

Article

Wind Speed Modeling for Wind Farms Based on Deterministic Broad Learning System

Lin Wang^{1,2,*} and Anke Xue¹¹ School of automation, Hangzhou Dianzi University, Hangzhou 310018, China; akxue@hdu.edu.cn² School of Electrical Engineering, Yancheng Institute of Technology, Yancheng 224051, China

* Correspondence: linwang@ycit.edu.cn

Abstract: As the penetration rate of wind power in the grid continues to increase, wind speed forecasting plays a crucial role in wind power generation systems. Wind speed prediction helps optimize the operation and management of wind power generation, enhancing efficiency and reliability. However, wind speed is a nonlinear and nonstationary system, and traditional statistical methods and classical intelligent algorithms struggle to cope with dynamically updating operating conditions based on sampled data. Therefore, from the perspective of optimizing intelligent algorithms, a wind speed prediction model for wind farms was researched. In this study, we propose the Deterministic Broad Learning System (DBLS) algorithm for wind farm wind speed prediction. It effectively addresses the issues of data saturation and local minima that often occur in continuous-time system modeling. To adapt to the continuous updating of sample data, we improve the sample input of the Broad Learning System (BLS) by using a fixed-width input. When new samples are added, an equivalent number of old samples is removed to maintain the same input width, ensuring the feature capture capability of the model. Additionally, we construct a dataset of wind speed samples from 10 wind farms in Gansu Province, China. Based on this dataset, we conducted comparative experiments between the DBLS and other algorithms such as Random Forest (RF), Support Vector Regression (SVR), Extreme Learning Machines (ELM), and BLS. The comparison analysis of different algorithms was conducted using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Among them, the DBLS algorithm exhibited the best performance. The RMSE of the DBLS ranged from 0.762 m/s to 0.776 m/s, and the MAPE of the DBLS ranged from 0.138 to 0.149.



Citation: Wang, L.; Xue, A. Wind Speed Modeling for Wind Farms Based on Deterministic Broad Learning System. *Atmosphere* **2023**, *14*, 1308. <https://doi.org/10.3390/atmos14081308>

Academic Editor: Sunil Jha

Received: 23 July 2023

Revised: 11 August 2023

Accepted: 15 August 2023

Published: 18 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: wind speed; modeling; Deterministic Broad Learning System (DBLS)

1. Introduction

Today, the collection of time series data is widely used in many fields. By processing and analyzing historical time series data, complex system models can be established [1]. These models can predict the development trends of systems and effectively solve problems where the operating mechanisms of complex systems are difficult to analyze [2–4]. Based on the nature of time series data, the requirements of the system problem, and the suitability of algorithms, common processing algorithms for time series data include univariate processing algorithms, multivariate processing algorithms [5], flat processing algorithms, linear processing algorithms, nonlinear processing algorithms, and algorithms for handling different time scales [6]. Researchers aim to develop models that accurately predict future values and thus forecast the numerical values, trends, or patterns of future time points or periods based on existing historical data [7–9].

Researchers have conducted extensive research on the analysis of complex system problems and the stability and accuracy of time series prediction models. Many well-performing research outcomes have been achieved. Common statistical algorithms include Auto Regression Moving Average (ARMA) [10], Auto Regressive Integrated Moving Average (ARIMA) [11], Seasonal Auto-Regressive Integrated Moving Average (SARIMA) [12],

Seasonal Trend Loss (STL) [13], Exponential Smoothing, Linear Regression, and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) [14]. Common intelligent algorithms include Artificial Neural Networks (ANNs) [15], Long Short-Term Memory (LSTM) [16] and its variants, Convolutional Neural Networks (CNNs) [17,18] and their variants, Support Vector Regression (SVR) [19,20] and its variants, Random Forest [21–23], Deep Learning [24,25] and its variants, and Genetic Algorithms [26,27] and their variants. These algorithms represent static nonlinear mapping relationships, and the stability and accuracy of prediction models are affected when sample data are dynamically updated. Additionally, most of these algorithms are based on iterative processes using gradient information, which results in time-consuming model updates and high computational resource utilization when sample data are dynamically updated.

Single-Layer Feedforward Neural Networks (SLFNs) are widely used for time series data modeling due to their simple structure and high computational efficiency [28]. However, SLFNs are sensitive to parameter settings and learning rates. In light of this, some researchers have proposed the Random Vector Functional-Link Neural Network (RVFLNN) [29], which is widely applied in time series modeling analysis and control systems due to its approximation performance and fast learning characteristics. However, the predictive performance and stability of the RVFLNN model are diminished when dealing with high-dimensional and time-varying datasets. Chen C L P and his colleagues proposed the Broad Learning System (BLS) [30–32], which is suitable for multidimensional and time-varying datasets. This algorithm enables fast modeling and rapid model updates, making it applicable in various fields. Cheng Y and others proposed the D-BLS algorithm for load trend prediction [33]. Wang M and his team combined adaptive Kalman filtering with the BLS for battery-charging-state detection, achieving excellent detection results [34]. Pu X and his colleagues developed an online semisupervised Broad Learning System and applied it to industrial fault detection, obtaining satisfactory performance [35]. Chu F and his collaborators applied the BLS to nonlinear systems, effectively capturing nonlinear features and system descriptions, thus improving the predictive accuracy and robustness of the model [36]. Additionally, the BLS has also found numerous applications in the energy sector.

Wind condition prediction in wind farms plays a crucial role in optimizing the power output of wind turbines, ensuring the efficient operation of the wind farm, and facilitating the grid integration and scheduling of wind power. Wind conditions in wind farms are influenced by factors such as temperature, pressure, humidity, altitude, season, atmospheric circulation, time, latitude and longitude, and geographical environment. A wind farm is a complex and nonlinear system with multiple couplings, representing a continuous-time series system. Currently, scholars have conducted extensive research to establish accurate and effective wind condition models for wind farms. Chen, CL Philip provided an approximation proof of the BLS in the literature [37] and applied it to wind speed prediction, greatly reducing the training time of samples. Wang S and others combined statistics with the Broad Learning System to effectively predict the formation of offshore storms [38]. Jiao, Xuguo, and their team developed the VMD-BLS wind speed prediction model and improved the model's prediction accuracy through error compensation [39]. However, when considering the continuous-time updating of sample data, the stability and accuracy of the prediction models for wind farm conditions and power generation have difficulty meeting the desired expectations.

When the sample data are continuously updated, the aforementioned algorithms are prone to data saturation and getting trapped in local minima. In most cases, the stability and accuracy of the prediction models are greatly reduced. It is difficult to achieve reliable predictions under the continuous updating of sample data. In the context of continuous updates to wind farm wind speed, the mentioned algorithms may experience data saturation, leading to a loss of the ability to capture dynamic features within the models. The algorithms mentioned above may encounter data saturation and lose the ability to capture dynamic features when the sample data are continuously updated. In

this paper, we propose the DBLS algorithm and apply it to wind speed prediction under continuous operating conditions, resulting in stable and accurate prediction models. To adapt to the continuous updating of sample data, the system input is changed to a fixed-width input. When new samples are added, an equal number of old samples are removed to maintain a constant input width for the system. This approach preserves some of the original features while incorporating new system characteristics through the addition of new samples. We established a sample dataset of wind speed from 10 wind farms in Gansu Province, China, and applied the DBLS algorithm to the constructed dataset. Furthermore, comparative experiments were conducted between the DBLS algorithm and the RF, SVR, ELM, and BLS algorithms. The experimental results demonstrate that the DBLS algorithm exhibits good stability and prediction accuracy.

The main contributions of this research are as follows:

1. The primary contribution of this research project is the introduction of the Deterministic Broad Learning System, addressing the issues of data saturation and local minima that commonly occur when the sample data are continuously updated. By adapting the system input to a fixed-width input, the proposed model achieves good prediction accuracy.
2. The collection and establishment of a sample dataset of wind speed from 10 wind farms in Gansu Province, China.
3. The application of the DBLS algorithm to wind speed prediction in wind farms, with comparative experiments conducted against RF, SVR, ELM, and BLS. The experimental results demonstrate that the DBLS algorithm performs well in terms of stability and prediction accuracy.

2. BLS

Chen [30] presented the BLS architecture based on RVFLNNs. Unlike deep networks, it does not have multiple layers of deep network superposition, only containing an input layer, feature layer, and output layer. The characteristics of the BLS are as follows:

- (1) The input of the BLS is to perform a linear random mapping of samples to form feature nodes, which facilitates the network to be easily integrated with other neural networks when required.
- (2) In the output layer, the BLS takes the feature node and the enhancement node together as the input of the output layer. Here, we notice that the feature node is a linear mapping of the input sample, which can effectively reduce the loss of sample features.

The BLS network structure is shown in Figure 1. The input sample X is mapped as a feature node through a linear random link function. The feature nodes as a whole map different enhancement nodes through nonlinear random functions. The middle layer of the BLS is composed of feature nodes and enhancement nodes. The specific operation principle will be analyzed in detail in the following content.

The mathematical theoretical support for the BLS is given in [37]. Therefore, only a brief overview of the principles of the BLS is provided. Feature nodes are calculated using Equation (1) [37].

$$Z_i = \phi_i(XW_{ei} + \beta_{ei}), i = 1, 2, \dots, n \quad (1)$$

In Equation (1), $\{X, Y\} \in \mathbb{R}^{N \times (M+C)}$ is the input sample set. Z_i is the i th feature window obtained by solving Formula (1). ϕ_i is the linear mapping function. W_{ei} and β_{ei} are randomly generated weights and offsets, respectively.

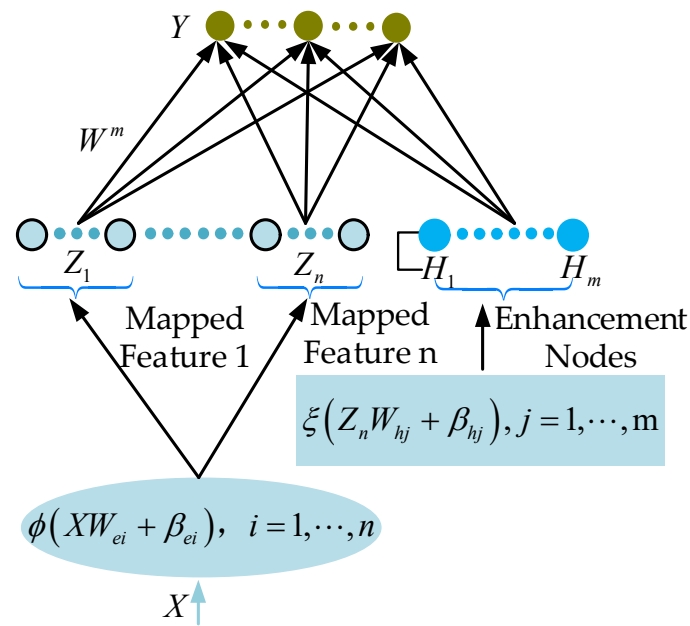


Figure 1. The structure of BLS [37].

The enhancement nodes are obtained via Equation (2) [37], which is a linear function.

$$H_j = \xi_j(XW_{hj} + \beta_{hj}), j = 1, 2, \dots, m \tag{2}$$

where W_{hj} and β_{hj} are randomly generated weights and offsets.

On the basis of Equations (1) and (2), the output layer of the BLS can be expressed via Equation (3) [37].

$$Y = [Z_1, Z_2, \dots, Z_n, H_1, H_2, \dots, H_m]W^m = [Z^n, H^m]W^m \tag{3}$$

where $W^m = [Z^n, H^m]^+ Y$. W^m is the output function weight.

3. Incremental Broad Learning System

In order to better apply the BLS network, the continuous updating of network input is studied. Generally, when the input sample is updated, most network structures update the weight and bias via retraining, while the BLS can update the weight and bias without retraining. This feature greatly reduces the training time and provides a good basis for online training.

Let the original input sample dataset be $X = [x_k, \dots, x_{k+n-1}]$, where x_{k+n} is an additional input sample. At this point, the structure of the BLS network is shown in Figure 2. The middle layer of the BLS can be represented as $A_{(k,k+n)}$. $A_{(k,k+n)}$ can be found using Equation (4).

$$A_{(k+1,k+n)} = \begin{bmatrix} A_{(k,k+n-1)} \\ A_{x_{k+n}}^T \end{bmatrix} = \begin{bmatrix} Z_k, Z_{k+1}, \dots, Z_{k+n-1} | H_1, H_2, \dots, Z_m \\ [\phi(X_{k+n}W_{e1} + \beta_{e1}), \dots, \phi(X_{k+n}W_{en} + \beta_{en}) | \xi(Z_{k+n}W_{h1} + \beta_{h1}), \dots, \xi(Z_{k+n}W_{hm} + \beta_{hm})]^T \end{bmatrix} \tag{4}$$

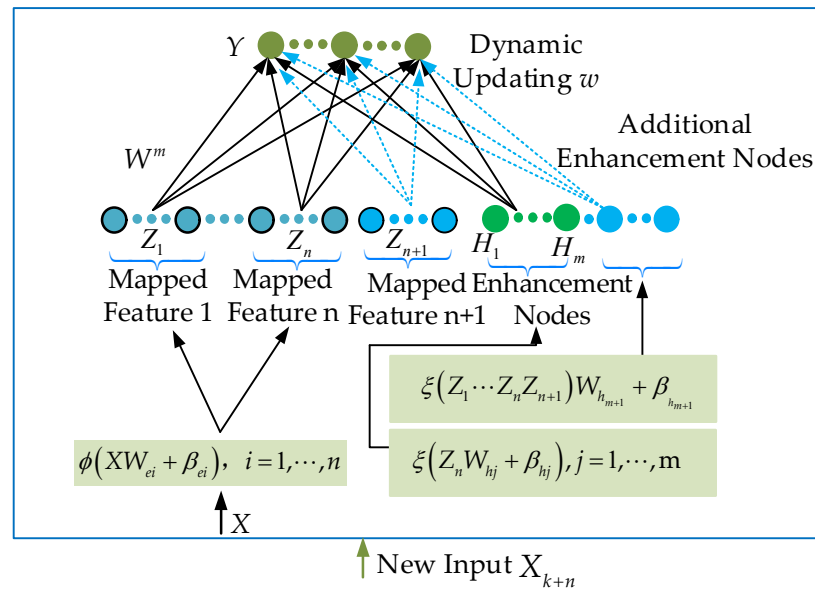


Figure 2. BLS structure with increasing input samples [37].

In Equation (4), $A_{(k,k+n-1)}$ represents the intermediate layer of the BLS at the time of the original sample set input. $A_{x_{k+n}}^T$ is the middle layer addition variable corresponding to the added input sample x_{k+n} .

The output weights of the BLS network are updated to $W_{(k,k+n)}$, which can be obtained by deriving Equation (5).

$$\min_{W(k,k+n)} = \|A_{(k,k+n)}W_{(k,k+n)} - Y_{(k,k+n)}\| \tag{5}$$

thus finding

$$W_{(k,k+n)} = (A_{(k,k+n)}^T A_{(k,k+n)})^{-1} A_{(k,k+n)}^T Y_{(k,k+n)} \tag{6}$$

$Y_{x_{k+n}}$ is the label for the new input sample x_{k+n+1} .

Let $B_{(k,k+n-1)} = A_{(k,k+n-1)}^T A_{(k,k+n-1)}$. Furthermore, it follows that

$$B_{(k,k+n)} = \begin{bmatrix} A_{(k,k+n-1)}^T \\ A_{x_{k+n}} \end{bmatrix}^T \begin{bmatrix} A_{(k,k+n-1)} \\ A_{x_{k+n}}^T \end{bmatrix} = B_{(k,k+n-1)} + A_{x_{k+n}} A_{x_{k+n}}^T \tag{7}$$

By deducing from Equation (6), it follows that

$$\begin{aligned} W_{(k,k+n)} &= (A_{(k,k+n)}^T A_{(k,k+n)})^{-1} A_{(k,k+n)}^T Y_{(k,k+n)} \\ &= B_{(k,k+n-1)}^{-1} \sum_{l=k}^{l=k+n} A_{x_l} Y_l \\ &= B_{(k,k+n-1)}^{-1} [A_{(k,k+n-1)}^T Y_{(k,k+n-1)} + A_{x_{k+n}} Y_{x_{k+n}}] \\ &= W_{(k,k+n-1)} + B_{(k,k+n)}^{-1} A_{x_{k+n}} [Y_{x_{k+n}} - A_{x_{k+n}}^T W_{(k,k+n-1)}] \end{aligned} \tag{8}$$

$$\begin{aligned} B_{(k,k+n)}^{-1} &= B_{(k,k+n-1)}^{-1} - B_{(k,k+n-1)}^{-1} A_{x_{k+n}} [\mathbf{I} + A_{x_{k+n}}^T B_{(k,k+n-1)}^{-1} A_{x_{k+n}}]^{-1} A_{x_{k+n}}^T B_{(k,k+n-1)}^{-1} \\ &= [\mathbf{I} - \frac{B_{(k,k+n-1)}^{-1} A_{x_{k+n}} A_{x_{k+n}}^T}{1 + A_{x_{k+n}}^T B_{(k,k+n-1)}^{-1} A_{x_{k+n}}}] B_{(k,k+n-1)}^{-1} \end{aligned} \tag{9}$$

4. Deterministic Broad Learning System

From Equations (8) and (9), it can be obtained that data saturation linearity will occur as n tends to infinity. Therefore, when introducing the BLS into dynamic systems, the Deterministic Broad Learning System (DBLS) is proposed to solve the data saturation problem. The structure of the DBLS is shown in Figure 3.

$$\begin{aligned}
 W_{(k+1,k+n)} &= (A_{(k+1,k+n)}^T A_{(k+1,k+n)})^{-1} A_{(k+1,k+n)}^T Y_{(k+1,k+n)} \\
 &= B_{(k+1,k+n)}^{-1} \sum_{l=k+1}^{l=k+n} A_{x_l} Y_l \\
 &= B_{(k+1,k+n)}^{-1} [B_{(k,k+n)} W_{(k,k+n)} - A_{x_k} Y_{x_k}] \\
 &= W_{(k,k+n)} - B_{(k+1,k+n)}^{-1} A_{x_k} [Y_{x_k} - A_{x_k}^T W_{(k,k+n)}]
 \end{aligned}
 \tag{10}$$

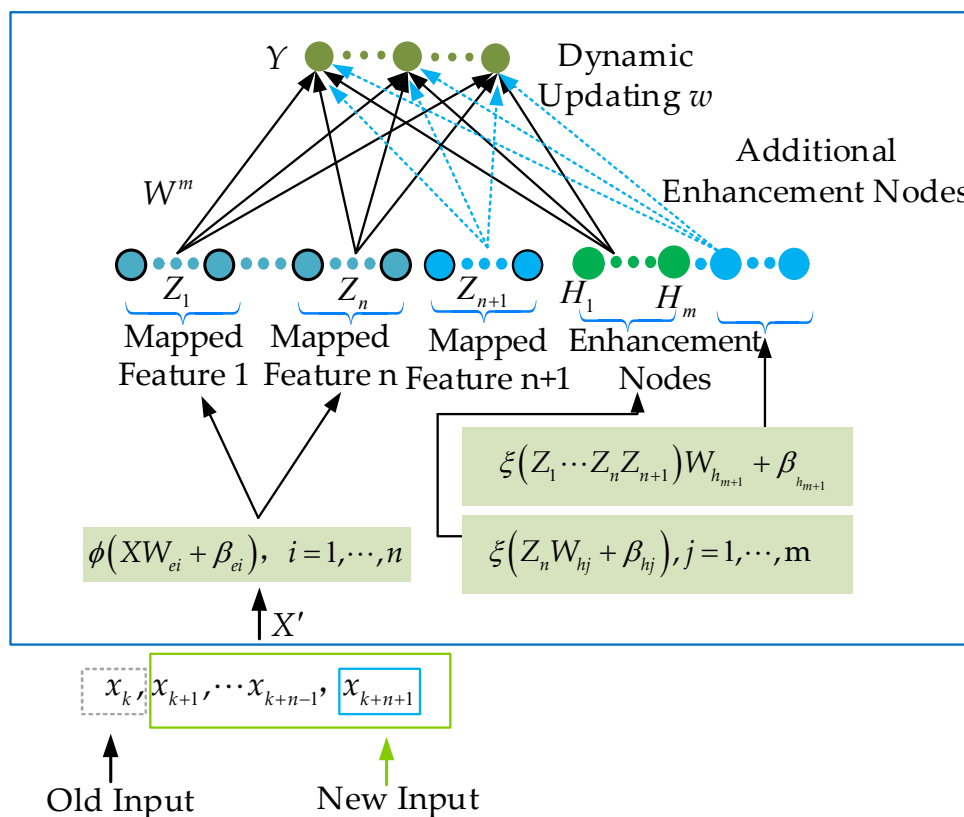


Figure 3. DBLS structure with input samples.

The DBLS uses a fixed input dimension, i.e., for each new set of data, an old set of data is removed. For example, let the input dimension of F be fixed as n . A new set of data (x_{k+n}, Y_{k+n}) is added, and a set of old data (x_k, Y_k) is removed to maintain the fixed dimension. Here, the weight of F can be calculated via Equations (10) and (11).

$$\begin{aligned}
 B_{(k+1,k+n)}^{-1} &= B_{(k,k+n)}^{-1} + B_{(k,k+n)}^{-1} A_{x_k} [I - A_{x_k}^T B_{(k,k+n)}^{-1} A_{x_k}]^{-1} A_{x_k}^T B_{(k,k+n)}^{-1} \\
 &= [I + \frac{B_{(k,k+n)}^{-1} A_{x_k} A_{x_k}^T}{1 - A_{x_k}^T B_{(k,k+n)}^{-1} A_{x_k}}] B_{(k,k+n)}^{-1}
 \end{aligned}
 \tag{11}$$

The DBLS is effective in overcoming data saturation, as demonstrated below. The relevant proof is made using the counterfactual method.

(1) The difference equation for the error vector $\tilde{W}_{(k+1,k+n+1)} = W_0 - W_{(k+1,k+n+1)}$ with respect to the DBLS output weights is constructed.

We perform a difference operation on both ends of Equation (10) with W_0 , and we derive

$$\begin{aligned}
 W_0 - W_{(k+1,k+n)} &= W_0 - W_{(k,k+n)} + B_{(k+1,k+n)}^{-1} A_{x_k} [Y_{x_k} - A_{x_k}^T W_{(k,k+n)}] \\
 \tilde{W}_{(k+1,k+n)} &= \tilde{W}_{(k,k+n)} + B_{(k+1,k+n)}^{-1} A_{x_k} [Y_{x_k} - A_{x_k}^T W_{(k,k+n)}] \\
 &= \tilde{W}_{(k,k+n)} + B_{(k+1,k+n)}^{-1} A_{x_k} [A_{x_k}^T W_0 + \varepsilon(k) - A_{x_k}^T W_{(k,k+n)}] \\
 &= [I + B_{(k+1,k+n)}^{-1} A_{x_k} A_{x_k}^T] \tilde{W}_{(k,k+n)} + B_{(k+1,k+n)}^{-1} A_{x_k} \varepsilon(k)
 \end{aligned} \tag{12}$$

such that

$$\begin{aligned}
 L_{(k+1,k+n)} &= I + B_{(k+1,k+n)}^{-1} A_{x_k} A_{x_k}^T \\
 &= B_{(k+1,k+n)}^{-1} [B_{(k+1,k+n)} + A_{x_k} A_{x_k}^T] \\
 &= B_{(k+1,k+n)}^{-1} B_{(k,k+n)}
 \end{aligned} \tag{13}$$

The difference equation for $\tilde{W}_{(k+1,k+n)}$ can be derived via

$$\tilde{W}_{(k+1,k+n)} = L_{(k+1,k+n)} \tilde{W}_{(k,k+n)} + B_{(k+1,k+n)}^{-1} A_{x_k} \varepsilon(k) \tag{14}$$

(2) Stability analysis of Equation (14)

Let the eigenvalue of $L_{(k+1,k+n)}$ be λ and the corresponding eigenvector be x_1 . We have that

$$L_{(k+1,k+n)} x_1 = \lambda x_1 \tag{15}$$

Taking Equation (15) into Equation (13), we have

$$\begin{aligned}
 L_{(k+1,k+n)} x_1 &= B_{(k+1,k+n)}^{-1} B_{(k,k+n)} x_1 \\
 &= [B_{(k,k+n)} - A_{x_k} A_{x_k}^T]^{-1} B_{(k,k+n)} x_1 \\
 &= \lambda x_1
 \end{aligned} \tag{16}$$

We further transform Equation (16) to obtain

$$(1 - \lambda) B_{(k,k+n)} x_1 = -\lambda A_{x_k} A_{x_k}^T x_1 \tag{17}$$

Since both $B_{(k,k+n)}$ and $A_{x_k} A_{x_k}^T$ are positive definite matrices, it can be deduced from Equation (17) that

$$(1 - \lambda) \lambda < 0 \tag{18}$$

We can obtain by solving Equation (18) that $\lambda < 0$ or $\lambda > 1$.

Equation (14) is unstable, so we can obtain

$$\lim_{k \rightarrow \infty} \tilde{W}_{(k+1,k+n)} \neq 0 \tag{19}$$

Clearly contradicting the previous assumptions, the DBLS does not suffer from data update saturation and is able to converge stably. The DBLS algorithm is presented in Algorithm 1. The DBLS algorithm flowchart is shown in Figure 4.

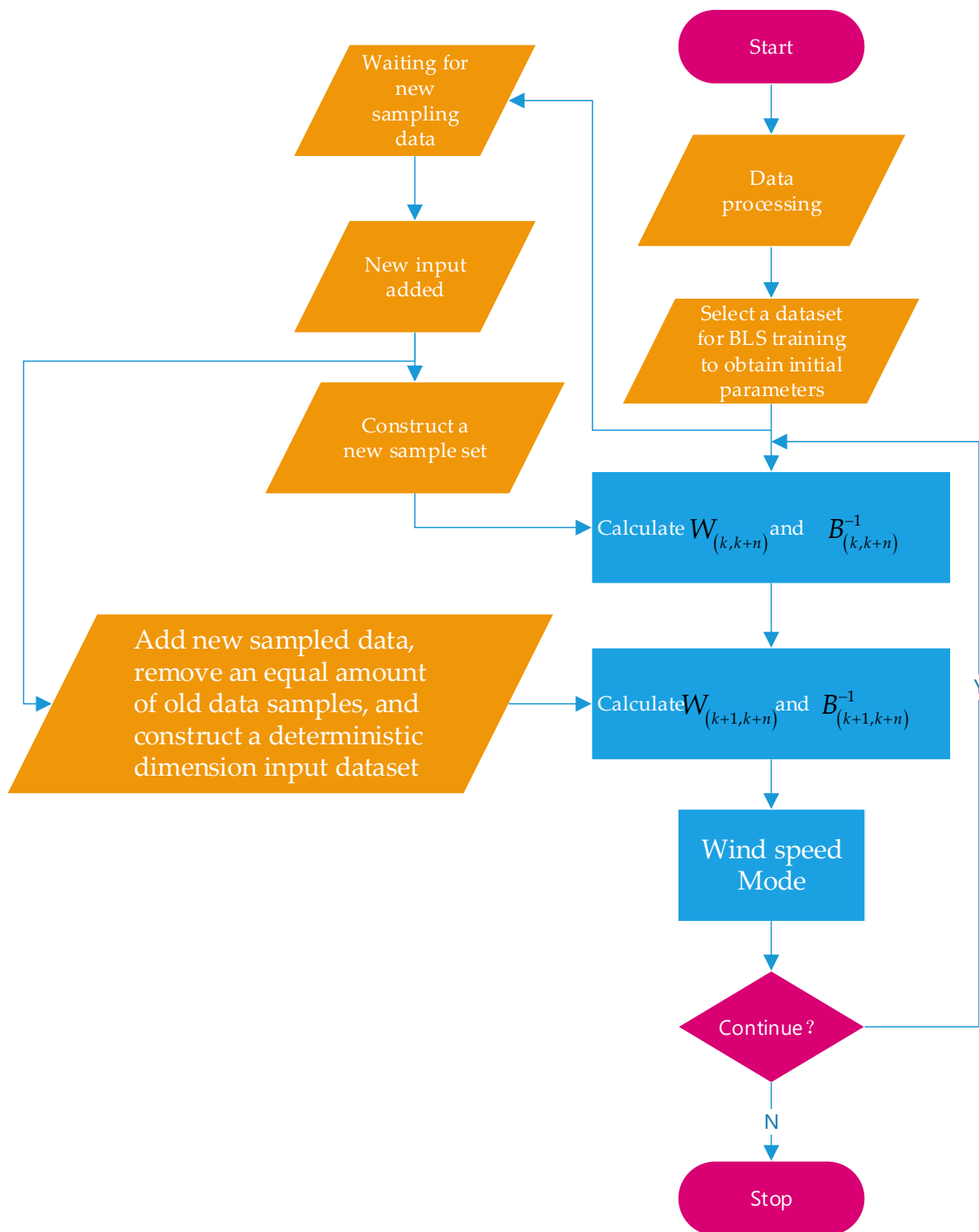


Figure 4. DBLS algorithm flowchart.

Algorithm 1 Deterministic Broad Learning System

Input: training sample X ;

Output: W

1. For $i = 0, i \leq n$, do
2. Random W_{ei}, β_{ei} ;
3. $Z_i = \phi_i(XW_{ei} + \beta_{ei}), i = 1, 2, \dots, n$;
4. end
5. Let $Z^n = [Z_1, \dots, Z_n]$;
6. For $j = 0, j \leq m$ do;
7. Random W_{hj}, β_{hj} ;
8. $H_j = \xi_j(XW_{hj} + \beta_{hj}), j = 1, 2, \dots, m$;
9. end
10. Let $H^m = [H_1, \dots, H_m]$;
11. Calculate $W_{(k,k+n-1)}$ with Equation (3);
12. while the training is doing do
13. New inputs are added as x_{k+n} ;
14. Calculate $B_{(k,k+n)}^{-1}$ with Equation (9);
15. Calculate $W_{(k,k+n)}$ by (8);
16. The original input x_k is removed;
17. Calculate $B_{(k+1,k+n)}^{-1}$ with Equation (11);
18. Calculate $W_{(k+1,k+n)}$ by Equation (10);
19. end

Set $W_{(k+1,k+n)}$

5. Validation and Analysis

The dataset was derived from ten wind farms located in Gansu Province, China. The dataset includes information such as time, wind speed, temperature, air pressure, and humidity at 70 m. It encompasses data from the year 2021. The training and testing sample information used during the experimental analysis is presented in Table 1. Table 1 provides the basic information of the dataset, where the four features related to wind speed are time, temperature, pressure, and humidity. Wind speed is the label in the dataset; therefore, the dataset encompasses a total of five categories, including time, temperature, pressure, humidity, and wind speed.

Table 1. The dataset information.

Dataset	Training Samples	Testing Samples	Features	Classifications
Wind farm 1	28,790	6240	4	5
Wind farm 2	28,800	6200	4	5
Wind farm 3	28,700	6300	4	5
Wind farm 4	28,650	6400	4	5
Wind farm 5	28,880	6100	4	5
Wind farm 6	28,780	6250	4	5
Wind farm 7	28,790	6250	4	5
Wind farm 8	28,700	6240	4	5
Wind farm 9	28,800	6200	4	5
Wind farm 10	28,850	6300	4	5

Before conducting the experiments, the dataset was preprocessed. The DBLS model was applied to the dataset consisting of 10 wind farms. Additionally, comparative experiments were conducted between the DBLS and other algorithms such as RF, SVR, ELM, and BLS. During the experiments, the parameters of the DBLS model were set as follows:

1. The initial values $B_{(k,k+n-1)}^{-1}$ and $W_{(k,k+n-1)}^{-1}$ of the DBLS algorithm were derived offline using batch BLS.
2. Number of feature windows (NnmWin): Selected from $\{5, 6, \dots, 20\}$, and in this experiment, NnmWin was set to 7.
3. Number of feature nodes (NumFea): Selected from $\{5, 6, \dots, 30\}$, and in this experiment, NumFea was set to 10.
4. Number of enhancement nodes (NumEnhan): Selected from $\{100, 200, \dots, 1000\}$, and in this experiment, NumEnhan was set to 300.
5. Scaling factor (s) for enhancement nodes: Set to 0.8 in this experiment.
6. Regularization parameter (C): Set to 2^{-30} in this experiment.

The experiment utilized the data from the first 25 days of each month as the training set, while the remaining days of the month were used as the testing set. When using the DBLS for wind speed prediction, the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are commonly adopted as evaluation metrics for assessing prediction performance. The formulas for calculating RMSE and MAPE are as follows:

$$RMSE = \sqrt{\frac{1}{l} \sum_{i=1}^l (y_i - \hat{y}_i)^2} \quad (20)$$

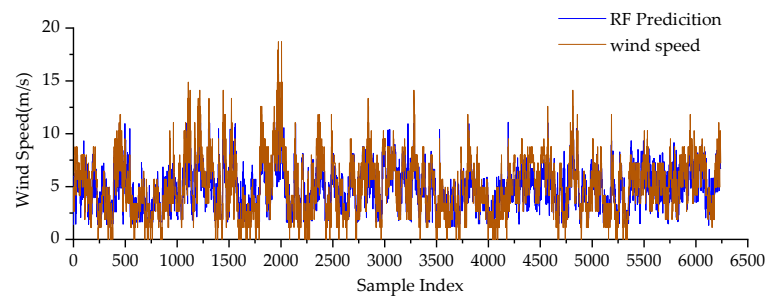
$$MAPE = \frac{1}{l} \sum_{i=1}^l \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (21)$$

where l represents the number of samples, y_i denotes the actual observed values, and \hat{y}_i represents the predicted values.

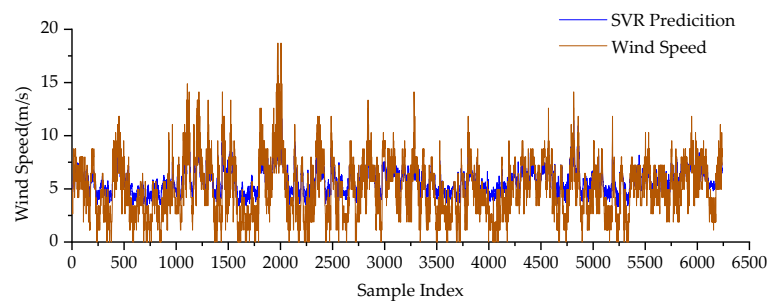
While conducting experiments on the wind speed of the established wind farm dataset, the DBLS model was configured with the following parameters: the feature layer window size was set to 7, each window had 10 nodes, the enhancement nodes were set to 300, and the regularization parameter ranged from 2^{-30} . In order to compare the predictive performance of the models, taking the wind speed dataset at 70 m as an example, the DBLS model was compared with the BLS, RF, SVR, and ELM models through experiments. Table 2 presents the parameter settings for the comparative experiments with different models. Figure 5 presents the prediction results of different algorithms.

Table 2. Parameter settings for different model experiments.

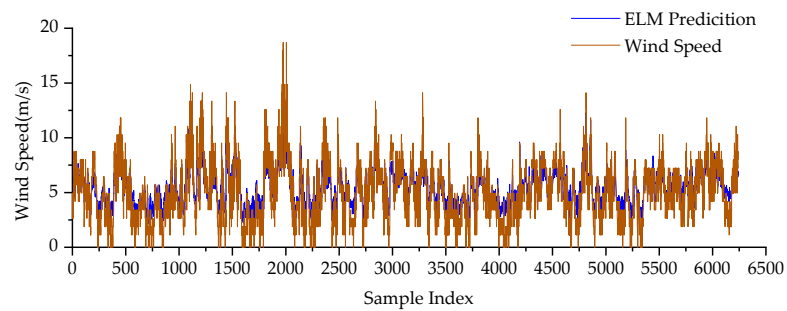
Dataset	RF			SVR		ELM		BLS			DBLS				
	Estimators	Min Samples Split	Max Depth	C	γ	C	Hidden Num	Num Win	Num Fea	Num Enhan	C	Num Win	Num Fea	Num Enhan	C
Wind farm 1	100	200	20	10	0.01	10	300	7	8	800	2^{-30}	6	10	400	2^{-30}
Wind farm 2	200	300	20	1	0.1	1	200	6	10	900	2^{-30}	8	12	500	2^{-30}
Wind farm 3	100	400	30	1	0.01	10	300	5	13	600	2^{-30}	7	10	300	2^{-30}
Wind farm 4	150	300	40	10	0.1	1	200	10	15	500	2^{-30}	9	12	500	2^{-30}
Wind farm 5	120	200	40	1	0.01	10	200	8	8	700	2^{-30}	10	10	600	2^{-30}
Wind farm 6	140	300	50	10	0.1	10	300	6	7	400	2^{-30}	6	9	400	2^{-30}
Wind farm 7	150	400	40	1	0.1	1	200	7	10	500	2^{-30}	5	8	500	2^{-30}
Wind farm 8	200	200	20	10	0.01	10	200	10	16	700	2^{-30}	6	9	600	2^{-30}
Wind farm 9	150	200	20	10	1	10	300	9	14	400	2^{-30}	7	9	400	2^{-30}
Wind farm 10	200	300	30	1	0.1	1	300	10	15	700	2^{-30}	7	10	300	2^{-30}



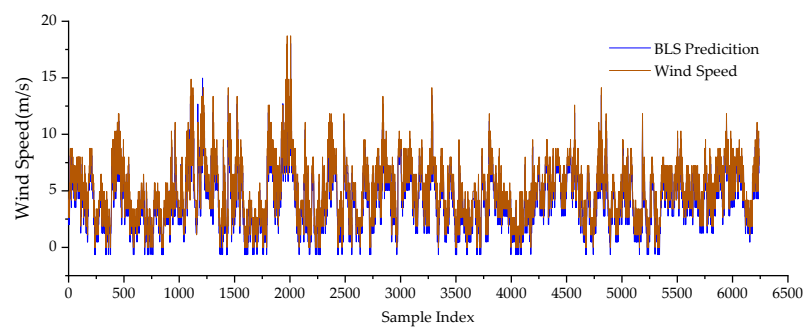
(a)



(b)



(c)



(d)

Figure 5. Cont.

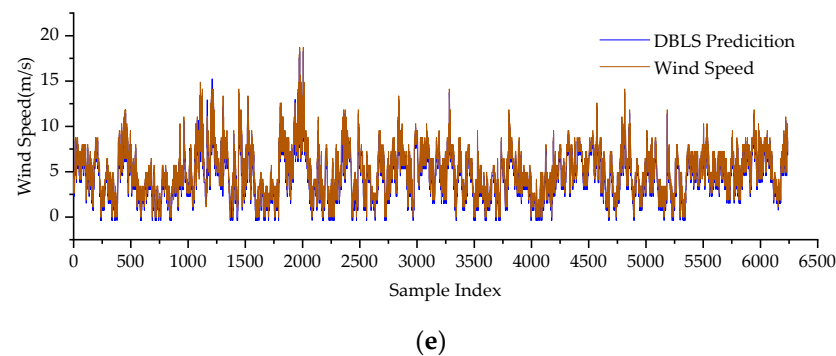


Figure 5. The prediction results of different algorithms: (a) Comparison graph between the wind speed model at 30 m based on the RF algorithm and the original wind speed. (b) Comparison graph between the wind speed model at 70 m based on the SVR algorithm and the original wind speed. (c) Comparison graph between the wind speed model at 70 m based on the ELM algorithm and the original wind speed. (d) Comparison graph between the wind speed model at 70 m based on the BLS algorithm and the original wind speed. (e) Comparison graph between the wind speed model at 70 m based on the DBLS algorithm and the original wind speed.

In Table 2, the RF model includes the following parameters: estimators represent the number of trees, `min_samples_split` represents the minimum number of samples required to split a node, and `max_depth` represents the maximum depth of the trees. In the SVR model, `C` represents the penalty factor for error terms, and γ represents the coefficient of the kernel function. For the ELM model, `C` represents the regularization coefficient and `Hidden Num` represents the number of hidden layers. In the BLS and DBLS models, `Num Win` represents the number of windows in the feature layer, `Num Fea` represents the number of nodes in each window, `Num Enhan` represents the number of enhancement nodes, and `C` represents the regularization parameter. After setting the parameters, the DBLS model was validated using the aforementioned dataset. The wind speed was used as the target variable in this experiment, and the remaining features were used as input data. The predictive metrics for different models on various datasets are presented in Table 3.

Table 3. RMSE and MAPE for different models.

Dataset	RF		SVR		ELM		BLS		DBLS	
	RMSE m/s	MAPE %	RMSE m/s	MAPE %	RMSE m/s	MAPE %	RMSE m/s	MAPE %	RMSE m/s	MAPE %
Wind farm 1	1.562	0.429	1.242	0.403	0.956	0.390	0.805	0.170	0.765	0.140
Wind farm 2	1.559	0.426	1.230	0.402	0.955	0.389	0.808	0.171	0.767	0.142
Wind farm 3	1.575	0.424	1.248	0.405	0.958	0.392	0.811	0.173	0.768	0.142
Wind farm 4	1.572	0.420	1.249	0.406	0.957	0.393	0.815	0.175	0.766	0.140
Wind farm 5	1.580	0.427	1.242	0.401	0.952	0.390	0.817	0.176	0.762	0.138
Wind farm 6	1.574	0.423	1.243	0.402	0.955	0.391	0.819	0.179	0.765	0.140
Wind farm 7	1.582	0.430	1.246	0.408	0.961	0.392	0.821	0.181	0.769	0.143
Wind farm 8	1.585	0.432	1.249	0.410	0.965	0.395	0.825	0.184	0.772	0.146
Wind farm 9	1.587	0.435	1.251	0.413	0.968	0.397	0.827	0.186	0.776	0.149
Wind farm 10	1.583	0.431	1.259	0.416	0.969	0.398	0.828	0.189	0.771	0.145

Based on the data in Table 3, it can be observed that the DBLS performs better than the other models in predicting wind speed. In the application experiments involving 10 wind farms, a comparison between the DBLS and BLS algorithms reveals a significant improvement in the learning accuracy of the DBLS, validating the findings in Figure 4. Under the condition of continuous system updates and dynamic prediction, the weight values of the BLS algorithm cannot be promptly adjusted, resulting in a weakened ability to capture new features. However, the DBLS algorithm overcomes the issue of data saturation

during continuous data updates by employing a fixed-width input and removing an equal amount of old data as new data are added. Therefore, in the context of nonlinear dynamic modeling and when system characteristics change, the weight values of the DBLS can be quickly adjusted to rapidly adapt to the system characteristics, leading to better prediction accuracy.

6. Discussion

This study investigated a predictive model for wind speed in wind farms. Wind speed in wind farms is a complex nonlinear system with multiple coupled parameters. Therefore, establishing a wind speed prediction model using intelligent algorithms has become an important research direction. The challenge of wind speed prediction models lies in whether they can maintain prediction accuracy and generalization stability under dynamically changing operational conditions.

Taking advantage of the online Broad Learning System, the DBLS algorithm is proposed from the perspective of dynamically updating weights. This algorithm addresses the stability issue of prediction accuracy and the generalization characteristics of wind speed prediction models under continuously changing operational conditions. Applying the DBLS algorithm to 10 wind farm datasets, the RMSE errors range from 0.762 m/s to 0.776 m/s, and the MAPE errors range from 0.138 to 0.149. Experimental results indicate high prediction accuracy and strong generalization ability.

A comparative analysis experiment was conducted by comparing the DBLS algorithm with the RF, SVR, ELM, and BLS algorithms. From the experimental results, it can be observed that the DBLS algorithm exhibits more prominent performance. In future research, the DBLS algorithm will be introduced into time series prediction applications under similar operational conditions to expand its application scope. Additionally, further research will delve into optimizing the optimal network structure of the DBLS algorithm to enhance its performance. Wind speeds in wind farms typically constitute complex nonlinear systems with multiple interdependent elements. Wind speeds within wind farms often manifest as nonstationary and nonuniform wind fields. References [40–42] thoroughly discuss and analyze nonstationary and nonuniform wind fields, providing directions and insights for our subsequent research. In the upcoming phases of our study, we will delve deeper into these aspects.

7. Conclusions

The working mechanism of an online BLS was analyzed, and based on that, the DBLS algorithm was proposed. The theoretical derivation process and convergence proof of the DBLS algorithm were provided. The DBLS algorithm effectively solves the issue of data saturation that occurs while continuously updating samples in nonlinear system modeling. The initial weight values of the algorithm are obtained through batch sample training. Subsequently, the algorithm uses fixed-width input, which means that when new samples are inputted, an equal amount of old sample data are removed. Therefore, the algorithm can better approximate time-varying nonlinear systems. Additionally, a dataset of 10 wind farms in Gansu Province, China, was established. Finally, a comparative experiment was conducted on the constructed dataset, comparing the performance of the DBLS with the BLS, RF, SVR, and ELM algorithms. The experimental results showed that the approximation performance of the DBLS algorithm was superior to that of other algorithms, further validating the reliability of the corresponding theoretical analysis.

The DBLS performs well in wind speed prediction for wind farms. Additionally, it is noted that the number of nodes in the DBLS network is currently determined through multiple experimental trials to achieve relative optimization. Therefore, how to obtain the optimal network structure for DBLS will be a focal point of future research. Currently, our research team is exploring the application of the DBLS in other temporal domains, and more relevant research outcomes will be provided in the future.

Author Contributions: Conceptualization, L.W. and A.X.; methodology, L.W. and A.X.; software, L.W.; validation, L.W.; formal analysis, L.W.; investigation, L.W. and A.X.; resources, L.W.; data curation, L.W.; writing—original draft preparation, L.W.; writing—review and editing, L.W. and A.X.; supervision, L.W.; project administration, L.W.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets of the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations and Symbols

DBLS	Deterministic Broad Learning System
BLS	Broad Learning System
RF	Random Forest
SVR	Support Vector Regression
ELM	Extreme Learning Machines
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
ARMA	Auto Regression Moving Average
ARIMA	Auto Regressive Integrated Moving Average
SARIMA	Seasonal Auto Regressive Integrated Moving Average
STL	Seasonal Trend Loss
GARCH	Generalized Auto Regressive Conditional Heteroskedasticity
ANNs	Artificial Neural Networks
LSTM	Long Short-Term Memory
CNNs	Convolutional Neural Networks
SVR	Support Vector Regression
SLFNs	Single-Layer Feedforward Neural Networks
RVFLNN	Random Vector Functional Link Neural Network
Z_i	the i th feature window
ϕ_i	the linear mapping function
W_{ei}	the weights of the ϕ_i function
β_{ei}	the biases of the ϕ_i function
ζ_j	the mapping function of the enhancement node
W_{hj}	the weights of the ζ_j function
β_{hj}	the biases of the ζ_j function
W^m	the output function weight
X	the input sample
$A_{(k,k+n-1)}$	the intermediate layer of the BLS
$A_{x_{k+n}}^T$	the middle layer addition variable corresponding to the added input sample x_{k+n}
$B_{(k,k+n-1)}$	$B_{(k,k+n-1)} = A_{(k,k+n-1)}^T A_{(k,k+n-1)}$
$B_{(k,k+n)}^{-1}$	the inverse of $B_{(k,k+n)}$
$B_{(k+1,k+n)}^{-1}$	the inverse of $B_{(k+1,k+n)}$
$\hat{W}_{(k+1,k+n+1)}$	the difference equation

References

1. Tao, T.; Shi, P.; Wang, H.; Ai, W. Short-term prediction of downburst winds: A double-step modification enhanced approach. *J. Wind Eng. Ind. Aerodyn.* **2021**, *211*, 104561. [[CrossRef](#)]
2. Dolatabadi, A.; Abdeltawab, H.; Mohamed, Y.A.R.I. Deep spatial-temporal 2-D CNN-BLSTM model for ultrashort-term LiDAR-assisted wind turbine's power and fatigue load forecasting. *IEEE Trans. Ind. Inform.* **2021**, *18*, 2342–2353. [[CrossRef](#)]
3. Ran, X.; Xu, C.; Ma, L.; Xue, F. Wind Power Interval Prediction with Adaptive Rolling Error Correction Based on PSR-BLS-QR. *Energies* **2022**, *15*, 4137. [[CrossRef](#)]
4. Wang, L.; Wang, Y.; Chen, J.; Shen, X. A PM2.5 Concentration Prediction Model Based on CART-BLS. *Atmosphere* **2022**, *13*, 1674. [[CrossRef](#)]
5. Jiang, Y.; Liu, S.; Zhao, N.; Xin, J.; Wu, B. Short-term wind speed prediction using time varying filter-based empirical mode decomposition and group method of data handling-based hybrid model. *Energy Convers. Manag.* **2020**, *220*, 113076. [[CrossRef](#)]
6. Han, L.; Li, M.J.; Wang, X.J.; Lu, P.P. Wind power forecast based on broad learning system and simplified long short term memory network. *IET Renew. Power Gener.* **2022**, *16*, 3614–3628. [[CrossRef](#)]
7. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 703–716.
8. Kim, H.S.; Han, K.M.; Yu, J.; Kim, J.; Kim, K.; Kim, H. Development of a CNN+LSTM Hybrid Neural Network for Daily PM2.5 Prediction. *Atmosphere* **2022**, *13*, 2124. [[CrossRef](#)]
9. Löwe, S.; Madras, D.; Zemel, R.; Welling, M. Amortized causal discovery: Learning to infer causal graphs from time-series data. In Proceedings of the Conference on Causal Learning and Reasoning, Eureka, CA, USA, 11–13 April 2022; pp. 509–525.
10. Pappas, S.S.; Ekonomou, L.; Karamousantas, D.C.; Chatzarakis, G.E.; Katsikas, S.K.; Liatsis, P. Electricity demand loads modeling using Autoregressive Moving Average (ARMA) models. *Energy* **2008**, *33*, 1353–1360. [[CrossRef](#)]
11. Nelson, B.K. Time series analysis using autoregressive integrated moving average (ARIMA) models. *Acad. Emerg. Med.* **1998**, *5*, 739–744. [[CrossRef](#)]
12. Adachi, Y.; Makita, K. Method of Time Series Analysis of Meat Inspection Data Using Seasonal Autoregressive Integrated Moving Average Model. *J. Jpn. Vet. Med. Assoc.* **2015**, *68*, 189–197. [[CrossRef](#)]
13. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A seasonal-trend decomposition. *J. Off. Stat.* **1990**, *6*, 3–73.
14. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [[CrossRef](#)]
15. Agatonovic-Kustrin, S.; Beresford, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. Pharm. Biomed. Anal.* **2000**, *22*, 717–727. [[CrossRef](#)] [[PubMed](#)]
16. Graves, A.; Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*; Long short-term memory; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
17. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [[CrossRef](#)]
18. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
19. Awad, M.; Khanna, R.; Awad, M.; Khanna, R. Support vector regression: Theories, concepts, and applications for engineers and system designers. In *Efficient Learning Machines*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 67–80.
20. Ding, S.; Zhang, Z.; Guo, L.; Sun, Y. An optimized twin support vector regression algorithm enhanced by ensemble empirical mode decomposition and gated recurrent unit. *Inf. Sci.* **2022**, *598*, 101–125. [[CrossRef](#)]
21. Biau, G.; Scornet, E. A random forest guided tour. *Test* **2016**, *25*, 197–227. [[CrossRef](#)]
22. Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [[CrossRef](#)]
23. Rigatti, S.J. Random forest. *J. Insur. Med.* **2017**, *47*, 31–39. [[CrossRef](#)]
24. Yoo, H.; Kim, E.; Chung, J.W.; Cho, H.; Jeong, S.; Kim, H.; Jang, D.; Kim, H.; Yoon, J.; Lee, G.H.; et al. Silent Speech Recognition with Strain Sensors and Deep Learning Analysis of Directional Facial Muscle Movement. *ACS Appl. Mater. Interfaces* **2022**, *14*, 54157–54169. [[CrossRef](#)]
25. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [[CrossRef](#)]
26. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [[CrossRef](#)]
27. Mueller, C.M.; Schatz, G.C. An Algorithmic Approach Based on Data Trees and Genetic Algorithms to Understanding Charged and Neutral Metal Nanocluster Growth. *J. Phys. Chem. A* **2022**, *126*, 5864–5872. [[CrossRef](#)] [[PubMed](#)]
28. Pao, Y.-H.; Takefuji, Y. Functional-link net computing: Theory, system architecture, and functionalities. *Computer* **1992**, *25*, 76–79. [[CrossRef](#)]
29. Pao, Y.-H.; Park, G.-H.; Sobajic, D.J. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* **1994**, *6*, 163–180. [[CrossRef](#)]
30. Liu, Z.; Zhou, J.; Chen, C.L.P. Broad learning system: Feature extraction based on K-means clustering algorithm. In Proceedings of the International Conference on Information, Cybernetics and Computational Social Systems (ICCS), Dalian, China, 24–26 July 2017; pp. 683–687.

31. Chen, C.L.P.; Liu, Z. Broad Learning System: An Effective and Efficient Incremental Learning System without the Need for Deep Architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 10–24. [[CrossRef](#)]
32. Igel'nik, B.; Pao, Y.H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans. Neural Netw.* **1995**, *6*, 1320–1329. [[CrossRef](#)]
33. Cheng, Y.; Le, H.; Li, C.; Huang, J.; Liu, P.X. A Decomposition-Based Improved Broad Learning System Model for Short-Term Load Forecasting. *J. Electr. Eng. Technol.* **2022**, *17*, 2703–2716. [[CrossRef](#)]
34. Wang, M.; Ge, Q.; Li, C.; Sun, C. Charging diagnosis of power battery based on adaptive STCKF and BLS for electric vehicles. *IEEE Trans. Veh. Technol.* **2022**, *71*, 8251–8265. [[CrossRef](#)]
35. Pu, X.; Li, C. Online semisupervised broad learning system for industrial fault diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6644–6654. [[CrossRef](#)]
36. Chu, F.; Liang, T.; Chen, C.L.P.; Wang, X.; Ma, X. Weighted broad learning system and its application in nonlinear industrial process modeling. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3017–3031. [[CrossRef](#)]
37. Chen, C.L.P.; Liu, Z.; Feng, S. Universal approximation capability of broad learning system and its structural variations. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 1191–1204. [[CrossRef](#)] [[PubMed](#)]
38. Wang, S.; Yuen, K.-V.; Yang, X.; Zhang, Y. Tropical Cyclogenesis Detection from Remotely Sensed Sea Surface Winds Using Graphical and Statistical Features-based Broad Learning System. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4203815. [[CrossRef](#)]
39. Jiao, X.; Zhang, D.; Song, D.; Mu, D.; Tian, Y.; Wu, H. Wind Speed Prediction Based on VMD-BLS and Error Compensation. *J. Mar. Sci. Eng.* **2023**, *11*, 1082. [[CrossRef](#)]
40. Tao, T.; He, J.; Wang, H.; Zhao, K. Efficient simulation of non-stationary non-homogeneous wind field: Fusion of multi-dimensional interpolation and NUFFT. *J. Wind Eng. Ind. Aerodyn.* **2023**, *236*, 105394. [[CrossRef](#)]
41. Xu, Y.-L.; Hu, L.; Kareem, A. Conditional Simulation of Nonstationary Fluctuating Wind Speeds for Long-Span Bridges. *J. Eng. Mech.* **2014**, *140*, 61–73. [[CrossRef](#)]
42. Tao, T.; Wang, H.; Zhao, K. Efficient simulation of fully non-stationary random wind field based on reduced 2D hermite interpolation. *Mech. Syst. Signal Process.* **2021**, *150*, 107265. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.