Article

Enhancing Wind Turbine Power Forecast via Convolutional Neural Network

Tianyang Liu ^{1,2} , Zunkai Huang ¹ , Li Tian ^{1,*}, Yongxin Zhu ^{1,*}, Hui Wang ^{1,*} and Songlin Feng ¹

¹ Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China; liutianyang@sari.ac.cn (T.L.); huangzk@sari.ac.cn (Z.H.); fengsl@sari.ac.cn (S.F.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: tianl@sari.ac.cn (L.T.); zhuyongxin@sari.ac.cn (Y.Z.); wanghui@sari.ac.cn (H.W.)

Abstract: The rapid development in wind power comes with new technical challenges. Reliable and accurate wind power forecast is of considerable significance to the electricity system's daily dispatching and production. Traditional forecast methods usually utilize wind speed and turbine parameters as the model inputs. However, they are not sufficient to account for complex weather variability and the various wind turbine features in the real world. Inspired by the excellent performance of convolutional neural networks (CNN) in computer vision, we propose a novel approach to predicting short-term wind power by converting time series into images and exploit a CNN to analyze them. In our approach, we first propose two transformation methods to map wind speed and precipitation data time series into image matrices. After integrating multi-dimensional information and extracting features, we design a novel CNN framework to forecast 24-h wind turbine power. Our method is implemented on the Keras deep learning platform and tested on 10 sets of 3-year wind turbine data from Hangzhou, China. The superior performance of the proposed method is demonstrated through comparisons using state-of-the-art techniques in wind turbine power forecasting.

Keywords: convolutional neural network; GWF matrices; ripple encoding; wind power generation forecast

Citation: Liu, T.; Huang, Z.; Tian, L.; Zhu, Y.; Wang, H.; Feng, S. Enhancing Wind Turbine Power Forecast via Convolutional Neural Network. *Electronics* **2021**, *10*, 261. <https://doi.org/10.3390/electronics10030261>

Received: 8 January 2021

Accepted: 19 January 2021

Published: 22 January 2021

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

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wind power generation is an essential part of renewable energy to mitigate some of the adverse effects of global warming. The Global Wind Energy Council (GWEC) estimated that total installed wind power capacity would increase to 792 million kW by 2020, and wind power would account for more than 20% of the total electricity generation all over the world by 2030. The inherent intermittence and randomness of wind energy inevitably make the power generated by wind turbines highly volatile. As the wind power capacity continues to grow, such power fluctuations will make it significantly challenging to keep the electric system stable and reliable. Generally, accurate prediction of wind power can effectively alleviate the pressure of peak and frequency modulation in the power system. There are four forecast categories based on the forecast horizon: long-term forecast, medium-term forecast, short-term forecast, and ultra-short-term forecast [1]. Specifically, short-term forecast refers to the forecast of electricity generation in the next day or a few days. It provides scientific and rational references for the power production, operation department, reducing production costs, ensuring power supply, setting the time-of-use price, and arranging equipment maintenance [2]. Hence, we mainly focus on the 24-h wind power generation prediction in this paper.

Traditional wind power forecasting methods generally employ turbine parameters and weather information parameters as model inputs. However, due to the complexity of real-world weather factors, it is not always possible to build a universally effective model by traditional means. Thanks to the development of machine learning and deep learning, researchers can develop accurate and stable models without physical model knowledge [3].

In some recent work, fuzzy prediction [4], wavelet analysis [5], least-square support vector machine (LSSVM) [6], and other methods have been widely applied to wind power prediction research. Meanwhile, deep learning techniques have also been adopted in wind power forecasts owing to their capability of thoroughly revealing nonlinear relationships in historical data [7]. Deep learning simulates the learning process of the human brain. Each computing unit is equivalent to one neuron in the brain. Through training and feedback from a large amount of data, the algorithm constantly corrects the connections between neurons until it reaches the optimal computational state. These networks include the general regression neural network (GRNN) [8], wavelet neural network (WNN), and generalized feed-forward neural network (GFNN) [9].

In this paper, we propose a new method to predict 24-h wind power. In contrast to existing studies, this paper is the first to convert multi-dimensional weather data into image matrices and utilize CNN to analyze them. The main contributions of this work can be summarized below:

- We propose Gramian wind field (GWF) matrices to map the zonal and meridional wind speed time series into image matrices with time characteristics retained and wind field features extracted.
- We propose a ripple encoding method to encode precipitation data to image matrices.
- We design a novel CNN network incorporating GWF matrices and ripple encoding matrices to forecast 24-h wind turbine power.

The rest of the paper is organized as follows. In Section 2, related work is reviewed briefly. In Section 3, we present the preliminaries and methodology. Experimental results are presented in Section 4, which are followed by the conclusions.

2. Related Work

The improvement in the accuracy of short-term wind speed prediction depends on the acquisition of more valuable input data and the adoption of appropriate methods to build prediction models. Wu and Alessio [10,11] conjectured that precipitation could affect the fan's operation and even damage the turbine blades.

Wind power ramps cause a large-amplitude power fluctuation, which harmfully affects the stability of the power system's operation. Ouyang et al. [12] combined the Markov chain and auto-regression (AR) model to correct the prediction residual. They also proposed an improved swinging door algorithm to extract linear segments. Zhao et al. [13] succeeded in capturing the spatial dependency between neighbor turbines' power output and utilized a graphical model to model the dependency of turbine-level ramp events. Many researchers also made contributions to regional wind power forecasting [14]. Ozkan et al. [15] proposed a method of power generation forecasting for offline plants within a specific region and time. Probabilistic wind power forecasting is another interesting problem. Compared with the deterministic prediction, probabilistic forecast considers the uncertainty, which helps manage risks and make decisions for power grids [16]. Wu et al. proposed a novel probabilistic forecasting method for wind power generation, which includes data preprocessing, a clustering algorithm, and post-processing of the predicted interval. Wang et al. [17] designed a framework that combined the hybrid feature selection and multi-objective optimization algorithms. The algorithm could perform deterministic and probabilistic forecasting together. With the development of deep learning technologies, recurrent networks become more and more popular in wind power forecast problems.

With unique time-varying vectors of hidden activations, recurrent networks [18] are usually used to deal with sequence problems such as machine translation and speech recognition. As an improved architecture for the basic recurrent neural network (RNN), long short-term memory (LSTM) networks solve the vanishing gradient problem and are more widely used [18]. Zhu et al. [1] combined a sequence to sequence model with LSTM to predict multistep electric signals. The gated recurrent unit (GRU) architecture is also popular because it achieves comparable performance to LSTM while keeping a relatively

low computation cost [19]. However, recurrent networks that maintain a hidden state of the entire past data are difficult to parallelize and accelerate [18].

Convolutional networks are often used in image processing, among which AlexNet [20], VGGNet [21], and ResNet [22] are the representative ones. AlexNet contains five convolutional layers and three dense layers. It also utilizes a dropout layer to avoid overfitting. Compared with AlexNet, VGGNet features a deeper structure and better performance. A typical VGGNet is formed by 16 layers, and the convolutional kernels in each layer have the same size. Furthermore, ResNet, which combines the best of VGGNet and a residual module, is an excellent convolutional network in target detection with high computation speed. Due to the residual module, ResNet can retain the information of the shallow layer even though it has a very deep structure. Recently, some convolutional networks have achieved brilliant results in sequence problems. Facebook AI Research proposed an architecture for a sequence to sequence modeling that is entirely convolutional to deal with the natural language processing (NLP) problem [18]. To capture the position of the words, they added position vectors as position embedding. They also proposed the stacked CNN architecture to analyze long-distance information with high efficiency, which also makes parallel acceleration available. At the same time, they utilized both gated linear units (GLU) as the gate mechanism and multistep attention that combines residual connection and linear mapping. Bai and Kolter [23,24] proposed a temporal convolutional network (TCN) which is more accurate, simpler, and clearer than recurrent architectures such as LSTMs and GRUs. They used causal convolution to retain the temporal correlation of sequences and utilized dilated convolution and residual blocks to remember historical information [24].

Recently, some convolutional networks have achieved good results in time series classification (TSC) problems. The main difference among these methods lies in the image matrices generated from time series, for example, GAF/MTF [25], recurrence plots [26], and relative position matrix [27]. Wang and Oates proposed Gramian angular field (GAF) matrices to map time series into images, where they represented time series in a polar coordinate system instead of the typical Cartesian coordinates. In the Gramian angular matrix, each element is the cosine of the summation of angles. They also proposed Markov transition field (MTF) matrices. In MTF, they encoded dynamic transition statistics and extended that idea by representing the Markov transition probabilities sequentially to preserve information in the time domain. Hatami et al. [27] proposed recurrence plots (RP) to transform time series into 2D images by revealing at which points some trajectories return to a previous state. RPs are a visualization tool used to explore the multiple dimensional phase space trajectory through a 2D representation of its recurrences.

3. Methodology

Wind speed and direction at different heights are strongly correlated with the internal atmospheric motion in the region, which means that the wind of a specific height at the current time is related to the historical conditions and other winds at different heights. Wind turbines and blades are also affected by rainfall [28]. Since the comprehensive analysis of these spatial and temporal data can make the prediction more accurate [29], we decided to combine wind data and precipitation data to forecast 24-h wind turbine power. The raw dataset contains ten wind turbine power and wind speed time series from 00:00 1 January 2012 to 24:00 31 December 2014, located in YuHang, Hangzhou, China. Wind speed data consist of hourly zonal and meridional wind component speed at 10 m and 100 m. Wind turbine power is presented as a percentage of the rated power. We obtained the daily precipitation data from the China Gauge-based Daily Precipitation Analysis (CGDPA) [30]. We propose the GWF matrix and ripple encoding to convert wind data and precipitation data, respectively, to image matrices, which are the data foundation of the CNN.

3.1. GWF Matrix

The Gram matrix is often used to calculate the linear dependence of a set of vectors. Given a set of input vectors, the Gram matrix is defined in Equation (1) [31]:

$$M = \begin{bmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \dots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \dots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \dots & \langle x_n, x_n \rangle \end{bmatrix} \quad (1)$$

where $\langle \rangle$ is an inner product of two vectors.

Inspired by the Gram matrix, we propose the Gramian wind field (GWF) matrix. Given time series, $\{u_1, u_2, \dots, u_n\}$ and $\{v_1, v_2, \dots, v_n\}$, of zonal and meridional wind component speed, we first calculated the hourly wind direction WD and an approximation of the wind energy WE from zonal and meridional wind components by Equation (2):

$$\begin{cases} WD_i = \arctan(u_i/v_i) \\ WE_i = 0.5 \times d \times \sqrt{(u_i^2 + v_i^2)}^3 \end{cases} \quad (2)$$

where d is air density. Since the pressure does not fluctuate enough for this to be a significant factor [32], we used a constant of 1.0 for d . After rescaling the wind energy component WE to $[0, 1]$ by min-max normalization, we designed a vector, $W = w_1, w_2, \dots, w_n$, that contains hourly wind features w_i in the polar coordinate system (WE'_i, WD_i). The min-max normalization formula is defined in Equation (3):

$$WE' = \frac{WE - WE_{\min}}{WE_{\max} - WE_{\min}} \quad (3)$$

where WE' is the data after min-max normalization and we is the original data. $WE_{\max}$ and $WE_{\min}$ are the maximum value and minimum value of the original data. The GWF is defined in Equation (4):

$$G = \begin{bmatrix} \langle w_1, w_1 \rangle & \langle w_1, w_2 \rangle & \dots & \langle w_1, w_n \rangle \\ \langle w_2, w_1 \rangle & \langle w_2, w_2 \rangle & \dots & \langle w_2, w_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle w_n, w_1 \rangle & \langle w_n, w_2 \rangle & \dots & \langle w_n, w_n \rangle \end{bmatrix} \quad (4)$$

where $\langle w_1, w_2 \rangle$ is the inner product between w_1 and w_2 .

$$\begin{aligned} & \langle w_1, w_2 \rangle \\ &= WE'_1 \cos WD_1 \times WE'_2 \cos WD_2 + WE'_1 \sin WD_1 \times WE'_2 \sin WD_2 \end{aligned} \quad (5)$$

Wind data in timestamp i are illustrated as the vector w_i in the polar coordinate system (WE'_i, WD_i). Items in diagonal represent the characteristic of the wind vector in each timestamp. Other items represent the relationship between wind vectors in different timestamps. In the raw dataset, there are two 2D time series from different heights: 100 m and 10 m. We mapped two 2D time series to two-channel images. An example 100 m GWF image is shown in Figure 1a.

3.2. Ripple Encoding

CGDPA is based on the optimal interpolation method and achieves a new high resolution ($0.25^\circ \times 0.25^\circ$ lat./lon.) [30]. From CGDPA, we gathered 3-year daily precipitation data (1st January 2012 to 31st December 2014) from the wind farm, and the maximum and minimum of precipitation data are 116.57 mm and 0 mm. We multiplied all the precipitations by 2.18 and then round the values to get a gray value series of (0,255). We propose ripple encoding to convert the new gray value series into an image matrix.

Figure 1. (a) An example 100 m Gramian wind field (GWF) image (top left) at 100 h (from 00:00 1 January 2012 to 04:00 5 January 2012), both axes are time axes; (b) an example ripple encoding image (top left) at 100 h (from 00:00 14 January 2012 to 04:00 18 January 2012), both axes are time axes.

On the reference of the GWF images' characteristics, the ripple encoding matrix is also symmetric, and the axes are in hours. For example, considering a 100-h precipitation data (5 days) series from 00:00 14 January 2012 to 04:00 18 January 2012, the raw precipitation series of each day is [8.75 7.62 0.31 0.17 7.26] and the gray value series after processing is [19 17 10 16]. The ripple encoding matrix of these 100 h data is shown in Equation (6), and its corresponding image is shown in Figure 1b.

$$R = \begin{bmatrix} 19 & \dots & 17 & \dots & 1 & \dots & 0 & \dots & 16 \\ \vdots & \ddots & & & & & & & \\ 17 & & 17 & & \vdots & & & & \\ \vdots & & & \ddots & & \vdots & & & \\ 1 & \dots & & 1 & & & & \vdots & \\ \vdots & & & & \ddots & & & & \\ 0 & & \dots & & & 0 & & & \\ \vdots & & & & & & \ddots & & \\ 16 & & & \dots & & & & & 16 \end{bmatrix} \quad (6)$$

3.3. Convolutional Neural Network Structure

After exploiting the GWF algorithm to map two 2D raw wind speed time series into two-channel images and mapping the precipitation data into the ripple encoding images, we propose a novel convolutional neural network to extract features from the above matrices and output a 24-h wind turbine power forecast. Figure 2 shows the proposed network.

Conv and FC denote the convolution layer and fully connected layer. The number following the layer name denotes the number of filters, and the numbers after the “at” symbol (@) denote the convolution kernel size. Each convolution layer consists of $64 \, 3 \times 3$ filters with a stride of 1 and zero paddings of 1. We utilized the GLU layer [33] following

the convolution layer. Like LSTMs, the GLU layer can control the information passed on in the hierarchy. The output of the GLU layer is calculated from Equation (7) [18]:

$$h_t(X) = (X \times W + b) \otimes \sigma(X \times V + c) \quad (7)$$

where X is the input and is the current output, W and V are convolution kernels, b and c are biases, σ is the sigma function, and $\otimes$ is the element-wise product operation.

Figure 2. The proposed forecast network framework.

Each fully connected layer consists of 256 units except that the final forecast layer consists of 24 units. Moreover, we applied the dropout layer (dropout rate is 0.5) and batch normalization layer to prevent overfitting and speed up the training procedure. We merged two networks to avoid potential interference and connect the final forecast layer with another fully connected layer.

4. Experimental Results and Evaluation

This section presents the details of the experimental setup, results, and evaluation of the proposed method.

4.1. Experimental Setup

The whole dataset contains 10 wind turbines, and each of them contains 26,304 hourly wind turbine power data, 26,304 hourly wind speed data, and 1096 daily precipitation data from 2012 to 2014. Each wind turbine is regarded as an independent case. We slice 20,000 samples from each wind turbine dataset and split them into training and test sets at a 7:3 ratio. We utilize the Keras deep learning framework and Python 3.7 for data processing and network design and training on the NVIDIA GeForce RTX 2080Ti.

4.2. Experimental Settings

To validate the efficiency of the proposed method, three popular recurrent network time series forecasting methods, including RNN [34], GRU [35], and LSTM [1], are taken as a comparison and assessed. The input data of three networks are the same: two 2D hourly wind speed data at 10 m and 100 m and 1D daily precipitation data. The experiment settings and parameters are listed in Table 1. We exploit the mean absolute error (MAE) and mean square error (MSE) as comparison metrics, which are defined in Equation (8).

$$\begin{cases} MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \\ MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \end{cases} \quad (8)$$

where $\hat{y}_t$ is the forecasted value, y_t is the actual value, and $y_{\max}$ and $y_{\min}$ are the maximum and minimum values among the test set. N refers to the test set size. Our assessment consists of two parts: (1) the performance of the proposed method is compared to the three previously mentioned typical time series forecasting methods to validate the efficacy in the different timesteps and different wind turbine datasets; (2) the effects of precipitation data are illustrated to reveal the effectiveness of ripple encoding.

Table 1. Experiment settings and parameters.

Settings	Parameters
wind turbine dataset number	10
training set size	14,000
test set size	6000
RNN network layer number	1
RNN neuron number	32
LSTM network layer number	2
LSTM neuron number in each layer	64
GRU network layer number	1
GRU neuron number	64
hidden dense layer number	1
hidden neuron number	200
dropout rate	0.5
training method	Adam Optimizer
loss function	MAE
forecast horizon	24

Figure 3. MAE and MSE comparison of 4 methods in different timesteps.

4.3. Comparison of Five Methods

To validate the efficiency of the proposed method, we compare the performance of different methods in the following timesteps: {24 40 50 60 70 75 80 82 85 90 95 100 110 120}. The results of ten wind turbines are averaged and presented in Figure 3. Furthermore, we present the detailed MAE and MSE comparison results for ten wind turbines under 75, 80, and 82 timesteps in Figures 4–6.

Figure 4. MAE and MSE comparison of ten wind turbines when timesteps = 75.

Figure 5. MAE and MSE comparison of ten wind turbines when timesteps = 80.

The results in Figures 3–6 indicate the following:

- In terms of the whole dataset of ten wind turbines, our method outperforms the other three methods among all the timesteps. Compared to the performance of GRU, LSTM, and RNN, the proposed method offers 27.52%, 28.67%, and 41.28% reductions in MAE, and 43.28%, 46.92%, and 57.84% reductions in MSE. Notably, all four methods receive better performance between 75 and 85 timesteps.
- In Figure 3, the other three networks all witness an obvious MAE and MSE decline from 24 to 50 timesteps. However, our method maintains outstanding forecasting performance without large fluctuations. It is notable that different timesteps will map out different image matrices in GWF matrices and ripple encoding. We can therefore conclude that our encoding method is effective.

- Regarding the results under fixed timesteps, the improvements in different turbines do not follow the same pattern. The improvements in MAE and MSE among different customers are largely diverse. While some customers receive 0.036 MAE reductions, the other customers may receive only half of that. Unlike the results of MAE, the reduction in MSE is more consistent.

We also compare our method with TCN, a popular convolution network for sequence problems [24]. Our method shows significant performance improvements in both different timesteps and different turbines. Figure 7 shows the detailed comparison results. In addition, training is the most time-consuming part of deep learning. We list the training times of five different methods in Table 2. Table 2 indicates that our method requires the longest training time.

Figure 6. MAE and MSE comparison of ten wind turbines when timesteps = 82.

Figure 7. (a) MAE and MSE comparison of the temporal convolutional network (TCN) and our method in different timesteps; (b) MAE and MSE comparison of the TCN and our method in ten wind turbines when timesteps = 82.

Table 2. Training time comparison of five different methods (CPU: Intel Xeon W-2145 @3.70 GHz, Intel, Santa Clara, CA, USA).

Method	Ours	LSTM	RNN	GRU	TCN
Training Time (s)	2774.6	23.25	13.67	43.1	251.9

4.4. Ripple Encoding

The proposed precipitation ripple encoding aims to supplement the dataset and obtain better forecasting performance since wind turbines and blades are affected by rainfall [11,28]. To investigate its efficiency, we compare the MAE and MSE results with and without ripple encoding in ten wind turbines. The results are presented in Figure 4 and Table 3 when the timesteps amount to 82.

Table 3. Metrics between methods with and without ripple encoding in ten wind turbines.

Metrics		Ripple Encoding	No Ripple Encoding
MAE	Variance	0.00012	0.00064
	Average	0.09551	0.12097
MSE	Variance	0.00001	0.00024
	Average	0.01877	0.04356

Figure 8 and Table 3 demonstrate that precipitation ripple encoding is effective. On average, ripple encoding brings a 0.02546 reduction in MAE and a 0.02479 reduction in MSE. However, the effect of ripple encoding varies between different wind turbines. The No.4 and No.10 wind turbines, respectively, reduce by 31.16% and 29.09% in MAE and 66.67% and 63.86% in MSE. Meanwhile, the No.8 wind turbine reduces by 0.21% in MAE and 4.37% in MSE. The significant differences in variance indicate that the effect of rainfall on wind turbines does not follow the same pattern. In summary, ripple encoding can effectively reduce the variance among different turbines and improve wind power forecasting performance.

Figure 8. MAE and MSE comparison of with and without ripple encoding. The curve denotes the rate of decline in different wind turbines.

5. Conclusions

This paper, for the first time, explores the potential of utilizing a convolutional neural network to forecast wind turbine power. In order to map weather data into images, we propose GWF matrices and ripple encoding to encode wind speed time series and precipitation data. Evaluation results indicate that our method delivers significant improvement for 24-h wind power generation forecast. Compared with state-of-the-art recurrent networks,

the proposed method outperforms LSTM by 28.67%, GRU by 27.53%, and RNN by 41.28%, in terms of MAE.

In addition, our dataset is extensible since the original dataset only records ten wind turbines at the same location, operation status, and weather conditions which are extremely similar. The performance of our model could be further improved with more turbine data and more wind information at different heights.

Author Contributions: Conceptualization, T.L.; Investigation, T.L.; Methodology, T.L. and Y.Z.; Supervision, T.L., Z.H., L.T., Y.Z., H.W. and S.F.; Validation, T.L.; Writing—original draft, T.L.; Writing—review & editing, T.L., Z.H. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Project (Grant No.2019YFC0117302), National Natural Science Foundation of China (Grant No.U2032125, Grant No.62004201), Shanghai Municipal Science and Technology Commission (Grant No.19511131202), Pudong Industry-University-Research Project (Grant PKX2019-D02), and Independent Deployment Project of Shanghai Advanced Research Institute (Grant E0560W1ZZ0, E052891ZZ1).

Acknowledgments: The authors would like to express their gratitude to all those who helped them during the writing of this manuscript. The authors' first gratitude goes to Zeyu Cao from Zhejiang University for the wind farm data supply. Second, they also owe their sincere gratitude to friends and colleagues who gave them much enlightening advice and encouragement.

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

References

1. Zhu, H.; Zhu, Y.; Wang, H.; Wang, S.; Liu, Z.; Balamurugan, B.; Xia, M. Multiscale deep network based multistep prediction of high-dimensional time series from power transmission systems. *Trans. Emerg. Telecommun. Technol.* **2020**, e3890. [\[CrossRef\]](#)
2. De Paula, A.; de Oliveira, E.; Oliveira, L.; Moraes, C. Reliability-constrained dynamic transmission expansion planning considering wind power generation. *Electr. Eng.* **2020**, *102*, 2583–2593. [\[CrossRef\]](#)
3. Shamshirband, S.; Rabczuk, T.; Chau, K.-W. A survey of deep learning techniques: Application in wind and solar energy resources. *IEEE Access* **2019**, *7*, 164650–164666. [\[CrossRef\]](#)
4. Zou, W.; Li, C.; Chen, P. An inter type-2 FCR algorithm based T-S fuzzy model for short-term wind power interval prediction. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4934–4943. [\[CrossRef\]](#)
5. Du, P.; Wang, J.; Yang, W.; Niu, T. A novel hybrid model for short-term wind power forecasting. *Appl. Soft Comput.* **2019**, *80*, 93–106. [\[CrossRef\]](#)
6. Yuan, X.; Tan, Q.; Lei, X.; Yuan, Y.; Wu, X. Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine. *Energy* **2017**, *129*, 122–137. [\[CrossRef\]](#)
7. Hao, Y.; Tian, C. A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. *Appl. Energy* **2019**, *238*, 368–383. [\[CrossRef\]](#)
8. Xiao, L.; Qian, F.; Shao, W. Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm. *Energy Convers. Manag.* **2017**, *143*, 410–430. [\[CrossRef\]](#)
9. Celik, A.N.; Kolhe, M. Generalized feed-forward based method for wind energy prediction. *Appl. Energy* **2013**, *101*, 582–588. [\[CrossRef\]](#)
10. Castorrini, A.; Corsini, A.; Rispoli, F.; Venturini, P.; Takizawa, K.; Tezduyar, T.E. Computational analysis of wind-turbine blade rain erosion. *Comput. Fluids* **2016**, *141*, 175–183. [\[CrossRef\]](#)
11. Wu, Z.; Cao, Y.; Nie, S.; Yang, Y. Effects of rain on vertical axis wind turbine performance. *J. Wind Eng. Ind. Aerodyn.* **2017**, *170*, 128–140. [\[CrossRef\]](#)
12. Ouyang, T.; Zha, X.; Qin, L.; He, Y.; Tang, Z. Prediction of wind power ramp events based on residual correction. *Renew. Energy* **2019**, *136*, 781–792. [\[CrossRef\]](#)
13. Zhao, J.; Chen, X.; He, M. Detection of Impending Ramp for Improved Wind Farm Power Forecasting. In Proceedings of the IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 7–8 February 2019; pp. 1–6.
14. Wang, Z.; Wang, W.; Liu, C.; Wang, B. Forecasted scenarios of regional wind farms based on regular vine copulas. *J. Mod. Power Syst. Clean Energy* **2019**, *8*, 77–85. [\[CrossRef\]](#)
15. Ozkan, M.B.; Karagoz, P. Data mining-based upscaling approach for regional wind power forecasting: Regional statistical hybrid wind power forecast technique (RegionalSHWIP). *IEEE Access* **2019**, *7*, 171790–171800. [\[CrossRef\]](#)
16. Wu, Y.-K.; Wu, Y.-C.; Hong, J.-S.; Phan, L.H.; Quoc, D.P. Probabilistic Forecast of Wind Power Generation with Data Processing and Numerical Weather Predictions. In Proceedings of the IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (I&CPS), Las Vegas, NV, USA, 29 June–28 July 2020; pp. 1–11.

17. Wang, J.; Niu, T.; Lu, H.; Yang, W.; Du, P. A novel framework of reservoir computing for deterministic and probabilistic wind power forecasting. *IEEE Trans. Sustain. Energy* **2019**, *11*, 337–349. [\[CrossRef\]](#)
18. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. *arXiv* **2017**, arXiv:1705.03122.
19. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
21. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.
24. Abueidda, D.W.; Koric, S.; Sobh, N.A.; Sehitoglu, H. Deep learning for plasticity and thermo-viscoplasticity. *Int. J. Plast.* **2020**, *136*, 102852. [\[CrossRef\]](#)
25. Wang, Z.; Oates, T. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*; AAAI: Menlo Park, CA, USA, 2015.
26. Hatami, N.; Gavet, Y.; Debye, J. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10696, p. 106960Y.
27. Chen, W.; Shi, K. A deep learning framework for time series classification using Relative Position Matrix and Convolutional Neural Network. *Neurocomputing* **2019**, *359*, 384–394. [\[CrossRef\]](#)
28. Ke, S.; Yu, W.; Wang, T.; Ge, Y. Aerodynamic performance and wind-induced effect of large-scale wind turbine system under yaw and wind-rain combination action. *Renew. Energy* **2019**, *136*, 235–253. [\[CrossRef\]](#)
29. De Oliveira, E.M.; Oliveira, F.L.C. Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. *Energy* **2018**, *144*, 776–788. [\[CrossRef\]](#)
30. Shen, Y.; Xiong, A. Validation and comparison of a new gauge-based precipitation analysis over mainland China. *Int. J. Climatol.* **2016**, *36*, 252–265. [\[CrossRef\]](#)
31. Barth, N. The gramian and k-volume in n-space: Some classical results in linear algebra. *J. Young Investig.* **1999**, *2*, 1–4.
32. Landry, M.; Erlinger, T.P.; Patschke, D.; Varrichio, C. Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. *Int. J. Forecast.* **2016**, *32*, 1061–1066. [\[CrossRef\]](#)
33. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 6–11 August 2017.
34. Kumar, D.; Mathur, H.; Bhanot, S.; Bansal, R.C. Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid. *Int. J. Model. Simul.* **2020**, 1–13. [\[CrossRef\]](#)
35. Li, C.; Tang, G.; Xue, X.; Saeed, A.; Hu, X. Short-term wind speed interval prediction based on ensemble GRU model. In *IEEE Transactions on Sustainable Energy*; IEEE: Piscataway, NJ, USA, 2019.