

Article

A Novel Hybrid Predictive Model for Ultra-Short-Term Wind Speed Prediction

Longnv Huang^{1,†}, Qingyuan Wang^{1,†}, Jiehui Huang^{1,†}, Limin Chen^{1,*}, Yin Liang¹, Peter X. Liu^{1,2}
and Chunquan Li¹

¹ School of Information Engineering, Nanchang University, Nanchang 330031, China; 6105119006@email.ncu.edu.cn (L.H.); wqy1476990135@163.com (Q.W.); jiehuihuang1107@163.com (J.H.); liangyin@ncu.edu.cn (Y.L.); xpliu@sce.carleton.ca (P.X.L.); lichunquan@ncu.edu.cn (C.L.)

² Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

* Correspondence: chenlimin@ncu.edu.cn

† These authors contributed equally to this work.

Abstract: A novel hybrid model is proposed to improve the accuracy of ultra-short-term wind speed prediction by combining the improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN), the sample entropy (SE), optimized recurrent broad learning system (ORBLS), and broadened temporal convolutional network (BTCN). First, ICEEMDAN is introduced to smooth the nonlinear part of the wind speed data by decomposing the raw wind speed data into a series of sequences. Second, SE is applied to quantitatively assess the complexity of each sequence. All sequences are divided into simple sequence set and complex sequence set based on the values of SE. Third, based on the typical broad learning system (BLS), we propose ORBLS with cyclically connected enhancement nodes, which can better capture the dynamic characteristics of the wind. The improved particle swarm optimization (PSO) is used to optimize the hyper-parameters of ORBLS. Fourth, we propose BTCN by adding a dilated causal convolution layer in parallel to each residual block, which can effectively alleviate the local information loss of the temporal convolutional network (TCN) in case of insufficient time series data. Note that ORBLS and BTCN can effectively predict the simple and complex sequences, respectively. To validate the performance of the proposed model, we conducted three predictive experiments on four data sets. The experimental results show that our model obtains the best predictive results on all evaluation metrics, which fully demonstrates the accuracy and robustness of the proposed model.

Keywords: wind speed forecast; optimized recurrent BLS; broadened TCN; data preprocessing



Citation: Huang, L.; Wang, Q.; Huang, J.; Chen, L.; Liang, Y.; Liu, P.X.; Li, C. A Novel Hybrid Predictive Model for Ultra-Short-Term Wind Speed Prediction. *Energies* **2022**, *15*, 4895. <https://doi.org/10.3390/en15134895>

Academic Editors: Ioannis Panapakidis, Minas Alexiadis and Aggelos S. Bouhouras

Received: 9 May 2022

Accepted: 17 June 2022

Published: 4 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/>).

1. Introduction

With the increasing global energy problems, wind energy is becoming one of the renewable energy sources of interest. However, the intermittent and fluctuating nature of the wind can make the power supply from wind power systems unstable. For example, a 10% difference in wind speed can produce a deviation of around 30% in wind power generation [1], which significantly affects the utilization of wind energy. Accurate short-term wind energy forecasts are essential to improving the reliability of wind power systems [2]. Wind speed forecasting is an important component of wind energy forecasting. To improve the efficiency of power systems and increase the utilization of wind energy, it is necessary to develop accurate short-term WSP models.

Wind speed forecasts can be classified into ultra-short-term forecasts (few minutes to one hour ahead), short-term forecasts (one hour to several hours ahead), mid-term forecasts (several hours to one week ahead), and long-term forecasts (one week to one year ahead) based on the forecast scale [3]. Currently, existing WSP models can be classified into physical models, statistical models, AI-based models, and hybrid models.

Physical models combine many physical factors such as terrain information, density, and temperature for WSP. It requires not only rich meteorological data but also physical conditions of data and wind field locations [4]. For example, numerical weather prediction [5] is a typical physical model. Physical methods are effective and accurate for long-term forecasting, but they are computationally intensive and costly for short-term forecasting.

Statistical models are simple in structure and fast in computation compared to physical models. There are widely used linear models such as the autoregressive model [6], autoregressive moving average [7], ARIMA [8], and Bayesian model [9]. Statistical models have better prediction results for the linear part of the wind speed series. However, the actual wind speed series usually exhibit prominent nonlinear and non-smooth properties, when statistical models may obtain undesirable prediction results [10].

AI-based models make good use of the ability of AI to capture the nonlinear features of the data. Wind speed prediction using BP neural networks [11], SVR [12], and fuzzy neural networks [13] have obtained good results. However, these models may have problems such as slow convergence rate and tend to fall into local minimization. Compared with shallow models, deep learning can accurately extract abstract features and potentially invariant architectures in the data. Therefore, to further improve the learning ability and prediction ability of prediction models, deep learning was introduced into WSP, such as ELM [14], deep belief network [15], CNN [16], graph neural network [17], LSTM neural networks [18], etc.

Hybrid models have become a trend in WSP in recent years. Hybrid models combine the different advantages of multiple techniques for prediction, which can carefully explore the features in wind speed data and effectively ensure prediction accuracy. Hybrid models can be broadly classified into the following four types:

- (1) Hybrid models based on signal decomposition. Advanced signal decomposition techniques are used to decompose wind sequences into a series of regular subsequences to smooth out the non-linear parts of the data. Wang et al. [11] proposed two WSP models, the EMD-based model and the EEMD-based model, and demonstrated experimentally that the EEMD-based model outperformed the EMD-based model. To solve the noise residuals problem of EEMD, Wang et al. [19] used CEEMD for WSP and obtained more satisfactory results. Ren et al. [20] experimentally demonstrated that the CEEMDAN-based model always performs best compared to the EMD-based model. Aside from EMD and its variants [3], wavelet transform [21,22] and VMD decomposition [23] are also common signal decomposition techniques in WSP.
- (2) Hybrid model based on weight assignment. This hybrid model typically uses multiple models to predict wind speed and assigns appropriate weights to each model. The final predicted values are obtained from a weighted combination of the predicted values of each model. To mitigate the adverse effects of multicollinearity of hybrid models, Jiang et al. [24] used GMDH to automatically identify the weights of three nonlinear models. The experimental results show that the application of GMDH can significantly improve the predictive power compared to the widely used equal-weighting scheme. Aytaç Altan et al. [25] used the gray wolf optimization algorithm to optimize the weighting of each IMF to create the optimal prediction model. Nie et al. [26] proposed a weight combination mechanism based on a multi-objective optimization algorithm, which further improved the prediction accuracy and predictive power of the model.
- (3) Hybrid model based on optimization algorithms. The model introduces some heuristic optimization algorithms to optimize the hyperparameters, weights, network structure, or thresholds of the model. Liu et al. [27] used Jaya to optimize the hyperparameters of SVM, which improved the SVM regression performance and effectively improved the prediction accuracy. Tian et al. [28] used PSO to optimize the weight coefficients of each prediction model, and the experimental results proved the necessity of the weight coefficient optimization strategy. Liu et al. [29] used GA to optimize the internal parameters of LSTM, thus improving the prediction efficiency and prediction

accuracy of the model. Huang et al. [30] used the modified Bayesian optimization algorithm to optimize the hyperparameters of the prediction model and obtained more satisfactory forecasting precision and computation cost. To obtain the best network structure and weights, Liu et al. [31] combined GA and PSO to optimize the CNN prediction model and effectively addressed the problem of poor prediction performance due to wind volatility.

- (4) Hybrid model based on error correction. Error correction is a post-processing technique for WSP, which predicts the residuals and superimposes the results on the original prediction to obtain the corrected final prediction. Duan et al. [32] used ICEEMDAN to decompose the errors, and the experimental results showed that the error decomposition correction method can significantly improve the prediction accuracy. Liu et al. [33] proposed an adaptive multiple error correction method, which makes full use of the deeper predictable components and effectively improves the reliability and accuracy of the model. Zhang et al. [34] demonstrated experimentally that the final predictions corrected by Markov chains are closer to the original wind field data, which proves the effectiveness of Markov chains.

Although the above WSP models have obtained promising prediction performance, they still have some problems that need to be further improved. For instance, the linear model [6–9] in statistical methods is difficult to extract the deep-level features of wind speed data well. AI-based models involving deep neural networks [15] cause huge computational costs. Hybrid methods based on weighting strategy [24] may have the problem of multicollinearity, which reduces the prediction accuracy. The performance of hybrid methods based on parameter optimization [27,28] is largely influenced by the understanding of the researcher of the optimization algorithm.

To deal with the above issues, we propose a new hybrid model for ultra-short-term WSP. First, ICEEMDAN decomposes the raw wind speed data into a series of sequences to smooth the nonlinear part of the wind speed data. Second, SE is applied to quantitatively assess the complexity of each sequence. All sequences are divided into simple sequence set and complex sequence set based on the values of SE. Third, the ORBLS is designed to predict the simple subsequence set, and the BTCN is designed to predict the complex subsequence set. This prediction relationship is based on the ability of the prediction network to match the sequence complexity, which was confirmed in the experimental results. Finally, the prediction values of all subsequences are summed up to obtain the final prediction results.

The contributions and innovations of this research are as follows:

- A novel hybrid model is proposed for ultra-short-term WSP, which quantitatively assesses the complexity of wind speed series by SE and builds different prediction models for different subseries with different complexity separately.
- The proposed ORBLS has cyclically connected enhancement nodes that can better capture the dynamic characteristics of the wind speed sequence, and the improved PSO is used to optimize the hyperparameters of ORBLS.
- The proposed BTCN adds a dilated causal convolution layer in parallel to each residual block, which alleviates the problem of massive local information loss from dilated causal convolution.
- The ICEEMDAN is introduced to smooth the nonlinear part of the wind speed data, which further improves the forecasting performance.
- Four wind speed datasets are used to comprehensively evaluate the effectiveness and robustness of the proposed model.

The rest of this article is organized as follows. The Section 2 introduces the model framework and methods involved in this article in detail. In the Section 3, the experimental cases and prediction results are elaborated in detail, which verifies the validity of the proposed model. The conclusions are presented in the Section 4.

2. Materials and Methods

2.1. Overall Framework of the Proposed Model

Since the wind is stochastic and volatile, the traditional single model does not predict well. A novel hybrid model is proposed to improve the accuracy of ultra-short-term wind speed prediction by combining ICEEMDAN, sample entropy (SE), ORBLS, and BTCN. Note that the proposed ORBLS has cyclically connected enhancement nodes compared to the typical BLS, and the improved PSO is used to optimize the hyper-parameters of ORBLS. Notably, in the proposed BTCN, a dilated causal convolution layer is added in parallel to each residual block to alleviate the local information loss.

The overall model framework diagram is shown in Figure 1. First, ICEEMDAN is introduced to decompose the wind speed series into multiple subseries to reduce the noise in the original wind sequence. Second, SE is applied to quantitatively assess the complexity of each sequence. All sequences are divided into simple sequence set and complex sequence set based on the values of SE. Third, the ORBLS is designed to predict the simple subsequence set, and the BTCN is designed to predict the complex subsequence set. This prediction relationship is based on the ability of the prediction network to match the sequence complexity, which was confirmed in the experimental results. Finally, the prediction values of all subsequences are summed up to obtain the final prediction results.

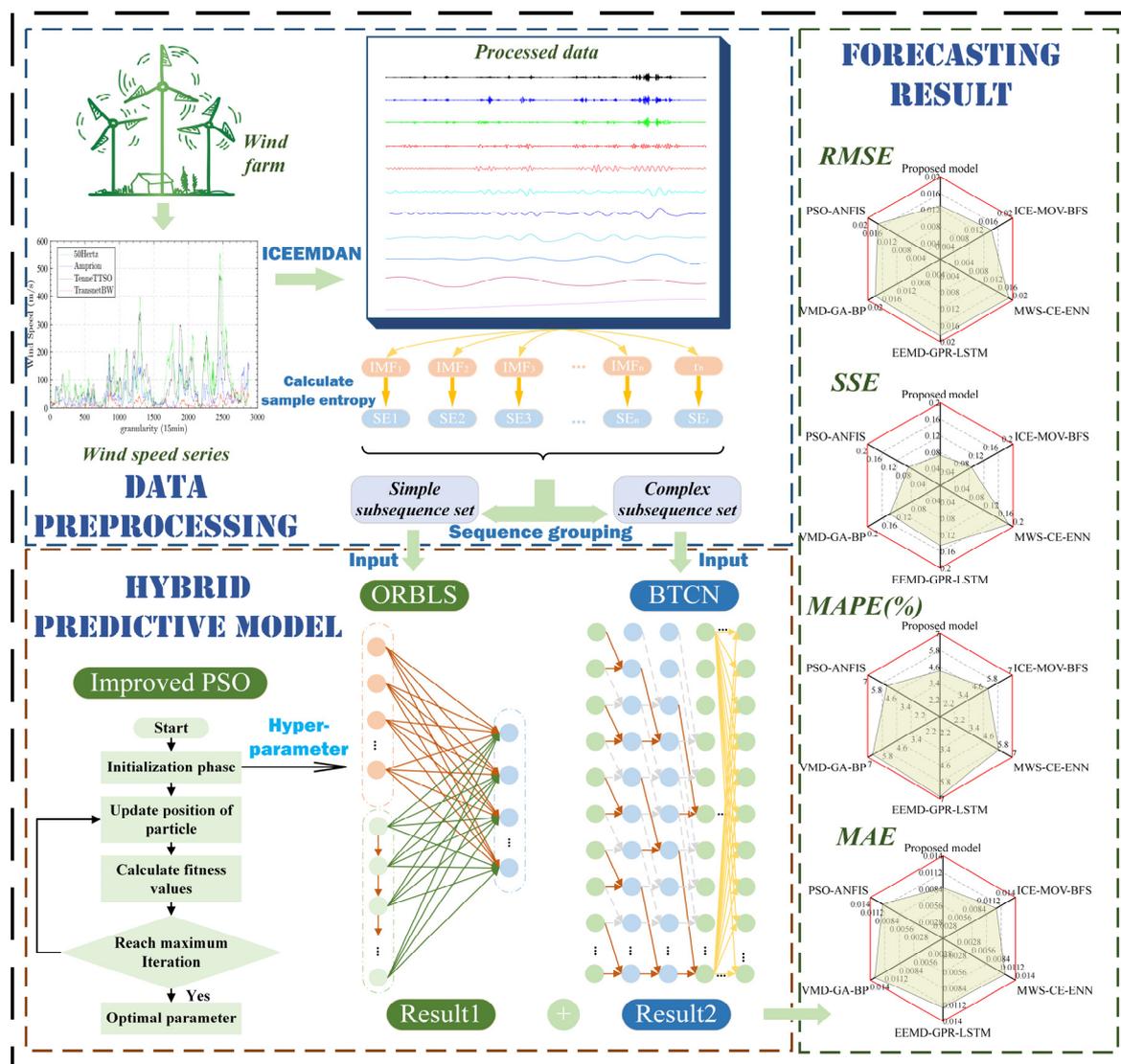


Figure 1. The proposed ultra-short-term wind speed forecasting framework.

2.2. ICEEMDAN Model

Due to the high volatility of the wind speed series, ICEEMDAN [35] is introduced to smooth the wind speed data. ICEEMDAN decomposes a signal into some intrinsic mode functions or modes.

The original wind speed series is defined as $X(n)$. Randomly generate white noises $w^i(n)$, $i = 1, 2, \dots, I$ with $\mu = 0$ and $\sigma^2 = 1$ where μ is the average value and σ^2 is the variance. Define an operator $E\{\ast\}$ which generates the IMFs by EMD and an operator $M\{\ast\}$ which denotes the local mean of the generated signal. Specific steps are as follows.

Step 1: Add $w^i(n)$ to the $X(n)$ and calculate by EMD the local means of I realizations $x^i(n) = X(n) + \beta_0 E\{w^i(n)\}$. The first-order residue is computed as follows:

$$r_1(n) = \left| M\{x^i(n)\} \right| \quad (1)$$

Step 2: Calculate the first modal component:

$$d_1(n) = X(n) - r_1(n) \quad (2)$$

Step 3: Add white noise $w^i(n)$ again and a second set of residuals $r_2(n) = r_1(n) + \beta_1 E\{w^i(n)\}$ is calculated by local mean decomposition. Define the second modal component:

$$d_2(n) = r_1(n) - \left| M\{r_1(n) + \beta_1 E\{w^i(n)\}\} \right| \quad (3)$$

Step 4: For $k = 3, 4, \dots, K$, compute the k th residual and modal components:

$$r_k(n) = \left| M\{r_{k-1}(n) - \beta_{k-1} E\{w^i(n)\}\} \right| \quad (4)$$

$$d_k(n) = r_{k-1}(n) - r_k(n) \quad (5)$$

Step 5: Go to step 4 for next k .

Throughout the implementation process, the coefficients $\beta_0 = \varepsilon_0 \text{std}(r_0(n)) / \text{std}(E\{w^i(n)\})$ and $\beta_k = \varepsilon_0 \text{std}(r_k(n))$, $j \geq 1$ are chosen to obtain a desired signal-to-noise ratio between the added noise and the residue, and ε_0 indicates the reciprocal of the desired signal-to-noise ratio between the k signals. The subsequence $IMF_k(n)$ obtained after decomposition is $d_k(n)$.

2.3. Sample Entropy

Sample entropy [36] is a quantitative description of the complexity of time series. The larger the SE value, the lower the autocorrelation of the series and the higher its complexity. The specific steps are as follows.

Step 1: Reconstruct the sequence $x(n) = \{x_1, x_2, x_3, \dots, x_N\}$ as an m -dimensional vector as follows:

$$X_m(j) = [X_{j+1}, X_{j+2}, X_{j+3}, \dots, X_{j+m-1}] \quad (6)$$

Step 2: For $1 \leq j \leq N - m + 1, j \neq k$, define the $D\langle X_m(j), X_m(k) \rangle$ as the biggest distance between $X_m(j)$, and $X_m(k)$ as follows:

$$D\langle X_m(j), X_m(k) \rangle = \max\{|X_{j+w} - X_{k+w}|\} \quad (7)$$

where $w = 0, 1, 2, \dots, m - 1$.

Step 3: Count the sum number of $D\langle X_m(j), X_m(k) \rangle < r$, for every j value and obtain $B_j^m(r)$ by calculating the ratio with $N - m + 1$. Calculate the mean for $B_j^m(r)$ to obtain $B^m(r)$.

$$B_j^m(r) = \frac{\text{num}\{D\langle X_m(j), X_m(k) \rangle < r\}}{N - m} \quad (8)$$

$$B^m(r) = \frac{\sum_{j=1}^{N-m+1} B_j^m(r)}{N - m + 1} \tag{9}$$

Step 4: Update the m as $m + 1$ and repeat steps 1–3 then get the mean value of $B^{m+1}(r)$. The sample entropy can be expressed as follows:

$$SE(m, r) = \lim_{n \rightarrow \infty} \left\{ -\ln \left(\frac{B^{m+1}(r)}{B^m(r)} \right) \right\} \tag{10}$$

When $N = \infty$, the estimated result can be obtained:

$$SE(m, r, N) = -\ln \left(\frac{B^{m+1}(r)}{B^m(r)} \right) \tag{11}$$

2.4. ORBLS Model

We designed ORBLS as one of the models to predict simple subsequence set. Compared to the conventional BLS [37], the proposed ORBLS has cyclically connected enhancement nodes, which can better capture the dynamic features of the time series. Moreover, the improved PSO in ORBLS is used to optimize the correlation weights. The following is the detailed process of the ORBLS algorithm.

Define the input wind speed series $X(n)$ and n feature mappings φ_i . The i th mapped feature Z_i is defined as follows:

$$Z_i = \varphi_i(X(n)W_{ei} + \beta_{ei}), i = 1, 2, \dots, n \tag{12}$$

where W_{ei} and β_{ei} represent random weights with the proper dimensions.

The feature nodes are denoted as $Z^n \triangleq [Z_1, Z_2 \dots Z_n]$, where the symbol \triangleq means “noted as”. We collect Z_i into Z^n , which is further input to the enhancement nodes H_j .

As shown in Figure 2, the enhancement nodes H_j are connected by a loop to form a cyclic structure, which processes the input sequence of one element at a time. Unlike the original BLS, the definition of H_j in ORBLS is as follows:

$$H_j = \psi_j \left(\partial_1 Z_i W_{hj} + \beta_{hj} + \partial_2 H_{j-1} W_{mj} \right), j = 1, 2, \dots, m \tag{13}$$

where W_{hj} , W_{mj} , and β_{hj} are random weights. When $j = 0$, $H_0 = 0$. The activation function ψ_j is typically chosen as the tansig function. ∂_1 and ∂_2 are parameters that balance the effect of the two components.

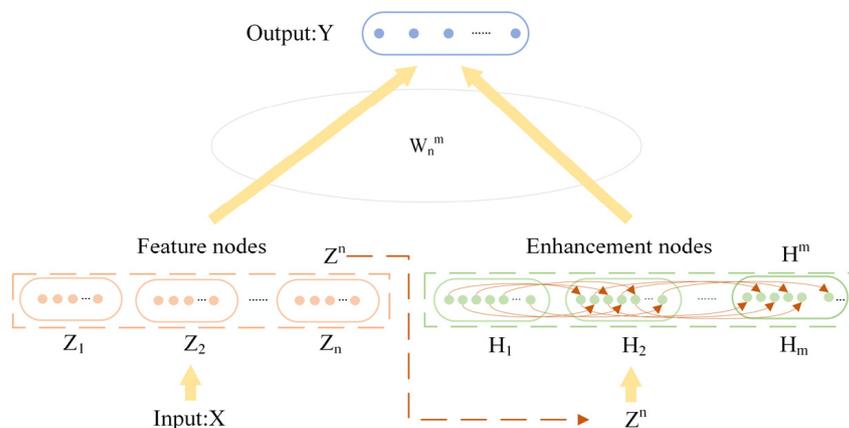


Figure 2. Structure of the ORBLS.

Let $H^m \triangleq [H_1, H_2 \dots H_m]$, then the output of the ORBLS can be denoted as:

$$Y = \{Z^n | H^m\} W_m^n \tag{14}$$

where the W_m^n is the final target weight needed by ORBLS and is obtained through the ridge regression algorithm, that is, $W_m^n \triangleq \{Z^n|H^m\}^+ Y$.

Let $\{M\} = \{Z^n|H^m\}$ then $\{Z^n|H^m\}^+$ can be expressed as follows:

$$\{Z^n|H^m\}^+ = \lim_{\lambda \rightarrow 0} \left\{ \lambda I + \{M\}\{M\}^T \right\}^{-1} \{M\}^T \quad (15)$$

where λ is l_2 regularization.

However, due to the recurrent connections in the enhancement nodes, incremental learning in ORBLS is too complicated to be feasible. Inspired by the idea of parameter optimization, the improved PSO [38] is introduced to find the parameters of ORBLS: $\{NF, NW, NE\}$, instead of incremental learning.

Randomly generate n particles with their dimensions of three dimensions correspond to the three parameters of ORBLS $\{NF, NW, NE\}$. Initialize the particle position $x_{id} \in (1, 100)$ and speed $v_{id} \in (-1, 1)$. Determine the value of each parameter, such as $c_1 = c_2 = 1.5$, $w_{max} = 1.0$, $w_{min} = 0.4$, and $iter_{max} = 100$, where c_1 and c_2 are learning factors, w_{max} and w_{min} are inertia weights, and $iter_{max}$ is the maximum number of iterations.

When the iteration of IPSO is consistently performed, the position and speed of the particles are continually updated through the following formula:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{gd}) \quad (16)$$

$$x_{id} = x_{id} + \gamma v_{id} \quad (17)$$

where γ is the velocity coefficient; the value of inertia weight w is $w = w_{max} - (w_{max} - w_{min}) \times 1/iter$. When reaching the maximum iterative number $iter_{max}$, the iteration is stopped and the best value of $\{NF, NW, NE\}$ can be obtained.

2.5. BTCN Model

We designed the BTCN as one of the models to predict the complex subsequence set. TCN [39] has powerful feature extraction capabilities and is well suited for processing time series. To alleviate the problem of loss of information of TCN, this paper proposes BTCN, where an expanded causal convolution layer is added in parallel to each residual block.

BTCN satisfies two principles: (1) the generated output is of the same length as the input; (2) the information cannot leak from the future to the past. BTCN mainly consists of dilated causal convolution and improved stacked residual blocks, where the former is used for historical information extraction and the latter is used for network architecture building of BTCN.

The first layer of BTCN is a one-dimensional fully convolutional network, and zero adding of length (kernel size 1) is added to keep subsequent layers the same length. Therefore, the hidden layer and the input layer of BTCN have the same length, which in turn satisfies principle (1). To satisfy principle (2), BTCN uses causal convolution. Let $\{x^1, x^2, \dots, x^t\}$ be the input, then the causal convolution is described as:

$$p(x) = \prod_{t=1}^T p(x^t | x^1, x^2, \dots, x^t) \quad (18)$$

To increase the receptive field, the dilated convolution is introduced into the causal convolution, which leads to the dilated causal convolution. Dilated causal convolution increases the receptive field by increasing the number of layers and parameters of the network. The relationship between the receptive field of the dilated causal convolution and the causal convolution is formulated as follows:

$$\tilde{h}_v = (d - v) \times (h_v - v) + h_v \quad (19)$$

where d denotes the dilated rate, and h_v and \tilde{h}_v are the receptive field sizes of the causal convolution and the dilated causal convolution in the v th layer, respectively. The receptive field of the latter layer is closely related to the receptive field of the former layer as follows:

$$\begin{aligned} w_v &= w_{v-1} + (h_v - 1) \times l_{v-1} \\ &= w_{v-1} - \left[(\tilde{h}_v - 1) \times \prod_{v=1}^{V-1} S_v \right] \end{aligned} \quad (20)$$

where w_v and w_{v-1} are the sizes of the receptive field at layer v and $v - 1$, respectively. S_v is the stride at layer v ($v = 1, 2, \dots, V - 1$).

Considering the problem of gradient decay during training, the residual connection is introduced in the output layer of TCN. The residual block applied in TCN consists of two layers of dilated causal convolution.

In order to alleviate the problem of massive local information loss in the dilated causal convolution, BTCN improves the residual block. As shown in Figure 3, BTCN adds a dilated causal convolution layer in parallel within each residual block. In residual block, the output o of transformation $R(x)$ is added to the input x as follows:

$$o = \text{Activation}[x + R_1(x) + R_2(x)] \quad (21)$$

where $\text{Activation}[\cdot]$ is activation function.

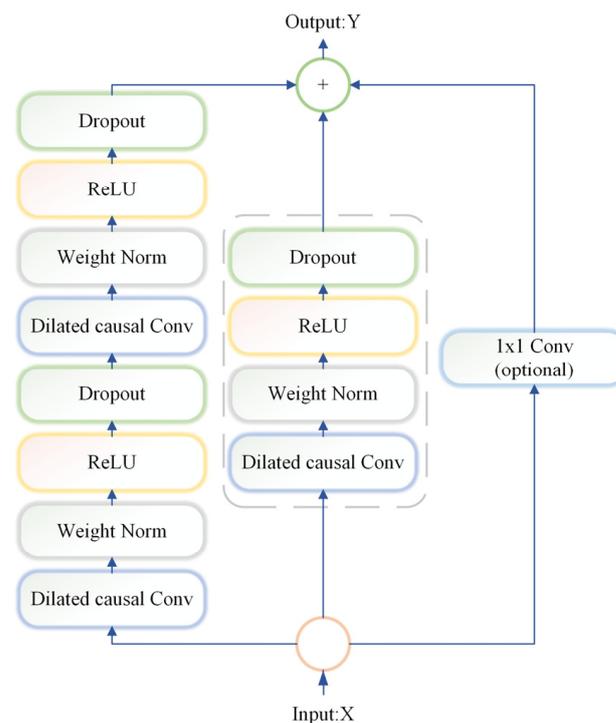


Figure 3. Residual block of BTCN.

Finally, the output of these two parallel causal convolutions is added to the input after 1×1 convolution operation, and the function of 1×1 convolution is used to ensure a uniform tensor shape.

2.6. The Prediction Steps

In this paper, we propose a novel model for ultra-short-term wind speed prediction. First, we use ICEEMDAN to decompose the original wind sequence $X(n) = \{x_1, x_2, \dots, x_n\}$ into multiple subseries $IMF_k(n), k = 1, 2, \dots, 11$. It should be noted that ICEEMDAN adaptively determines the number of decomposed subsequences based on the size and volatility of the data. The sample entropy $SE(m, r)$ is then calculated for each sequence.

The higher the SE value, the lower the autocorrelation of the series and the higher the complexity. We sort all sequences according to the sample entropy value from small to large, and take the first five sequences to form a complex subsequence set, and the rest are simple subsequence sets.

Then, the ORBLS prediction network is used to predict a set of simple subsequences, where PSO is used to optimize the hyper-parameters {NF, NW, NE} of ORBLS. The BTCN prediction network is designed to make predictions on complex subsets of sequences. It is worth noting that this predictive relationship, i.e., ORBLS predicts the simple subsequence set and BTCN predicts the complex subsequence set, is determined by matching the predictive power of the network with the complexity of the sequence set. Subsequent experiments confirmed this fact. Finally, the predicted values of all subsequences are superimposed to obtain the final predicted wind speed: $Y(n) = \{y_1, y_2, \dots, y_n\}$.

The entire steps of the proposed predictive framework are as follows:

Step 1: ICEEMDAN is used to decompose the raw wind speed data $\{x_1, x_2, \dots, x_n\}$ into a series of subseries $IM\check{F}_k(n), k = 1, 2, \dots, 11$.

Step 2: Calculate the SE value of each $IM\check{F}_k(n)$.

Step 3: Sort the $IM\check{F}_k(n)$ by SE value from largest to smallest and take the top five $IM\check{F}_k(n)$ to form a complex sequence set. The rest are classified as the simple sequence set.

Step 4: Build ORBLS model to predict simple sequence set and build BTCN model to predict complex sequence set.

Step 5: The final forecasted results can be obtained by stacking all forecasted values.

To show the working principle of the model more clearly, we list the specific steps in Algorithm 1.

Algorithm 1: The proposed predictive framework for WSP

Input:

$$X(n) = \{x_1, x_2, \dots, x_n\}$$

$$n_p = 40, e_p = 50, iter_{max} = 100.$$

Output:

$$Y(n) = \{y_1, y_2, \dots, y_i\}$$

Process:

1: **for** $k = 1:K$

2: get $IM\check{F}_k(n)$ by Equation (5)

3: get SE_k by Equation (10)

4: **end**

5: //ORBLS algorithm main steps

6: **for** $i = 1:n$ **do**

7: Calculate $Z_i = \varphi_i(X(n)W_{ei} + \beta_{ei})$

8: **end**

9: Obtain the final Z^n

10: **for** $j = 1:n$ **do**

11: Calculate $H_j = \psi_j(s_1 Z_i W_{hj} + \beta_{hj} + s_2 H_{j-1} W_{mj})$

12: **end**

13: Obtain the final H^m

14: Calculate connecting weights: $W_m^n \triangleq \{Z^n | H^m\}^+ Y$

15: Obtain the $Y = \{Z^n | H^m\} W_m^n$

16: //BTCN algorithm main steps

17: Calculate causal convolution $\prod_{t=1}^T p(x^t | x^1, x^2, \dots, x^t)$

18: Increase the receptive field: $\tilde{h}_v = (d - v) \times (h_v - v) + h_v$

19: Update the receptive field: $w_v = w_{v-1} - \left[(\tilde{h}_v - 1) \times \prod_{v=1}^{V-1} S_v \right]$

20: Calculate basic block $o = Activation|x + R(x)|$

3. Case Analysis

3.1. Data Description

To demonstrate the predictive performance of the proposed model in ultra-short-term WSP, we used four wind speed datasets from four German power companies including 50Hertz-power-company (HER), Amprion-power-company (AMP), TenneTTSO-power-company (TEN), and TransnetBW-power-company (TRA). Those datasets with 15 min intervals were collected from 23 August 2019 to 22 September 2019. Each dataset contains 2880 samples. The datasets are freely available at <http://www.netztransparenz.de/> (accessed on 20 March 2021).

We perform single-step ultra-short-term WSP experiments with a step length of 15 min. In our experiments, the first 80% of the wind speed sequence is used as the training set, and the rest is done as the test set for ultra-short-term wind prediction. Table 1 displays the information of the four datasets. Experiments are implemented in MATLAB R2021b on a 64-bit personal computer with Intel(R) core i5-9300 CPU/16.00 GB RAM.

Table 1. The statistical information of wind speed data.

| Dataset | Time-Step (min) | Mean (m/s) | Median (m/s) | Max (m/s) | Min (m/s) | Standard Deviation (m/s) |
|---------|-----------------|------------|--------------|-----------|-----------|--------------------------|
| HER | 15 | 0.89764 | 0.65000 | 5.60000 | 0 | 0.88480 |
| AMP | 15 | 0.42297 | 0.30000 | 2.06000 | 0 | 0.38803 |
| TEN | 15 | 0.84532 | 0.59420 | 4.76590 | 0.01420 | 0.78575 |
| TRA | 15 | 0.12757 | 0.07445 | 0.86230 | 0 | 0.14548 |

3.2. Evaluation Index

To comprehensively evaluate the prediction performance of the proposed model, four evaluation indicators were given. *MAE* can accurately reflect the average value of the absolute error. *MAPE* divides the absolute error by the corresponding actual value. *RMSE* represents the sample standard deviation between the predicted value and the actual observation value, which has a very sensitive reflection and can well reflect the accuracy of the prediction. *SSE* represents the total error of the model. Their definitions are as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (22)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (23)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (24)$$

$$SSE = \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (25)$$

where \hat{y}_i is the predicted value and y_i is the actual value.

3.3. Comparable Methods

To verify the effectiveness and advancement of the proposed model, it was compared with twelve advanced predictive models, involving PSO-ANFIS [40], VMD-GA-BP [41], EEMD-GPR-LSTM [42], MWS-CE-ENN [19], ICE-MOV-BFS [26], BLS [37], TCN [39], optimized BLS (OBLS), ORBLS, BTCN, ICEEMDAN-ORBLS, and ICEEMDAN-BTCN. Table 2 lists the parameter settings of six comparison methods. BLS, TCN, OBLS, ORBLS, BTCN,

ICEEMDAN-ORBLs, and ICEEMDAN-BTCN are used to perform the ablation experiment for proposed model.

Table 2. Parameter settings of the models.

| Model | Parameter Setting |
|----------------|---|
| PSO-ANFIS | $iter_{max} = 300, n_p = 40, c_1 = 1.0, c_2 = 2.0, n_m = 4, n_r = 4, n_v = 52$ |
| VMD-GA-BP | $k = 11, iter_{max} = 150, e_p = 100, n_p = 40, l_r = 0.1, n_{b1} = 9$ |
| EEMD-GPR-LSTM | $k = 11, e_p = 200, n_{b1} = 100, n_{b2} = 100, s_1 = 50, \sigma = 20$ |
| MWS-CE-ENN | $e_p = 1000, l_r = 0.1, p_r = 0.000001, n_p = 40, n_i = 5, n_{b1} = 6, n_o = 1,$ $iter_{max} = 100, n_{std} = 0.2$ |
| ICE-MOV-BFS | $e_p = 100, n_i = 3, n_{b1} = 5, n_{b2} = 5, n_o = 3$ |
| Proposed Model | $n_{std} = 0.01, n_p = 40, iter_{max} = 100, c_1 = 1.5, c_2 = 1.5, e_p = 50, l_r = 0.002,$ $\lambda = 10_{-30}, w_d = 0.05, n_f = 32, k_s = 3, d_f = [1, 2, 4, 8, 16]$ |

In Table 2, $iter_{max}$ is the iterative number; e_p is the number of network iterations; n_p is population size; c_1 and c_2 are personal and global learning coefficients, respectively; $n_r, n_v, n_f, n_i,$ and n_o are the number of rules, variables, filters, input nodes, and output nodes, respectively; k is decomposition number of VMD/EEMD; n_{bi} is the number of the i -th hidden nodes; l_r is the learning rate of the network; p_r is the training requirement accuracy; w_d is the weight of dropout; s_1 is the length scale of GPR; σ is the parameter in GPR; n_{std} is the noise standard deviation in ICEEMDAN/CEEMDAN; λ is the regularization parameter for ridge regression, k_s is the kernel size of filter, d_f is the dilation factor.

3.4. Experimental Results

3.4.1. Experiment I: Comparison between Different Forecasting Methods

We verified the effectiveness and advancement of the proposed model by comparing it with PSO-ANFIS, VMD-GA-BP, EEMD-GPR-LSTM, MWS-CE-ENN, and ICE-MOV-BFS. The training and testing processes of each were repeated 10 times for all models. The average values of the evaluation indicators are listed Table 3, where the best predictions are highlighted with dark gray backgrounds.

Table 3. Forecasting performances of the proposed model and reference models.

| Dataset | Metrics | Proposed Model | PSO-ANFIS | VMD-GA-BP | EEMD-GPR-LSTM | MWS-CE-ENN | ICE-MOV-BFS |
|---------|---------|----------------|-----------|-----------|---------------|------------|-------------|
| HER | RMSE | 0.0086 | 0.0120 | 0.0137 | 0.0126 | 0.0132 | 0.0097 |
| | SSE | 0.0317 | 0.0615 | 0.0806 | 0.0676 | 0.0746 | 0.0403 |
| | MAPE | 4.1366 | 5.1392 | 5.2415 | 11.7694 | 6.1941 | 4.7523 |
| | MAE | 0.0053 | 0.0077 | 0.0092 | 0.0089 | 0.0082 | 0.0069 |
| AMP | RMSE | 0.0147 | 0.0186 | 0.0187 | 0.0296 | 0.0231 | 0.0167 |
| | SSE | 0.0930 | 0.1481 | 0.1505 | 0.3743 | 0.2284 | 0.1197 |
| | MAPE | 4.1848 | 5.0249 | 6.3716 | 5.9358 | 5.8755 | 4.6723 |
| | MAE | 0.0107 | 0.0133 | 0.0146 | 0.0143 | 0.0127 | 0.0124 |
| TEN | RMSE | 0.0087 | 0.0107 | 0.0114 | 0.0118 | 0.0129 | 0.0094 |
| | SSE | 0.0324 | 0.0490 | 0.0558 | 0.0591 | 0.1601 | 0.0378 |
| | MAPE | 3.5071 | 4.1562 | 6.3884 | 5.4492 | 5.2321 | 4.8391 |
| | MAE | 0.0055 | 0.0066 | 0.0083 | 0.0073 | 0.0081 | 0.0069 |
| TRA | RMSE | 0.0259 | 0.0369 | 0.0357 | 0.0266 | 0.0327 | 0.0264 |
| | SSE | 0.2879 | 0.5829 | 0.5460 | 0.3029 | 0.4577 | 0.2983 |
| | MAPE | 5.7290 | 7.9526 | 8.8673 | 5.7345 | 6.3028 | 5.7326 |
| | MAE | 0.0183 | 0.0271 | 0.0266 | 0.0197 | 0.0244 | 0.0191 |

Interestingly, we can find from Table 3 that the proposed model has the best prediction performance on each of the RMSE, SSE, MAPE, and MAE evaluation indicators of the four datasets among all models. It is worth noting that even on the most volatile TRA dataset, our model still outperforms all the benchmark models, which confirms the superiority of our model.

To visualize the effectiveness of the proposed model in fitting the wind speed series, we selected one day from each of the four datasets and plotted the predicted versus actual values of the model as shown in Figure 4. From Figure 4, it can be seen that the proposed model shows a high fit on all datasets, which reflects the effectiveness of our model. Moreover, our model fits best on the HER dataset, and the prediction curve is close to the actual value curve, indicating a good prediction.

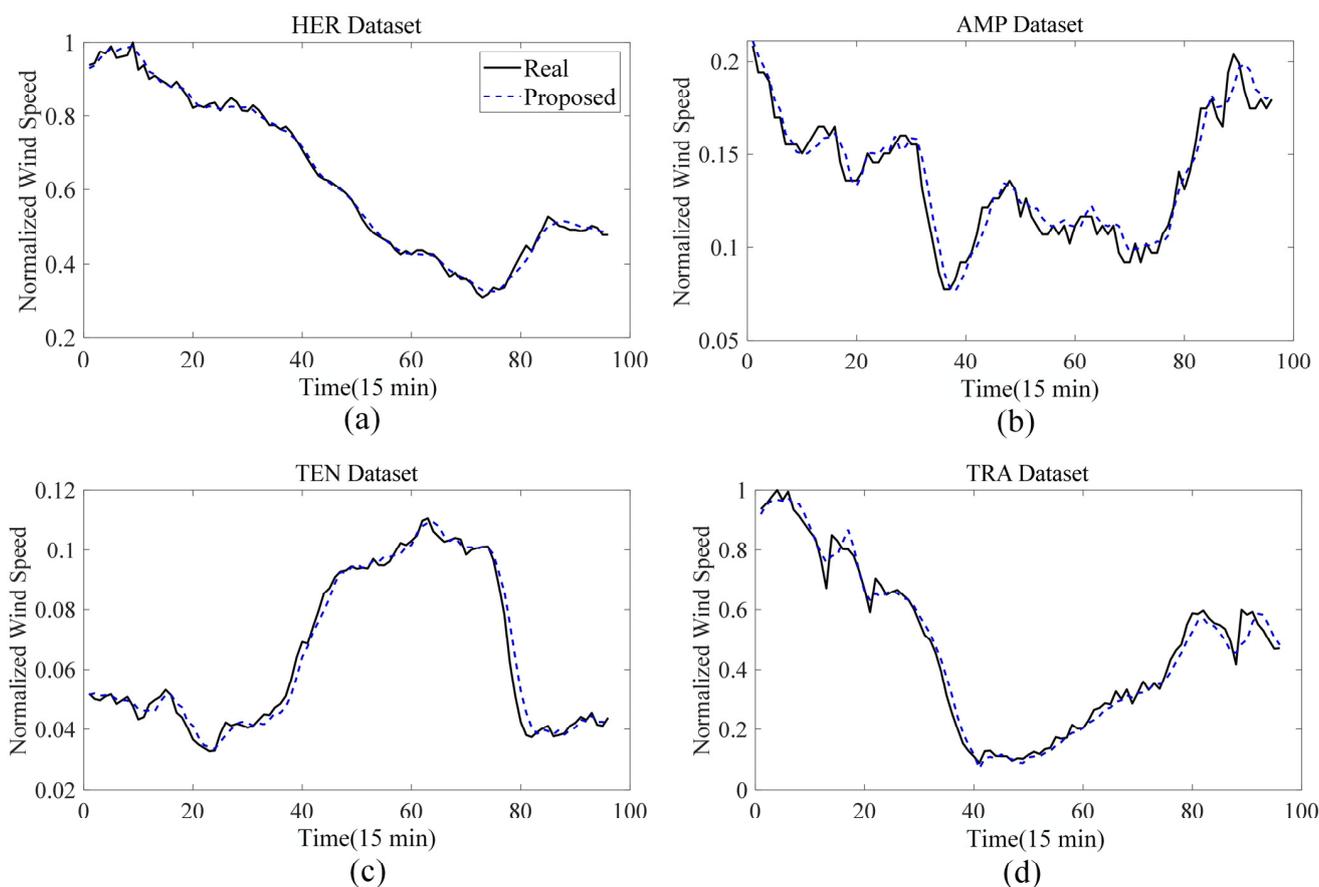


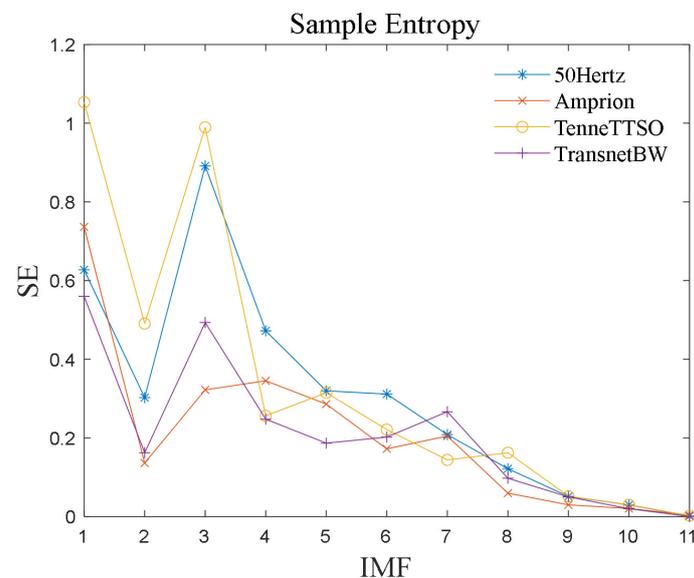
Figure 4. Wind speed fitting curves of the proposed model for four wind datasets: (a) experiments results on HER wind datasets; (b) experiments results on AMP wind datasets; (c) experiments results on TEN wind datasets; (d) experiments results on TRA wind datasets.

3.4.2. Experiment II: Experiments on Sample Entropy to Quantify Sequence Complexity

We designed ORBLS and BTCN to predict the simple sequence set and complex sequence set respectively, and to verify their reasonableness, we designed Experiment II. We use ORBLS to predict the two sequence sets, the simple sequence set and the complex sequence set, which are divided by sample entropy, and BTCN also predicts these two sequence sets. The experimental results are shown in Table 4, where all models were repeatedly trained and tested 10 times. In addition, the SE values of each IMF are plotted in Figure 5.

Table 4. Forecasting performances of BTCN and ORBLS.

| Model | Subsequence Set | RMSE | SSE | MAE |
|-------|----------------------|---------------|---------------|---------------|
| BTCN | high subsequence set | 0.0031 | 0.0042 | 0.0016 |
| | low subsequence set | 0.0085 | 0.0310 | 0.0009 |
| ORBLS | high subsequence set | 0.0036 | 0.0056 | 0.0020 |
| | low subsequence set | 0.0007 | 0.0002 | 0.0005 |

**Figure 5.** The value of SE for all subsequences.

According to Figure 5, the top five IMFs with the highest SE values are classified as complex sequence sets; the remaining ones are classified as simple sequence sets. According to Table 4, BTCN predicts better for complex sequence sets compared to simple sequence sets, and ORBLS predicts the simple sequence set much better than the complex sequence set.

3.4.3. Experiment III: Ablation Experiment between Single Models and Hybrid Models

To verify the rationality of the proposed model, it was compared with BLS, TCN, optimized BLS (OBLS), ORBLS, BTCN, ICEEMDAN-ORBLS, and ICEEMDAN-BTCN. Similarly, all models were repeatedly trained and tested 10 times. The experimental results are shown in Table 5, where the best predictions are highlighted with dark gray backgrounds. The forecast results for 22 September 2019 are plotted in Figure 6, which also shows the forecast errors in superimposed shades.

Table 5. Forecasting performances of the proposed model and reference models.

| Model | RMSE | SSE | MAPE (%) | MAE |
|----------------|---------------|---------------|---------------|---------------|
| TCN | 0.0175 | 0.1310 | 8.4706 | 0.0108 |
| BLS | 0.0308 | 0.4058 | 6.4823 | 0.0164 |
| BTCN | 0.0149 | 0.0952 | 5.6806 | 0.0091 |
| OBLS | 0.0122 | 0.0642 | 5.4614 | 0.0080 |
| ORBLS | 0.0117 | 0.0588 | 5.0518 | 0.0075 |
| ICEEMDAN-BTCN | 0.0146 | 0.0916 | 6.0596 | 0.0094 |
| ICEEMDAN-ORBLS | 0.0091 | 0.0356 | 4.6292 | 0.0061 |
| Proposed | 0.0086 | 0.0317 | 4.1366 | 0.0053 |

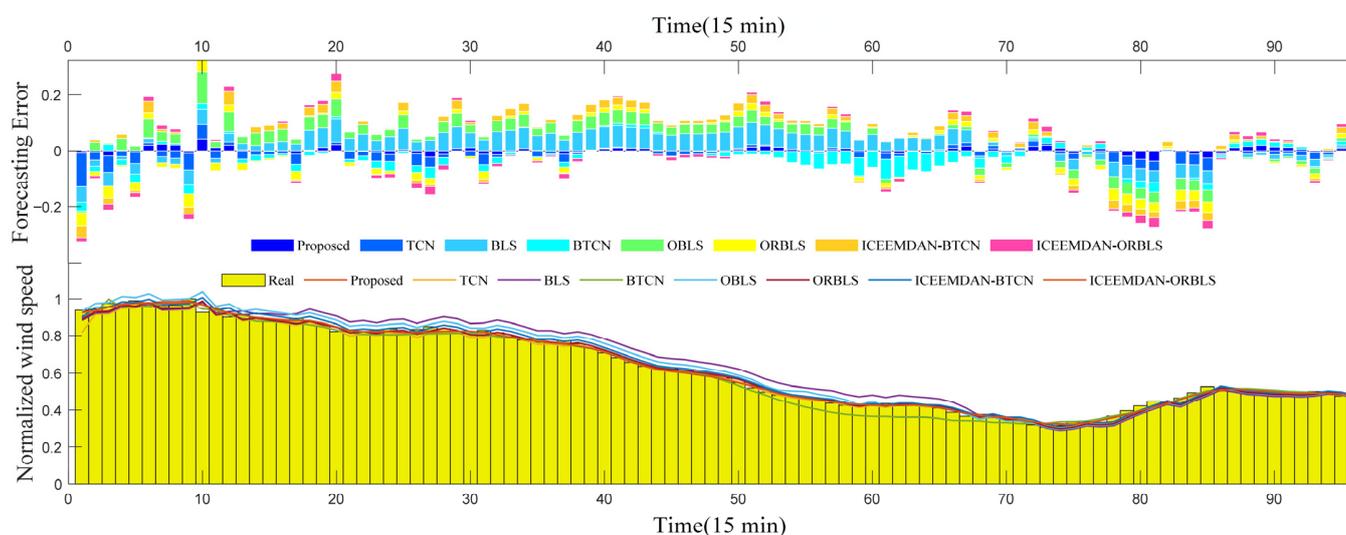


Figure 6. Forecasting results in HER dataset (22 September 2019).

It can be found in Figure 6 that the prediction error of our proposed model is the smallest and the curve fitting is the closest. According to Table 5, the proposed model is consistently superior to BLS, TCN, optimized BLS (OBLS), ORBLS, BTCN, ICEEMDAN-ORBLS, and ICEEMDAN-BTCN. This further proves the advantages of our proposed model due to its combining ICEEMDAN, ORBLS, and BTCN.

Furthermore, first, compared with ICEEMDAN-ORBLS or ICEEMDAN-BTCN, the proposed model provides better predictive performance due to combining ORBLS and BTCN, thus showing the effectiveness of sample entropy. Second, compared with BLS or OBLS, ORBLS obtains better predictive performance due to cyclic enhancement nodes and IPSO optimization hyperparameters. Third, BTCN offers better predictive performance compared to TCN, which proves the advantages of BTCN. Fourth, compared with ORBLS and BTCN, ORBLS and BTCN based on ICEEMDAN decomposition have better prediction results, which reflects the importance of decomposition for improving model prediction performance.

4. Conclusions

In this paper, we propose a novel strategy based on ICEEMDAN, sample entropy, ORBLS, and BTCN for ultra-short-term WSP. Experimental results show that the proposed model achieves better performance than the compared advanced predictive models. That can be attributed to the following several reasons: (1) The proposed model combines ORBLS and BTCN well for prediction by sample entropy and obtained more satisfactory forecasting precision. (2) ICEEMDAN decomposes the raw wind sequence to effectively smooth the nonlinear part of the wind speed data, which enables the model to more accurately simulate the fluctuations of wind and thus further improve the prediction performance. (3) ORBLS has cyclically connected enhancement nodes to better capture the dynamic characteristics of the wind speed sequence. Note that improved PSO replaces incremental learning to better update the hyper-parameters of ORBLS, which can provide high-quality sample generation performance; therefore, ORBLS obtains high prediction accuracy and stability. (4) BTCN adds a dilated causal convolution layer in parallel to each residual block to alleviate the problem of massive local information loss, which further improves its predictive performance.

In future work, we will consider the effect of the other variables such as vertical wind speed, air temperature, spatial location, and turbulent energy transport on the ultra-short-term wind speed prediction, which may further improve the predictive performance.

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

Funding: This work was supported in part by the National Natural Science Foundation of China under Grants 62173176 and 61863028, and in part by the Science and Technology Department of Jiangxi Province of China under Grants 20204ABC03A39.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Nomenclature

| | |
|----------|--|
| AI | Artificial intelligence |
| ANFIS | Adaptive-network-based fuzzy inference system |
| ARIMA | Autoregressive integrated moving average |
| BLS | Broad learning system |
| BP | Back propagation |
| BTCN | Broadened TCN |
| CEEMD | Complementary ensemble empirical mode decomposition |
| CEEMDAN | Complete ensemble empirical mode decomposition with adaptive noise |
| CNN | Convolutional neural network |
| EEMD | Ensemble empirical mode decomposition |
| EMD | Empirical mode decomposition |
| ELM | Extreme learning machine |
| GA | Genetic algorithm |
| GMDH | Group method of data handling neural network |
| GPR | Gaussian process regression |
| ICEEMDAN | Improved CEEMDAN |
| IMF | Intrinsic mode functions |
| IOWA | Induced ordered weighted averaging |
| LSTM | Long short-term memory |
| MAE | Mean absolute error |
| MAPE | Mean absolute percentage error |
| PSO | Particle swarm optimization |
| ORBLS | Optimized recurrent BLS |
| RMSE | Root mean square error |
| SE | Sample entropy |
| SSE | Sum of squared error |
| SVM | Support vector machine |
| TCN | Temporal convolutional network |
| VMD | Variational mode decomposition |
| WSP | Wind speed prediction |

References

1. Hu, J.; Wang, J.; Zeng, G. A hybrid forecasting approach applied to wind speed time series. *Renew. Energy* **2013**, *60*, 185–194. [[CrossRef](#)]
2. Kavasseri, R.G.; Seetharaman, K. Day-ahead wind speed forecasting using f-ARIMA models. *Renew. Energy* **2009**, *34*, 1388–1393. [[CrossRef](#)]
3. Singh, S.; Mohapatra, A. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew. Energy* **2019**, *136*, 758–768. [[CrossRef](#)]
4. Shi, J.; Liu, Y.; Li, Y.; Liu, Y.; Roux, G.; Shi, L.; Fan, X. Wind Speed Forecasts of a Mesoscale Ensemble for Large-Scale Wind Farms in Northern China: Downscaling Effect of Global Model Forecasts. *Energies* **2022**, *15*, 896. [[CrossRef](#)]

5. Wang, H.; Han, S.; Liu, Y.; Yan, J.; Li, L. Sequence transfer correction algorithm for numerical weather prediction wind speed and its application in a wind power forecasting system. *Appl. Energy* **2019**, *237*, 1–10. [[CrossRef](#)]
6. Hill, D.C.; McMillan, D.; Bell, K.R.W.; Infield, D. Application of Auto-Regressive Models to U.K. Wind Speed Data for Power System Impact Studies. *IEEE Trans. Sustain. Energy* **2011**, *3*, 134–141. [[CrossRef](#)]
7. Do, D.-P.N.; Lee, Y.; Choi, J. Hourly Average Wind Speed Simulation and Forecast Based on ARMA Model in Jeju Island, Korea. *J. Electr. Eng. Technol.* **2016**, *11*, 1548–1555. [[CrossRef](#)]
8. Yunus, K.; Thiringer, T.; Chen, P. ARIMA-Based Frequency-Decomposed Modeling of Wind Speed Time Series. *IEEE Trans. Power Syst.* **2015**, *31*, 2546–2556. [[CrossRef](#)]
9. Wang, Y.; Wang, H.; Srinivasan, D.; Hu, Q. Robust functional regression for wind speed forecasting based on Sparse Bayesian learning. *Renew. Energy* **2018**, *132*, 43–60. [[CrossRef](#)]
10. Elsaraiti, M.; Merabet, A. A Comparative Analysis of the ARIMA and LSTM Predictive Models and Their Effectiveness for Predicting Wind Speed. *Energies* **2021**, *14*, 6782. [[CrossRef](#)]
11. Wang, S.; Zhang, N.; Wu, L.; Wang, Y. Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. *Renew. Energy* **2016**, *94*, 629–636. [[CrossRef](#)]
12. Alkesaiberi, A.; Harrou, F.; Sun, Y. Efficient Wind Power Prediction Using Machine Learning Methods: A Comparative Study. *Energies* **2022**, *15*, 2327. [[CrossRef](#)]
13. Ma, X.; Jin, Y.; Dong, Q. A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting. *Appl. Soft Comput.* **2017**, *54*, 296–312. [[CrossRef](#)]
14. Luo, X.; Sun, J.; Wang, L.; Wang, W.; Zhao, W.; Wu, J.; Wang, J.-H.; Zhang, Z. Short-Term Wind Speed Forecasting via Stacked Extreme Learning Machine with Generalized Correntropy. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4963–4971. [[CrossRef](#)]
15. Wang, H.; Wang, G.; Li, G.; Peng, J.; Liu, Y. Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Appl. Energy* **2016**, *182*, 80–93. [[CrossRef](#)]
16. Shang, Z.; Wen, Q.; Chen, Y.; Zhou, B.; Xu, M. Wind Speed Forecasting Using Attention-Based Causal Convolutional Network and Wind Energy Conversion. *Energies* **2022**, *15*, 2881. [[CrossRef](#)]
17. Khodayar, M.; Wang, J. Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting. *IEEE Trans. Sustain. Energy* **2018**, *10*, 670–681. [[CrossRef](#)]
18. Zhu, Q.; Chen, J.; Shi, D.; Zhu, L.; Bai, X.; Duan, X.; Liu, Y. Learning Temporal and Spatial Correlations Jointly: A Unified Framework for Wind Speed Prediction. *IEEE Trans. Sustain. Energy* **2019**, *11*, 509–523. [[CrossRef](#)]
19. Wang, J.; Yang, Z. Ultra-short-term wind speed forecasting using an optimized artificial intelligence algorithm. *Renew. Energy* **2021**, *171*, 1418–1435. [[CrossRef](#)]
20. Ren, Y.; Suganthan, P.N.; Srikanth, N. A Comparative Study of Empirical Mode Decomposition-Based Short-Term Wind Speed Forecasting Methods. *IEEE Trans. Sustain. Energy* **2014**, *6*, 236–244. [[CrossRef](#)]
21. Yang, Y.; Fan, C.; Xiong, H. A novel general-purpose hybrid model for time series forecasting. *Appl. Intell.* **2021**, *52*, 2212–2223. [[CrossRef](#)] [[PubMed](#)]
22. Domínguez-Navarro, J.; Lopez-Garcia, T.; Valdivia-Bautista, S. Applying Wavelet Filters in Wind Forecasting Methods. *Energies* **2021**, *14*, 3181. [[CrossRef](#)]
23. Zhang, Y.; Pan, G. A hybrid prediction model for forecasting wind energy resources. *Environ. Sci. Pollut. Res.* **2020**, *27*, 19428–19446. [[CrossRef](#)] [[PubMed](#)]
24. 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](#)]
25. Altan, A.; Karasu, S.; Zio, E. A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Appl. Soft Comput.* **2020**, *100*, 106996. [[CrossRef](#)]
26. Nie, Y.; Liang, N.; Wang, J. Ultra-short-term wind-speed bi-forecasting system via artificial intelligence and a double-forecasting scheme. *Appl. Energy* **2021**, *301*, 117452. [[CrossRef](#)]
27. Liu, M.; Cao, Z.; Zhang, J.; Wang, L.; Huang, C.; Luo, X. Short-term wind speed forecasting based on the Jaya-SVM model. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106056. [[CrossRef](#)]
28. Tian, Z. Modes decomposition forecasting approach for ultra-short-term wind speed. *Appl. Soft Comput.* **2021**, *105*, 107303. [[CrossRef](#)]
29. Liu, Z.; Hara, R.; Kita, H. Hybrid forecasting system based on data area division and deep learning neural network for short-term wind speed forecasting. *Energy Convers. Manag.* **2021**, *238*, 114136. [[CrossRef](#)]
30. Huang, H.; Jia, R.; Shi, X.; Liang, J.; Dang, J. Feature selection and hyper parameters optimization for short-term wind power forecast. *Appl. Intell.* **2021**, *51*, 6752–6770. [[CrossRef](#)]
31. Liu, J.; Shi, Q.; Han, R.; Yang, J. A Hybrid GA-PSO-CNN Model for Ultra-Short-Term Wind Power Forecasting. *Energies* **2021**, *14*, 6500. [[CrossRef](#)]
32. Duan, J.; Zuo, H.; Bai, Y.; Duan, J.; Chang, M.; Chen, B. Short-term wind speed forecasting using recurrent neural networks with error correction. *Energy* **2021**, *217*, 119397. [[CrossRef](#)]
33. Liu, H.; Yang, R.; Wang, T.; Zhang, L. A hybrid neural network model for short-term wind speed forecasting based on decomposition, multi-learner ensemble, and adaptive multiple error corrections. *Renew. Energy* **2020**, *165*, 573–594. [[CrossRef](#)]

34. Zhang, Y.; Han, J.; Pan, G.; Xu, Y.; Wang, F. A multi-stage predicting methodology based on data decomposition and error correction for ultra-short-term wind energy prediction. *J. Clean. Prod.* **2021**, *292*, 125981. [[CrossRef](#)]
35. Colominas, M.A.; Schlotthauer, G.; Torres, M.E. Improved complete ensemble EMD: A suitable tool for biomedical signal processing. *Biomed. Signal Process. Control* **2014**, *14*, 19–29. [[CrossRef](#)]
36. Wu, Q.; Lin, H. Daily urban air quality index forecasting based on variational mode decomposition, sample entropy and LSTM neural network. *Sustain. Cities Soc.* **2019**, *50*, 101657. [[CrossRef](#)]
37. 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.* **2017**, *29*, 10–24. [[CrossRef](#)]
38. Zhang, Y.; Sun, H.; Guo, Y. Wind Power Prediction Based on PSO-SVR and Grey Combination Model. *IEEE Access* **2019**, *7*, 136254–136267. [[CrossRef](#)]
39. Bai, S.; Kolter, J.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2019**, arXiv:1803.01271.
40. Dokur, E.; Yuzgec, U.; Kurban, M. Performance Comparison of Hybrid Neuro-Fuzzy Models using Meta-Heuristic Algorithms for Short-Term Wind Speed Forecasting. *Electrica* **2021**, *21*, 305–321. [[CrossRef](#)]
41. Zhang, Y.; Pan, G.; Chen, B.; Han, J.; Zhao, Y.; Zhang, C. Short-term wind speed prediction model based on GA-ANN improved by VMD. *Renew. Energy* **2020**, *156*, 1373–1388. [[CrossRef](#)]
42. Huang, Y.; Liu, S.; Yang, L. Wind Speed Forecasting Method Using EEMD and the Combination Forecasting Method Based on GPR and LSTM. *Sustainability* **2018**, *10*, 3693. [[CrossRef](#)]