{l,\rho} \leq \dots \leq \lambda_{l,q_l} < x_l^{\max}, \quad (9)$$

FIGURE 5: Prediction model of BSNN.

where $\lambda_{l,\rho}$ is the ρ th internal node, and the external node can be defined as follows:

$$\begin{aligned} \dots \lambda_{l,\rho-1} &\leq \lambda_{l,0} \leq x_l^{\min}, \\ x_l^{\max} &\leq \lambda_{l,q_l+1} \leq \lambda_{l,q_l+2} \dots \end{aligned} \quad (10)$$

The interval $\mathbf{U}_l$ can be divided into $q_l + 1$ subinterval by all of the nodes as follows:

$$\mathbf{U}_{l,\rho} = x_l : \begin{cases} x_l \in [\lambda_{l,\rho}, \lambda_{l,\rho+1}), & \rho = 0, 1, \dots, q_l - 1, \\ x_l \in [\lambda_{l,\rho}, \lambda_{l,\rho+1}], & \rho = q_l. \end{cases} \quad (11)$$

(2) *The Multivariable Basis Function of BSNN.* Let $\alpha(x_l)$ indicate k -order basis function for x_l , its ρ th basis function is $N_\rho^k(x_l)$ which is defined by $\lambda_{l,(\rho-k\eta)}, \lambda_{l,(\rho-k\eta+1)}, \dots, \lambda_{l,\rho}$, η is expansion factor, and the recurrence relations can be expressed as follows:

$$\begin{aligned} N_\rho^k(x_l) &= \left(\frac{x_l - \lambda_{l,(\rho-k\eta)}}{\lambda_{l,(\rho-\eta)} - \lambda_{l,(\rho-k\eta)}} \right) N_{\rho-\eta}^{k-1}(x_l) \\ &\quad + \left(\frac{\lambda_{l,\rho} - x_l}{\lambda_{l,\rho} - \lambda_{l,(\rho-(k-1)\eta)}} \right) N_\rho^{k-1}(x_l), \quad (12) \\ N_\rho^1(x_l) &= \begin{cases} 0, & x_l \in [\lambda_{l,(\rho-\eta)}, \lambda_{l,\rho}), \\ 1, & x_l \notin [\lambda_{l,(\rho-\eta)}, \lambda_{l,\rho}). \end{cases} \end{aligned}$$

If the number of internal nodes is 2 and expansion factor $\eta = 1$, the basis function graph of 3 degrees BSNN for single variable x_l is shown in Figure 6.

The number of basis function for $\mathbf{X}$ is

$$q = \prod_{l=0}^{m-1} q_l. \quad (13)$$

The basis function of $\mathbf{X}$ is got as the result that single variable basis functions combine with each other:

$$\alpha_{0,1,\dots,m-1}(\mathbf{X}) = \prod_{l=0}^{m-1} \alpha(x_l) = \prod_{l=0}^{m-1} \prod_{\rho=1}^{q_l} N_\rho^k(x_l). \quad (14)$$

FIGURE 6: The basis function of BSNN.

3.2. *The Mapping of BSNN.* The network from input to output can be divided into two parts.

(1) $\mathbf{X} \rightarrow \boldsymbol{\alpha}(\mathbf{X})$.

In this section, it realizes the nonlinear mapping from input layer to hidden layer and set $\boldsymbol{\alpha}(\mathbf{X}) = [\alpha_1(\mathbf{X}), \dots, \alpha_r(\mathbf{X}), \dots, \alpha_q(\mathbf{X})]^T \in R^q$, where $\alpha_r(\mathbf{X})$ is output value of the r th basis function for $\mathbf{X}$. According to the positivity and tightness of basis function, the number of output nonzero values is $b = (k\eta)^m$, $\boldsymbol{\alpha}(\mathbf{X})$ belongs to the interval $[0, 1]$, and the other output values are zero. If we let $\eta = 1$, $b = k^m$ is got.

(2) $\boldsymbol{\alpha}(\mathbf{X}) \rightarrow y$.

In this section, it realizes linear mapping from hidden layer to output layer:

$$y = \boldsymbol{\omega}(t) \boldsymbol{\alpha}(\mathbf{X}), \quad (15)$$

where $\boldsymbol{\omega}(t) = [\omega_0(t), \dots, \omega_r(t), \dots, \omega_q(t)]$ is the weight value of the network. The weight value can be updated as follows:

$$\boldsymbol{\omega}(t+1) = \boldsymbol{\omega}(t) + \frac{\beta [y^d - y] \boldsymbol{\alpha}^T(\mathbf{X})}{\boldsymbol{\alpha}^T(\mathbf{X}) \boldsymbol{\alpha}(\mathbf{X})}, \quad (16)$$

where t is the number of iteration, y^d is the expected value, y is the actual value, and β ($0 < \beta < 2$) is learning rate and ensures that the iterative algorithm can converge. The verification is as follows.

Set $\varepsilon(t) = y^d - y = y^d - \boldsymbol{\omega}(t) \boldsymbol{\alpha}(\mathbf{X})$; then

$$\begin{aligned} \Delta \varepsilon(t) &= \varepsilon(t+1) - \varepsilon(t) = -[\boldsymbol{\omega}(t+1) - \boldsymbol{\omega}(t)] \boldsymbol{\alpha}(\mathbf{X}) \\ &= -\Delta \boldsymbol{\omega}(t) \boldsymbol{\alpha}(\mathbf{X}). \end{aligned} \quad (17)$$

Set the Lyapunov function as $V(t) = \varepsilon^2(t)$; then

$$\begin{aligned} \Delta V(t) &= V(t+1) - V(t) = \varepsilon^2(t+1) - \varepsilon^2(t) \\ &= [\varepsilon(t+1) - \varepsilon(t)] [2\varepsilon(t) + \varepsilon(t+1) - \varepsilon(t)] \\ &= \Delta \varepsilon(t) [2\varepsilon(t) + \Delta \varepsilon(t)] \\ &= -\Delta \boldsymbol{\omega}(t) \boldsymbol{\alpha}(\mathbf{X}) [2\varepsilon(t) - \Delta \boldsymbol{\omega}(t) \boldsymbol{\alpha}(\mathbf{X})] \\ &= -\beta \varepsilon(t) [2\varepsilon(t) + \beta \varepsilon(t)] = -\beta(2 - \beta) \varepsilon^2(t). \end{aligned} \quad (18)$$

Since $0 < \beta < 2$, then $\Delta V(t) < 0$, and the parameters can converge.

4. B-Spline Neural Network Optimized by PSO

The internal nodes $\lambda_{l,\rho}$ ($\rho = 1, \dots, q_l$) of BSNN have a great relationship with the prediction accuracy. The traditional method which divides the input space into linear space is thoughtless and it is difficult to get higher approximation accuracy. PSO is used to optimize the parameters, which can improve the accuracy and avoid falling into local minimum.

4.1. *The Basic Principle of PSO.* PSO stems from the enlightenment of birds foraging behavior, which is mainly used to solve the optimization problem. Each particle is a point with a certain speed in the solution space and it also has the related individual fitness which corresponds to the objective function; the effectiveness of the solution is determined by the fitness function that is defined on the basis of optimized goals. Particles are following the current optimum particles in solution space and finding the optimal solution by iteration.

Set the current position as $\mathbf{p}_s = (p_{s1}, \dots, p_{sd}, \dots, p_{sD})$, the current speed as $\mathbf{v}_s = (v_{s1}, \dots, v_{sd}, \dots, v_{sD})$, and the best position that the particle has experienced as $\mathbf{p}_{\text{best}} = (p_{\text{best}1}, \dots, p_{\text{best}d}, \dots, p_{\text{best}D})$ in D dimension search space. Set the number of all particles as N ; the best position of all the particles experienced is $\mathbf{g}_{\text{best}} = (g_{\text{best}1}, \dots, g_{\text{best}d}, \dots, g_{\text{best}D})$. The particles can update themselves by tracking individual extremum $\mathbf{p}_{\text{best}}$ and global extremum $\mathbf{g}_{\text{best}}$ during each iteration. In the process of finding the two extremums, the particles update their own speed and position as follows:

$$\begin{aligned} v_{sd}(t+1) &= w v_{sd}(t) + c_1 r_1 (p_{\text{best}d}(t) - p_{sd}(t)) \\ &\quad + c_2 r_2 (g_{\text{best}d}(t) - p_{sd}(t)), \\ p_{sd}(t+1) &= p_{sd}(t) + v_{sd}(t+1), \end{aligned} \quad (19)$$

where c_1, c_2 are acceleration constant; $v_{sd}(t)$ is the speed of the d th particle; $p_{sd}(t)$ is the position of the d th particle; r_1, r_2 are random number and they belong to the interval $[0, 1]$; w is weight value and is used to balance global searching and local searching, and w can update itself as follows:

$$w = w_{\max} - \frac{t}{t_{\max}} (w_{\max} - w_{\min}), \quad (20)$$

where $w_{\max}$ is the maximum of w and $w_{\min}$ is the minimum of w ; t is the current number of iteration and $t_{\max}$ is the maximum of iteration.

In order to prevent the particle far from the search space, the speed of each particle will be limited in a maximum $v_{\max}$ ($v_{\max} > 0$); if the updated speed exceeds the specified speed, then set the speed of the particle as $v_{\max}$, $v_{\max} = \delta p_{\max}$, where δ belongs to the interval $(0, 1]$ and $p_{\max}$ is the maximum of p_{sd} .

4.2. *PSO-BSNN Algorithm Implementation.* The traditional method for BSNN to divide the internal nodes of input space is thoughtless. If the internal nodes cannot be determined

accurately, the BSNN is easy to fall into local minimum and then the prediction accuracy is influenced. PSO is a kind of intelligent search method; the internal nodes can be optimized by using PSO algorithm and then get the optimal location. The dimension of each particle vector is D , and each vector is composed of internal and external nodes $\lambda_{l,\rho}$ of BSNN. For the single input variable x_{il} , the number of internal nodes is 2 and the external nodes is 4; then the vector dimension $D = 6$. The optimization process is as follows.

(1) Initialization: the basic initial parameters include the position $\mathbf{p}_s = (p_{s1}, \dots, p_{s6}) = (\lambda_{il,-1}, \lambda_{il,0}, \lambda_{il,1}, \lambda_{il,2}, \lambda_{il,3}, \lambda_{il,4})$, speed of each particle $\mathbf{v}_s$, the population of particles N , the largest iteration number $t_{\max}$, and the acceleration constants c_1, c_2 .

(2) Fitness Evaluation: the error between expected value and actual value is selected as objective function which is used to measure the effectiveness of optimization. The mean square error e_{MSE} is adopted as the fitness function:

$$f = e_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i^d - y_i)^2 \quad (0 \leq i \leq N), \quad (21)$$

where N is the number of sample date, y_i^d is the expected value for the i th particle, and y_i is the actual output value for the i th particle.

(3) Update the parameters according to formula (17) and limit the speed according to $v_{\max}$.

(4) Calculate the fitness evaluation value, comparing it with the optimal fitness evaluation value; if better, update the particle location; otherwise do not change.

(5) Judge whether the result satisfies the termination condition. The termination condition is whether it reaches the maximum number of iterations or satisfies mean square error. If it does, then it is over, or return to step (2).

5. Simulation Results and Analysis

Let the largest iteration number $t_{\max} = 100$, the acceleration constant $c_1 = c_2 = 3$, and the population of particles $N = 30$, and let $r_1 = 0.3$, $r_2 = 0.6$, and the initial position and speed of each particle are got randomly in the defined interval. Adopting 3 orders B-spline neural network, the number of neural network input is 5. The internal node number of each input variable is 2 and expansion factor is 1; then the nodes number of hidden layer units is 3^5 and the nonzero output nodes number of hidden layer is 2^5 . There is only one output unit.

Three kinds of prediction error are adopted as criteria for evaluation:

$$\begin{aligned} e_{\text{RMSE}} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2}, \\ e_{\text{MAE}} &= \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*|, \\ e_{\text{MRE}} &= \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y_i^*}{y_i} \right|, \end{aligned} \quad (22)$$

FIGURE 7: Prediction result of 3 days.

TABLE 1: Forecasting error of each prediction model.

Prediction model	$e_{\text{RMSE}}/(\text{m/s})$	$e_{\text{MAE}}/(\text{m/s})$	$e_{\text{MRE}}/\%$
PSO-BSNN	0.0319	0.0348	0.0826
BSNN	0.0700	0.0664	0.1555
BPNN	0.1060	0.1017	0.2267

where e_{RMSE} is root mean square error and it is used to measure the deviation between prediction value and real value, e_{MAE} is mean absolute error that reflects the error of prediction value deviating from real value, and e_{MRE} is mean relative error that reflects the reliability of mean absolute error value.

According to the wind speed date of 10 days in Figure 1 that is provided from the wind form in Colorado State, the first data of 7 days is used as training sample to train neural network and the last date of 3 days is used as test sample to examine the results of prediction. Simulation results based on PSO-BSNN, BSNN, and BPNN are displayed in Figure 7. They all adopt 3-layer network structures.

The prediction error is listed in Table 1.

It can be seen from Table 1 that the prediction model based on BSNN has better performance than the prediction model based on BPNN, e_{RMSE} and e_{MAE} have fallen by 0.2455 m/s and 0.2614 m/s, respectively, e_{MRE} has fallen by 14.54%, and the prediction effects have some improvements. Comparing with the prediction model based on BSNN, the prediction model based on PSO-BSNN enhances the prediction performance largely, because the input space of BSNN has been optimized by particle swarm. It gets higher prediction accuracy than the prediction model based on BPNN, e_{RMSE} and e_{MAE} have fallen by 0.4198 m/s and 0.4233 m/s, respectively, and e_{MRE} has fallen by 36.50%.

Now BPNN is set 4-layer network structures, and the simulation result is shown in Figure 8.

The prediction error is listed in Table 2.

It can be seen from Table 2 that the simulation result based on 4-layer BPNN structure is very similar to BSNN, but still poorer than the prediction result based on PSO-BSNN.

FIGURE 8: The prediction result of 3 days with 4-layer BPNN.

TABLE 2: Forecasting error of BPNN prediction model.

Prediction model	$e_{RMSE}/(m/s)$	$e_{MAE}/(m/s)$	$e_{MRE}/\%$
BPNN	0.0810	0.0697	0.1603

TABLE 3: Forecasting error of each prediction model.

Prediction model	$e_{RMSE}/(m/s)$	$e_{MAE}/(m/s)$	$e_{MRE}/\%$
PSO-BSNN	0.0419	0.0448	0.0926
BSNN	0.0912	0.1064	0.1555
BPNN	0.1377	0.1410	0.2431

The prediction of 3-day wind speed date belongs to short-term wind speed prediction. Now, the wind speed date of 10 hours from Sinkiang in China is used in prediction. The first data of 7 hours is used as training sample and last wind speed date of 3 hours is used as test sample. It belongs to super short-term wind speed prediction. The sample interval of wind speed is still set as 10 min, there will be 6 sample data in each hour, and the number of sample dates of 3 hours is 18. The simulation result is shown in Figure 9.

The prediction error is listed in Table 3.

It can be seen from Table 3 that comparing with BPNN, in the simulation results based on BSNN, e_{RMSE} and e_{MAE} have fallen by 0.0465 m/s and 0.0346 m/s, respectively, and e_{MRE} has fallen by 8.76%. Comparing with BSNN, in the simulation results based on PSO-BSNN, e_{RMSE} and e_{MAE} have fallen by 0.0493 m/s and 0.0616 m/s, respectively, and e_{MRE} has fallen by 6.29%.

Very short-term wind speed prediction results based on 4-layer BP neural network structure are shown in Figure 10.

The prediction error is listed in Table 4.

It can be seen from Table 4 that the super short-term wind speed prediction based on 4-layer BPNN is very similar to BSNN, but still poorer than the prediction result based on PSO-BSNN.

According to the simulation results of short-term and super short-term wind speed prediction, the prediction model based on PSO-BSNN always gets better results than BSNN and BPNN.

FIGURE 9: Prediction result of 3 hours.

FIGURE 10: Prediction result of 3 hours with 4-layer BPNN.

TABLE 4: Forecasting error of BPNN prediction model.

Prediction model	$e_{RMSE}/(m/s)$	$e_{MAE}/(m/s)$	$e_{MRE}/\%$
BPNN	0.1023	0.1152	0.1825

6. Conclusions

A prediction model based on BSNN optimized by PSO is present in this paper. BSNN is designed based on B-spline interpolation theory, it can change the input space division flexibly, and the traditional method to divide the input space into linear space is thoughtless which will influence the final prediction accuracy. PSO is adopted to solve the problem. Simulation results show that the PSO-BSNN has better performance than BSNN and BPNN, and the forecasting accuracy is improved effectively.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] R. Belu and D. Koracin, "Wind characteristics and wind energy potential in western Nevada," *Renewable Energy*, vol. 34, no. 10, pp. 2246–2251, 2009.
- [2] K. Xie and R. Billinton, "Energy and reliability benefits of wind energy conversion systems," *Renewable Energy*, vol. 36, no. 7, pp. 1983–1988, 2011.
- [3] M. C. Alexiadis, P. S. Dokopoulos, H. S. Sahsamanoglou, and I. M. Manousaridis, "Short-term forecasting of wind speed and related electrical power," *Solar Energy*, vol. 63, no. 1, pp. 61–68, 1998.
- [4] A. Yona, T. Senjyu, F. Toshihisa, and C. Kim, "Very short-term generating power forecasting for wind power generators based on time series analysis," *Smart Grid and Renewable Energy*, vol. 4, no. 2, pp. 181–186, 2013.
- [5] E. A. Bossanyi, "Short-term wind prediction using Kalman filters," *Wind Engineering*, vol. 9, no. 1, pp. 1–8, 1985.
- [6] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renewable Energy*, vol. 29, no. 6, pp. 939–947, 2004.
- [7] S. Salcedo-Sanz, E. G. Ortiz-García, and A. M. Pérez-Bellido, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4052–4057, 2011.
- [8] O. Kisi, J. Shiri, and O. Makarynsky, "Wind speed prediction by using different wavelet conjunction models," *The International Journal of Ocean and Climate Systems*, vol. 2, no. 3, pp. 189–208, 2011.
- [9] H. Leung, T. Lo, and S. Wang, "Prediction of noisy chaotic time series using an optimal radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1163–1172, 2001.
- [10] D. S. K. Karunasinghe and S.-Y. Liong, "Chaotic time series prediction with a global model: artificial neural network," *Journal of Hydrology*, vol. 323, no. 1–4, pp. 92–105, 2006.
- [11] J. Zhang, K. C. Lam, W. J. Yan, H. Gao, and Y. Li, "Time series prediction using Lyapunov exponents in embedding phase space," *Computers & Electrical Engineering*, vol. 30, no. 1, pp. 1–15, 2004.
- [12] H. Ma and C. Han, "Selection of embedding dimension and delay time in phase space reconstruction," *Frontiers of Electrical and Electronic Engineering in China*, vol. 1, no. 1, pp. 111–114, 2006.
- [13] A. Kechroud, J. J. H. Paulides, and E. A. Lomonova, "B-spline neural network approach to inverse problems in switched reluctance motor optimal design," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 4179–4182, 2011.
- [14] L. D. S. Coelho and M. W. Pessôa, "Nonlinear identification using a B-spline neural network and chaotic immune approaches," *Mechanical Systems and Signal Processing*, vol. 23, no. 8, pp. 2418–2434, 2009.
- [15] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [16] Y. Marinakis, M. Marinaki, and G. Dounias, "A hybrid particle swarm optimization algorithm for the vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 463–472, 2010.
- [17] B. Brandstätter and U. Baumgartner, "Particle swarm optimization—mass-spring system analogon," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 997–1000, 2002.
- [18] J. Botzheim, C. Cabrita, and L. T. K&ocaron;oczy, "Genetic and bacterial programming for B-spline neural networks design," *Journal of Advanced Computational Intelligence and Intelligence Informatics*, vol. 11, no. 2, pp. 220–231, 2007.
- [19] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, pp. 81–86, IEEE, Seoul, South Korea, May 2001.
- [20] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 110–117, Indianapolis, Ind, USA, April 2003.