sustainability

MDPI

Article

A Hybrid Generative Adversarial Network Model for Ultra Short-Term Wind Speed Prediction

Qingyuan Wang [1,†], Longnv Huang [1,†], Jiehui Huang [1,†], Qiaoan Liu [2], Limin Chen [1,*], Yin Liang [1], Peter X. Liu [1,3] and Chunquan Li [1]

[1] The School of Information Engineering, Nanchang University, Nanchang 330031, China; 7904119023@email.ncu.edu.cn (Q.W.); 6105119006@email.ncu.edu.cn (L.H.); 7803018161@email.ncu.edu.cn (J.H.); liangyin@ncu.edu.cn (Y.L.); xpliu@sce.carleton.ca (P.X.L.); lichunquan@ncu.edu.cn (C.L.)
[2] The School of Future Technology, Nanchang University, Nanchang 330031, China; 5811121050@email.ncu.edu.cn
[3] The 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: To improve the accuracy of ultra-short-term wind speed prediction, a hybrid generative adversarial network model (HGANN) is proposed in this paper. Firstly, to reduce the noise of the wind sequence, the raw wind data are decomposed using complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN). Then the decomposed modalities are entered into the HGANN network for prediction. HGANN is a continuous game between the generator and the discriminator, which in turn allows the generator to learn the distribution of the wind data and make predictions about it. Notably, we developed the optimized broad learning system (OBLS) as a generator for the HGANN network, which can improve the generalization ability and error convergence of HGANN. In addition, improved particle swarm optimization (IPSO) was used to optimize the hyperparameters of OBLS. To validate the performance of the HGANN model, experiments were conducted using wind sequences from different regions and at different times. The experimental results show that our model outperforms other cutting-edge benchmark models in single-step and multi-step forecasts. This demonstrates not only the accuracy and robustness of the proposed model but also the applicability of our model to more general environments for wind speed prediction.

Keywords: wind speed forecast; OBLS; data preprocessing; optimized hyper-parameters

Citation: Wang, Q.; Huang, L.; Huang, J.; Liu, Q.; Chen, L.; Liang, Y.; Liu, P.X.; Li, C. A Hybrid Generative Adversarial Network Model for Ultra Short-Term Wind Speed Prediction. *Sustainability* **2022**, *14*, 9021. https://doi.org/10.3390/su14159021

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 5 June 2022
Accepted: 20 July 2022
Published: 22 July 2022

1. Introduction

Energy demand has always been one of the main problems of human development since the increasing consumption of energy with the improvement of living standards. In recent years, renewable energy has gradually become a research hotspot. Wind energy is valued for its clean, pollution-free, renewable, and abundant availability. However, wind is highly random and volatile, which may affect the stability of the power system and hinder the efficient use of wind energy [1]. Accurate ultra-short-term wind speed prediction models are therefore crucial in power dispatch planning and power market operations [2]. Thus, reliable wind speed prediction has drawn a lot of interest.

The three common wind speed prediction models are physical models, statistical models, and hybrid models. Physical models take into account the physical conditions and locations of wind farms, which require abundant meteorological data. Numerical weather prediction is a typical physical model, as it takes into account temperature pressure and obstacles for wind speed prediction, so it has a long calculation period [3]. Physical models

are efficient and accurate for long-term forecasting, but they are computationally intensive and expensive for small-scale forecasting.

Statistical models make better use of historical wind speed data to predict future wind speeds than physical models. Statistical models include both traditional statistical models and neural network-based models. Traditional statistical models include the autoregressive moving average model [4], the autoregressive integrated moving average model [5], the Bayesian model [6], etc. However, the non-linear nature of wind makes it difficult for traditional statistical models to extract the deeper features of wind speed data. Neural networks are introduced into the field of wind speed prediction for their ability to fit the non-linear part of the data well. Neural network-based models can extract deeper features from wind speed data than traditional statistical models—for example, BP [7], RBF [8], artificial neural network [9], SVR [10], etc. To improve the learning ability and predictive ability of predictive models, deep neural networks are introduced into wind speed prediction, such as the deep belief network [11], RNN [12], GNN [13], and LSTM [14].

In recent years, hybrid models have gradually become the mainstream wind speed prediction models. Hybrid models typically use one or more auxiliary strategies to assist the main forecasting network in wind speed prediction. Therefore, hybrid models can achieve better prediction performance than physical models and statistical models. The auxiliary strategies involved in hybrid models include data preprocessing techniques, optimization algorithms, error correction, and weighting strategies.

(1) Data preprocessing techniques. Zhang et al. [15] used EMD for data pre-processing of wind speed, which effectively reduced the volatility of the wind speed series. However, EMD suffers from the problem of modal confusion, which leads to unsatisfactory decomposition results. Santhosh et al. [16] used EEMD to process the raw wind speed series, which effectively mitigated the EMD problem. However, EEMD has a noise residual problem affected by noise residuals. Wang et al. [17] used CEEMD for wind speed prediction. CEEMD cancels out the residual noise with a pair of white noises, effectively improving the efficiency of the calculation. Ren et al. [18] experimentally demonstrated that the CEEMDAN-based model always performs best compared to the EMD-based model.

(2) Optimization algorithms. Optimization algorithms can be used to optimize the hyper-parameters, weights, network structure, and thresholds of predictive models. Li et al. [19] used PSO to optimize two hyper-parameters of LSTM, which solved the problem of wide intervals caused by interval superposition and thus improved the wind speed prediction accuracy. Tian [20] used PSO to optimize the weight coefficients of each prediction model, and the experimental results demonstrate the necessity of introducing the weight coefficient optimization strategy. Liu et al. [21] used GA to optimize the internal parameters of LSTM, which improved the efficiency and accuracy of the prediction model. Cui et al. [22] used the Bat algorithm to optimize the thresholds of BP networks, effectively improving the generalization ability and nonlinear mapping ability of BP networks.

(3) Error correction. Error correction is a post-processing technique for wind forecasting. It predicts the residuals and superimposes the predictions on the original predictions to obtain the final predictions. Duan et al. [23] used improved CEEMDAN to decompose the errors, and the experimental results showed that the error decomposition correction method can significantly enhance the prediction accuracy. Liu et al. [24] 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. [25] demonstrated experimentally that the final predicted values after Markov chain correction are closer to the original wind field data, which proves that the use of the Markov chain is effective.

(4) Weighting strategies. To scientifically determine the weights of different prediction networks in a hybrid model, many scholars have proposed different weighting strategies. To alleviate the adverse effects of multi-collinearity in combinatorial prediction models, Jiang et al. [26] used a GMDH neural network to automatically identify the weights of three nonlinear models. The application of GMDH can significantly improve the predictive

capability compared to the widely used equal-weighting scheme. Wang et al. [27] used MTO to minimize the error sum of squares of the IOWA operator, which obtains the optimal weight vector for the combined prediction model and ensures the stability of the prediction results. Altan et al. [28] optimized the weighting coefficients for each IMF using the gray wolf optimizer algorithm.

Although the above models achieve good predictive performance, they still have some problems. Methods involving deep neural networks [27] cause huge computational costs. Hybrid methods based on weighting strategy [28] may have the problem of multicollinearity, which reduces the prediction accuracy. The performance of hybrid methods based on parameter optimization [26] is largely influenced by the understanding of the researcher of the optimization algorithm.

Considering the above issues, we propose a hybrid model combining data preprocessing techniques and optimization algorithms for ultra-short-term wind speed prediction. We design the hybrid generative adversarial network (HGANN) as the prediction master network for the proposed hybrid model. The contributions and innovations of this research are concluded as follows:

(1) A hybrid generative adversarial network model (HGANN) is proposed for ultra-short-term wind speed prediction, which learns the distribution of wind data and predicts it through a continuous game between generators and discriminators.
(2) To improve the error convergence of the model, the OBLS was developed as a generator for HGANN. The IPSO was used to optimize the hyperparameters of the OBLS. To maintain the stability of the generated samples, we used the discriminator of WGAN as the discriminator of HGANN.
(3) A wind data decomposition and denoising process was carried out using CEEMDAN to reduce the randomness and instability in the original wind series.

The rest of this article is organized as follows. Section 2 introduces the model framework and methods involved in this article in detail. In Section 3, the experimental cases and prediction results are elaborated in detail, which verifies the validity of the framework we propose. Section 4 contains a discussion of the results of the experiment. The conclusions are presented in Section 5.

2. Proposed Predictive Framework

2.1. Overall Framework of HGANN

Generative adversarial networks (GANs) [29] are deep learning networks, which are composed of a generator and discriminator that confront each other. The role of the generator is to generate false samples that are close to the real ones. The role of the discriminator is to distinguish between true and false samples as correctly as possible. However, GANs often suffer from the problem of target confusion. Our proposed HGANN alleviates this problem to a great extent.

We developed a hybrid generative adversarial network model (HGANN) for ultra-short-term wind speed prediction, which uses the two networks to compete with each other to achieve highly accurate wind speed predictions. The proposed model is shown in Figure 1. First, CEEMDAN decomposes the raw wind speed data into multiple modalities. These modalities are separately fed into the generator of HGANN, the OBLS. The generator is used to obtain virtual samples that are similar to real samples. The virtual samples and real samples are then fed into the discriminator, which consists of convolutional layers and fully connected layers. The discriminator extracts the high-dimensional features of the input samples through the convolutional layer, and then further extracts the effective features by the fully connected layer. The outputs scalars "1" or "0" of the discriminator are passed to the generator and the discriminator to perform the iterative update of HGANN. Via the continuous iterative update, OBLS obtains the best parameters and performs wind speed prediction. Finally, the final wind speed forecast can be obtained by stacking all forecast values.

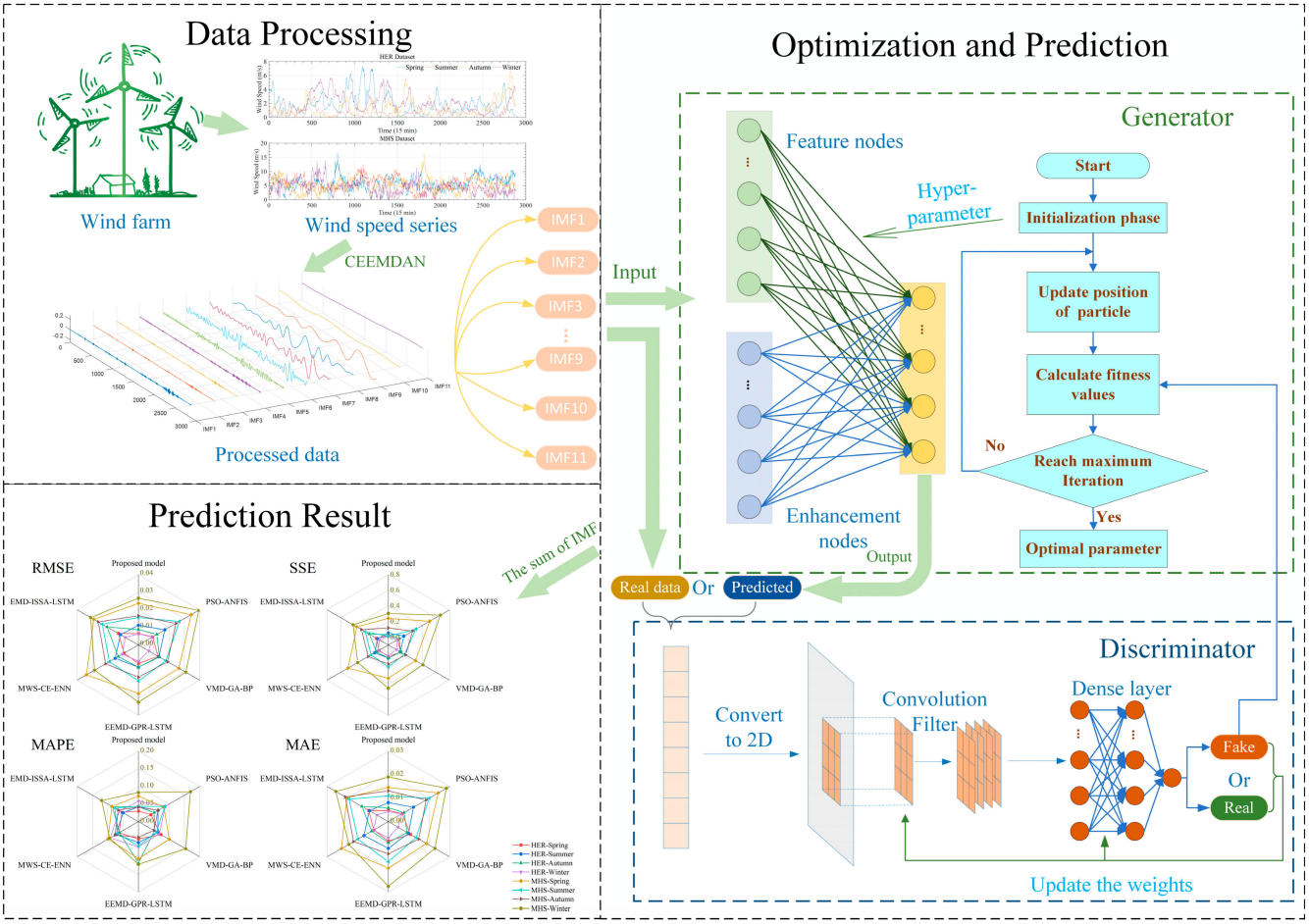

Figure 1. The proposed short-term wind speed forecasting framework. In the data processing step, CEEMDAN turns the wind data into multiple modalities. The HGANN network consisting of a generator and discriminator predicts these modalities. The final wind speed prediction result can then be obtained by stacking the prediction results of all modalities.

2.2. CEEMDAN Model

Due to the high volatility of the wind speed series, CEEMDAN [30] is introduced to smooth the wind speed data. CEEMDAN decomposes a signal into some modalities.

The original wind speed series is defined as $X(n)$. CEEMDAN decomposes $X(n)$ into $\overline{IMF_j}(n), j = 1, 2, 3, \ldots J$ and residue $r_j(n)$. Figure 2 shows the flow chart of the CEEMDAN algorithm. The specific steps of the algorithm are as follows.

Randomly generate white noise with (0, 1), which is defined as $w^i(n)$, $i = 1, 2, \ldots I$. Define an operator $E\{*\}$, which generates the IMFs by EMD. We set the noise standard deviation to $\varepsilon = 0.2$ and the ensemble size to $I = 500$.

Add $w^i(n)$ to the $X(n)$ and generate a new series with noisy signals $X(n) + \varepsilon_0 w^i(n)$. For j= 1, the first-order $\overline{IMF_1}(n)$ that is decomposed by EMD is expressed as:

$$\overline{IMF_1}(n) = \frac{1}{I}\sum_{i=1}^{I} E\left\{X(n) + \varepsilon_0 w^i(n)\right\} \quad (1)$$

The first-order residue is computed as follows:

$$r_1(n) = X(n) - \overline{IMF_1}(n) \quad (2)$$

For $j = 2,\ 3, \ldots J$, calculate the $\overline{IMF_j}(n)$ and the jth residue as follows:

$$\overline{IMF_j}(n) = \frac{1}{I}\sum_{i=1}^{I} E\Big\{\ r_{j-1}(n) + \varepsilon_{j-1}w^i(n)\Big\} \tag{3}$$

$$r_j(n) = r_{j-1}(n) - \overline{IMF_j}(n) \tag{4}$$

Decompose $E\{r_{j-1}(n) + \varepsilon_{j-1}w^i(n)\}$ until the residue $r_j(n)$ cannot be decomposed and has only one extreme value. Then we can get $\overline{IMF_j}(n) = \frac{1}{I}\sum_{i=1}^{I} E\{\ r_{j-1}(n) + \varepsilon_{j-1}w^i(n)\}$ and the final residue $r_j(n) = X(n) - \sum_{j=1}^{J}\overline{IMF_j}(n)$.

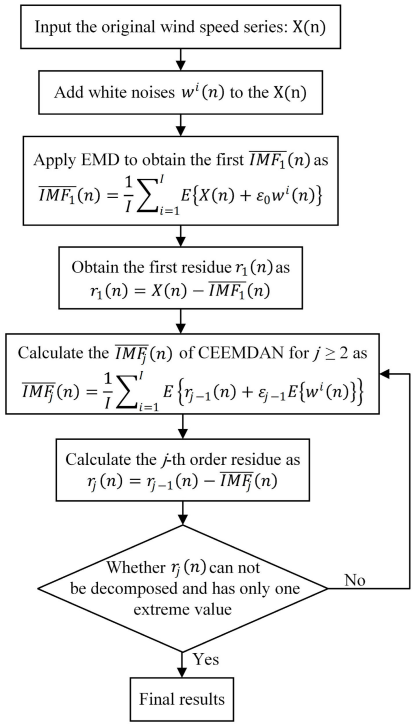

Figure 2. Flowchart of the CEEMDAN algorithm procedure.

The original wind speed time series $X(n)$ can be decomposed as $X(n) = \sum_{j=1}^{J}\overline{IMF_j}(n) + r_j(n)$, where $\overline{IMF_j}(n)$ or $r_j(n)$ can represent different features of the wind speed.

2.3. Generator OBLS for HGANN

BLS [31] can provide incremental structural learning. It achieves better forecasting results in time-series forecasting. Furthermore, because of its shallow network structure, BLS has higher error convergence performance than CNN. Compared with BLS, OBLS can provide both higher convergence performance and predictive accuracy. This is because OBLS uses IPSO to improve the network hyper-parameter of optimization. Therefore, OBLS has faster convergence and higher error convergence than CNN. Therefore, instead of using CNN as the generator of GAN, we use OBLS as the generator of HGANN to solve the problem of target confusion during HGANN training, which can improve the generalization ability and error convergence of HGANN, and thus make HGANN more suitable for wind speed prediction. The following is the detailed process of the OBLS algorithm.

Randomly generate n particles so that the dimensions of the particles are a three-dimensional vector $\{NF, NW, NE\}$ corresponding to the three parameters of BLS, respectively. Initialize the particle position $x_{id} \in (1, 100)$, and speed $v_{id} \in (-1,\ 1)$. Determine the learning factors $c_1 = 1.5$ and $c_2 = 1.5$, inertia weights $w_{max} = 1.0$ and $w_{min} = 0.4$, and the maximum number of iterations $iter_{max} = 100$.

Assume the input wind speed series data $X(n)$ and project the data using $\varnothing_i(X(n)W_{ei}+\beta_{ei})$ to represent ith mapped feature Z_i, where W_{ei} represents random weight with the proper dimensions. The jth group of enhancement nodes $\varnothing_j\left(Z_iW_{hj}+\beta_{hj}\right)$ is denoted as H_i. $\varnothing_i$ and $\varnothing_j$ can be different functions. The ith mappings can be denoted as:

$$Z_i = \varnothing_i(X(n)W_{ei}+\beta_{ei}), i = 1,2,\ldots n \tag{5}$$

The feature nodes are denoted as $Z^n \triangleq [Z_1, Z_2 \ldots Z_n]$, where W_{hj} and β_{hj} are random weights. The enhanced nodes are denoted as:

$$H_j = \varnothing_j\left(Z_iW_{hj}+\beta_{hj}\right), j = 1,2,\ldots m \tag{6}$$

Let $H^m \triangleq [H_1, H_2 \ldots H_m]$ where the symbol $\triangleq$ means "noted as"; then the output of the BLS can be denoted as:

$$Y = \{Z^n|H^m\}W^n \tag{7}$$

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

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

$$\{Z^n|H^m\}^+ = \lim_{\breve{}\to 0}\left\{\breve{}I + \{M\}\{M\}^T\right\}^{-1}\{M\}^T \tag{8}$$

where λ is l_2 regularization.

The IPSO [32] is introduced to iterate to optimize the parameters of BLS: $\{NF, NW, NE\}$. When the iteration of IPSO is consistently performed, the position and speed of the particles are continually updated through the following equation:

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2\left(p_{gd} - x_{gd}\right) \tag{9}$$

$$x_{id} = x_{id} + \gamma v_{id} \tag{10}$$

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

The generator takes the wind speed subsequence $\{x_1, x_2, \ldots x_n\}$ as input, which is generated by CEEMDAN. Then the generator generates a new wind speed sequence $\{y'_1, y'_2, \ldots y'_n\}$, which is statistically similar to the wind speed sequence $\{y_1, y_2, \ldots y_n\}$.

From Equations (8)–(10), OBLS does not require layer-to-layer coupling. Since there are no multi-layer connections, OBLS does not need to use gradient descent to update the weights, so the computational cost of OBLS is significantly lower than that of deep learning. When the accuracy of OBLS does not meet the requirements, its accuracy can be improved by increasing the "width" of the network nodes. Compared with the increase in the amount of calculation by increasing the number of layers in the deep network, the increase in that by increasing the "width" of the network nodes in OBLS is negligible.

2.4. Discriminators for HGANN

To maintain the stability of the generated samples, we used the discriminator of WGAN [33] as the discriminator of HGANN. In HGANN, the discriminator takes $\{x_i, y'_i\}$ or $\{x_i, y_i\}$ as input. The training goal of discriminator is to discriminate $\{x_i, y'_i\}$ as false and $\{x_i, y_i\}$ as true. The discriminator is trained by minimizing the distance function ($\mathcal{L}_D$) (loss function), which is defined as follows:

$$\mathcal{L}_D = \mathcal{L}(D(\{x_i, y_i\}), 1) + \mathcal{L}\left(D\left(\{x_i, y'_i\}\right), 0\right) + GP \tag{11}$$

$$P = \frac{\breve{}}{m} \sum_{i}^{m} \left[\nabla_{x_i, y_i'} D\left(x_i, y_i'\right)^i \right]^2 \tag{12}$$

here, $D(*)$ represents the output of the D; $\mathcal{L}$ is the binary cross entropy, defined as:

$$\mathcal{L} = -[klog(s) + (1-k)log(1-s)] \tag{13}$$

Based on this loss function, the discriminator can achieve an output of 1 when the input is $\{x_i, y_i\}$ and an output of 0 when the input is $\{x_i, y_i'\}$, and then discriminates the wind speed sequence $\{y_1, y_2, \ldots y_n\}$.

The discriminator outputs a scalar of "0" or "1." The scalar of "0" or "1" has two purposes: (1) It can influence and then adjust the weights of the neural network in the discriminator and maximize Equation (14) through a backpropagation algorithm. (2) It can be passed to the generator to assist the PSO algorithm to find the optimal hyper-parameters of the OBLS and then calculate the value of the fitness function F_c, which is defined as follows:

$$F_c = \frac{1}{n} \sum_{i=1}^{n} D(G(x(n))) \tag{14}$$

where $G(*)$ represents the output of the generator.

2.5. Prediction Steps of the Proposed HGANN Model

We propose the HGANN model for ultra-short-term wind speed prediction. The flow chart of the prediction process of the proposed model is shown in Figure 3. CEEMDAN is used to decompose the raw wind speed data $\{x_1, x_2, \ldots x_n\}$ into multiple modes $\overline{IMF_j}(n)$. These $\overline{IMF_j}(n)$ are separately sent into the generator (OBLS) of HGANN to obtain virtual samples $\{y_1', y_2', \ldots y_n'\}$. The discriminator (WGAN) takes $\{x_i, y_i'\}$ or $\{x_i, y_i\}$ as input and then outputs scalars "1" or "0." The scalars "1" or "0" are passed to the generator (OBLS) and the discriminator (WGAN) to participate in iterative model updates. Through the continuous iterative update, OBLS obtains the optimal value of $\{NF, NW, NE\}$. The final wind speed forecasting values $\{y_1, y_2, \ldots y_n\}$ can be obtained by stacking all forecast values.

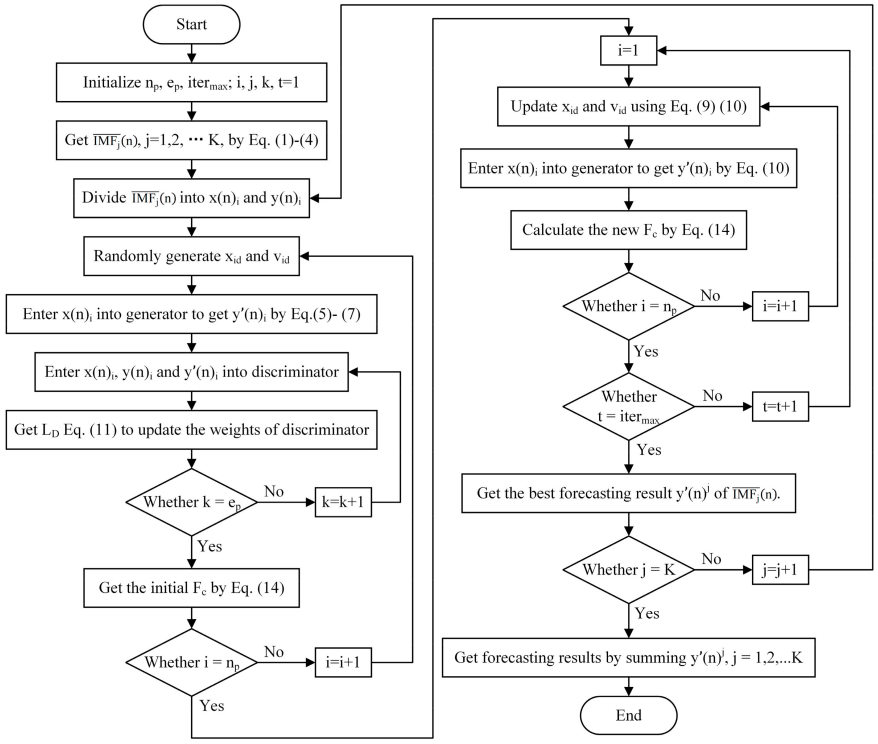

Figure 3. Flowchart of the proposed HGANN model prediction procedure.

3. Case Analysis

3.1. Data Description

To demonstrate the applicability of the proposed model in different locations, we used datasets from the 50Hertz wind farm in Germany and the Mahuangshan wind farm in China [26]: HER and MHS, respectively. The HER datasets are freely available at http://www.netztransparenz.de/ (accessed on 5 October 2021). Both data sets are recorded for one year and wind speeds are measured in 15 min intervals. We selected wind speed series from both HER and MHS datasets for March, June, September, and December, representing spring, summer, autumn, and winter, respectively. Experiments using the four wind speed series of spring, summer, autumn, and winter can verify the applicability of our model at different periods. Each series contains 2880 samples. Table 1 shows information on the selected wind speed data for spring, summer, autumn, and winter.

Table 1. Seasonal statistics of the wind speed data.

Season	Mean (m/s)	Median (m/s)	Max (m/s)	Min (m/s)	Standard Deviation (m/s)
HER data					
Spring	1.89	1.53	7.25	0.03	1.47
Summer	0.82	0.63	4.27	0	0.75
Autumn	1.23	0.83	6.65	0	1.19
Winter	1.90	1.60	5.68	0.07	1.35
MHS data					
Spring	5.78	5.70	16.50	0.40	2.25
Summer	5.46	5.45	12.50	0.40	2.21
Autumn	5.18	5.20	16.50	0.40	2.33
Winter	4.50	4.27	14.07	0	2.28

In our experiments, the first 80% of the wind speed sequence was used as the training set, and the rest was used as the test set for ultra-short-term wind prediction. Table 1 displays the information of the four datasets. The experiments were implemented in MATLAB R2021b on a 64-bit personal computer with Intel(R) core i5-9300 CPU/16.00 GB RAM.

3.2. Evaluation Index

To comprehensively evaluate the prediction performance of HGANN, four evaluate 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 reflect the accuracy of the prediction well. SSE represents the total error of the model. Their definitions are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{15}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{17}$$

$$SSE = \sum_{j=1}^{n}(y_i - \hat{y}_i)^2 \tag{18}$$

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

3.3. Comparable Methods

To verify the prediction performance of the proposed HGANN, it was compared with 10 advanced predictive models, involving PSO-ANFIS [34], VMD-GA-BP [35], EEMD-GPR-LSTM [36], EMD-ISSA-LSTM [37], MWS-CE-ENN [20], CNN [38], WGAN [39], BLS [31], OBLS, and WGAN-OBLS. Table 2 lists the parameter settings of six comparison methods. BLS, WGAN, PSO-BLS, and PSO-WGAN-OBLS are the same as those of HAGNN to perform the ablation experiment for HAGNN.

Table 2. Parameter settings of the models.

Model	Parameter Setting
PSO-ANFIS	$iter_{max} = 300$, $n_p = 40$, $c_1 = 1.0$, $c_2 = 2.0$, $n_r = 4$, $n_v = 52$
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	$e_p = 1000$, $l_r = 0.1$, $p_r = 0.000001$, $n_p = 40$, $n_i = 5$, $n_{b1} = 6$, $n_o = 1$, $iter_{max} = 100$, $n_{std} = 0.2$
EMD-ISSA-LSTM	$e_p = 100$, $l_r = 0.005$, $n_p = 10$, $iter_{max} = 20$, $v_e = 0.6$, $p_d = 0.7$, $p_e = 0.2$
Proposed Model	$n_{std} = 0.01, n_p = 40$, $iter_{max} = 100$, $c_1 = 1.5$, $c_2 = 1.5$, $e_p = 50$, $l_r = 0.002$, $\lambda = 10^{-30}$, $n_{b1} = 48$, $n_{b2} = 96$, $n_{b3} = 384$

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_i, and n_o are the number of rules, variables, input nodes, and output nodes, respectively; k is the decomposition number of VMD/EEMD; n_{bi} is the number of the ith hidden nodes; l_r is the learning rate of the network; p_r is the training requirement accuracy; n_{std} is the noise standard deviation in ICEEMDAN/CEEMDAN; v_e is the early warning value; p_d and p_s are the proportion of discoverers and sparrows aware of danger, respectively; and $\check{}$ is the regularization parameter for ridge regression.

3.4. Experimental Results

(1) Experiment I: Comparison Between Different Forecasting Methods

We experimentally verified the effectiveness and advancement of the proposed HGANN by comparing it with PSO-ANFIS, VMD-GA-BP, EEMD-GPR-LSTM, MWS-CE-ENN, and EMD-ISSA-LSTM. Considering that wind data characteristics show strong seasonality, experiments were conducted using wind series from multiple seasons to further validate the predictive performance of the model. We chose the HER and MHS datasets for March, June, September, and December for this experiment. The training and testing processes of each of the compared models were repeated 10 times. The experimental results of the different datasets are presented in Tables 3 and 4, where the first-best predictions are highlighted. Figure 4 depicts the wind speed prediction results of the proposed model for the HER dataset.

Table 3. Forecast results of different models for the HER data.

Season	Metrics	Proposed Model	PSO-ANFIS	VMD-GA-BP	EEMD-GPR-LSTM	MWS-CE-ENN	EMD-ISSA-LSTM
Spring	RMSE	**0.0065**	0.0092	0.0128	0.0105	0.0093	0.0131
	SSE	**0.0180**	0.0365	0.0817	0.0471	0.0370	0.0737
	MAPE	**0.0300**	0.0402	0.0732	0.0493	0.0632	0.0536
	MAE	**0.0048**	0.0068	0.0108	0.0083	0.0071	0.0096
Summer	RMSE	**0.0112**	0.0172	0.0134	0.0126	0.0152	0.0117
	SSE	**0.0537**	0.1279	0.0768	0.0679	0.0989	0.0586
	MAPE	**0.0420**	0.0641	0.0632	0.0596	0.0730	0.0500
	MAE	**0.0081**	0.0123	0.0099	0.0114	0.0119	0.0093

Table 3. *Cont.*

Season	Metrics	Proposed Model	PSO-ANFIS	VMD-GA-BP	EEMD-GPR-LSTM	MWS-CE-ENN	EMD-ISSA-LSTM
Autumn	RMSE	**0.0088**	0.0120	0.0137	0.0126	0.0132	0.0204
	SSE	**0.0331**	0.0615	0.0806	0.0676	0.0746	0.1779
	MAPE	**0.0422**	0.0514	0.0524	0.1177	0.0619	0.0851
	MAE	**0.0059**	0.0077	0.0092	0.0089	0.0082	0.0129
Winter	RMSE	**0.0063**	0.0090	0.0067	0.0093	0.0104	0.0086
	SSE	**0.0172**	0.0354	0.0190	0.0370	0.0463	0.0317
	MAPE	**0.0575**	0.0880	0.0683	0.0722	0.0923	0.0620
	MAE	**0.0043**	0.0064	0.0047	0.0069	0.0073	0.0056

Table 4. Forecast results of different models for the MHS data.

Season	Metrics	Proposed Model	PSO-ANFIS	VMD-GA-BP	EEMD-GPR-LSTM	MWS-CE-ENN	EMD-ISSA-LSTM
Spring	RMSE	**0.0237**	0.0344	0.0264	0.0276	0.0339	0.0291
	SSE	**0.2404**	0.5054	0.2977	0.3250	0.4918	0.3624
	MAPE	**0.0724**	0.0830	0.1007	0.1055	0.0990	0.0866
	MAE	**0.0146**	0.0244	0.0187	0.0198	0.0229	0.0203
Summer	RMSE	**0.0156**	0.0267	0.0179	0.0205	0.0189	0.0236
	SSE	**0.0104**	0.3045	0.1375	0.1799	0.1529	0.2384
	MAPE	**0.0433**	0.0857	0.0588	0.0697	0.0680	0.0765
	MAE	**0.0110**	0.0200	0.0136	0.0171	0.0149	0.0192
Autumn	RMSE	**0.0166**	0.0245	0.0190	0.0183	0.0214	0.0262
	SSE	**0.1179**	0.2570	0.1543	0.1433	0.1960	0.2937
	MAPE	**0.0410**	0.0641	0.0505	0.0439	0.0762	0.0658
	MAE	**0.0131**	0.0185	0.0149	0.0137	0.0170	0.0211
Winter	RMSE	**0.0266**	0.0394	0.0309	0.0325	0.0286	0.0312
	SSE	**0.3034**	0.6645	0.4075	0.4521	0.3501	0.4166
	MAPE	**0.0828**	0.1698	0.1540	0.1215	0.0937	0.1200
	MAE	**0.0190**	0.0284	0.0228	0.0276	0.0193	0.0257

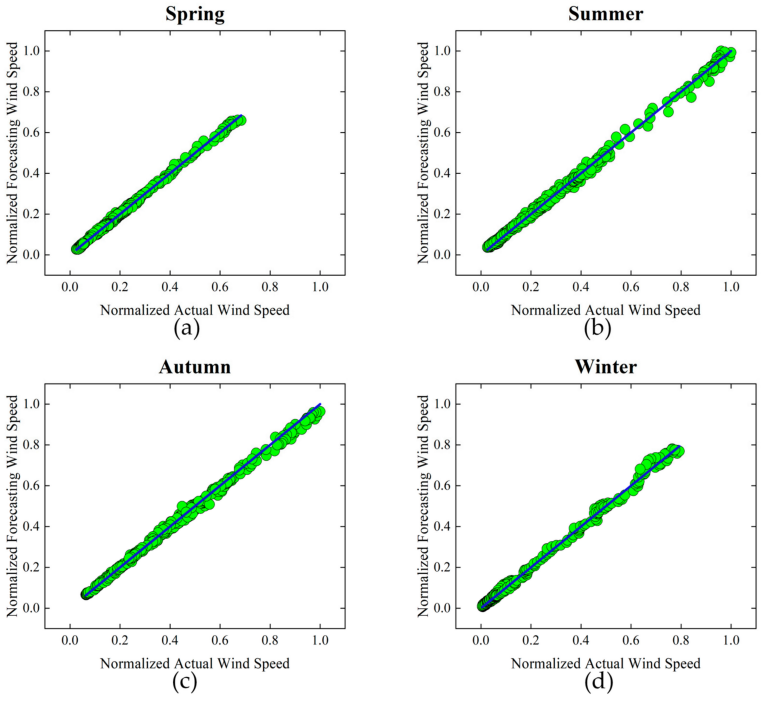

Figure 4. Forecasting results of HER wind speed data sets: (**a**) experiment results of spring wind speed sequences; (**b**) experiment results of summer wind speed sequences; (**c**) experiment results of autumn wind speed sequences; (**d**) experiment results of winter wind speed sequences.

Interestingly, it can be seen from Tables 3 and 4 that the proposed HGANN had the best prediction performance for each of the RMSE, SSE, MAPE, and MAE indicators on four datasets among all models. Wind speed forecasts at different times or at different locations may yield different results. Notably, our model achieved promising predictive results on both the geographically distinct German dataset HER and the Chinese dataset MHS. Furthermore, our model showed competitive prediction performance for wind series in different seasons. This indicates that our model can be extended to more general environments for wind speed prediction.

The abscissa and ordinate in Figure 4 represent the actual wind speed and the predicted wind speed, respectively; the blue line indicates that the predicted value is equal to the actual value. The ordinate of the green point is the predicted value, so the fit of the green point to the straight line reflects the accuracy of the prediction. As can be seen from Figure 4, the green points are very close to the blue line, which indicates that our model can predict wind speed effectively.

(2) Experiment II: Multi-Step Prediction Experiment

Multi-step forecasting can be built based on single-step forecasting. Compared to single-step forecasting, multi-step forecasting is more practical for power systems. Therefore, in wind speed prediction, multi-step prediction is of high practical value. The experiment aimed to demonstrate the predictive performance of the HGANN model in multi-step forecasting. We selected 2880 samples from 23 August to 22 September from the HER dataset for the one-step, two-step, and three-step experiments. Performance metrics involved RMSE, SSE, MAPE, and MAE. Benchmark models covered PSO-ANFIS, VMD-GA-BP, EEMD-GPR-LSTM, MWS-CE-ENN, and EMD-ISSA-LSTM. The training and testing processes of each model were repeated 10 times. The experimental results of the proposed model and the benchmark models are shown in Table 5.

Table 5. Multi-step prediction results for 15 min wind speed.

Model	RMSE			SSE			MAPE			MAE		
	1-Step	**2-Step**	**3-Step**	**1-Step**	**2-Step**	**3-Step**	**1-Step**	**2-Step**	**3-Step**	**1-Step**	**2-Step**	**3-Step**
Proposed	**0.0091**	**0.0136**	**0.0178**	**0.0356**	**0.0679**	**0.1361**	**0.0463**	**0.0712**	**0.1087**	**0.0061**	**0.0098**	**0.0134**
PSO-ANFIS	0.0120	0.0174	0.0238	0.0615	0.1299	0.2428	0.0514	0.0885	0.1270	0.0077	0.0119	0.0161
VMD-GA-BP	0.0137	0.0151	0.0203	0.0806	0.0976	0.1764	0.0524	0.0721	0.1123	0.0092	0.0101	0.0213
EEMD-GPR-LSTM	0.0126	0.0164	0.0192	0.0676	0.1151	0.1578	0.1177	0.1353	0.1536	0.0089	0.0129	0.0157
MWS-CE-ENN	0.0132	0.0176	0.0213	0.0746	0.1326	0.1942	0.0619	0.0826	0.1121	0.0082	0.0172	0.0224
EMD-ISSA-LSTM	0.0109	0.0139	0.0183	0.0508	0.0827	0.1433	0.0503	0.0919	0.1325	0.0086	0.0134	0.0192

From Table 5, it can be seen that the one-step, two-step, and three-step prediction results of the proposed HGANN model provided lower RMSE, SSE, MAPE, and MAE values than those of the benchmark models. For instance, the proposed HGANN provided 0.0091 (one-step), 0.0136 (two-step), and 0.0178 (three-step) on RMSE, compared with EMD-ISSA-LSTM, which provided the predictive results of 0.0109 (one-step), 0.0139 (two-step), and 0.0183 (three-step). Furthermore, we also provide clear visual results of multi-step predictions for the six models in Figure 5. The results from Table 5 and Figure 5 indicate that the proposed HGANN model had the best robustness and the highest wind speed prediction accuracy among all compared models.

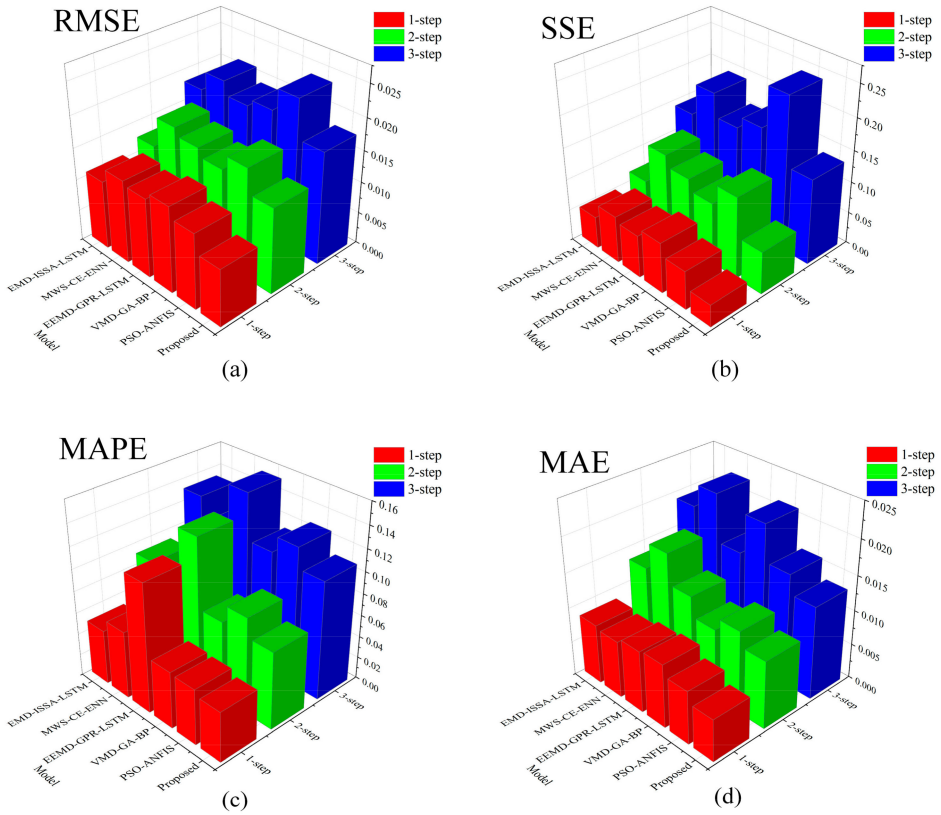

Figure 5. Multi-step forecasting experiments under RMSE, SSE, MAPE, and MAE indicators: (**a**) experiment results on the RMSE indicator; (**b**) experiment results on the SSE indicator; (**c**) experiment results on the MAPE indicator; (**d**) experiment results on the MAE indicator.

(3) Experiment III: Ablation Experiment Between Single Models and Hybrid Models.

To verify the rationality of the proposed HGANN model, it was compared with WGAN-OBLS, OBLS, WGAN, BLS, and CNN on the HER dataset. The generator of HGANN is OBLS and its discriminator is the discriminator of WGAN. To emphasize the effectiveness of OBLS, it was compared with the generator of WGAN, namely, CNN. Similarly, all compared models were repeatedly trained and tested 10 times. In the HER dataset, 2880 data from 23 August to 22 September were selected for this experiment. The experimental results are shown in Table 6, where the first-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 6. Forecasting performances of the proposed model and reference models.

Indicators	Proposed Model	WGAN-OBLS	OBLS	WGAN	CNN	BLS
RMSE	**0.0088**	0.0119	0.0122	0.0138	0.0221	0.0308
SSE	**0.0331**	0.0603	0.0642	0.0819	0.2090	0.4058
MAPE	**0.0422**	0.0506	0.0546	0.0571	0.0601	0.0648
MAE	**0.0059**	0.0076	0.0080	0.0086	0.0161	0.0164

Figure 6 shows that among all the compared models, our proposed model had the best curve fitting and the smallest predicted error. The suggested model consistently outperformed WGAN-OBLS, PSO-BLS, WGAN, OBLS, CNN, and BLS, as shown in Table 6. This further demonstrates the advantages of our proposed model, as it combines CEEMDAN, OBLS, and WGAN.

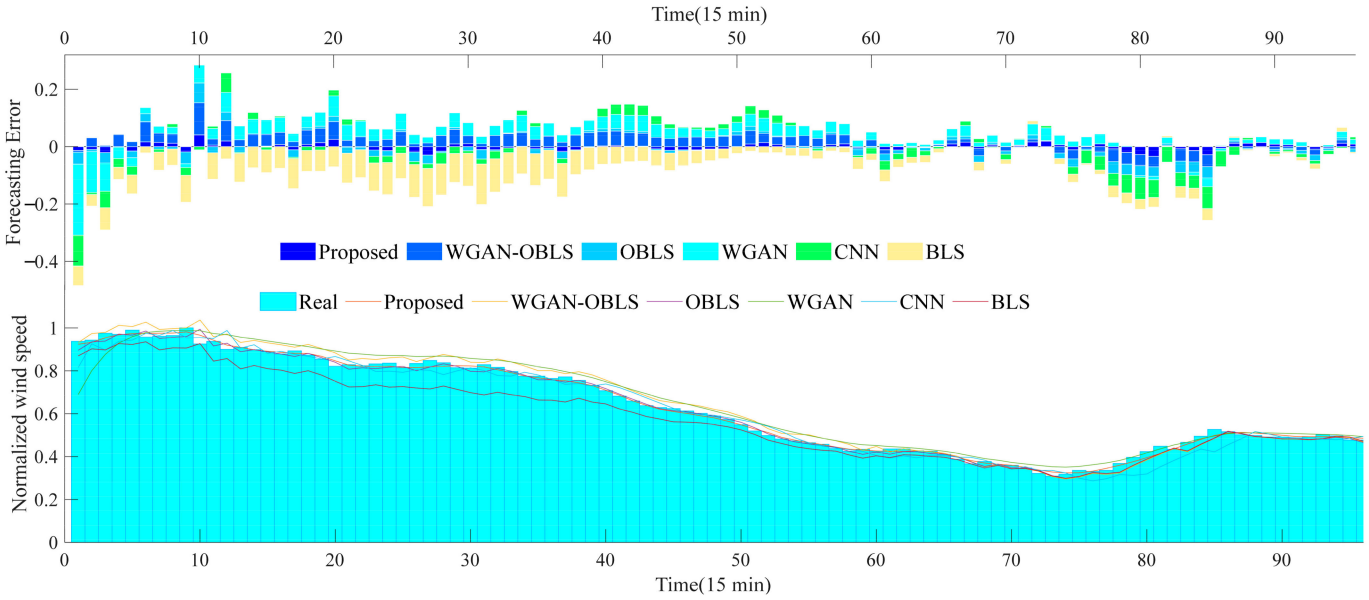

Figure 6. Forecasting results in HER dataset (22 September 2019).

Furthermore, first, compared with WGAN-OBLS without CEEMDAN, the proposed model had better predictive performance due to its covering CEEMADN and WGAN-OBLS, thus showing the effectiveness of CEEMDAN. Second, compared with WGAN or OBLS, WGAN-OBLS provided better predictive performance due to combing OBLS and WGAN, thus showing the effectiveness of OBLS and WGAN in WGAN-OBLS. Third, compared with BLS, OBLS provided better predictive performance due to using the improved PSO, thus showing the effectiveness of PSO in OBLS. Fourth, OBLS had better predictive results compared to the generator of WGAN, namely, CNN. This demonstrates the advantage of OBLS over CNN as a generator. This may be due to the flexible structure and better error convergence of OBLS.

4. Discussion

Our model was compared with five advanced models to evaluate its performance and advantages in various wind sequence experiments. Experimental results show that the proposed model had better predictive performance. The reasons behind this fact are given as follows.

First, the wind speed data were one-year data from wind farms in Germany and China, which cover complex fluctuation characteristics. Therefore, our HGANN model uses CEEMDAN to smoothen the volatility of the data and improve the predictive performance.

Second, HGANN uses OBLS as the generator to provide a special shallow broad incremental learning network structure, which can not only be beneficial for improving prediction accuracy for one-dimensional wind speed prediction compared to CNN but also greatly decrease computational cost using pseudo-inverse operations to determine the network weights instead of using convolution operations.

Third, in our HGANN model, the proposed OBLS uses an improved PSO to optimize the hyper-parameters of its network, which can search in a wider range and obtain the optimal parameters over BLS. Therefore, OBLS has better generalization ability than BLS.

Finally, HGANN can better extract the deeper features of wind speed data by playing a minimum–maximum game between the generator and discriminator for wind speed prediction.

5. Conclusions

Although existing various hybrid predictive models have provided competitive performance in ultra-short-term wind speed prediction, they still need to be further improved—

for instance, how to effectively reduce the computational cost of hybrid predictive models, and how to effectively deal with the multicollinearity problem of the hybrid forecasting model based on weighted strategy, which leads to the problem of reduced forecasting accuracy. To enhance the predictive power and decrease the computational cost, this paper proposes the HGANN model for ultra-short-term wind speed forecasting. HGANN is a generative adversarial network in which the generator and discriminator play against each other to obtain wind speed predictions with high accuracy. In HGANN, we developed OBLS and the convolutional structures as the generator and the discriminator, respectively, which enables them to obtain effective synergies to improve predictive performance. Particularly, OBLS involves a special shallow broad incremental learning network structure, which can effectively deal with one-dimensional wind speed data. Furthermore, the shallow network structure of OBLS can also significantly decrease computational cost via using pseudo-inverse operations rather than convolution operations. In addition, the proposed OBLS applies an improved PSO to obtain the optimal network hyper-parameters. CEEMDAN performs noise reduction and decomposition of the wind data. Via the above rational combination, the proposed HGANN provides high predictive accuracy and generalization ability with low computational cost in ultra-short-term wind speed prediction. The experimental results indicate the above fact. For instance, the RMSE predictive errors of the proposed model were 29.35%, 49.22%, 38.09%, and 30.10% compared to the four state-of-art predictive models PSO-ANFIS, VMD-GA-BP, EEMD-GPR-LSTM, and MWS-CE-ENN on the spring wind data of the HER dataset, respectively. In the future, we plan to use parallel computing to speed up the process of PSO optimization of BLS during training. Furthermore, the proposed HGANN will be extended to a wider range of applications, such as financial time-series forecasting, electricity-load forecasting, traffic forecasting, etc.

Author Contributions: Conceptualization, Q.W. and L.H.; methodology, Q.W.; software, J.H.; validation, Q.W., J.H. and L.H.; formal analysis, L.H.; investigation, Q.W. and Q.L.; resources, J.H.; data curation, Q.W. and Q.L.; writing—original draft preparation, L.H.; writing—review and editing, L.C. and Y.L.; visualization, 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 grant 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

ANFIS	Adaptive-network-based fuzzy inference system
BLS	Broad learning system
BP	Back propagation
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
GA	Genetic algorithm
GMDH	Group method of data handling neural network
GNN	Graph neural network
GAN	Generative adversarial network
GPR	Gaussian process regression
ICEEMDAN	Improved CEEMDAN

IMF	Intrinsic mode function
IOWA	Induced ordered weighted averaging
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MTO	Multi-tracker optimizer
PSO	Particle swarm optimization
OBLS	Optimized broad learning system
RBF	Radial basis function
RMSE	Root mean square error
SSE	Sum of squared error
SVR	Support vector regression
VMD	Variational mode decomposition
WGAN	Wasserstein generative adversarial network

References

1. Zhen, Z.; Qiu, G.; Mei, S.; Wang, F.; Zhang, X.; Yin, R.; Li, Y.; Osório, G.J.; Shafie-khah, M.; Catalão, J.P.S. An ultra-short-term wind speed forecasting model based on time scale recognition and dynamic adaptive modeling. *Int. J. Electr. Power Energy Syst.* **2022**, *135*, 107502. [CrossRef]
2. Wang, F.; Tong, S.; Sun, Y.; Xie, Y.; Zhen, Z.; Li, G.; Cao, C.; Duić, N.; Liu, D. Wind process pattern forecasting based ultra-short-term wind speed hybrid prediction. *Energy* **2022**, *255*, 124509. [CrossRef]
3. 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]
4. 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]
5. Yunus, K.; Thiringer, T.; Chen, P. ARIMA-Based Frequency-Decomposed Modeling of Wind Speed Time Series. *IEEE Trans. Power Syst.* **2016**, *31*, 2546–2556. [CrossRef]
6. Wang, Y.; Wang, H.; Srinivasan, D.; Hu, Q. Robust functional regression for wind speed forecasting based on Sparse Bayesian learning. *Renew. Energy* **2019**, *132*, 43–60. [CrossRef]
7. 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]
8. Zhang, Y.; Zhang, C.; Zhao, Y.; Gao, S. Wind speed prediction with RBF neural network based on PCA and ICA. *J. Electr. Eng.* **2018**, *69*, 148–155. [CrossRef]
9. Son, Y.; Yoon, Y.; Cho, J.; Choi, S. Cloud Cover Forecast Based on Correlation Analysis on Satellite Images for Short-Term Photovoltaic Power Forecasting. *Sustainability* **2022**, *14*, 4427. [CrossRef]
10. Hameed, S.S.; Ramadoss, R.; Raju, K.; Shafiullah, G.M. A Framework-Based Wind Forecasting to Assess Wind Potential with Improved Grey Wolf Optimization and Support Vector Regression. *Sustainability* **2022**, *14*, 4235. [CrossRef]
11. Khodayar, M.; Wang, J.; Manthouri, M. Interval Deep Generative Neural Network for Wind Speed Forecasting. *IEEE Trans. Smart Grid* **2019**, *10*, 3974–3989. [CrossRef]
12. Beigi, M.; Beigi Harchegani, H.; Torki, M.; Kaveh, M.; Szymanek, M.; Khalife, E.; Dziwulski, J. Forecasting of Power Output of a PVPS Based on Meteorological Data Using RNN Approaches. *Sustainability* **2022**, *14*, 3104. [CrossRef]
13. Khodayar, M.; Wang, J. Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting. *IEEE Trans. Sustain. Energy* **2019**, *10*, 670–681. [CrossRef]
14. 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* **2020**, *11*, 509–523. [CrossRef]
15. Zhang, Y.; Pan, G.; Zhang, C.; Zhao, Y. Wind speed prediction research with EMD-BP based on Lorenz disturbance. *J. Electr. Eng.* **2019**, *70*, 198–207. [CrossRef]
16. Santhosh, M.; Venkaiah, C.; Kumar, D.M.V. Short-term wind speed forecasting approach using Ensemble Empirical Mode Decomposition and Deep Boltzmann Machine. *Sustain. Energy Grids Netw.* **2019**, *19*, 100242. [CrossRef]
17. Wang, J.; Yang, Z. Ultra-short-term wind speed forecasting using an optimized artificial intelligence algorithm. *Renew. Energy* **2021**, *171*, 1418–1435. [CrossRef]
18. 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* **2015**, *6*, 236–244. [CrossRef]
19. Li, Y.; Chen, X.; Li, C.; Tang, G.; Gan, Z.; An, X. A Hybrid Deep Interval Prediction Model for Wind Speed Forecasting. *IEEE Access* **2021**, *9*, 7323–7335. [CrossRef]
20. Tian, Z. Modes decomposition forecasting approach for ultra-short-term wind speed. *Appl. Soft Comput.* **2021**, *105*. [CrossRef]
21. 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]

22. Cui, Y.; Huang, C.; Cui, Y. A novel compound wind speed forecasting model based on the back propagation neural network optimized by bat algorithm. *Environ. Sci. Pollut. Res. Int.* **2020**, *27*, 7353–7365. [CrossRef]
23. 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]
24. 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* **2021**, *165*, 573–594. [CrossRef]
25. 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]
26. 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]
27. Wang, J.; Wang, Y.; Li, Z.; Li, H.; Yang, H. A combined framework based on data preprocessing, neural networks and multi-tracker optimizer for wind speed prediction. *Sustain. Energy Technol. Assess.* **2020**, *40*, 100757. [CrossRef]
28. 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.* **2021**, *100*, 106996. [CrossRef]
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M. Generative adversarial nets. In Proceedings of the International Conference on Neural Information Processing Systems, Kuching, Malaysia, 8–13 December 2014; pp. 2672–2680.
30. Torres, M.; Colominas, M.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 4144–4147. [CrossRef]
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. Zhang, Y.; Sun, H.; Guo, Y. Wind Power Prediction Based on PSO-SVR and Grey Combination Model. *IEEE Access* **2019**, *7*, 136254–136267. [CrossRef]
33. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875v3.
34. 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]
35. 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]
36. 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]
37. Tian, Z.; Chen, H. A novel decomposition-ensemble prediction model for ultra-short-term wind speed. *Energy Convers. Manag.* **2021**, *248*, 114775. [CrossRef]
38. Huang, L.; Li, L.; Wei, X.; Zhang, D. Short-term prediction of wind power based on BiLSTM–CNN–WGAN-GP. *Soft Comput.* **2022**, *26*, 1–15. [CrossRef]
39. Wang, H.; Yi, H.; Peng, J.; Wang, G.; Liu, Y.; Jiang, H.; Liu, W. Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network. *Energy Convers. Manag.* **2017**, *153*, 409–422. [CrossRef]