Short-Term Nacelle Orientation Forecasting Using Bilinear Transformation and ICEEMDAN Framework

Huajin Li[1,2], Jiahao Deng[3]*, Peng Feng[1], Chuanhao Pu[2], Dimuthu D. K. Arachchige[3] and Qian Cheng[4]

[1]School of Architecture and Civil Engineering, Chengdu University, Chengdu, China, [2]State Key Laboratory of Geo-hazard Prevention and Geo-environment Protection, Chengdu University of Technology, Chengdu, China, [3]College of Computing and Digital Media, DePaul University, Chicago, IL, United States, [4]College of Civil Engineering, Sichuan University of Science and Engineering, Zigong, China

To maximize energy extraction, the nacelle of a wind turbine follows the wind direction. Accurate prediction of wind direction is vital for yaw control. A tandem hybrid approach to improve the prediction accuracy of the wind direction data is developed. The proposed approach in this paper includes the bilinear transformation, effective data decomposition techniques, long-short-term-memory recurrent neural networks (LSTM-RNNs), and error decomposition correction methods. In the proposed approach, the angular wind direction data is firstly transformed into time-series to accommodate the full range of yaw motion. Then, the continuous transformed series are decomposed into a group of subseries using a novel decomposition technique. Next, for each subseries, the wind directions are predicted using LSTM-RNNs. In the final step, it decomposed the errors for each predicted subseries to correct the predicted wind direction and then perform inverse bilinear transformation to obtain the final wind direction forecasting. The robustness and effectiveness of the proposed approach are verified using data collected from a wind farm located in Huitengxile, Inner Mongolia, China. Computational results indicate that the proposed hybrid approach outperforms the other single approaches tested to predict the nacelle direction over short-time horizons. The proposed approach can be useful for practical wind farm operations.

Keywords: wind direction, bilinear transformation, ICEEMDAN, LSTM-RNN, error correction

INTRODUCTION

Wind energy generation is expanding with about 12% of world's electricity to be supplied by 2020 (Kodama and Burls 2019). Compared with the traditional form of power generation, wind energy has the advantages of zero pollution and low operation cost. Hence, it has become one of the fastest growing renewable energy power supplies globally (Duan et al., 2021).

Although it has obvious advantages over others, wind energy still faces technical challenges due to the characteristics of chaos, randomness, and intermittence which make the wind data complex. The wind direction is one of the most complex aspect of the wind data due to its high dynamics in both spatial and temporal domains. To follow the wind direction, the nacelle of a wind turbine orientes the controlling of yaw and maximizes the energy output. For most efficient energy extraction, the nacelle

orientation of a wind turbine needs to agree with wind direction which calls for accurate and prediction of the wind direction (Hu et al., 2016).

According to literature review, statistical approaches based on meteorological and geographic information are widely applied to forecast wind direction (Mcwilliams and Sprevak 1982; Castino et al., 1998; Erdem and Shi 2011). Liu et al. (2010) applied a neural Kriging method to spacially estimate the distribubiton of wind directions. Erdem and Shi (2011) developed autoregressive moving average (ARMA) model to forecast the short-term wind directions. Masseran et al. (2013) used a mixture of Von Mises models to fit the wind direction series.

Therefore, machine learning adoptions for wind direction forecasting have evolved from the classic approach to deep learning, which is then improved in this study (Mohandes et al., 2004; Bilgili et al., 2007). In the wind direction forecasting sector, Zhou et al. (2011) selected least-square support vector machines (LS-SVM) to predict the wind directions. Tagliaferri et al. (2015) developed artificial neural networks to forecast the short-term wind directions. Khosravi et al. (2018) developed an adaptive neuro-fuzzy inference system to predict the wind directions. Amin et al. (2018) improved the wind direction forecasting using the echo state network (ESN) which is a deep-learning algorithm. Tang et al. (2021) integrated the ESN network with IFPA optimization algorithm and developed a two-step deep-learning wind direction framework.

Considering the complexity and high dynamics of the wind direction series, additional measures are essential to study in the pattern inside. Even though deep learning algorithms have achieved promising results in the field of time-series prediction, it is still challenging for a single deep-learning approach to adapt all wind direction patterns. To further improve the prediction performance, hybrid prediction models are considered to be the mainstream since last year. The signal decomposition is one of the most popular components within the hybrid models published. It contains wavelet decomposition (Liu et al., 2014), empirical mode decomposition (EMD) (Santhosh et al., 2018), complete ensemble empirical model decomposition (CEEMD) (Zhang et al., 2017), complete EEMD with adaptive noise (CEEMDAN) (Yang and Wang 2018), and the improved CEEMDAN (ICEEMDAN) (Rong et al., 2019). In particular, the ICEEMDAN has demonstrated its superior performance in decomposing a complex signal into a finite number of intrinsic mode functions with transient frequencies. The decomposed subseries contains the detailed characteristics of the signal and can essentially reflect the spatial and temporal patterns of the wind direction series (Kou et al., 2020).

Based on the above considerations, in this research, we propose a new hybrid approach combining ICEEMDAN and error correction methods for short-term wind direction forecasting. First, the angular wind direction data has been transformed *via* bilinear transformation. Then, the transformed wind direction series are decomposed into a series of relatively simple subseries by the ICEEMDAN modules. Next, the LSTM-RNN is established as the prediction module to predict each sub-series. After that, the prediction errors are obtained and decomposed by ICEEMDAN modules.

The statistical ARIMA model is used to predict the error subsequence and compute the prediction error. In the last step, the final prediction of the wind direction is made by summing all predicted subseries together with current predicted error and then transformed into angular data by inverse bilinear transformation.

The major contribution of this research can be summarized as follows: First, the wind direction forecasting system based on ICEEMDAN decomposition, LSTM-RNN and error correction has been proposed; Second, the comparative analysis is performed against other benchmarking deep-learning algorithm; Third, the experiments were performed in different seasons to explore seasonal patterns of wind directions.

The remainder of the manuscript is configured as follows. In Section "dataset description and transformation", it summarizes the data collection process and patterns inside the wind direction dataset. In Section "methodologies", it introduces the ICEEMDAN decomposition, LSTM-RNN, error correction, benchmarking deep-learning algorithms, and error correction procedures. The experimental results are provided in Section "experimental results" and the Conclusion is made in Section "conclusions" respectively.

DATASET DESCRIPTION AND TRANSFORMATION

Data Analysis

In this study, the data has been collected during the year of 2020 from a wind farm namely Huitengxile wind power plant in Inner Mongolia, Northern China. It is one of the largest wind farms in Asian and it's located in the suburbs between Chaha'er youyi zhongqi and Ulanqab city. The whole wind farm has multiple wind turbines that are distributed in an open flat grassland which provides rich wind resources. The prevailing wind directions are northwest and southeast which are very stable in recent years. The location and the annual wind rose diagrams has been illustrated in **Figure 1** below.

According to **Figure 1B**, the two prevailing wind directions, around $180°$ and $315°$ are visible. The geographic center coordinate is $112°40'E$ and $41°05'N$. It's annual average wind speed at 10 m height is 7.2 m/s and its annual average wind speed at 40 m height is 8.8 m/s. In the wind farm, the annual average air density is 1.07 kg/m and it contains an effective wind speed of 5–25 m/s with strong stability and high quality.

The data used in this research has been collected by the supervisory control and data acquisition (SCADA) system. Usually, data on more than 100 parameters at 10 s intervals is collected and stored in a SCADA system. The SCADA collected data of individual wind turbines is streamed to a central computer for condition monitoring, performance evaluation, and other forms of analysis.

In this research, the SCADA data collected at 20 wind turbines over the period of 2020 has been analyzed. According to **Figure 1B**, there are two annual prevailing wind directions and it can be partitioned into four seasons independently as illustrated with the wind roses in **Figure 2**. In the fall and spring,
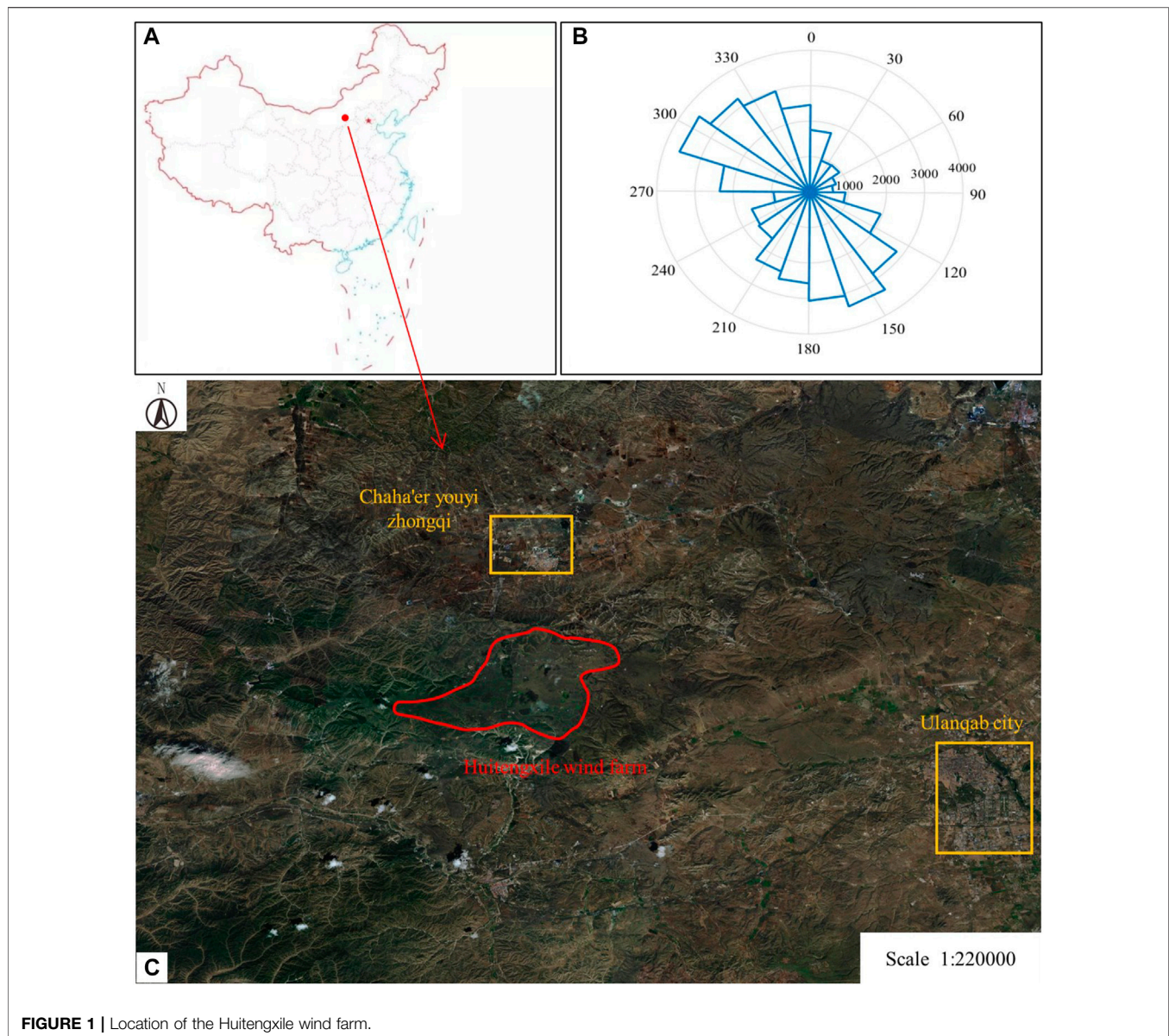
FIGURE 1 | Location of the Huitengxile wind farm.

two prevailing wind directions around 150˚ and 315˚are observed. In the winter and summer, one prevailing wind direction is noted. Since the wind direction data is captured as a discrete angular variable, it needs to be transformed for modeling. A bilinear transformation of the angular wind direction is applied in the next section.

Bilinear Transformation of Angular Data

The value of wind direction ranges from 0˚ to 360˚. It is likely that the wind direction may change from the interval, i.e (0˚, 10˚) to (350˚, 360˚). Practice shows that bilinear transformation is a better way for transforming discrete wind direction data to continuous data than the sine and cosine transformation (Peng et al., 2020). Geometrically, the two intervals are close to each other and therefore this change would lead to a large prediction error (Bilgili et al., 2007). To avoid such error, transformation of the

discrete angular variable into a standardized continuous variable is essential. One option is to use a sine and cosine transformation which is not the best approach due to two variables needed for prediction which enlarges the prediction errors. A better option is to apply a bilinear transformation (Jury 1973).

The bilinear transformation maps the analog plane (s-plane) into the digital plane (z-plane) (Groutage et al., 2003) (see **Figure 3**). The transformation function, the ratio of two polynomials (Davies 1974), is expressed in **Eq. 1**.

$$H(s)_Z = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s} \qquad (1)$$

where: s is the original value of angular variable in s-plane; T is the time interval of the transformation. The bilinear transformation expressed the angular variable between 0˚ and 360˚ as continuous
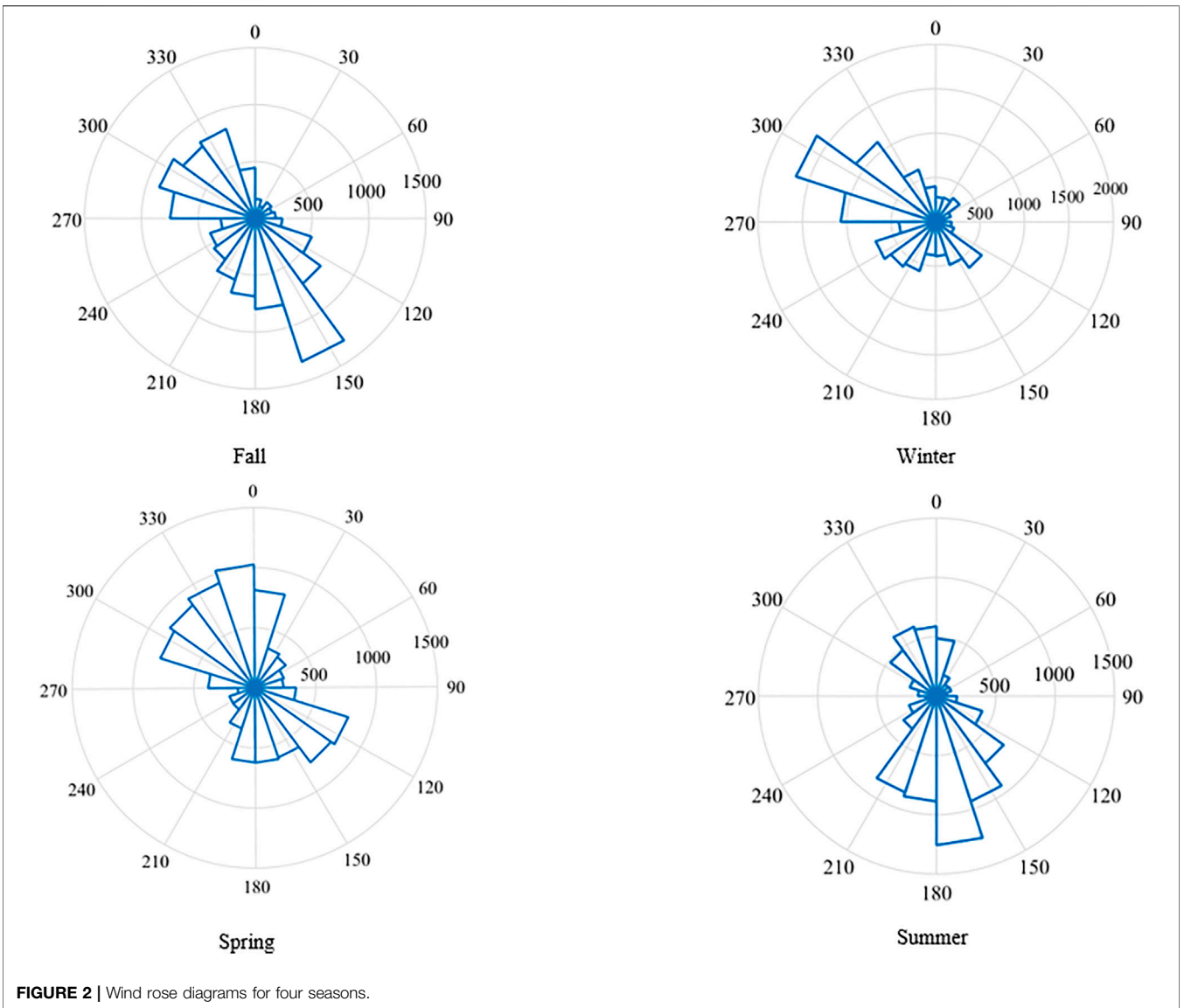
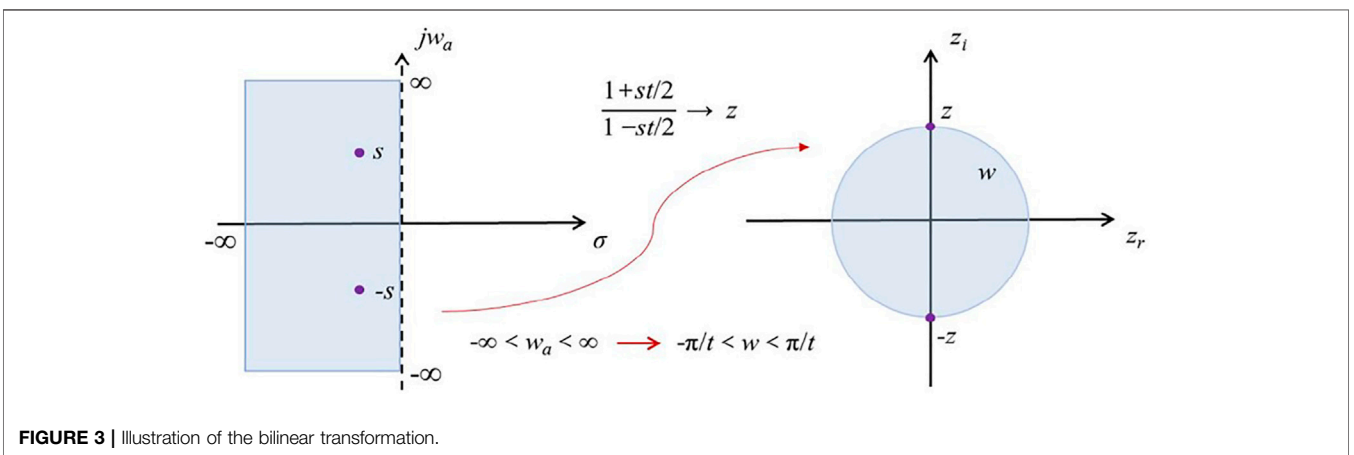FIGURE 2 | Wind rose diagrams for four seasons.

FIGURE 3 | Illustration of the bilinear transformation.

and normalized. The inverse bilinear transformation is expressed in **Eq. 2**.

$$s = \frac{1}{T} \frac{2 - 2H(s)_Z^{-1}}{2 + 2H(s)_Z^{-1}} \tag{2}$$

where: s is the inversed value of angular variable in s-plane; and $H(s)z$ is the transformed angular variable.

Since the wind direction data is noisy, a bilinear transformation function acting as a low-pass filter in the continuous-time domain reduces the noise (Davies 1974). A prediction model developed with the transformed data is more accurate than the model based on the discrete time-series angular data.

METHODOLOGIES

The use of deep learning algorithms in regression, multi-class classification, collaborative filtering, and graphic learning is growing (Lecun et al., 2015). The concept of deep learning originates from research in neural networks and it avoids the local optima dilemma. However, any single deep learning algorithms can offer limited extraction of patterns inside the dataset. Hybrid frameworks containing multiple deep learning algorithm is becoming the new mainstream in academia.

ICEEMDAN

In this research, the improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) is served as the major module in the hybrid forecasting framework. It is considered as an improvement on empirical mode decomposition (EMD) which decomposes the wind directions in the temporal domain (Colominas et al., 2014).

The time-series of wind direction can be expressed as the sum of multiple IMFs and the residual after the ICEEMDAN decomposition which can be expressed in **Eq. 3** as follows:

$$H(t) = \sum_{j=1}^{n} IMF_j(t) + r(t) \tag{3}$$

The amplitude energy $E_1, E_2, \ldots, E_n$ of the IMFs is calculated as **Eq. 4**:

$$E_j = \sum_{k=1}^{N} |IMF_j(k)|^2 \tag{4}$$

where N denotes the total number of sampling points of the jth IMF. Assuming that the energy carried by $r(t)$ can be ignored, the total energy of the transformed direction series can be expressed as **Eq. 5** as follows:

$$E = \sum_{j=1}^{n} \sum_{k=1}^{N} |IMF_j(k)|^2 \tag{5}$$

To remain the data in the same magnitude, the amplitude of the IMFs is normalized to facilitate the subsequent calculations and the impact of singular data has been reduced. Hence, the energy entropy of the ICEEMDAN framework can be expressed as **Eq. 6** below:

$$E = -\sum_{i=1}^{n} p_j \ln p_j \tag{6}$$

Compared with other decomposition methods, the ICEEMDAN can not only reduce the noise in the original time-series data but also reduce the residual spurious pattern problems based by signal overlap. Thus, the decomposed subseries gains more orthogonality among each other and it can provide more accurate reconstruction of the original series.

Short-Term Wind Direction Forecasting Using ICEEMDAN

To integrate the wind direction series with the ICEEMDAN modules, the implementations are introduced as follows (Duan et al., 2021):

Step 1: Compute the local means of realizations using the EMD algorithm described in **Eq. 7**:

$$x^i = x + \beta_0 E_1(w^i) \tag{7}$$

where $\beta_0 = \varepsilon_0 \sigma(x)/\sigma(E_1(w^i))$; $\sigma()$ compute the standard deviation; and ε_0 is the reciprocal of the desired signal-to-noise ratio between the first added noise and the analyzed signal.

Step 2: Compute residual term R_1 in the first component using **Eq. 8**:

$$R_1 = M(x^i) \tag{8}$$

Step 3: Compute the first mode at the first stage (k = 1) using **Eq. 9**:

$$d_1 = x - R_1 \tag{9}$$

Step 4: Estimate the second residue as the average of local means of the realizations $R_1 + \beta_1 E_2(w^i)$ and then define the second mode using **Eq. 10** as follows:

$$d_2 = R_1 - M(R_1 + \beta_1 E_2(w^i)) \tag{10}$$

Step 5: For the other terms (k = 3, . . . ,K) of residuals, they can be computed by **Eq. 11**:

$$R_k = M(R_{k-1} + \beta_{k-1} E_k(w^i)) \tag{11}$$

Step 6: Compute the other terms (k = 3, . . . ,K) of the mode by **Eq. 12**:

$$d_k = R_{k-1} - R_k \tag{12}$$

Step 7: Implement step 4 for the next iteration.

For the transformed wind direction series, the IMF components are obtained *via* the above steps which can be illustrated by the diagram presented in **Figure 4**.
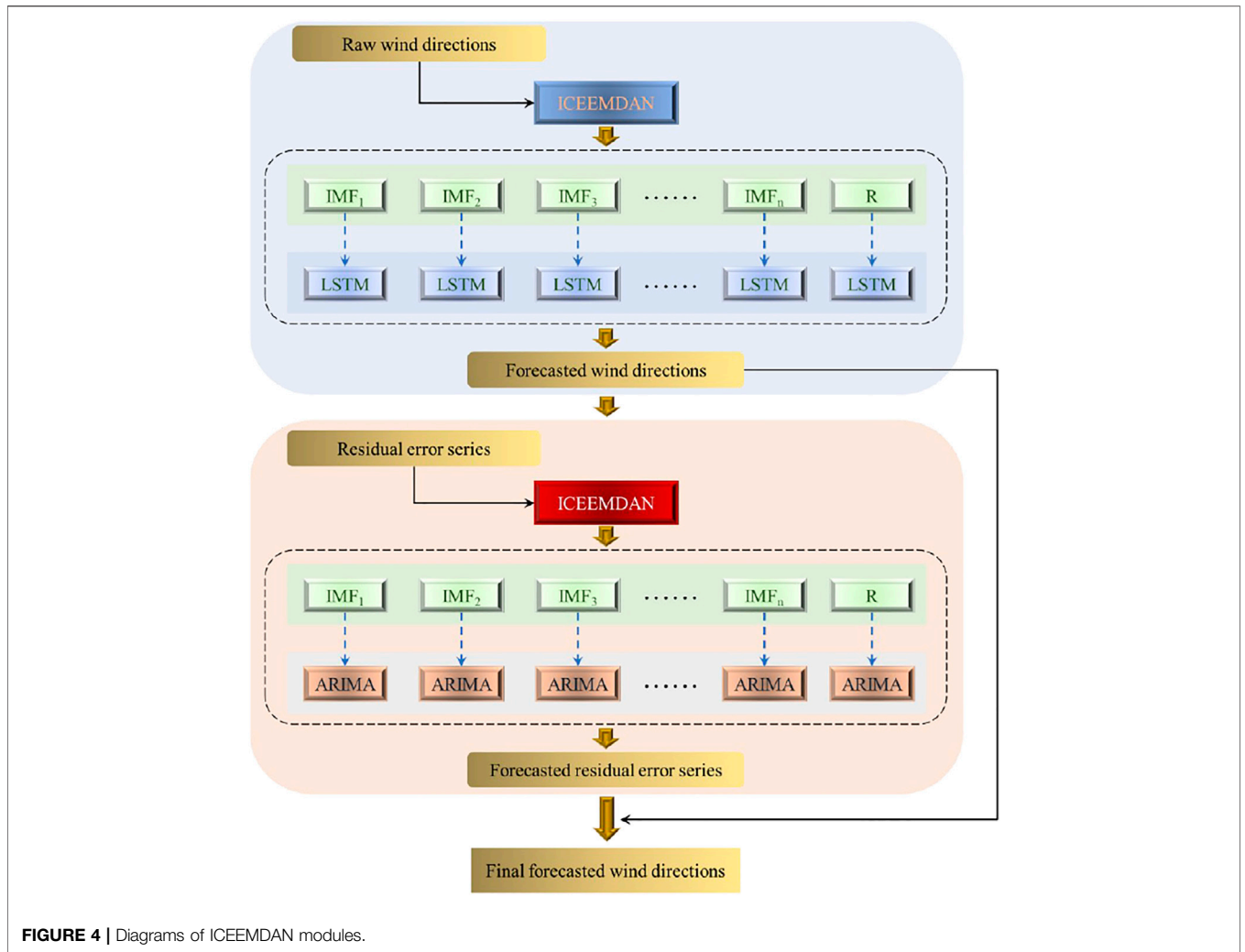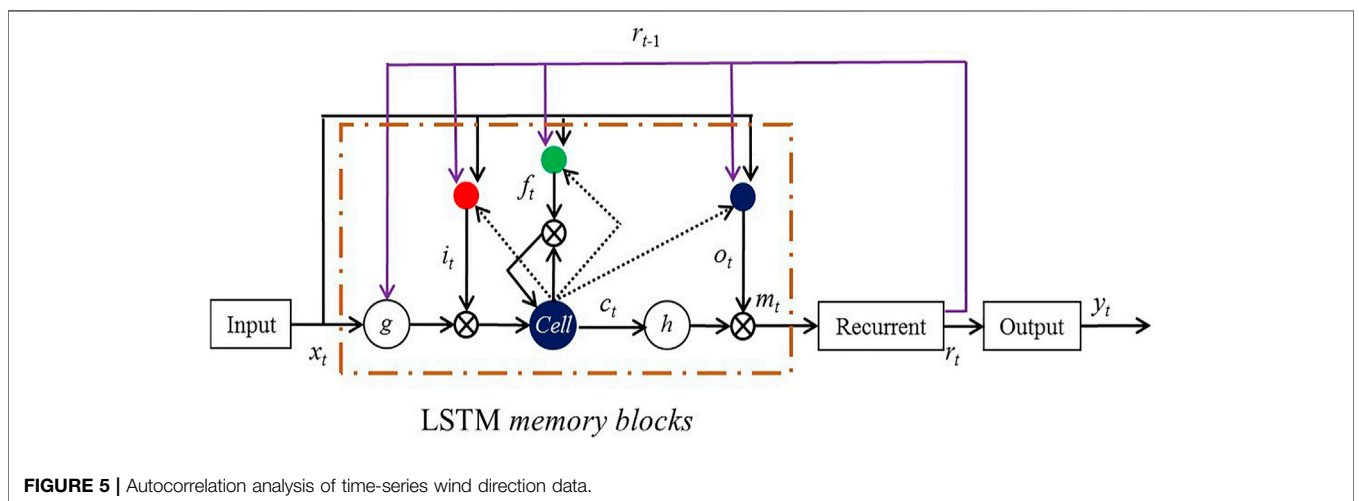
FIGURE 4 | Diagrams of ICEEMDAN modules.

FIGURE 5 | Autocorrelation analysis of time-series wind direction data.

Long-Short-Term Memory Recurrent Neural Network

A major drawback of the classical deep neural networks is that they do not have memory of the past periods. The time series information such as the past clusters of seasonal patterns and seasonal trend may not be reflected (Lee et al. 1993). Introduced by Hochreiter and Schmidhuber (1997) and Gers et al. (2003), the long-short-term memory recurrent neural network (LSTM-RNN) matches the needs and it is used in this paper to predict wind direction.

The long-short-term memory recurrent neural network (LSTM-RNN) contains units called memory blocks composed of memory cells with self-connections storing temporal states. Each memory block includes an input and output gate. The input gate controls the flow of input data into the cell. The output gate controls the output data flow into the rest of the network (Sak et al. 2014). In addition, the LSTM-RNN has peephole connections (Gers et al. 2003) from its internal cells to the gates in the same cell to learn precise timing of the output. The architecture of LSTM-RNN is illustrated in **Figure 5**.

With a long-short-term memory recurrent neural network (LSTM-RNN) architecture, the mapping from an input to an output layer is iteratively computed from **Eqs 13–18** (Gers et al. 2003).

$$i_t = sig\left(W_{ix}\boldsymbol{x}_t + W_{im}\boldsymbol{m}_{t-1} + W_{ic}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i\right) \tag{13}$$

$$f_t = sig\left(W_{fx}\boldsymbol{x}_t + W_{fm}\boldsymbol{m}_{t-1} + W_{ic}\boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \tag{14}$$

$$c_t = f_t \circ \boldsymbol{c}_{t-1} + i_t \circ g\left(W_{cx}\boldsymbol{x}_t + W_{cm}\boldsymbol{m}_{t-1} + \boldsymbol{b}_c\right) \tag{15}$$

$$o_t = sig\left(W_{ox}\boldsymbol{x}_t + W_{om}\boldsymbol{m}_{t-1} + W_{oc}\boldsymbol{c}_t + \boldsymbol{b}_o\right) \tag{16}$$

$$\boldsymbol{m}_t = o_t \circ h\left(\boldsymbol{c}_t\right) \tag{17}$$

$$\boldsymbol{y}_t = \phi\left(W_{ym}\boldsymbol{m}_t + \boldsymbol{b}_y\right) \tag{18}$$

where: W are the weight matrices (i.e., W_{ix} is the weight matrix from the input to the input layer; W_{ic}, W_{fc}, W_{oc} are diagonal weight matrices of the peephole connections (Gers et al., 2003)); b_i, b_f, b_o, and b_c are the bias vectors; m is the cell output activation vector; sig () is the sigmoid function; i, f, o, and c are the input gate, forget gate, output gate, and cell activation vectors, respectively, with all having the same size as the cell output activation vector m; $\circ$ is the element-wise product of the vectors; and g () and h () are the cell input and cell output activation functions, respectively.

Benchmarking Machine Learning Algorithms

Comparative analysis is performed in this research against the other benchmarking popular deep learning algorithms. All algorithms tested here are using the same ICEEMDAN framework as described in Section "Short-term Wind Direction Forecasting using ICEEMDAN". The benchmarking deep learning algorithms compared includes deep neural network (DNN) (Xu et al., 2018; Sun et al., 2020; Yi and Xu, 2020), deep belief network (DBN) (Ouyang et al., 2019; Li et al., 2020), kernel-based extreme learning machine (KELM) (Li et al., 2018; Ouyang

et al., 2018), and gated recurrent unit network (GRU) (Pan et al., 2019; Tang and Zhang, 2019).

The DNN is a fully connected feedforward network that consists of a cascade of multiple layers and hidden units. It's structure with multiple processing layers enables it to handle highly nonlinear patterns inside the dataset. The deep temporal representations in the temporal domain can be effectively extracted by DNN.

Similar to DNN, the DBN consists of multiple layers of restricted Boltzmann machines (RBMs). It also contains a supervised regression layer stacked on the top of all RMBs for classification or regression tasks. Inside each RBM, it contains an input layer and a hidden layer with hidden-to-all-visible connections.

The ELM is a single hidden layer feedforward network. Instead of conventional back-propagation, it uses Penn-Moore pseudo-inverse to compute the wights between the hidden layer and output layer. The KELM is the improvement of vanilla ELM which uses the kernel matrix to replace the randomly initialized weights between the input layer and output layer. The most popular applied kernel functions include RBF, linear function, and polynomial function.

The GRU is another type of recurrent neural network other than LSTM-RNN proposed by Cho et al. (2014). In a typical GRU unit, it has one less gate than the LSTM unit and consists of two gates: the reset gate and the update gate. Hence, the GRU is also popular in modeling time-series dataset.

Measurement Matrices

To assess prediction accuracy of the proposed deep learning model, four metrics are computed: the MAE [Mean absolute error (**Eq. 19**)], the MAPE [Mean absolute percentage error (**Eq. 20**)], the MSE [Mean square error (**Eq. 21**)], and the RMSE [Root mean square error (**Eq. 22**)].

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|o_j - t_j\right| \tag{19}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{o_j - t_j}{t_j}\right| \tag{20}$$

$$\text{MSE} = \frac{1}{N}\sum_{j=1}^{N}o_j - t_j^2 \tag{21}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}o_j - t_j^2} \tag{22}$$

where: o_j is the jth predicted wind direction; t_j is the jth measured wind direction; and N denotes total number of samples.

Error Correction

To improve the forecasting accuracy, the error correction is implemented in this research. First, after the forecasting outcome produced by each LSTM-RNN, the error series E(t) of the training dataset can be computed by comparing the original transformed wind direction series. The step can be expressed in **Eq. 23** as follows:

$$E\left(t\right) = W_{LSTM}\left(t\right) - W_{Actual}\left(t\right) \tag{23}$$
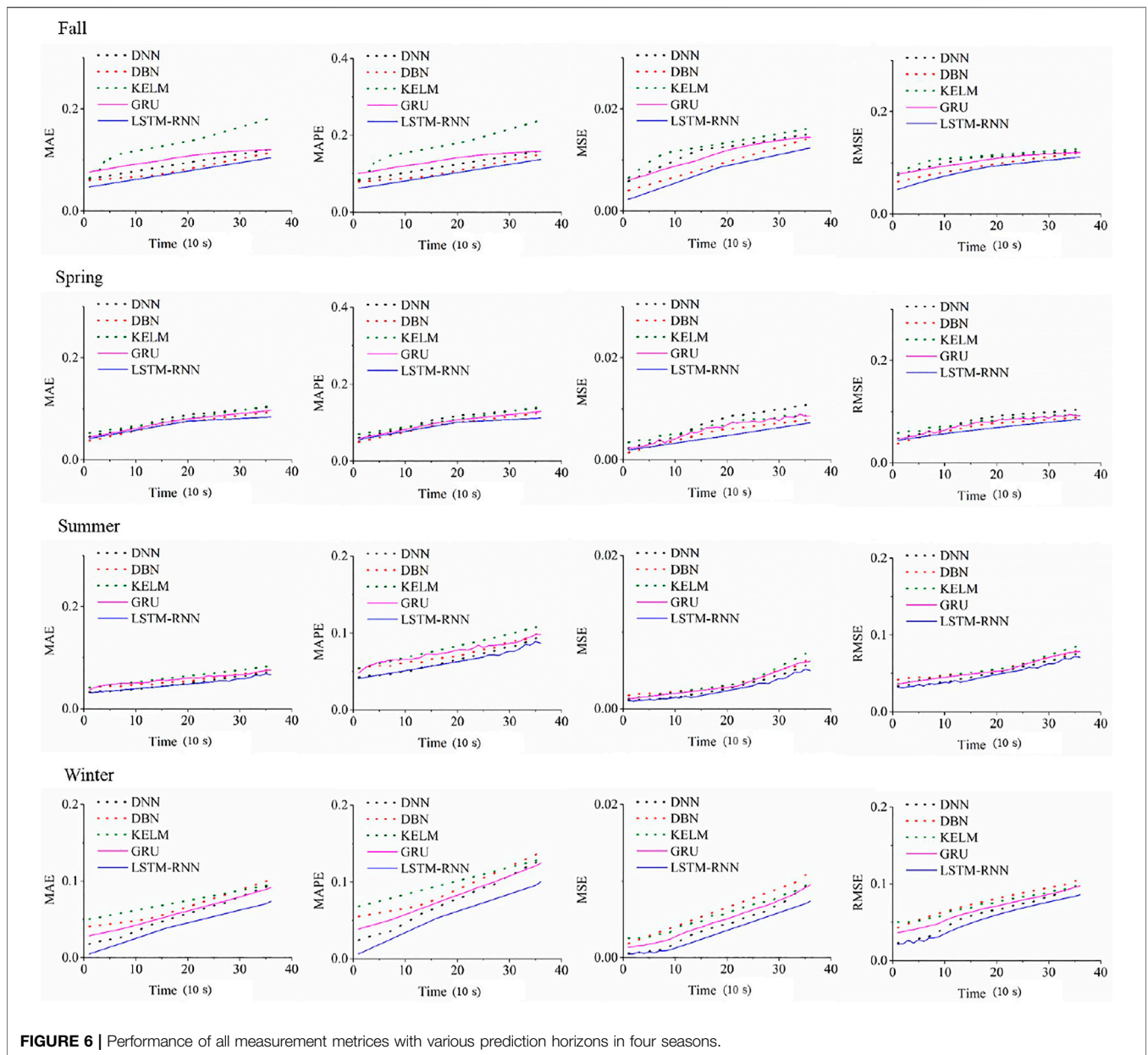
FIGURE 6 | Performance of all measurement metrices with various prediction horizons in four seasons.

where $W_{LSTM}(t)$ is the final forecasted transformed series in the first module; and $W_{Actual}(t)$ is the actual transformed wind direction series after bilinear transformation.

The forecasted errors of wind direction series E(t) are oscillatory in the time-series domain (Wasynczuk et al., 1981). The relationship between oscillatory and decaying property of the wind direction errors can be represented by an ARIMA model which predicts the errors. In detail, the ARIMA can be constructed by computing autocorrelation expressed with the autocorrelation factor (ACF) (See. **Eq. 24**) and the partial autocorrelation factor (PACF)) (See. **Eq. 25**). Here, Cov() denotes the covariance; Var() denotes the variance; and *Corr* () denotes the Pearson's correlation coefficient.

$$\rho_k = \frac{Cov(E(t), E(t-k))}{Var(E(t))} \tag{24}$$

$$\rho_k = Corr(E(t), E(t-k)|E(t-1), \dots, E(t-k+1)) \tag{25}$$

For each IMF, an ARIMA model is developed and then all outcomes of each ARIMA are integrated to obtain the final error series. Last, as illustrated in **Figure 4**, the final prediction is achieved by **Eq. 26** as follows:

$$\widehat{W_{Final}(t)} = \widehat{W_{LSTM}(t)} + \hat{E}(t) \tag{26}$$

where $\widehat{W_{LSTM}(t)}$ denotes the forecasting results from the LSTMs; $\hat{E}(t)$ denotes the errors forecasted by ARIMA models; and $\widehat{W_{Final}(t)}$ is the final forecasting outcomes.

TABLE 1 | Mape of the five algorithms for 2020 before error correction.

Season	Resolution (s)	Algorithm	Prediction horizon				Season	Resolution (s)	Algorithm	Prediction horizon			
			2 min	5 min	10 min	1 h				2 min	5 min	10 min	1 h
Spring	10	DNN	0.082	0.106	0.133	0.211	Summer	10	DNN	0.047	0.057	0.092	0.181
		DBN	0.072	0.09	0.121	0.207			DBN	0.056	0.064	0.097	0.182
		KELM	0.081	0.098	0.137	0.224			KELM	0.062	0.074	0.106	0.186
		GRU	0.077	0.095	0.125	0.211			GRU	0.059	0.07	0.095	0.181
		LSTM-RNN	**0.072**	**0.091**	**0.108**	**0.195**			LSTM-RNN	**0.047**	**0.056**	**0.084**	**0.171**
	20	DNN	0.088	0.112	0.139	0.269		20	DNN	0.049	0.061	0.099	0.221
		DBN	0.077	0.093	0.127	0.252			DBN	0.056	0.065	0.105	0.227
		KELM	0.085	0.102	0.141	0.273			KELM	0.065	0.077	0.113	0.228
		GRU	0.082	0.101	0.131	0.264			GRU	0.064	0.073	0.104	0.227
		LSTM-RNN	**0.076**	**0.094**	**0.117**	**0.231**			LSTM-RNN	**0.049**	**0.058**	**0.096**	**0.214**
	30	DNN	0.096	0.114	0.15	0.33		30	DNN	0.052	0.066	0.112	0.278
		DBN	0.082	0.099	0.139	0.316			DBN	0.058	0.072	0.116	0.277
		KELM	0.089	0.109	0.153	0.328			KELM	0.068	0.083	0.125	0.29
		GRU	0.087	0.105	0.142	0.311			GRU	0.063	0.078	0.116	0.286
		LSTM-RNN	**0.081**	**0.096**	**0.128**	**0.305**			LSTM-RNN	**0.051**	**0.063**	**0.108**	**0.261**
Fall	10	DNN	0.094	0.112	0.154	0.239	Winter	10	DNN	0.049	0.069	0.125	0.212
		DBN	0.079	0.095	0.145	0.226			DBN	0.06	0.078	0.134	0.213
		KELM	0.14	0.163	0.232	0.302			KELM	0.077	0.091	0.125	0.213
		GRU	0.109	0.128	0.153	0.232			GRU	0.055	0.073	0.12	0.209
		LSTM-RNN	**0.075**	**0.092**	**0.133**	**0.211**			LSTM-RNN	**0.035**	**0.053**	**0.097**	**0.183**
	20	DNN	0.098	0.116	0.156	0.285		20	DNN	0.053	0.072	0.127	0.247
		DBN	0.082	0.101	0.147	0.279			DBN	0.064	0.084	0.136	0.264
		KELM	0.145	0.167	0.224	0.365			KELM	0.08	0.094	0.131	0.256
		GRU	0.113	0.131	0.158	0.285			GRU	0.059	0.077	0.124	0.243
		LSTM-RNN	**0.079**	**0.095**	**0.137**	**0.266**			LSTM-RNN	**0.041**	**0.057**	**0.102**	**0.234**
	30	DNN	0.102	0.124	0.168	0.364		30	DNN	0.058	0.084	0.14	0.314
		DBN	0.086	0.11	0.16	0.353			DBN	0.069	0.095	0.149	0.33
		KELM	0.149	0.179	0.237	0.459			KELM	0.083	0.101	0.142	0.313
		GRU	0.117	0.138	0.169	0.357			GRU	0.064	0.087	0.136	0.321
		LSTM-RNN	**0.083**	**0.104**	**0.149**	**0.332**			LSTM-RNN	**0.046**	**0.066**	**0.114**	**0.276**

Bold only shows the optimal solution results.
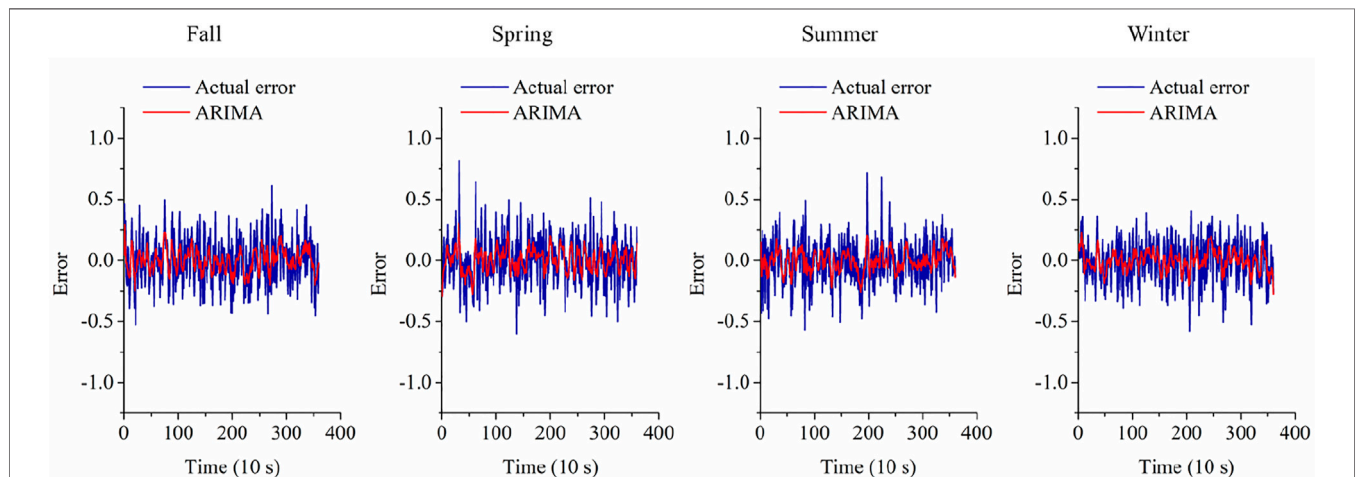

FIGURE 7 | Actual error versus forecasted error by ARIMA models.

EXPERIMENTAL RESULTS

Training Strategies

In this section, computational experience with models predicting wind direction is presented. Wind data from four seasons, spring, summer, fall, and winter are used. Prediction of wind direction is conducted using dataset at 10, 20, and 30 s resolution. The prediction horizons are 2, 5, 10 min, and 1 h. The prediction model is expressed in **Eqs 27–29**.

$$Y = f(D) \qquad (27)$$

TABLE 2 | Mape of the five algorithms for 2020 after error correction.

Season	Resolution (s)	Algorithm	Prediction horizon				Season	Resolution (s)	Algorithm	Prediction horizon			
			2 min	5 min	10 min	1 h				2 min	5 min	10 min	1 h
Spring	10	DNN	0.082	0.106	0.133	0.211	Summer	10	DNN	0.047	0.057	0.092	0.181
		DBN	0.072	0.094	0.121	0.207			DBN	0.056	0.064	0.097	0.182
		KELM	0.081	0.098	0.137	0.224			KELM	0.062	0.074	0.106	0.186
		GRU	0.077	0.095	0.125	0.211			GRU	0.059	0.077	0.095	0.181
		LSTM-RNN	**0.072**	**0.091**	**0.108**	**0.195**			LSTM-RNN	**0.047**	**0.056**	**0.084**	**0.171**
	20	DNN	0.088	0.117	0.139	0.269		20	DNN	0.049	0.061	0.099	0.221
		DBN	0.077	0.093	0.127	0.252			DBN	0.056	0.065	0.105	0.227
		KELM	0.085	0.102	0.141	0.273			KELM	0.065	0.077	0.113	0.228
		GRU	0.082	0.146	0.131	0.264			GRU	0.064	0.073	0.104	0.227
		LSTM-RNN	**0.076**	**0.094**	**0.117**	**0.231**			LSTM-RNN	**0.049**	**0.058**	**0.096**	**0.214**
	30	DNN	0.096	0.114	0.154	0.332		30	DNN	0.052	0.066	0.112	0.278
		DBN	0.082	0.099	0.139	0.316			DBN	0.058	0.072	0.116	0.277
		KELM	0.089	0.109	0.153	0.328			KELM	0.068	0.083	0.125	0.294
		GRU	0.087	0.105	0.142	0.311			GRU	0.063	0.078	0.116	0.286
		LSTM-RNN	**0.081**	**0.096**	**0.128**	**0.305**			LSTM-RNN	**0.051**	**0.063**	**0.108**	**0.261**
Fall	10	DNN	0.071	0.096	0.117	0.204	Winter	10	DNN	0.036	0.056	0.092	0.181
		DBN	0.06	0.083	0.113	0.214			DBN	0.044	0.065	0.098	0.182
		KELM	0.107	0.142	0.176	0.254			KELM	0.056	0.076	0.092	0.179
		GRU	0.083	0.115	0.116	0.202			GRU	0.041	0.061	0.088	0.179
		LSTM-RNN	**0.057**	**0.079**	**0.101**	**0.185**			LSTM-RNN	**0.026**	**0.044**	**0.071**	**0.161**
	20	DNN	0.074	0.121	0.124	0.255		20	DNN	0.039	0.06	0.099	0.221
		DBN	0.062	0.086	0.117	0.243			DBN	0.047	0.071	0.106	0.234
		KELM	0.117	0.144	0.175	0.319			KELM	0.059	0.079	0.101	0.215
		GRU	0.086	0.114	0.125	0.239			GRU	0.044	0.064	0.097	0.212
		LSTM-RNN	**0.061**	**0.083**	**0.109**	**0.228**			LSTM-RNN	**0.03**	**0.048**	**0.081**	**0.205**
	30	DNN	0.077	0.107	0.135	0.305		30	DNN	0.043	0.074	0.111	0.275
		DBN	0.065	0.095	0.129	0.291			DBN	0.051	0.079	0.118	0.277
		KELM	0.113	0.155	0.187	0.387			KELM	0.061	0.084	0.113	0.285
		GRU	0.089	0.119	0.136	0.314			GRU	0.047	0.072	0.108	0.274
		LSTM-RNN	**0.063**	**0.089**	**0.122**	**0.282**			LSTM-RNN	**0.034**	**0.055**	**0.092**	**0.242**

Bold only shows the optimal solution results.

$$D = [D_{t-1}, D_{t-2}, D_{t-3}, D_{t-4}, D_{t-5}, D_{t-6}] \tag{28}$$

$$D_{t-i} = (x_{t-i}, \ldots, x_{t-i-359}) \tag{29}$$

where: $f(D)$ represents the whole framework illustrated in **Figure 4**; D_{t-i} is the ith lagged vector containing 1 hour of the historical wind direction data; and x_{t-i} denotes the ith lagged transformed wind direction series.

The wind speed and wind direction are 10 s data. One hour of data (360 data points) is used as the input vector. The six-time lagged vectors containing 1 hour of the historical wind direction data are selected as the input vectors. The wind direction of the 2, 5, 10 min, and 1 h horizon is predicted. The input vector is normalized beforehand and the predicted values are inverse-normalized.

Short-Term Predictions

Based on the training strategy stated in Section "Training Strategies", experiments with the five selected algorithms have been performed. In all experiments, the wind direction has been predicted for the next 2, 5, 10 min, and 1 h. Experiments have been conducted in each of the four seasons of 2020.

The prediction accuracy results in **Figure 6** demonstrate that the long-short-term memory recurrent neural network (LSTM-RNN) performs better over short-term horizons than the other

algorithms. Since the LSTM-RNN contains long/short term memory, it produces smaller prediction errors than the DNN, DBN, KELM, and GRU. For the short-term horizons (i.e., 2 and 5 min), prediction accuracy of all five algorithms is similar. However, the LSTM-RNN provides more promising results for longer-term predictions (i.e., 10 min and 1 h) of wind direction.

The prediction accuracy in four seasons varies. In the fall and spring season, the prediction errors are larger than the errors in the summer and winter season. This is due to a larger variability of the wind direction over short-term horizons. Hence, training specific prediction models in different seasons is necessary.

Table 1 provide the MAPE for different resolution data (i.e., 10, 20, 30 s) and different prediction horizons (i.e., 2, 5, 10 min, and 1 h) before the error correction. Obviously, the MAPE errors are smaller for the 10 s data than for 20 and 30 s. With the increase of the prediction horizon, the MAPEs increase. The LSTM-RNN algorithm has the smallest MAPE at all resolutions and all prediction horizons. Therefore, it is an effective algorithm for wind direction prediction at short-term horizons.

Error Correction

To correct the errors made by the ICEEMDAN modules, the ARIMA model has been developed to forecast the errors. In the
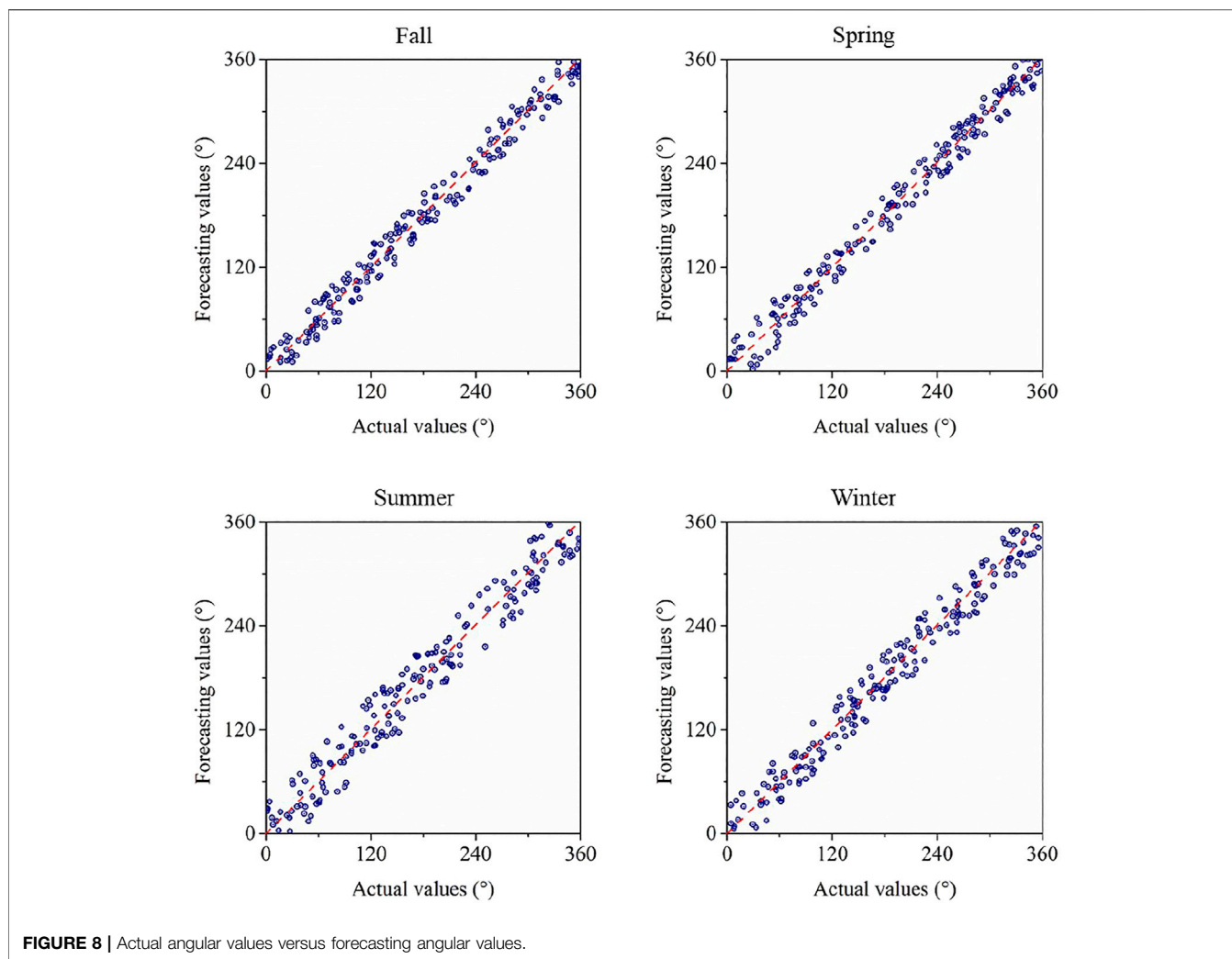
FIGURE 8 | Actual angular values versus forecasting angular values.

TABLE 3 | Angular prediction error at four seasons.

Season	Fall	Winter	Spring	Summer
MAE (2 min)	1.88°	1.67°	1.92°	1.65°
MAPE (2 min)	0.86%	0.79%	0.95%	0.74%
MAE (5 min)	5.18°	5.01°	5.57°	4.93°
MAPE (5 min)	2.37%	2.28%	2.75%	2.21%
MAE (10 min)	6.53°	6.25°	6.97°	6.10°
MAPE (10 min)	3.18%	3.01%	3.36%	2.96%
MAE (1 h)	10.64°	9.05°	10.58°	9.34°
MAPE (1 h)	5.62%	4.39%	5.29%	4.71%

second part of **Figure 4**, To illustrate this step, the forecasted errors using ARIMA versus the actual errors produced by LSTM-RNNs are visualized in **Figure 7**. It is obvious that the aggregated results from ARIMAs can represent the temporal trend of the forecasted errors produced from the first component of the proposed framework.

Table 2 provides the MAPE for different resolution data (i.e., 10, 20, 30 s) and different prediction horizons (i.e., 2, 5, 10 min, and 1 h) after the error correction. There exists significant performance for all algorithms tested with respect to the MAPE computed before and after error correction. It validates the effectiveness of implementing error correction in improving the forecasting accuracy of time-series dataset. Meanwhile, the LSTM produces the smallest errors which also demonstrates its superior performance in forecasting wind directions.

Error Analysis

The experiments reported in Section "Short-term Predictions" have been conducted using the transformed wind direction data from four seasons. An inverse bilinear transformation, expressed in **Eq. 2**, is applied to transform the predicted transformed wind direction into the original angular range [0°, 360°]. The actual angular values versus the forecasted angular values by the proposed framework using ICEEMDAN and LSTM-RNN are presented in **Figure 8**. It can be seen that the majority of the forecasted values fall within a relatively small range with respect
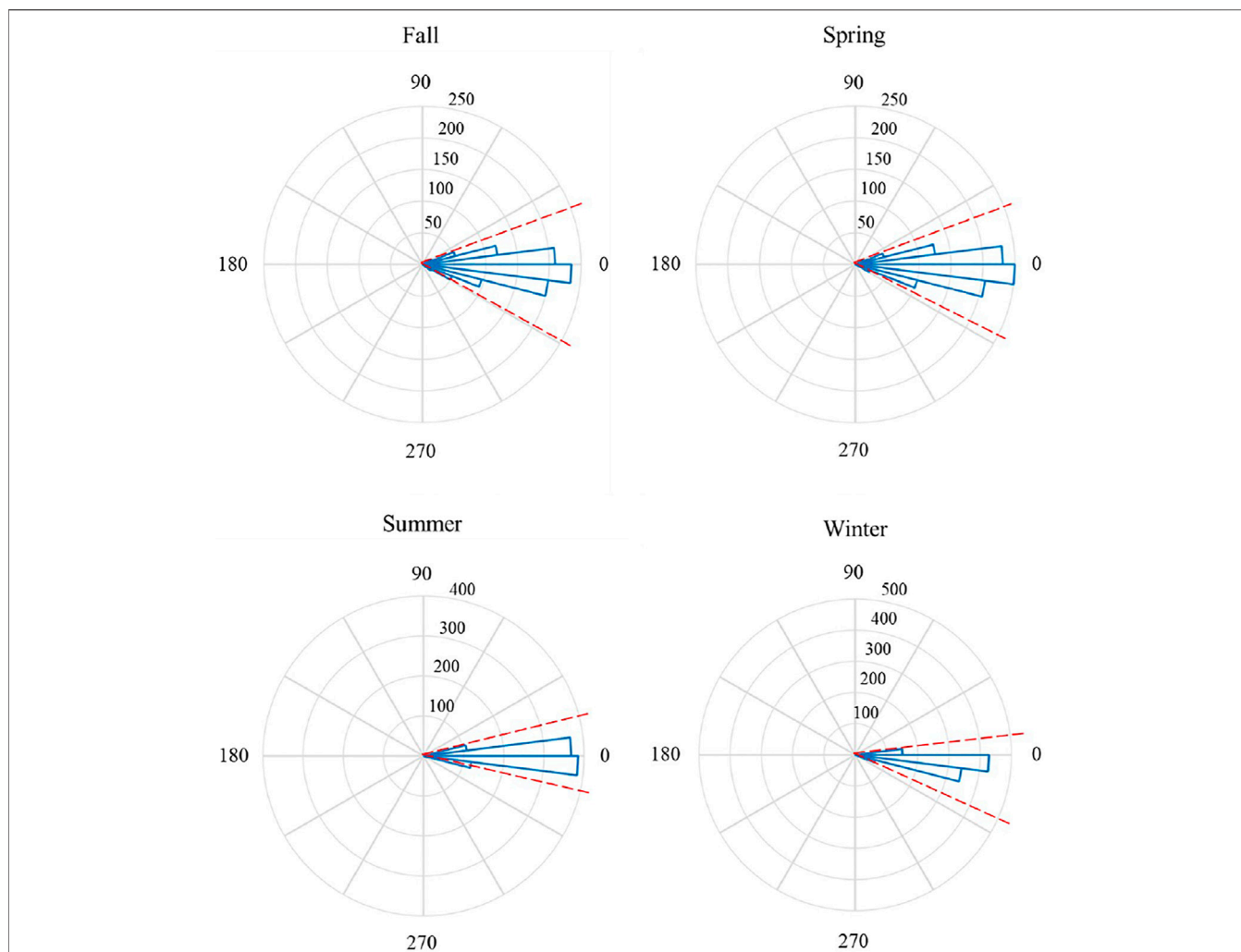
FIGURE 9 | Wind rose of the prediction errors for the four seasons of interest in 2020.

TABLE 4 | Angular prediction error at four seasons.

Season	Fall	Winter	Spring	Summer
MAE (2 min)	1.61°	1.63°	1.62°	1.59°
MAPE (2 min)	0.79%	0.78%	0.80%	0.76%
MAE (5 min)	4.94°	4.41°	5.19°	4.51°
MAPE (5 min)	2.08%	1.93%	2.58%	1.97%
MAE (10 min)	5.92°	5.86°	6.18°	5.94°
MAPE (10 min)	2.79%	2.78%	2.81%	2.71%
MAE (1 h)	9.81°	9.02°	9.75°	9.32°
MAPE (1 h)	4.77%	4.37%	4.92%	4.74%

to the actual values. It demonstrates the proposed framework can sufficiently provide accurate forecasting performances.

In this section, performance of the ICEEMDAN framework integrated with the long-short-term memory recurrent neural network (LSTM-RNN) for prediction of wind direction at four seasons is discussed. The prediction error of the inverse transformed wind direction at 2, 5, 10 min, and 1 h horizons are presented in **Table 3**. The mean absolute error (MAE) and mean absolute percentage error (MAPE) of wind direction are smaller in the summer and winter.

The wind direction error shows less variability over short horizons. The changes of a nacelle position are usually made within 5 min and the prediction error should be under 3% (Ouyang et al., 2017). A control chart is applied to monitor the prediction error and facilitate changing the nacelle position. A control chart with lower and upper limits enables monitoring the yaw error. Any prediction error that exceeds the bound (i.e., 3%) may trigger a change of the nacelle position. The final forecasting errors in the angular perspectives are illustrated in **Figure 9**.

Validation

The long-short-term memory recurrent neural network (LSTM-RNN) has been demonstrated to perform better than other algorithms. To validate the effectiveness and robustness of the LSTM-RNN, the data collected from another wind farm located

in Shandong Province in the year 2020 has been used. The experiments are conducted following the similar training strategies as described in Section "Data Analysis". The computational results are presented in **Table 4**.

The prediction error (see **Table 4**) in winter and summer seasons of 2020 from the wind farm in Shandong Province produced by the LSTM-RNN is similar to the one based on the 2020 data (see **Table 3**) in the wind farm in Inner Mongolia. More accurate performance has been observed in the fall and spring seasons with two prevailing wind directions. The favorable prediction error validates the effectiveness and robustness of the LSTM-RNN to predict the nacelle orientation.

CONCLUSION

A hybrid short-term forecasting framework to orient nacelle based on the predicted wind direction was presented. Industrial data collected from a wind farm in Inner Mongolia, China was utilized to train and validate the prediction models. A bilinear transformation was applied to transform the wind direction data from an angular variable into a continuous time-series. The forecasting framework was developed using ICEEMDAN integrated with LSTM-RNN. Also, the error corrections are implemented to improve the forecasting accuracy. The wind direction was predicted at short-term horizons, i.e., 2, 5, 10 min, and 1 h. Five algorithms, the deep neural network, deep belief network, kernel-based extreme learning machine, gated recurrent unit network, and long-short-term memory recurrent neural network were applied to predict wind direction at short-term horizons. The results of performance analysis of the five algorithms at four seasons were reported.

It was demonstrated that the long-short-term memory recurrent neural network outperformed the other four algorithms tested to predict wind direction. The results

presented are of paramount importance in yaw control and can improve the efficiency of energy extraction process.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

HL conceptualized the study, contributed to the study methodology, and wrote the original draft. JD contributed to the study methodology, data curation and investigation. PF contributed to data analysis and investigation. HL contributed to software and formal analysis. CP and DA contributed to investigation and writing-original draft. QC contributed to editing. All authors have read and agreed to the published version of the manuscript.

FUNDING

REFERENCES

Amin, C. M., Sami, F. M., and Trzynadlowski, A. M. (2018). Wind Speed and Wind Direction Forecasting Using echo State Network with Nonlinear Functions. *Renew. Energ.* 131 (2), 879–889. doi:10.1016/j.renene.2018.07.060

Bilgili, M., Sahin, B., and Yasar, A. (2007). Application of Artificial Neural Networks for the Wind Speed Prediction of Target Station Using Reference Stations Data. *Renew. Energ.* 32 (14), 2350–2360. doi:10.1016/j.renene.2006.12.001

Castino, F., Festa, R., and Ratto, C. F. (1998). Stochastic Modelling of Wind Velocities Time Series. *J. Wind Eng. Ind. Aerodynamics* 74-76, 141–151. doi:10.1016/s0167-6105(98)00012-9

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). *Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation*. doi:10.3115/v1/D14-1179

Colominas, M. A., Schlotthauer, G., and Torres, M. E. (2014). Improved Complete Ensemble EMD: A Suitable Tool for Biomedical Signal Processing.. *Biomed. Signal Proce. Contr.* 146, 19–29.

Davies, A. (1974). Bilinear Transformation of Polynomials. *IEEE Trans. Circuits Syst.* 21 (6), 792–794. doi:10.1109/tcs.1974.1083929

Duan, J., Zuo, H., Bai, Y., Duan, J., Chang, M., and Chen, B. (2021). Short-term Wind Speed Forecasting Using Recurrent Neural Networks with Error Correction. *Energy* 217, 119397. doi:10.1016/j.energy.2020.119397

Erdem, E., and Shi, J. (2011). ARMA Based Approaches for Forecasting the Tuple of Wind Speed and Direction. *Appl. Energ.* 88 (4), 1405–1414. doi:10.1016/j.apenergy.2010.10.031

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2003). Learning Precise Timing with Lstm Recurrent Networks. *J. Machine Learn. Res.* 3 (1), 115–143.

Groutage, F., Volfson, L., and Schneider, A. (2003). S-plane to Z-Plane Mapping Using a Simultaneous Equation Algorithm Based on the Bilinear Transformation. *IEEE Trans. Automatic Control.* 32 (7), 635–637. doi:10.1109/TAC.1987.1104664

Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.* 9 (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

Hu, Q., Zhang, R., and Zhou, Y. (2016). Transfer Learning for Short-Term Wind Speed Prediction with Deep Neural Networks. *Renew. Energ.* 85 (JAN), 83–95. doi:10.1016/j.renene.2015.06.034

Jury, E. (1973). Remarks on "The Mechanics of Bilinear Transformation". *IEEE Trans. Audio Electroacoust.* 21 (4), 380–382. doi:10.1109/tau.1973.1162485

Khosravi, A., Koury, R. N. N., Machado, L., and Pabon, J. J. G. (2018). Prediction of Wind Speed and Wind Direction Using Artificial Neural Network, Support Vector Regression and Adaptive Neuro-Fuzzy Inference System. *Sustainable Energ. Tech. Assessments* 25, 146–160. doi:10.1016/j.seta.2018.01.001

Kodama, K., and Burls, N. J. (2019). An Empirical Adjusted Enso Ocean Energetics Framework Based on Observational Wind Power in the Tropical pacific. *Clim. Dyn.* 53 (5), 3271–3288. doi:10.1007/s00382-019-04701-8

Kou, Z., Yang, F., Wu, J., and Li, T. (2020). Application of ICEEMDAN Energy Entropy and AFSA-SVM for Fault Diagnosis of Hoist Sheave Bearing. *Entropy* 22 (12), 1347. doi:10.3390/e22121347

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature* 521 (7553), 436–444. doi:10.1038/nature14539

Lee, T.-H., White, H., and Granger, C. W. J. (1993). Testing for Neglected Nonlinearity in Time Series Models. *J. Econom.* 56 (3), 269–290. doi:10.1016/0304-4076(93)90122-l

Li, H., Xu, Q., He, Y., and Deng, J. (2018). Prediction of Landslide Displacement with an Ensemble-Based Extreme Learning Machine and Copula Models. *Landslides* 15 (10), 2047–2059. doi:10.1007/s10346-018-1020-2

Li, H., Xu, Q., He, Y., Fan, X., and Li, S. (2020). Modeling and Predicting Reservoir Landslide Displacement with Deep Belief Network and EWMA Control Charts: a Case Study in Three Gorges Reservoir. *Landslides* 17 (3), 693–707. doi:10.1007/s10346-019-01312-6

Liu, D., Niu, D., Wang, H., and Fan, L. (2014). Short-term Wind Speed Forecasting Using Wavelet Transform and Support Vector Machines Optimized by Genetic Algorithm. *Renew. Energ.* 62, 592–597. doi:10.1016/j.renene.2013.08.011

Liu, H., Shi, J., and Erdem, E. (2010). Prediction of Wind Speed Time Series Using Modified Taylor Kriging Method. *Energy* 35 (12), 4870–4879. doi:10.1016/j.energy.2010.09.001

Masseran, N., Razali, A. M., Ibrahim, K., and Latif, M. T. (2013). Fitting a mixture of von Mises distributions in order to model data on wind direction in Peninsular Malaysia. *Energ. Convers. Manag.* 72, 94–102. doi:10.1016/j.enconman.2012.11.025

Mcwilliams, B., and Sprevak, D. (1982). The Simulation of Hourly Wind Speed and Direction. *Mathematics Comput. Simulation* 24 (1), 54–59. doi:10.1016/0378-4754(82)90050-7

Mohandes, M. A., Halawani, T. O., Rehman, S., and Hussain, A. A. (2004). Support Vector Machines for Wind Speed Prediction. *Renew. Energ.* 29 (6), 939–947. doi:10.1016/j.renene.2003.11.009

Ouyang, T., He, Y., and Huang, H. (2018). Monitoring Wind Turbines' Unhealthy Status: a Data-Driven Approach. *IEEE Trans. Emerging Top. Comput. Intelligence* 3 (2), 163–172. doi:10.1109/TETCI.2018.2872036

Ouyang, T., He, Y., Li, H., Sun, Z., and Baek, S. (2019). Modeling and Forecasting Short-Term Power Load with Copula Model and Deep Belief Network. *IEEE Trans. Emerg. Top. Comput. Intell.* 3 (2), 127–136. doi:10.1109/tetci.2018.2880511

Ouyang, T., Kusiak, A., and He, Y. (2017). Predictive Model of Yaw Error in a Wind Turbine. *Energy* 123, 119–130. doi:10.1016/j.energy.2017.01.150

Pan, C., Yi, J., Yin, C., Yu, J., and Li, X. (2019). Joint 3D UAV Placement and Resource Allocation in Software-Defined Cellular Networks with Wireless Backhaul. *IEEE Access* 7, 104279–104293. doi:10.1109/access.2019.2927521

Peng, G., Si, C. A., Jc, B., and Di, C. (2020). Wind Direction Fluctuation Analysis for Wind Turbines. *Renew. Energ.* 162, 1026–1035. doi:10.1016/j.renene.2020.07.137

Rong, H., Gao, Y., Guan, L., Zhang, Q., Zhang, F., and Li, N. (2019). Gam-based Mooring Alignment for Sins Based on an Improved Ceemd Denoising Method. *Sensors* 19 (16), 3564. doi:10.3390/s19163564

Sak, H., Senior, A., and Beaufays, F. (2014). *Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition.* arXiv preprint arXiv:1402.1128

Santhosh, M., Venkaiah, C., and Vinod Kumar, D. M. (2018). Ensemble Empirical Mode Decomposition Based Adaptive Wavelet Neural Network Method for Wind Speed Prediction. *Energ. Convers. Manag.* 168, 482–493. doi:10.1016/j.enconman.2018.04.099

Sun, Z., He, Y., Gritsenko, A., Lendasse, A., and Baek, S. (2020). Embedded Spectral Descriptors: Learning the point-wise Correspondence Metric via Siamese Neural Networks. *J. Comput. Des. Eng.* 7 (1), 18–29. doi:10.1093/jcde/qwaa003

Tagliaferri, F., Viola, I. M., and Flay, R. G. J. (2015). Wind Direction Forecasting with Artificial Neural Networks and Support Vector Machines. *Ocean Eng.* 97, 65–73. doi:10.1016/j.oceaneng.2014.12.026

Tang, Z., and Zhang, Z. (2019). The Multi-Objective Optimization of Combustion System Operations Based on Deep Data-Driven Models. *Energy* 182, 37–47. doi:10.1016/j.energy.2019.06.051

Tang, Z., Zhao, G., and Ouyang, T. (2021). Two-phase Deep Learning Model for Short-Term Wind Direction Forecasting. *Renew. Energ.* 173, 1005–1016. doi:10.1016/j.renene.2021.04.041

Wasynczuk, O., Man, D. T., and Sullivan, J. P. (1981). Dynamic Behavior of a Class of Wind Turbine Generators during Random Wind Fluctuations. *IEEE Power Eng. Rev.* PER-1 (6), 47–48. doi:10.1109/mper.1981.5511593

Xu, W., Yi, J., Dasgupta, S., Cai, J. F., Jacob, M., and Cho, M. (2018). "Sepration-free Super-resolution from Compressed Measurements Is Possible: an Orthonormal Atomic Norm Minimization Approach," in 2018 IEEE International Symposium on Information Theory (ISIT) (IEEE), 76–80.

Yang, Z., and Wang, J. (2018). A Hybrid Forecasting Approach Applied in Wind Speed Forecasting Based on a Data Processing Strategy and an Optimized Artificial Intelligence Algorithm. *Energy* 160, 87–100. doi:10.1016/j.energy.2018.07.005

Yi, J., and Xu, W. (2020). Necessary and Sufficient Null Space Condition for Nuclear Norm Minimization in Low-Rank Matrix Recovery. *IEEE Trans. Inform. Theor.* 66 (10), 6597–6604. doi:10.1109/tit.2020.2990948

Zhang, W., Qu, Z., Zhang, K., Mao, W., Ma, Y., and Fan, X. (2017). A Combined Model Based on CEEMDAN and Modified Flower Pollination Algorithm for Wind Speed Forecasting. *Energ. Convers. Manag.* 136, 439–451. doi:10.1016/j.enconman.2017.01.022

Zhou, J., Shi, J., and Li, G. (2011). Fine Tuning Support Vector Machines for Short-Term Wind Speed Forecasting. *Energ. Convers. Manag.* 52 (4), 1990–1998. doi:10.1016/j.enconman.2010.11.007