



Article Wind Speed Prediction for Offshore Sites Using a Clockwork Recurrent Network

Yuxuan Shi * D, Yanyu Wang and Haoran Zheng

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; wangyanyu@shu.edu.cn (Y.W.); zhrzhr@shu.edu.cn (H.Z.)

* Correspondence: shiyuxuan@shu.edu.cn

Abstract: Offshore sites show greater potential for wind energy utilization than most onshore sites. When planning an offshore wind power farm, the speed of offshore wind is used to estimate various operation parameters, such as the power output, extreme wind load, and fatigue load. Accurate speed prediction is crucial to the running of wind power farms and the security of smart grids. Unlike onshore wind, offshore wind has the characteristics of random, intermittent, and chaotic, which will cause the time series of wind speeds to have strong nonlinearity. It will bring greater difficulties to offshore wind speed predictions, which traditional recurrent neural networks cannot deal with for lacking in long-term dependency. An offshore wind speed prediction method is proposed by using a clockwork recurrent network (CWRNN). In a CWRNN model, the hidden layer is subdivided into several parts and each part is allocated a different clock speed. Under the mechanism, the long-term dependency of the recurrent neural network can be easily addressed, which can furthermore effectively solve the problem of strong nonlinearity in offshore speed winds. The experiments are performed by using the actual data of two different offshore sites located in the Caribbean Sea and one onshore site located in the interior of the United States, to verify the performance of the model. The results show that the prediction model achieves significant accuracy improvement.



Citation: Shi, Y.; Wang, Y.; Zheng, H. Wind Speed Prediction for Offshore Sites Using a Clockwork Recurrent Network. *Energies* **2022**, *15*, 751. https://doi.org/10.3390/en15030751

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

Received: 2 January 2022 Accepted: 15 January 2022 Published: 20 January 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/). Keywords: clockwork recurrent network; offshore site; strong nonlinearity; wind speed prediction

1. Introduction

With the increasingly severe global climate problem, the sustainability of traditional fossil fuels is facing huge challenges, and the development of renewable energy (RE) is becoming inevitable [1]. RE, including wind energy, geothermal energy, and solar energy, cannot only reduce carbon emissions, but also achieve sustainable development [2,3]. As one form of RE, wind energy is widely used around the world on account of its wide distribution, huge reserves, and environmental friendliness [4]. At the same time, wind power is also one of the most commercially viable and dynamic RE sources due to its low cost and permanent nature. On account of its relatively mature technology and commercial conditions for large-scale development, wind energy has been the fastest growing energy source in recent years. [5]. According to the data from the Global Wind Energy Council, global wind power is accelerating its deployment, driven by the carbon-neutral trend. The latest data show that the total global wind power bidding volume in the first quarter of 2021 is 6970 MW, 1.6 times that of the same period last year [6].

However, wind energy resources are susceptible to environmental changes, such as geography, climate, and seasons. It brings great difficulties to wind power utilization. In addition, the ecological problem with wind power is that it may disturb birds. Therefore, accurate offshore wind speed prediction is of great help to the development of wind power. However, there are still some factors that affect the prediction accuracy, among which the major challenge is historical data. Regrettably, potential offshore sites have not had enough records of wind speed for various reasons in the past. Consequently, it is a major

technical challenge for risk assessment using only short-term records of historical wind speed data. Nevertheless, unlike onshore wind, offshore wind has the characteristics of random, intermittent, and chaotic, which will cause the time series of wind speeds to have strong nonlinearity [7], inevitably bringing greater difficulties to offshore wind speed predictions.

Within past studies, scholars have proposed various wind speed prediction methods. There are three main categories, including physical models, statistical models, and machine learning models. Physical models make predictions by monitoring the terrain, climate, and other factors. Among the physical models, numerical weather prediction (NWP) is a commonly used model that simulates physical interactions in the atmosphere based on conservation equations (kinetic energy, potential energy, and mass) [8,9]. However, different locations and fields bring about variability in the NWP models and their model resolutions. The resolution of the model data seriously affects the prediction accuracy and the datasets are hard to obtain [10]. Statistical models mainly use historical data to make predictions. The commonly used statistical models are Gaussian process regression (GPR) [11,12], autoregressive (AR) [13], autoregressive moving average (ARMA) [14], autoregressive integral moving average (ARIMA) [15], and seasonal ARIMA [16]. However, when the nonlinear characteristics are prominent, the prediction performance of these models decreases significantly [17]. Comparatively, machine learning is often performed to predict wind speed because of its ability to fit stronger nonlinearity, which includes the multi-layer perceptron (MLP) [18], back propagation neural network (BPNN) [19], radial basis function neural network (RBFNN) [20], support vector machine (SVM)/support vector regression (SVR) [21–26], echo state network [27], deep belief networks [28], and convolutional neural network (CNN) [29]. However, these models still have various problems in their application, such as getting stuck in local optimum solutions, overfitting, and low convergence rates.

Recently, the recurrent neural network (RNN) is proposed to model sequential data or time series data [30]. RNN, as a type of artificial neural network that uses a simple but elegant mechanism, addresses the drawback of vanilla neural networks and keeps the characteristic of the autoregressive model. It brings to RNN the ability to solve the nonlinear problem of time series data. Therefore, RNNs achieve great performances when modeling sequential data and have become one of the most valuable breakthroughs in deep learning model preparation in recent decades. Meanwhile, many studies on wind speed prediction have emerged in recent years, which use RNN models [30,31] or hybrid RNN models [32–36]. At the same time, researchers constantly optimized the network structure of the RNN to improve its performance. Several new models based on RNNs, such as long and short term memory networks (LSTMs) [37–48], bidirectional LSTM (BiLSTM) [49], gated recurrent units (GRUs) [50], clockwork recurrent neural networks (CWRNNs) [51], and dilated recurrent neural networks (DRNNs) [52], have been proposed to solve problems of RNN, including vanishing gradients and the long-term dependency, and improve the performance of RNNs.

CWRNN, which adopts a special mechanism to solve problems of simple RNNs and contains an even smaller number of parameters than simple RNNs, was proposed in 2014 [53]. CWRNN breaks up neurons in the hidden layer into different parts, and neurons in the same part work at a given clock speed. At the same time, only a few parts are activated. It makes CWRNN have a certain memory mechanism that can solve the long-term dependency problem. Additionally, it has shown better performances than common RNNs and even LSTM in various tasks. Xie et al. applied CWRNN to muscle perimysium segmentation. They utilized CWRNN to handle biomedical data, and experiment results show that CWRNN outperforms the other machine learning models [54]. Feng et al. used CWRNN to estimate the state-of-charge of lithium batteries and showed that this method achieves impressive results [51]. Lin et al. proposed a trajectory generation method for unmanned vehicles based on CWRNN. The performance of the CWRNN method is verified by experiments. The study also compared CWRNN with LSTM in several

metrics [55]. Achanta et al. investigated CWRNN for statistical parametric speech synthesis. The experimental results show that the architecture of the CWRNN is equivalent to the RNN with LI units, and outperforms the RNN with dense initialization and LI units [56]. Presently, the methods based on CWRNN have been used in various fields, such as speech recognition and stock prediction [57]. As far as we know, it has not been used in wind speed prediction.

To solve the strong nonlinear problem and achieve a higher prediction accuracy, an offshore wind speed prediction method is proposed, which is based on the CWRNN. In the proposed method, the hidden layer is subdivided into several parts and each part is allocated a different clock speed. Under the mechanism, the long-term dependency of RNNs can be easily addressed. The trained CWRNN model can output an instantaneous prediction for data from the previous sampling step. The experiments are performed to validate the performance of the model by the actual wind speed data of two different offshore sites and one onshore site.

The main contributions of this study are as follows:

- An offshore wind speed prediction method is proposed based on the CWRNN. Compared with the other RNNs, the CWRNN adopts a special mechanism to solve longterm dependency. The experiments prove that the method can effectively solve the problem of strong nonlinearity in offshore wind speed, and improve the prediction accuracy by over 38% in terms of the different kinds of evaluation criteria compared with simple RNNs.
- The method fully exploits the ability of RNNs to solve nonlinear problems with time series data. Compared with the traditional machine learning models, the proposed method keeps the characteristics of the autoregressive model, which improves the performance in prediction accuracy.
- Hyperparameters, such as the number of network parts that are the key influencing factors of the model, and the different part periods are thoroughly analyzed, which seriously affect the performance of predicting the offshore wind speed.

The rest of the paper is organized as follows: Section 2 introduces the related theory; Section 3 describes the overall implementation process of this method; Section 4 presents the experiment results; the results are discussed in Section 5; and Section 6 summarizes the whole paper.

2. Theoretical Background

There is an inherent concept of sequential data that incrementally progresses over time. As we all know, traditional neural networks (NNs) are good at solving nonlinear problems and perform well in most cases. However, they lack the inherent trend for the persistence of sequential data. For example, a simple feedforward NN cannot really understand the meaning of a sentence according to the order of input data in the context. The RNNs settle the shortcomings of the original NNs with an ingenious mechanism, which gives them the advantage in time modeling. This section provides a brief overview of the RNN, LSTM, and CWRNN.

2.1. RNN

RNN is a specific NN that is designed to model sequential data or time- series data. The principle of RNN is to feed the output of the previous layer back to the input of the next layer, which gives RNN the ability to predict the output of the layer. In the RNN, the neurons in different layers of the NN are compressed into a single layer, as shown in Figure 1.



Figure 1. The structure of a simple RNN.

As seen in Figure 1, at time *t*, the input is a combination of the input at *t* and the output at a previous time, t - 1. This feedback mechanism improves the output of the time step *t*. The calculation formula for output y_t^O at time step *t* is:

$$y_t^H = f_H \Big(W_H \cdot y_{t-1}^H + U_I \cdot x_t \Big) \tag{1}$$

$$y_t^O = f_O \left(W_O \cdot y_{t-1}^H \right) \tag{2}$$

where W_H , U_I , W_O are the weight matrices of the hidden layers, input layer, and output layer; x_t is defined as the input vector at t; and y_t^H and y_{t-1}^H are defined as the hidden neurons at different times. $f_H(\cdot)$ and $f_O(\cdot)$ are defined as different activation functions. Here, the biases of the neurons are omitted.

RNNs must use a context when making predictions and, in this case, must also learn the required context. The shortcoming of the RNN is that, when training the model, the gradient can easily vanish or explode, which is mainly because of the lack of long-term dependency. Researchers proposed some techniques to solve the problems, such as LSTM, which uses a gate mechanism.

2.2. LSTM

LSTM, as a special type of RNN, can keep long-term information from the input sequence, which makes up for the difficulties of RNN in learning long-term information, and solves the problems of RNN gradient disappearance and gradient explosion. The framework of the LSTM unit is shown in Figure 2. LSTM and RNN have the same chain structures, but their repeating modules are different. Unlike the repeating module in a standard RNN that contains a single layer, LSTM has multiple layers of neurons. These neurons constitute the forgetting gate, the input gate, and the output gate of LSTM. The status updates and output updates for the three gates are described below.



Figure 2. The structure of LSTM.

Forgetting gate: this gate control unit determines how much information the cell state discards. The status update, f_t , of the forgetting gate at the time, t, is as follows:

$$f_t = f_O \left(W_f \cdot y_{t-1}^H + U_I^f \cdot x_t \right)$$
(3)

where W_f is defined as the weight matrix of the forgetting gate, and U_I^f is defined as the weight matrix between the hidden layer of the forgetting gate and the input layer.

Input gate: this gate control unit determines to what extent the input information, x_t , at the current moment is added to the memory cell stream. The status update, i_t , of the input gate is as follows:

$$i_t = f_O \left(W_i \cdot y_{t-1}^H + U_I^i \cdot x_t \right) \tag{4}$$

where W_i is defined as the weight matrix of the input gate, and U_I^i is the weight matrix between the hidden layer of the input gate and the input layer.

After the work of the input gate and the forgetting gate is completed, the state of the memory cells, c_t , is updated as follows:

$$\widetilde{c}_t = f_H \Big(W_c \cdot y_{t-1}^H + U_I^c \cdot x_t \Big)$$
(5)

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \widetilde{c_t} \tag{6}$$

where W_c represents the weight matrix of the memory cells, and U_I^c is the weight matrix between the hidden layer of the memory cells and the input layer.

Output gate: after the internal memory cell state is updated, the output gate controls how much memory can be used in the network update at the next moment. The state update, o_t , of the output gate at the time, t, is as follows:

$$o_t = f_O \Big(W_o \cdot y_{t-1}^H + U_I^o \cdot x_t \Big) \tag{7}$$

where W_o is defined as the weight matrix of the output gate; U_I^o is the weight matrix between the hidden layer of the output gate and the input layer; and b_o represents the offset.

Finally, the network output at moment *t* is:

$$y_t^H = o_t \cdot f_H(c_t) \tag{8}$$

$$y_t^O = f_O\left(W_O \cdot y_t^H\right) \tag{9}$$

To alleviate the gradient exploding and vanishing problems, an LSTM block that embeds three gates into the hidden neurons of the RNN is generally applied to process the time series data, and achieves a good result in most cases. It is easier to understand that the complex network structure increases the stability and ability of the model. However, it also makes the network computationally more expensive. Meanwhile, the performance of the complex deep learning neural network models, especially LSTMs, depends on the quantity and diversity of the data.

2.3. CWRNN

The structure of the CWRNN is close to that of a simple RNN with three layers. The difference between these two models is that the CWRNN divides the neurons of the hidden layers into *n* parts; each part has a clock speed, T_i , where $T_i \in \{T_1, T_2, \dots, T_n\}$. Therefore, each part handles the input data at a different frequency, as shown in Figure 3. The parts with a long clock speed can handle long-term information, and the parts with a short clock speed are used to handle the continuous information.





 W_H and W_i are defined as the weight matrices of the hidden and input layers, respectively, which are divided into *n* blocks. At the same time, W_H is also an upper triangular matrix, as shown in Figure 4. At any time step, *t*, only the related rows of the work parts W_H and W_i are activated. Then, the output vector, y_H , was updated in the same way. The other parts keep the output values unchanged. The update mechanism is shown in Figure 4.



Figure 4. Update process of the hidden units at t = 6.

$$W_{H} = \begin{pmatrix} W_{H_{1}} \\ \vdots \\ W_{H_{n}} \end{pmatrix} \qquad W_{i} = \begin{pmatrix} W_{I_{1}} \\ \vdots \\ W_{I_{n}} \end{pmatrix}$$
(10)

$$W_{H_i} = \begin{cases} W_{H_i} & \text{for } (t \text{ MOD } T_i) = 0\\ 0 & otherwise \end{cases}$$
(11)

Therefore, the parts with a long clock-speed handle the long-term information, and the parts with a short clock-speed handle the continuous information. The two parts are independent of each other and work well.

Having the same number of hidden neurons, the CWRNN processes much faster than a simple RNN, because only the corresponding parts are updated at each step. In the case of this exponential clock setting, when n > 4, the CWRNN can run faster than the RNN, which has the same neurons [53].

3. Framework of the Prediction Method

3.1. *The Procedure*

The framework of the proposed method is described in Figure 5. The procedure is divided into four steps.



Figure 5. The procedure of the proposed method.

Step 1: data processing. Wind speed raw data are normalized to [0, 1] at first, then preprocessed to the format required for the CWRNN model.

Step 2: model setting. The hyperparameters are set to fit the model, including the hidden layer parts, length of series input, and number of neurons. The influence of these hyperparameters will be discussed later, in detail.

Step 3: train model. For model training, we used a mini-batch stochastic gradient descent and Adam optimizer to minimize the mean square error (MSE) for the prediction vectors. The parameters can be trained through the back propagation of standard error.

Step 4: model test. Some prediction and evaluation indexes of the training model, such as the mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and coefficient of determination (R²), are performed to verify the prediction performance.

3.2. Dataset

The experimental datasets are from three wind speed measure sites, among which two are located offshore in the Virgin Islands, between the Atlantic Ocean and the Caribbean Sea, and the other onshore site is located in Humeston, Iowa, U.S.A. [58,59]. This study first conducts experiments on two offshore wind speed datasets to verify the proposed model, and then conducts experiments on the onshore wind speeds to verify the generalization of the model. Three data sets and their division in the model are described in Figure 6. The data are collected from 2012–2014. The sampling period in the data set is 10 min and each dataset has 3000 points. Table 1 shows the data of the wind speed at three different locations. It depicts the minimum, average, maximum, and standard deviation values (Stdev).



Figure 6. Datasets of Site1, Site2, Site3, and the data segmentation method.

Table 1. Data statistics on the wind speed at the three locations.

Site	Metrics					
one	Average(m/s)	Maximum(m/s)	Minimum(m/s)	Stdev (m/s)		
Site1	5.6655	11.7630	0.3600	2.0553		
Site2	7.4647	14.4030	1.8014	1.7486		
Site3	9.1397	17.4560	0.3870	3.3416		

3.3. Evaluation Metrics

To quantitatively describe the performance of all the methods, four different indicators, MAE, MAPE, RMSE, and R², are used to analyze the results. The calculation formula of each indicator is shown in Table 2. For all the formulas, y_i is the true value, \hat{y}_i is the predicted value, \bar{y}_i is the average of the samples, and *N* is the length of the samples.

Equations
$MAE = \frac{1}{n} \sum_{i=1}^{n} y_i - \hat{y}_i $
$ ext{MAPE} = rac{1}{N}\sum\limits_{i=1}^{N} \left rac{y_i - \hat{y_i}}{y_i} ight $
$ ext{RMSE} = \sqrt{rac{1}{n}\sum\limits_{i=1}^n (y_i - \hat{y}_i)^2}$
${ m R}^2 = 1 - rac{{\sum\limits_{i = 1}^N {{\left({{y_i} - {{\hat y_i}}} ight)}^2 } }}{{\sum\limits_{i \le 1} {{\left({{y_i} - {{\overline y_i}}} ight)}^2 } }}$

Table 2. Calculation formulas for the four evaluation indicators of the experiment.

4. Results

The proposed method was programmed with Python using Tensorflow and Keras. The following results and discussions were accomplished on a laptop computer with a system of Windows 10, an Intel Core i5-1135G7 @2.40 GHz, and 16 GB of memory. The source codes of the baseline models will be publicly available on the website [60].

4.1. Comparison with the RNNs

In reference [53], the CWRNN demonstrates that it outperforms both the RNN and LSTM networks in the experiments. In this study, to verify the advantages of CWRNNs, three other RNN models, including simple RNNs, LSTMs, and BiLSTMs, were used to make offshore wind speed predictions. The same dataset was used to train and evaluate the models. All the models have the same hyperparameters, which are shown in Table 3. The prediction results are shown and described in Figure 7 and Table 4.

Table 3. The numerical metrics of the prediction results by CWRNNs and RNNs of Site1.

Hyperparameters	Settings (All Models, including the RNN, LSTM, BiLSTM, and CWRNN, have the Same Hyperparameters)
Input numbers	60
Hidden layers	1
Hidden neurons	200
Dense layers	1
Optimizer	RMSprop
Learning rate	10-3
Epoch	200
Batch size	100



Figure 7. Comparison results of the proposed model with RNNs of Site1.

Model	Deverseters	Run Times	Evaluation Metrics (Mean Value of 10 Times)			
	rarameters	(Mean Value of 10 Times)	MAE	MAPE	RMSE	R ²
Simple RNN	40,601	128.3919 s	0.7207	10.8733	1.0116	0.5988
LSTM	161,801	743.7911 s	0.6222	8.5401	0.8304	0.7296
BiLSTM	323,601	1666.8021 s	0.5443	7.8204	0.7551	0.7764
CWRNN	40,801	77.7866 s	0.4572	6.7873	0.6566	0.8310

Table 4. The numerical metrics of the prediction results by CWRNNs and RNNs of Site1.

As shown in Figure 7, compared with the true data for Site1, the prediction curves of all the RNNs are close to the real curve of the true wind speed data, which means they have all captured the tendency of true wind speed. It relies on the powerful ability of RNNs in a modeling time series. In contrast to other RNNs, the prediction curve of the CWRNN appears to be closer to the real curve, which verifies that the CWRNN has a better performance in solving strong nonlinear problems.

Table 4 lists the corresponding MAE, MAPE, RMSE, and R² values. The indexes of the RNN are the worst because the RNN cannot remember long-term dependency due to the vanishing gradient. In comparison to the other RNNs, CWRNNs achieves great accuracy, with lower MAE, MAPE, RMSE and higher R². Furthermore, it can be observed from Table 4 that the CWRNN almost has the same parameters as the simple RNN, but the LSTM and BiLSTM have large parameters, which are computationally expensive; hence, the LSTMs are slow, which is also shown in Table 5. In comparison to all the RNNs, the CWRNN resulted in fewer runtimes because only parts were updated at every step.

Table 5. Average and standard deviation of prediction results by the CWRNNs and RNNs of Site1.

				Evaluatio	on Metrics			
Model	Model MAE		MAPE		RMSE		R ²	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
Simple RNN	0.7207	0.0974	10.8733	1.5279	1.0116	0.0896	0.5988	0.0749
LSTM	0.6222	0.0539	8.5401	0.6652	0.8304	0.0715	0.7296	0.0492
BiLSTM CWRNN	0.5443 0.4572	0.0334 0.0044	7.8204 6.7873	0.5151 0.0311	0.7551 0.6566	0.0362 0.0044	0.7764 0.8310	0.0217 0.0023

Table 5 shows the mean and standard deviation values of the metrics of the prediction results. All the metrics data in the following figures are the average of 10 times.

As shown in Figure 8, compared with the true data of site2, the same conclusion as Site1 can be obtained. Compared with the other RNNs, the prediction curve of the CWRNN still appears to be closer to the real curve, by which the performance of the CWRNN has been verified again. These numerical results can also be obtained from Table 6. Compared with the other RNNs, the CWRNN also achieves better accuracy, with a lower MAE, MAPE, and RMSE, and a higher R², which shows that the CWRNN can deal with strong nonlinear problems.

To verify the generalization of the proposed model, Site3, which is an onshore wind power station, was selected for verification. Compared with the offshore sites, the wind speed of Site3 changes more slowly, as is shown in Figure 9. From the figure, it can be observed that the RNN is still the worst model among all the RNNs. The reason may be that we set the same hyperparameters in the experiments, which included the input length. The RNN has a poor ability in its long-term dependency. The numerical result in Table 7 also verifies the conclusion. The CWRNN continues to show the best prediction results in both the onshore and offshore wind speed data, which verified that the CWRNN has a better performance in wind speed predictions.



Figure 8. Comparison results of the proposed model with the RNNs of Site2.

Table 6.	The numerical	metrics of the	e prediction	results by the	e CWRNNs an	d RNNs of Site2
			1	2		

Model	Eva	aluation Metrics (M	ean Value of 10 Tim	ies)
	MAE	MAPE	RMSE	R ²
Simple RNN	0.6719	9.5407	0.8794	0.6362
LSTM	0.4952	6.3373	0.7256	0.7523
CWRNN	0.4430	5.8871	0.6799	0.7825



Figure 9. Comparison results of the proposed model with the RNNs of Site3.

Table 7. The numerical metrics of the prediction results by the CWRNNs and RNNs of Site3.

Model	Eva	aluation Metrics (M	ean Value of 10 Tim	ies)
	MAE	MAPE	RMSE	R ²
Simple RNN	1.6685	22.5562	1.7986	0.5955
LSTM	0.4315	5.7073	0.7715	0.9288
CWRNN	0.3843	5.0672	0.6446	0.9480

The evaluation metrics of all three sites are recorded together, as shown in Figure 10. It can be seen that the model achieves a better performance at all three sites, which means

the proposed method has good generalization. Furthermore, Site3, which was an onshore site, achieved the best performance out of all of the sites; its wind speed could be more easily predicted in comparison to the other offshore sites.



Figure 10. Evaluation metrics of Site1, Site2, and Site3.

4.2. Comparison with the Traditional Neural Networks

In order to verify the powerful ability of CWRNNs for time series prediction, the proposed method was compared with the traditional neural networks. In this experiment, the MLP, BPNN, and CNN, as traditional neural networks that are powerful machine learning models often used in different fields, were tested to perform the time series prediction task. The results are shown and described in Figure 11 and Table 8.



Figure 11. Comparison results of the proposed model with traditional NNs.

Table 8. The numerical metrics of the pre	ediction results by	V CWRNNs and	traditional NNs
---	---------------------	--------------	-----------------

Model	Evaluation Metrics (Mean Value of 10 Times)				
	MAE	MAPE	RMSE	R ²	
MLP	0.86	24.9	1.18	0.45	
BPNN	0.53	7.99	0.76	0.78	
CNN	0.61	8.58	0.79	0.76	
CWRNN	0.46	6.79	0.66	0.83	

It is obvious from the figure that MLP achieves the worst result. MLP, as a typical simple NN, has shortcomings, such as a slow learning speed, easily falling into local

extremum, and learning may not be sufficient. The result shows that MLP fails to learn from the wind speed data. The results also show that BPNN and CNN have worse performances in wind speed prediction. In most cases, BPNN and CNN have the powerful ability to solve nonlinear problems. However, they are not good at dealing with time series. Compared with the traditional neural networks, CWRNN appears to be more powerful in time series processing. Table 8 shows the numerical metrics of the prediction results, which further illustrates the above conclusion.

4.3. Comparison with Different Hyperparameters

There are many hyperparameters to set up a CWRNN model. Some hyperparameters are shared by RNN models, such as hidden layer parts, hidden layer neurons, the number of hidden layers, and the length of time series inputs. In essence, the CWRNN is a type of RNN that has the same network framework and mechanism of the backward pass of the error propagation. Therefore, the influence of the shared hyperparameters on the network is roughly the same. However, the CWRNN has some unique hyperparameters. The following experiments will focus on the specific parameters of CWRNNs.

4.3.1. Comparison with Different Part Numbers

The number of hidden layer parts is an important hyperparameter of the CWRNN, which has a great impact on the performance of the model. In the experiment, by changing the value of the hyperparameter, the influence on the accuracy of the model is evaluated. By setting different numbers for the hidden layer parts and training the model, we then used the evaluation metrics to evaluate the model's accuracy. The number of parts was set as (2, 4, 5), with all other parameters being the same.

The results are shown and described in Figure 12 and Table 9. From the results, we find that the least number of parts has the worst accuracy. When the number of parts increase to 4, we achieved the highest prediction accuracy. When the number raised to 5, the accuracy was lower than 4 parts, and higher than 2 parts. However, at the same time, the cost time of training the model significantly increased. Therefore, the value of four parts was the best choice in this study.



Figure 12. Comparison results of the proposed model with different part numbers.

Part Number	Values	Evaluation Metrics (Mean Value of 10 Times)			
	values	MAE	MAPE	RMSE	R ²
2	[1,2]	0.4835	6.9156	0.686	0.8155
4	[1,2,4,8]	0.4572	6.7873	0.6566	0.8310
5	[1,2,4,8,16]	0.4719	6.8052	0.6725	0.8227

4.3.2. Comparison with Different Part Periods

The part period is another hyperparameter that is unique to CWRNNs. The exponential series is often used as the part period. However, some other functions can be used for the part period, such as the linear function, Fibonacci function, logarithmic functions, or even fixed random periods. Different part periods will cause the different performances of the model. In this experiment, four different part periods were used to test the performance of the CWRNN. All the hidden layer parts were set to 4 and the other parameters were the same.

The results are shown in Figure 13 and Table 10. The four part periods were the linear series, odd series, triple series, and exponential series. Compared with the other series, the part period using the exponential series resulted in the model achieving the best performance. The result of the triple series shows great competitiveness, which means that the series gap increases with the increase in the number of periods and is thus a better choice.



Figure 13. The results comparison of the proposed model with different part periods.

Part Periods	Values	Evaluation Metrics (Mean Value of 10 Times)			
		MAE	MAPE	RMSE	R ²
1	[1,2,3,4]	0.4948	7.0096	0.6973	0.8094
2	[1,3,9,27]	0.4655	6.7413	0.6673	0.8254
3	[1,3,5,7]	0.4806	6.9184	0.6822	0.8175
4	[1,2,4,8]	0.4572	6.7873	0.6566	0.8310

Table 10. The numerical metrics of the prediction results with different part periods.

5. Discussion

An offshore wind speed prediction method using CWRNNs is proposed and is verified by the wind speed dataset of offshore and onshore sites. The results are further discussed and analyzed in the following contexts:

(1) As is commonly known, RNN is excellent at modeling sequential data with a simple mechanism. However, with the increase in the dependency length, which means more context is needed, the RNN cannot learn from the input data. There are some techniques to improve the RNN. LSTM, which uses the gating mechanism, is proposed to solve problems, including vanishing gradients and long dependency. It is easier to understand that the complex network structure increases the model stability. However, the performance of most machine learning models, especially complex deep learning neural network models, depends on the quantity and diversity of the data. Naturally, if a machine learning model has a lot of parameters, it needs a proportional number of samples to perform well.

The CWRNN is another type of RNN, which breaks up the neurons in the hidden layer into different parts, and the neurons in the same part work at a given clock speed to address long term dependency. The parameters of the CWRNN are close to the simple RNN. This indicates that the CWRNN is more suitable for the case of a small sample size than LSTM. Meanwhile, the CWRNN employs an ingenious mechanism for activating neurons parts at different clock speeds, which can efficiently learn the long-term time series information, thus solving strongly nonlinear problems. At the same time, the CWRNN only updates neuron parts at a specific clock rate, which reduces the computation cost.

(2) There is an inherent concept of sequential data or time series data that incrementally progresses over time. As we know, traditional NNs are good at solving the nonlinear problem and perform well in most cases. However, they lack the inherent trend of persistence for obtaining sequential data. A simple feedforward NN cannot really understand the meaning of a sentence according to the order of input data in the context. CNNs have been extremely successful in the computer vision field. However, they have difficulties in dealing with time series data. The RNN, as a type of neural network, keeps the characteristics of the autoregressive model, and also has the ability to model sequential data. Furthermore, for the human neural system, the vision channel and the memory channel are different channels that have different mechanisms.

Recently, the attention mechanism is one of the most valuable breakthroughs in deep learning model preparation in the last few decades. Unlike the vanilla RNN approach, it proposes to help monitor all the hidden states in the encoder sequence for making predictions. It can assign the weight values to the extracted information to highlight the important information that the attention mechanism seems to break the barriers between the vision channel and memory channel. However, it still has a great number of parameters, which also need a large number of sample data. For now, the CWRNN is a good choice to solve strong nonlinear problems with limited samples.

(3) Hyperparameters can directly impact the performance of machine learning models. Therefore, to achieve the best performance, the optimization of the hyperparameters plays a crucial role. In addition to the common parameters of the RNNs, the CWRNN has some unique parameters. The setting of these parameters requires a complex parameter tuning process and the appropriate parameters will result in a great improvement to its performance.

In this study, some unique parameters were discussed, which were based on the experiment results. However, the common parameters of the RNNs still affect the model performance. Considering the shared RNN parameters together with the intrinsic parameters of the CWRNN will be a big project. Tuning these parameters requires further research.

6. Conclusions

This study proposes an offshore wind speed prediction method based on CWRNNs. The CWRNN breaks up neurons in the hidden layer into different parts, and neurons in the same part work at a given clock speed to address long term dependency, which can effectively solve the problem of strong nonlinearity in offshore wind speed. The performance of the proposed method is verified by three datasets from two different offshore sites and one onshore site. The experimental results show that the proposed model achieves a significant improvement in its prediction accuracy.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; software, Y.S.; validation, Y.S., Y.W. and H.Z.; data curation, Y.W.; writing—original draft preparation, Y.S.; writing—review and editing, Y.W. and H.Z.; visualization, Y.S.; supervision, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China under Grant No. 2018YFB1307400.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the support from the DER AI Lab of Shanghai University and the State Grid Intelligence Technology Corporation of China for the development of the machine learning model and the dataset.

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

References

- Jung, C.; Taubert, D.; Schindler, D. The temporal variability of global wind energy—Long-term trends and inter-annual variability. Energy Convers. Manag. 2019, 188, 462–472. [CrossRef]
- 2. Wang, J.; Song, Y.; Liu, F.; Hou, R. Analysis and application of forecasting models in wind power integration: A review of multi-step-ahead wind speed forecasting models. *Renew. Sust. Energ. Rev.* **2016**, *60*, 960–981. [CrossRef]
- 3. Yang, K.; Tang, Y.; Zhang, Z. Parameter Identification and State-of-Charge Estimation for Lithium-Ion Batteries Using Separated Time Scales and Extended Kalman Filter. *Energies* **2021**, *14*, 1054. [CrossRef]
- Qian, Z.; Pei, Y.; Zareipour, H.; Chen, N. A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. *Appl. Energy* 2019, 235, 939–953. [CrossRef]
- 5. Zhang, J.; Draxl, C.; Hopson, T.; Monache, L.D.; Vanvyve, E.; Hodge, B.M. Comparison of numerical weather prediction based deterministic and probabilistic wind resource assessment methods. *Appl. Energy* **2015**, *156*, 528–541. [CrossRef]
- Wang, J.; Qin, S.; Jin, S.; Wu, J. Estimation methods review and analysis of offshore extreme wind speeds and wind energy resources. *Renew. Sust. Energ. Rev.* 2015, 42, 26–42. [CrossRef]
- Morgan, E.C.; Lackner, M.; Vogel, R.M.; Baise, L.G. Probability distributions for offshore wind speeds. *Energy Convers. Manag.* 2011, 52, 15–26. [CrossRef]
- 8. Cai, H.; Jia, X.; Feng, J.; Yang, Q.; Li, W.; Li, F.; Lee, J. A unified Bayesian filtering framework for multi-horizon wind speed prediction with improved accuracy. *Renew. Energy* **2021**, *178*, 709–719. [CrossRef]
- 9. Li, L.; Liu, Y.-Q.; Yang, Y.-P.; Han, S.; Wang, Y.-M. A physical approach of the short-term wind power prediction based on CFD pre-calculated flow fields. *J. Hydrodyn. B* 2013, 25, 56–61. [CrossRef]
- Zhang, K.; Qu, Z.; Dong, Y.; Lu, H.; Leng, W.; Wang, J.; Zhang, W. Research on a combined model based on linear and nonlinear features—A case study of wind speed forecasting. *Renew. Energy* 2019, 130, 814–830. [CrossRef]
- 11. Zhang, C.; Wei, H.; Zhao, X.; Liu, T.; Zhang, K. A Gaussian process regression-based hybrid approach for short-term wind speed prediction. *Energy Convers. Manag.* **2016**, *126*, 1084–1092. [CrossRef]
- 12. Dhiman, H.S.; Deb, D.; Foley, A.M. Bilateral Gaussian wake model formulation for wind farms: A forecasting based approach. *Renew. Sust. Energ. Rev.* **2020**, *127*, 109873. [CrossRef]
- Karakuş, O.; Kuruoğlu, E.E.; Altınkaya, M.A. One-day ahead wind speed/power prediction based on polynomial autoregressive model. *IET Renew.* 2017, 11, 1430–1439. [CrossRef]
- 14. Tian, Z.; Wang, G.; Ren, Y. Short-term wind speed forecasting based on autoregressive moving average with echo state network compensation. *Wind Eng.* **2019**, *44*, 152–167. [CrossRef]
- 15. Liu, M.D.; Ding, L.; Bai, Y.L. Application of hybrid model based on empirical mode decomposition, novel recurrent neural networks and the ARIMA to wind speed prediction. *Energy Convers. Manag.* **2021**, 233, 113917. [CrossRef]
- 16. Liu, X.; Lin, Z.; Feng, Z. Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM. *Energy* **2021**, 227, 120492. [CrossRef]
- 17. Cai, H.; Jia, X.; Feng, J.; Li, W.; Hsu, Y.M.; Lee, J. Gaussian Process Regression for numerical wind speed prediction enhancement. *Renew. Energy* **2020**, *146*, 2112–2123. [CrossRef]
- 18. Ak, R.; Li, Y.F.; Vitelli, V.; Zio, E. Adequacy assessment of a wind-integrated system using neural network-based interval predictions of wind power generation and load. *Int. J. Electr. Power* **2018**, *95*, 213–226. [CrossRef]
- 19. 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]
- 20. Rani, R.H.; Victoire, T.A. Training radial basis function networks for wind speed prediction using PSO enhanced differential search optimizer. *PLoS ONE* **2018**, *13*, e0196871. [CrossRef]
- 21. Lu, P.; Ye, L.; Tang, Y.; Zhao, Y.; Zhong, W.; Qu, Y.; Zhai, B. Ultra-short-term combined prediction approach based on kernel function switch mechanism. *Renew. Energy* **2021**, *164*, 842–866. [CrossRef]
- 22. Dhiman, H.S.; Deb, D.; Guerrero, J.M. Hybrid machine intelligent SVR variants for wind forecasting and ramp events. *Renew. Sust. Energ. Rev.* **2019**, *108*, 369–379. [CrossRef]
- 23. Dhiman, H.S.; Anand, P.; Deb, D. Wavelet transform and variants of SVR with application in wind forecasting. In *Innovations in Infrastructure*; Springer: Singapore, 2018; Volume 757, pp. 501–511. [CrossRef]
- 24. Dhiman, H.S.; Deb, D. Machine intelligent and deep learning techniques for large training data in short-term wind speed and ramp event forecasting. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12818. [CrossRef]
- Dhiman, H.S.; Deb, D.; Balas, V.E. Supervised Machine Learning in Wind Forecasting and Ramp Event Prediction; Academic Press: Salt Lake City, UT, USA, 2020. [CrossRef]

- Patel, P.; Shandilya, A.; Deb, D. Optimized hybrid wind power generation with forecasting algorithms and battery life considerations. In Proceedings of the IEEE Power and Energy Conference at Illinois (PECI), Champaign, IL, USA, 23–24 February 2017. [CrossRef]
- Bai, Y.; Liu, M.-D.; Ding, L.; Ma, Y.-J. Double-layer staged training echo-state networks for wind speed prediction using variational mode decomposition. *Appl. Energy* 2021, 301, 117461. [CrossRef]
- 28. Xu, W.; Liu, P.; Cheng, L.; Zhou, Y.; Xia, Q.; Gong, Y.; Liu, Y. Multi-step wind speed prediction by combining a WRF simulation and an error correction strategy. *Renew. Energy* **2021**, *163*, 772–782. [CrossRef]
- 29. Zhu, X.; Liu, R.; Chen, Y.; Gao, X.; Wang, Y.; Xu, Z. Wind speed behaviors feather analysis and its utilization on wind speed prediction using 3D-CNN. *Energy* 2021, 236, 121523. [CrossRef]
- 30. Ma, Q.-L.; Zheng, Q.-L.; Peng, H.; Zhong, T.-W.; Qin, J.-W. Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network. *Chin. Phys. B* 2008, 17, 536. [CrossRef]
- 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]
- Wang, S.; Wang, J.; Lu, H.; Zhao, W. A novel combined model for wind speed prediction—Combination of linear model, shallow neural networks, and deep learning approaches. *Energy* 2021, 234, 121275. [CrossRef]
- Saeed, A.; Li, C.; Gan, Z.; Xie, Y.; Liu, F. A simple approach for short-term wind speed interval prediction based on independently recurrent neural networks and error probability distribution. *Energy* 2022, 238, 122012. [CrossRef]
- Liu, L.; Wang, J. Super multi-step wind speed forecasting system with training set extension and horizontal-vertical integration neural network. *Appl. Energy* 2021, 292, 116908. [CrossRef]
- Xiong, D.; Fu, W.; Wang, K.; Fang, P.; Chen, T.; Zou, F. A blended approach incorporating TVFEMD, PSR, NNCT-based multimodel fusion and hierarchy-based merged optimization algorithm for multi-step wind speed prediction. *Energy Convers. Manag.* 2021, 230, 113680. [CrossRef]
- Neshat, M.; Nezhad, M.M.; Abbasnejad, E.; Seyedali, M.; Lina, B.T.; Davide, A.G.; Bradley, A.; Markus, W. A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm. *Energy Convers. Manag.* 2021, 236, 114002. [CrossRef]
- Ahmad, T.; Zhang, D. A data-driven deep sequence-to-sequence long-short memory method along with a gated recurrent neural network for wind power forecasting. *Energy* 2022, 239, 122109. [CrossRef]
- 38. Zhang, Z.; Ye, L.; Qin, H.; Liu, Y.; Wang, C.; Yu, X.; Li, J. Wind speed prediction method using Shared Weight Long Short-Term Memory Network and Gaussian Process Regression. *Appl. Energy* **2019**, *247*, 270–284. [CrossRef]
- 39. Chen, Y.; Dong, Z.; Wang, Y.; Su, J.; Han, Z.; Zhou, D.; Zhang, K.; Zhao, Y.; Bao, Y. Short-term wind speed predicting framework based on EEMD-GA-LSTM method under large scaled wind history. *Energy Convers. Manag.* **2021**, 227, 113559. [CrossRef]
- 40. Tian, Z. Modes decomposition forecasting approach for ultra-short-term wind speed. *Appl. Soft Comput.* **2021**, *105*, 107303. [CrossRef]
- 41. Li, F.; Ren, G.; Lee, J. Multi-step wind speed prediction based on turbulence intensity and hybrid deep neural networks. *Energy Convers. Manag.* **2019**, *186*, 306–322. [CrossRef]
- 42. Zhang, Y.; Chen, B.; Pan, G.; Zhao, Y. A novel hybrid model based on VMD-WT and PCA-BP-RBF neural network for short-term wind speed forecasting. *Energy Convers. Manag.* 2019, 195, 180–197. [CrossRef]
- 43. Tian, Z.; Ren, Y.; Wang, G. Short-term wind speed prediction based on improved PSO algorithm optimized EM-ELM. *Energy* Sources Part A Recovery Util. Environ. Eff. 2018, 41, 26–46. [CrossRef]
- 44. Song, J.; Wang, J.; Lu, H. A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. *Appl. Energy* **2018**, *215*, 643–658. [CrossRef]
- Ma, Z.; Chen, H.; Wang, J.; Yang, X.; Yan, R.; Jia, J.; Xu, W. Application of hybrid model based on double decomposition, error correction and deep learning in short-term wind speed prediction. *Energy Convers. Manag.* 2020, 205, 112345. [CrossRef]
- 46. Wang, J.; Li, Y. Multi-step ahead wind speed prediction based on optimal feature extraction, long short-term memory neural network and error correction strategy. *Appl. Energy* **2018**, 230, 429–443. [CrossRef]
- 47. Liang, Z.; Liang, J.; Wang, C.; Dong, X.; Miao, X. Short-term wind power combined forecasting based on error forecast correction. *Energy Convers. Manag.* **2016**, *119*, 215–226. [CrossRef]
- 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]
- 49. Liang, T.; Zhao, Q.; Lv, Q.; Sun, H. A novel wind speed prediction strategy based on Bi-LSTM, MOOFADA and transfer learning for centralized control centers. *Energy* 2021, 230, 120924. [CrossRef]
- Li, C.; Tang, G.; Xue, X.; Saeed, A.; Hu, X. Short-term wind speed interval prediction based on ensemble GRU model. *IEEE Trans.* Sustain. Energy 2019, 11, 1370–1380. [CrossRef]
- Feng, X.; Chen, J.; Zhang, Z.; Miao, S.; Zhu, Q. State-of-charge estimation of lithium-ion battery based on clockwork recurrent neural network. *Energy* 2021, 236, 121360. [CrossRef]
- 52. Luo, J.; Fu, Y. Dilated Recurrent Neural Network. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
- 53. Koutnik, J.; Greff, K.; Gomez, F.; Schmidhuber, J. A clockwork rnn. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.

- 54. Xie, Y.; Zhang, Z.; Sapkota, M.; Yang, L. Spatial clockwork recurrent neural network for muscle perimysium segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Athens, Greece, 11–21 October 2016. [CrossRef]
- 55. Lin, C.; Wang, H.; Yuan, J.; Yu, D.; Li, C. Research on UUV obstacle avoiding method based on recurrent neural networks. *Complexity* **2019**, *2019*, 6320186. [CrossRef]
- Achanta, S.; Godambe, T.; Gangashetty, S.V. An investigation of recurrent neural network architectures for statistical parametric speech synthesis. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015. [CrossRef]
- Liu, W.; Gu, Y.; Ding, Y.; Lu, W.; Rui, X.; Tao, L. A Spatial and Temporal Combination Model for Traffic Flow: A Case Study of Beijing Expressway. In Proceedings of the 2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE), Beijing, China, 11–13 September 2020. [CrossRef]
- 58. Roberts, O.; Andreas, A. United States Virgin Islands: St. Thomas & St. Croix (Data); NREL Report No. DA-5500-64451; NREL-DATA: Golden, CO, USA, 1997. [CrossRef]
- 59. NREL: Measurement and Instrumentation Data Center (MIDC). Available online: https://midcdmz.nrel.gov (accessed on 1 January 2022).
- SHU DER AI Lab. Available online: https://github.com/SHU-DeepEnergyResearch/Time-Series-Prediction (accessed on 1 January 2022).