Computer Systems Science & Engineering
DOI:10.32604/csse.2021.016911
Article

Tech Science Press

Short-term Wind Speed Prediction with a Two-layer Attention-based LSTM

Jingcheng Qian[1], Mingfang Zhu[1], Yingnan Zhao[2,*] and Xiangjian He[3]

[1]Wujiang Power Supply Company, Suzhou, 320500, China
[2]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[3]School of Computing and Communications, University of Technology, Sydney, Australia
*Corresponding Author: Yingnan Zhao. Email: zh_yingnan@126.com
Received: 15 January 2021; Accepted: 08 March 2021

Abstract: Wind speed prediction is of great importance because it affects the efficiency and stability of power systems with a high proportion of wind power. Temporal-spatial wind speed features contain rich information; however, their use to predict wind speed remains one of the most challenging and less studied areas. This paper investigates the problem of predicting wind speeds for multiple sites using temporal and spatial features and proposes a novel two-layer attention-based long short-term memory (LSTM), termed 2Attn-LSTM, a unified framework of encoder and decoder mechanisms to handle temporal-spatial wind speed data. To eliminate the unevenness of the original wind speed, we initially decompose the preprocessing data into IMF components by variational mode decomposition (VMD). Then, it encodes the spatial features of IMF components at the bottom of the model and decodes the temporal features to obtain each component's predicted value on the second layer. Finally, we obtain the ultimate prediction value after denormalization and superposition. We have performed extensive experiments for short-term predictions on real-world data, demonstrating that 2Attn-LSTM outperforms the four baseline methods. It is worth pointing out that the presented 2Atts-LSTM is a general model suitable for other spatial-temporal features.

Keywords: Wind speed prediction; temporal-spatial features; VMD; LSTM; attention mechanism

1 Introduction

Due to its cleanliness, low cost, and sustainability, wind energy has become the mainstream new energy source. According to the latest data released by the Global Wind Energy Council (GWEC), the world's installed wind power capacity reached 651 GW in 2019 [1]. However, it poses significant challenges to the power system's operation control with a high proportion of wind power because of the randomness, volatility, and intermittency of wind farms [2]. Accurate wind speed prediction is the basis of operation control [3]. Wind speed forecasts can be divided into short-term (minutes, hours, days), medium-term (weeks, months), and long-term (years) forecasts according to different time intervals. Among them, short-term forecasting is essential for the power system to make daily dispatch plans. It has a significant impact on the economical and reliable operation of the power system.

Wind speed prediction techniques fall into the following three categories: physical, statistical, and artificial intelligence models. Physical modeling methods [4–6] mainly predict wind speed by establishing formulas between wind speed and air pressure, air density, and air humidity. The modeling process involves a large amount of calculation. Due to the complexity of wind speed and regional differences, it is challenging to establish high-precision short-term forecasts for different regions using physical models. Therefore, they are usually applied for long-term wind speed prediction in specific areas. Compared with physical models, statistical models are simple, easy, and better, so they are widely adopted in short-term wind speed prediction. Statistical models use historical wind speed data to establish a linear mapping relationship between system input and output to make predictions, for example, the kriging interpolation method [7] and the von Mises distribution [8]. There are still some commonly used methods, such as autoregressive (AR) [9] and autoregressive moving average (ARMA) [10].

Machine learning technologies are the basis of artificial intelligence models. They describe the complicated nonlinear relationship between system input and output based on a large amount of wind speed temporal data. For example, [11] used the least squares support vector machine to predict wind speed. With the vigorous development of machine learning technology, technologies in this field have been rapidly applied to short-term wind speed prediction, such as CNN, RNN, GRU, LSTM, etc. Combining the existing wind speed prediction technology and the hybrid neural network model has obtained a promising prediction result [12–14]. However, the current short-term wind speed prediction models only focus on time series data, and the wind speed data of the sites near the target wind farm also contain rich information. Data analysis based on spatial-temporal correlation has become a research hot spot [15,16]. In addition to temporal data, geographic spatial relationships are also considered to improve prediction accuracy. Moreover, the attention mechanism (AM) has recently become a research hot spot [17,18]. It builds an attention matrix to enable deep neural networks to focus on crucial features during training to avoid the impact of insensitive features.

In this paper, we introduce two-layer attention-based LSTM (2Atts-LSTM) networks. Experiments on real-world data show that they are superior to other baselines.

The main contributions of this article can be summarized as follows:

(1) 2Atts-LSTM, a novel deep architecture for short-term wind speed prediction, is proposed, which integrates the attention mechanism and LSTM into a unified framework. This model achieves spatial feature and temporal dependency extraction automatically.

(2) VMD technology is combined with 2Attn-LSTM to obtain a relatively stable subsequence. It can eliminate the uncertainty of the actual wind speed.

The rest of the paper is organized as follows: Section 2 gives relevant background theories, including VMD and LSTM networks; Section 3 illustrates the algorithm proposed in the article; Section 4 presents the experimental results, compared with the baselines; Section 5 concludes this paper and provides further work.

2 Background Theories

2.1 VMD

Based on empirical mode decomposition (EMD), the variational mode decomposition (VMD) proposed by Dragomirestskiy et al. [19] is a new type of complicated signal decomposition method. It decomposes the signal into limited bandwidths with different center frequencies according to the preset number of modes.

Using VMD, the original wind speed sequence with strong nonlinearity and randomness can be decomposed into a series of stable mode components. Fig. 1 shows the flowchart of the VMD algorithm. Suppose the wind speed data after preprocessing are $\tilde{X}(t)_l$. The process is as follows:

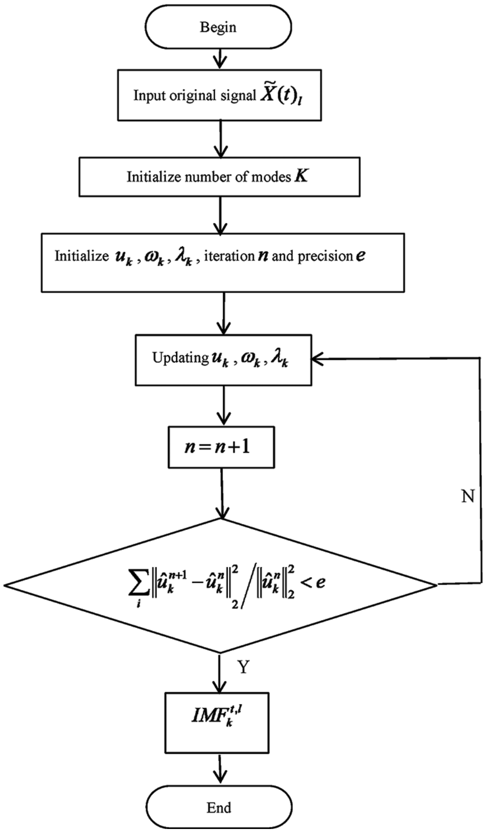
Begin
Input original signal $\widetilde{X}(t)_l$
Initialize number of modes K
Initialize u_k, ω_k, λ_k, iteration n and precision e
Updating u_k, ω_k, λ_k
$n=n+1$
N
$\sum_i \left\|\hat{u}_k^{n+1}-\hat{u}_k^n\right\|_2^2 \Big/ \left\|\hat{u}_k^n\right\|_2^2 < e$
Y
$IMF_k^{t,l}$
End

Figure 1: The flowchart of VMD algorithm

(1) Assuming that each mode has a limited bandwidth with a center frequency, we now look for modes so that the sum of each mode's estimated bandwidth is the lowest, expressed as

$$\min_{|u_k|.|\omega_k|}\{\sum_{k=1}^{K}\left\|\partial_t[(\delta(t)+\frac{j}{\pi t})*u_k(t)_l]e^{-jw_kt}\right\|_2^2$$
$$s.t.\sum_{k=1}^{K}u_k(t)_l=\tilde{X}(t)_l \tag{1}$$

(2) Solving the above model, introduce the penalty factor α and Lagrangian penalty operator $\lambda(t)$, transform the constraint problem into the nonconstraint problem, and obtain the augmented Lagrangian expression.

$$L(\{u_k\},\{\omega_k\},\lambda)=\alpha\sum_K\left\|\partial_t[(\delta(t)+\frac{j}{\pi t})*u_k(t)]\bullet e^{-jw_kt}\right\|_2^2$$
$$+\left\|f(t)-\sum_K u_k(t)\right\|_2^2+\left\langle\lambda(t)f(t)-\sum_K u_k(t)\right\rangle \tag{2}$$

(3) Update parameters u_k, ω_k and λ_k iteratively by the alternating direction method of multipliers, which is defined as

$$\hat{u}_k^{n+1} = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \tag{3}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega \left|\hat{u}_k^{n+1}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{u}_k^{n+1}(\omega)\right|^2 d\omega} \tag{4}$$

where $\hat{f}(\omega)$, $\hat{u}_i(\omega)$, $\hat{\lambda}(\omega)$ and $\hat{u}_k^{n+1}(\omega)$ represent the Fourier transforms of $f(\omega)$, $u_i(\omega)$, $\lambda(\omega)$and $u_k^{n+1}(\omega)$, and n is the number of iterations.

(4) For a given precision e > 0, if $\sum_i \left\|\hat{u}_k^{n+1} - \hat{u}_k^n\right\|_2^2 \Big/ \left\|\hat{u}_k^n\right\|_2^2 < e$, then stop the iteration. Otherwise, return to

(5) Finally, we can get K decomposed $F_k^{t,l}$.

Fig. 2 shows the wind speed subsequences, IMF, with different frequencies but stronger regularity by VMD.

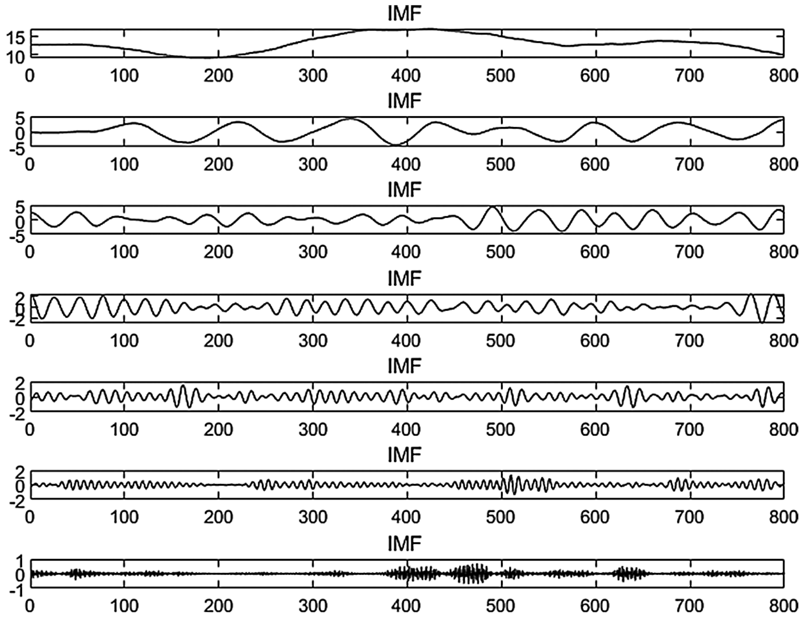

Figure 2: Wind speed is decomposed by VMD

2.2 LSTM

LSTM [20], a variant of the recurrent neural network (RNN), shows superior performance in processing sequential data. It overcomes the problem of "long-term dependencies" [21]. Due to its tremendous learning capacity, LSTM has been widely used in various kinds of tasks, such as speech recognition [22], software-defined network (SDN) [23], and some prediction cases, i.e., trajectory [24], oil price [25], and even the number of confirmed COVID-19 cases [26]. In the usual applications, the stacked LSTM network is the most basic and simplest structure with high performance. In this paper, the proposed 2Attn-LSTM falls into this category.

Each LSTM cell unit consists of an internal memory cell c_t and three gates, i.e., forget gate f_t, input gate i_t, and output gate o_t. h_t is the final state determined by c_t and o_t. The memory cell will store the previous data for a long time controlled by the input and output gates. At the same time, the information stored in the memory cell can be cleared by the forget gate. The formulations in the LSTM are given by Eqs. (5)–(9).

$$i_t = sigmoid(W_{hi}h_{t-1} + W_{xi}F_k^{t,l} + b_i) \tag{5}$$

$$f_t = sigmoid(W_{hf}h_{t-1} + W_{xf}F_k^{t,l} + b_f) \tag{6}$$

$$o_t = sigmoid(W_{ho}h_{t-1} + W_{hx}F_k^{t,l} + w_{co}c_t) \tag{7}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes (\tanh(W_{hc}h_{t-1} + W_{xc}F_k^{t,l} + b_c)) \tag{8}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{9}$$

where w_{hi}, w_{xi}, b_i, w_{hf}, w_{xf}, b_f, w_{hc}, w_{xc}, b_c, w_{ho}, w_{hx}, w_{co} are learnable parameters of input gate, forget gate, memory cell, output gate and final state, respectively.

3 Methodology

The proposed 2Attn-LSTM method, illustrated in Fig. 3, consists of data preprocessing, decomposition of VMD, LSTM encoder with Attention1 and LSTM decoder with Attention2. Then, it gets each IMF's prediction value. After denormalization and superposition, we can obtain the final wind speed prediction. The preprocessing stage contains data cleaning and normalization. Then, it decomposes the preprocessed data into components, $F_k^{t,l}$, by VMD. The model training phase contains an encoder and decoder; that is, the first layer handles the spatial features, and the second layer manages the temporal features. Here, we adopt an attention mechanism into the architecture, which has been widely applied recently.

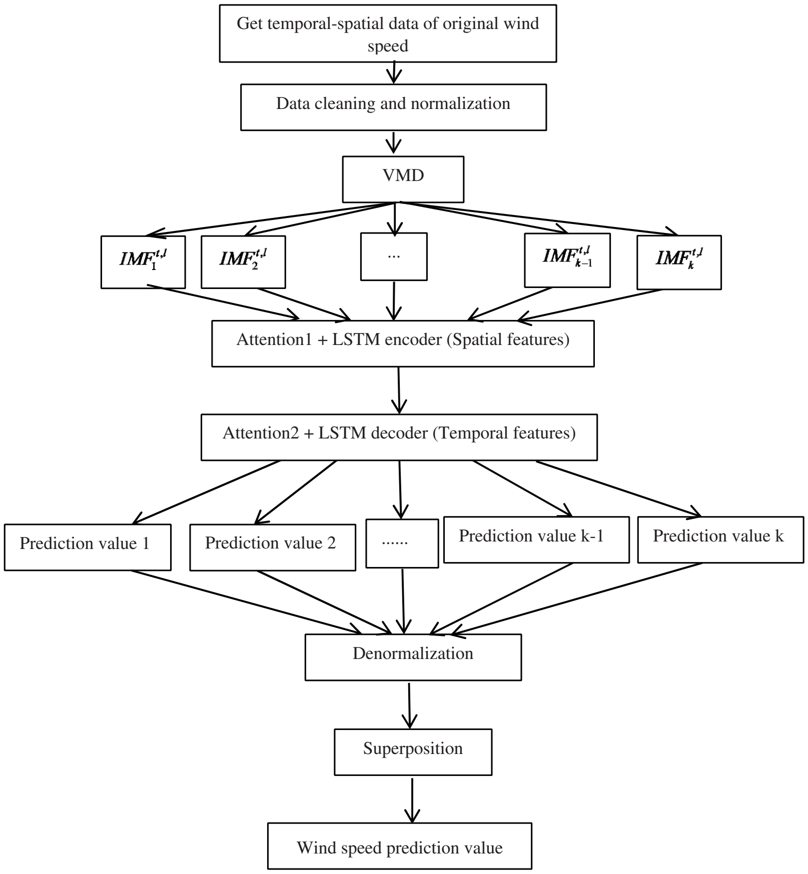
Get temporal-spatial data of original wind speed
Data cleaning and normalization
VMD
$IMF_1^{t,l}$
$IMF_2^{t,l}$
...
$IMF_{k-1}^{t,l}$
$IMF_k^{t,l}$
Attention1 + LSTM encoder (Spatial features)
Attention2 + LSTM decoder (Temporal features)
Prediction value 1
Prediction value 2
......
Prediction value k-1
Prediction value k
Denormalization
Superposition
Wind speed prediction value

Figure 3: Framework of the presented approach

3.1 Data Processing

Obtain the original space-time wind speed sequence of the target site $X(t)_l$. For missing data, repeated data, and jump data, replace with the average wind speed near the value. After normalization, we obtain $\tilde{X}(t)_l$, where $t \in R^T$, $l \in R^L$, T is the time lag, and L is the number of neighboring sites of the target site. The normalization formula is

$$\tilde{X}(t)_l = \frac{X(t)_l - X_{\min}(t)_l}{X_{\max}(t)_l - X_{\min}(t)_l} \tag{10}$$

where $X_{\max}(t)_l$ is the maximum temporal wind speed of site l, and $X_{\min}(t)_l$ is the minimum temporal wind speed of site l. $X(t)_l$ is the value before normalization, and $\tilde{X}(t)_l$ is the value after normalization of site l.

After the handling of the 2-layer LSTM network, we need denormalization and superposition. The denormalization formula is given as follows:

$$Y(t)_l = \tilde{Y}(t)_l(IMF_{\max}(t)_l - IMF_{\min}(t)_l) + IMF_{\min}(t)_l \tag{11}$$

where $IMF_{\max}(t)_l$ and $IMF_{\min}(t)_l$ are the maximum and minimum IMF components of site l, respectively. $\tilde{Y}(t)_l$ is the normalization value, and $Y(t)_l$ is the denormalized result.

3.2 Temporal-Spatial Feature Model

In the proposed 2Attn-LSTM framework, we process the temporal-spatial data. Except for the general sequential features, the spatial data do have plenty of information helpful for wind speed prediction. Zhu et al. [15] proposed a deep architecture, termed PSTN, integrating CNN and LSTM, to learn temporal and spatial correlations jointly for short-term wind speed prediction.

However, Zhu et al. [15] embedded the temporal-spatial features into a 2D matrix, named SWSM. The item in SWSM is defined by $x(i,j)_t \in R^{M \times N}$, where $M \times N$ is the spatial square of the target site. Instead of SWSM, we specify one IMF time series as the target series for making predictions, while other IMF series are used as features. Furthermore, we separated the spatiotemporal features into spatial data, served as the input of the encoder of 2Attn-LSTM, and temporal data, served as the input of the decoder of 2Attn_LSTM. The scheme is superior to PSTN in both space requirements and time complexity. Suppose the time window length is T, the number of neighboring sites is L, and the number of IMF components is K. We use $F_k^{t,l} = (f_k^{1,l}, f_k^{2,l}, \ldots, f_k^{T,l}) \in R^T$ to denote the temporal features and $F_k^{t,l} = (f_k^{t,1}, f_k^{t,2}, \ldots, f_k^{t,L}) \in R^L$ to describe the spatial features.

As illustrated in Fig. 4, we decompose the original wind speed into K IMF components, denoted by $IMF_k^{t,l}$. The features along the x-direction are temporal features, i.e., $IMF_k^{t,l} = (IMF_k^{1,l}, IMF_k^{2,l}, \ldots, IMF_k^{T,l}) \in R^T$. The y-direction features are $IMF_k^{t,l} = (IMF_1^{t,l}, IMF_2^{t,l}, \ldots, IMF_K^{t,l}) \in R^K$, and they denote the K IMF components. The z-direction features are spatial features, i.e., $IMF_k^{t,l} = (IMF_k^{t,1}, IMF_k^{t,2}, \ldots, IMF_k^{t,L}) \in R^L$.

3.3 Network Architecture

Fig. 5 depicts the hierarchy of 2Attn-LSTM, which follows the encoder-decoder architecture. We adopt two separate LSTMs. One is to encode the spatial features, and the other decodes the temporal features. The encoder captures the temporal correlations of IMF components at each time by referring to the previous hidden state of the encoder, previous values of sensors and the spatial information. In the decoder, we use temporal attention to adaptively select the relevant previous time intervals for making predictions.

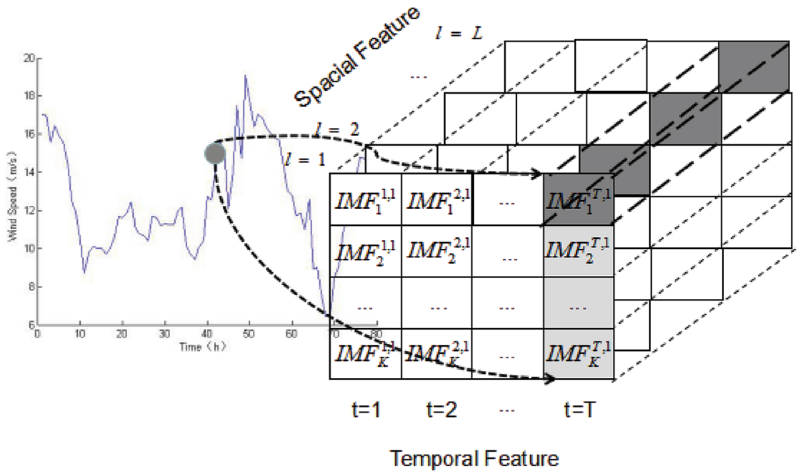

Figure 4: Model of spatial-temporal feature

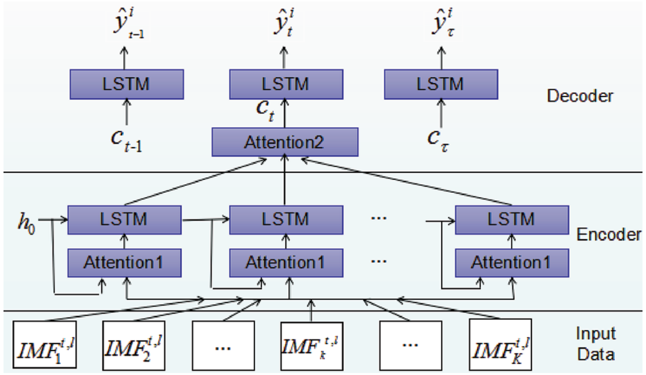

Figure 5: 2Attn-LSTM hierarchy

In the encoder part, we calculate Attention1 as follows:

$$a_t^l = \frac{\exp((1-\lambda)g_t^l + \lambda I_{i,j})}{\sum_{j=1}^{K} \exp((1-\lambda)g_t^j + \lambda I_{i,j})} \tag{12}$$

where g_t^l is the weight attention of i and j sites, which is calculated as follows:

$$g_t^l = v_g^T \tanh(W_g[h_{t-1}; c_{t-1}] + U_g F_k^{t,j} + W'_g F_k^{t,i} u_g + b_g) \tag{13}$$

Here, v_g, u_g, b_g, W_g and U_g are learnable parameters, and [;] is the connection computation. h_{t-1} and c_{t-1} are the hidden state and memory unit cell at time $t-1$ of the LSTM encoder, respectively. $I_{i,j}$ is the mutual information of i, j sets. It is computed as follows:

$$I_{i,j} = H(F_k^{t,i}) + H(F_k^{t,j}) - H(F_k^{t,i}, F_k^{t,j}) \tag{14}$$

$$H(F_k^{t,i}) = -\sum_{f \in F_k^{t,i}} P_{F_k^{t,i}}(f) \log(P_{F_k^{t,i}}(f)) \tag{15}$$

$$H(F_k^{t,i},F_k^{t,j}) = -\sum_{f\in F_k^{t,i},f'\in F_k^{t,i}} P_{F_k^{t,i}F_k^{t,i}}(f,f')\log(P_{F_k^{t,i}F_k^{t,i}}(f,f')) \tag{16}$$

where $H(F_k^{t,i})$is the entropy of $F_k^{t,i}$, $H(F_k^{t,i},F_k^{t,j})$is the union entropy of $F_k^{t,i}$and $F_k^{t,j}$, and $P(\bullet)$ is the probability density function.

In the encoder, the following formula is used to update the hidden state at time t:

$$h_t = f_e(h_{t-1}, a_t^l) \tag{17}$$

where f_e is the LSTM cell of the encoder, and h_{t-1} is the hidden state at time $t-1$.

In the decoder, we use the following equation to update the hidden state at time t':

$$h'_{t'} = f_d(h'_{t'-1}, [\hat{f}^i_{t'-1}; a_{t'}]) \tag{18}$$

where f_d is the LSTM cell of the decoder, and $\hat{f}^i_{t'-1}$ is the prediction component at time $t'-1$. Attention2 is calculated as follows:

$$u^o_{t'} = v_d^T \tanh(W'_d[h'_{t'-1}; c'_{t'-1}] + W_d h_o + b_d) \tag{19}$$

$$\gamma^o_{t'} = \frac{\exp(u^o_{t'})}{\sum_{j=1}^{T}\exp(u^j_{t'})} \tag{20}$$

$$\hat{a}_{t'} = \sum\nolimits_{o=1}^{T} \gamma^o_{t'} h_o \tag{21}$$

where W_d, W'_d, v_d and b_d are learnable parameters. $h'_{t'-1}$ and $c'_{t'-1}$ are the hidden state and memory cell of the decoder in LSTM at time $t'-1$, respectively.

The final prediction component is

$$\hat{f}^i_{t'} = v_y^T(W_m[a_{t'}; h_{t'}] + b_m) + b_y \tag{22}$$

where W_m, b_m, v_y and b_y are parameters.

4 Experiments

4.1 Settings

We perform our experiments over the Wind Integration National Data set (WIND), provided by the National Renewable Energy Laboratory (NREL). It contains wind speed data for more than 126,000 sites in the United States for the years 2007–2013. We consider 6 different datasets based on WIND, as depicted in Tab. 1. They belong to Wyming and Texas states. In each state, we choose 5, 3, and 1 wind farms with different time intervals (i.e., 1 hour, 30 minutes, 15 minutes) and time spans (i.e., 1 year, six months, three months) to guarantee plenty of instances. For example, 5 wind farms of 286 sites in Wyoming state are conducted in the experiment. The D1 dataset has 2,514,120 instances with a 1-hour time interval during 2012.

We use general criteria to evaluate the proposed 2Attn-LSTM model, that is, the mean absolute error (MAE) and root mean squared error (RMSE), which are widely adopted as the evaluation indices in the task of wind speed prediction. They are given by the following:

Table 1: Dataset details

Dataset	State	Time Spans	Time Intervals	Wind farms	Sites	Instances
D1	Wyoming	1/1/2012–31/12/2012	60 minutes	5	287	2,514,120
D2		1/1/2012–30/6/2012	30 minutes	3	185	1,607,280
D3		1/7/2012–30/9/2012	15 minutes	1	80	706,560
D4	Texas	1/1/2009–31/12/2009	60 minutes	5	305	2,671,800
D5		1/7/2008–31/12/2008	30 minutes	3	147	1,298,304
D6		1/1/2009–31/3/2009	15 minutes	1	75	640,800

$$E_{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \hat{Y}_i\right| \tag{23}$$

$$E_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2} \tag{24}$$

where N is the number of predictions, and i is the sequence number of the forecast point. Y_i and $\hat{Y}_i$ denote the ground truth and predicted wind speeds, respectively.

4.2 Baselines

We compare our model with 4 baselines. They are BP, ARIMA [27], LSTM and PSTN [15]. The back propagation (BP) neural network algorithm is a multilayer feedforward network trained according to the error back propagation algorithm and is one of the most widely applied neural network models. Autoregressive Integrated Moving Average (ARIMA) is actually a class of models that explains a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values. It is a well-known model for forecasting future values in a time series. As a variant of RNN, LSTM shows superior performance in processing sequential data.

The three methods mentioned above are classical models in short-term wind speed prediction. The PSTN was recently proposed to leverage both temporal and spatial correlations. It integrates CNN and LSTM to form a unified framework. To evaluate the presented 2Attn-LSTM with PSTN, we choose the same configuration in Zhu et al. [15], as shown in Tab. 2.

Table 2: PSTN configuration

Index	Type	Configurations
–	Input	data block size 10 × 10
1	Convolution layer	kernels: 20; kernel size: 3 × 3; stride: 1 × 1
2	Max-Pooling layer	pooling size: 2 × 2; stride: 2 × 2
3	Convolution layer	kernels: 50; kernel size: 3 × 3; stride:1 × 1
4	Convolution layer	kernels: 200; kernel size: 2 × 2; stride:1 × 1
5	Fully connected layer	units: 200; activation function: none
6	LSTM layer	hidden unites:{100, 200, 300, 400, 500}
7	LSTM layer	hidden unites: 100

4.3 Implementation Details

The determination of the optimal hyperparameters is still an open issue. Specifically, we divided the dataset into three subsets, i.e., training set, validation set and testing set at a ratio of 6:1:3. The training set serves for model training, including searching for optimal hyperparameters, and the validation set is used for model selection and overfitting prevention. We use testing data to test the model performance. All the baselines are determined in this way as well.

In the presented 2Attn-LSTM, there are many hyperparameter settings during the training phase. We set the batch size to 256 and the learning rate to 0.01. We set $\tau = 6$ to make short-term predictions. The trade-off parameter λ is empirically fixed from 0.1 to 0.5. For the length of window size T, we set $T \in \{6, 12, 24, 36, 48\}$. For simplicity, we use the same hidden dimensionality at the encoder and the decoder and conduct a grid search over {32, 64, 128, 256}. Moreover, we use stacked LSTMs (the number of layers is denoted as q) as the units of the encoder and decoder to enhance our performance. The setting is in which $q = 2$, $m = n = 64$ and $\lambda = 0.2$ outperform the others in the validation set.

The TensorFlow deep learning framework based on the Python platform builds our model as well as the baselines. All the methods are carried out on a 64-bit PC with an Intel Core i5-7600 CPU/32.00 GB RAM. We test different hyperparameters to find the best setting for each.

4.4 Short-term Wind Speed Prediction

To evaluate the prediction performance of the presented model, we conduct experiments with a prediction horizon ranging from 10 minutes to 1 hour. The prediction performance of all models is evaluated on 6 testing sets by MAE and RMSE indices.

The results shown in Tab. 3 illustrate that the proposed 2Attn-LSTM model holds the dominant position over the other models, while BP produces the worst prediction results. BP performs fairly poor with longer prediction horizons. For example, BP is 3.0% lower than the ARIMA 15-minute ahead prediction, while it increases to 10% when performing the 1-hour ahead prediction in terms of MAE. Although ARIMA outperforms BP, it is still inferior to LSTM, which implies that LSTM is more efficient in capturing temporal information. This mainly benefits from the working mechanism, i.e., the gates and the memory cell update information and prevent the model from vanishing the gradient. Specifically, PSTN improves the average MAE and RMSE by 14% and 3%, respectively, compared to LSTM. Integrating spatial and temporal features in the PSTN contributes to the best performance. The proposed 2Attn-LSTM method outperformed the PSTN in MAE by 8% in the 15-min horizon and 27.5% in the 1-hour ahead prediction task. The reasons for this may lie in the following two aspects. (1) The 2Attn-LSTM model handles the VMD first, which decomposes the original wind speed sequence with strong nonlinearity, and randomness can be decomposed into a series of stable modes. It plays a more critical role when the prediction horizon increases. (2) It considers both spatial and temporal features, such as PSTN, which is helpful for prediction.

Figs. 6 and 7 show the comparison of these five methods in the Wyoming dataset by RMSE and in the Texas dataset by MAE. Fig. 6 implies that for the same method, the shorter the time interval is, the higher the prediction performance. Fig. 7 lists the comparison among 5 models by MAE. It can be concluded that 2Attn-LSTM achieves the best performance.

Table 3: Performance comparison among different methods

Method	Wyoming						Texas					
	D1		D2		D3		D4		D5		D6	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BP	1.054	1.382	0.993	1.124	0.928	0.948	1.012	1.279	0.948	1.058	0.905	0.976
ARIMA	0.948	1.253	0.927	1.108	0.899	1.002	0.982	1.163	0.957	1.028	0.895	0.962
LSTM	0.870	1.161	0.764	0.997	0.731	0.723	0.844	1.094	0.824	0.923	0.744	0.851
PSTN	0.813	1.067	0.658	0.916	0.603	0.642	0.709	0.947	0.736	0.810	0.670	0.693
2Attn-LSTM	**0.746**	**1.043**	**0.609**	**0.870**	**0.519**	**0.549**	**0.658**	**0.832**	**0.683**	**0.749**	**0.589**	**0.528**

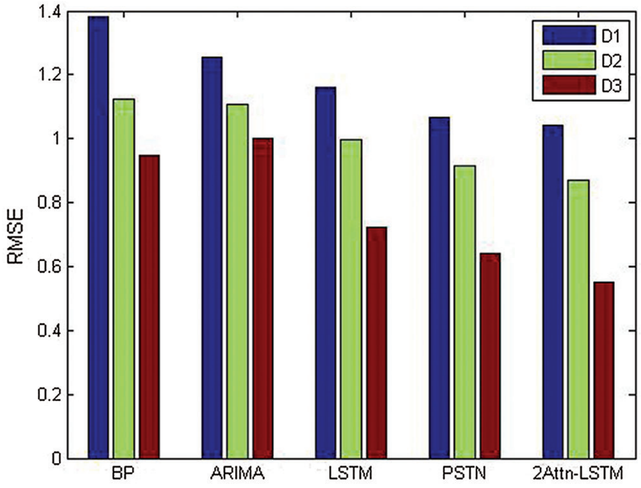

Figure 6: RMSE in the Wyoming dataset

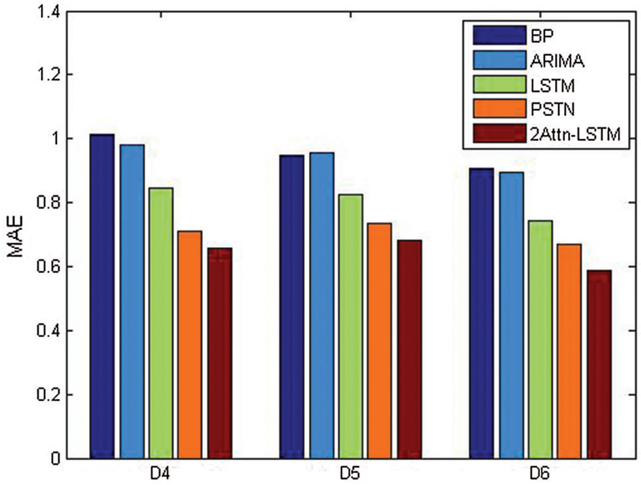

Figure 7: MAE in the Texas dataset

5 Conclusion and Future Work

We propose a deep 2Atts-LSTM architecture for short-term wind prediction, which integrates spatial-temporal features into a unified framework. In the first layer, an encoder of LSTM with mutual-information-based attention is adopted to extract the spatial features from the IMF components by VMD of wind speed. In the second layer, we employ temporal attention to select the relevant time step to make predictions adaptively. Experiments on real-world data illustrate the superior performance against 4 baselines in terms of MAE and RMSE simultaneously.

It is worth pointing out that the presented 2Atts-LSTM is a general model suitable for other spatial-temporal features. Furthermore, we will investigate how to integrate more sensor data into the model, such as atmospheric pressure and temperature. We think it is feasible to combine more variables; although, it is challenging to achieve the input selection and train the more complicated framework.

Acknowledgement: Thank you for the Wind Integration National Data set (WIND) provided by National Renewable Energy Laboratory (NREL).

Funding Statement: This work is supported in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions, Natural Science Foundation of China (No. 61103141, No. 61105007 and No. 51405241) and NARI Nanjing Control System Ltd. (No. 524608190024).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Global Wind Energy Council. Global wind report annual market update 2019. [Online]. Available at: https://gwec.net/wp-content/uploads/2020/08/Annual-Wind-Report_2019_digital_final_2r.pdf

[2] S. Fan, J. R. Liao, R. Yokoyama, L. Chen and W. Lee, "Forecasting the wind generation using a two-stage network based on meteorological information," *IEEE Transactions on Energy Conversion*, vol. 24, no. 2, pp. 474–482, 2009.

[3] M. R. Chen, G. Q. Zeng, K. D. Lu and J. Weng, "A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN, and LSTM," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6997–7010, 2019.

[4] L. Langberg, "Short-term prediction of the power production from wind farms," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 80, no. 2, pp. 207–220, 1999.

[5] L. Langberg, "Shot-term prediction of local wind conditions," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 89, no. 4, pp. 235–245, 2001.

[6] M. Lange and F. Ulrich, *Physical approach to short-term wind power prediction*. New York, NY, USA: Springer, 2006.

[7] M. Cellura, G. Cirrincione, A. Marvuglia and A. Miraoui, "Wind speed spatial estimation for energy planning in Sicily: Introduction and statistical analysis," *Renewable Energy*, vol. 33, no. 6, pp. 1237–1250, 2008.

[8] J. A. Carta, C. Bueno and P. Ramirez, "Statistical modelling of directional wind speeds using mixtures of von Mises distributions: Case study," *Energy Conversion and Management*, vol. 49, no. 5, pp. 897–907, 2008.

[9] Z. Huang and M. Gu, "Characterizing nonstationary wind speed using the ARMA-GRACH model," *Journal of Structural Engineering*, vol. 145, no. 1, Article 04018226, 2019.

[10] S. N. Singh and A. Mohapatra, "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting," *Renewable Energy*, vol. 136, no. 1, pp. 758–768, 2019.

[11] J. Zhou, J. Shi and G. Li, "Fine tuning support vector machines for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 52, no. 4, pp. 1990–1998, 2011.

[12] C. Y. Zhang, C. L. P. Chen, M. Gan and L. Chen, "Predictive deep Boltzmann machine for multiperiod wind speed forecasting," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1416–1425, 2015.

[13] M. Khodayar, O. Kaynak and M. E. Khodayar, “Rough deep neural architecture for short-term wind speed forecasting,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2270–2779, 2017.

[14] C. M. Amin, F. M. Sami and A. W. Trzynadlowski, “Wind speed and wind direction forecasting using echo state network with nonlinear functions,” *Renewable Energy*, vol. 131, no. 2, pp. 879–889, 2019.

[15] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai *et al.,* “Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction,” *IEEE Transactions On Sustainable Energy*, vol. 11, no. 1, pp. 509–523, 2020.

[16] Y. Liang, S. Ke, J. Zhang, X. Yi and Y. Zheng, “GeoMan: Multi-level attention networks for geo-sensory time series prediction,” in *Proc. IJCAI-18*, Stockholm, Sweden, pp. 3428–3434, 2018.

[17] H. Ling, J. Wu, P. Li and J. Shen, “Attention-aware network with latent semantic analysis for clothing invariant gait recognition,” *Computers Materials & Continua*, vol. 60, no. 3, pp. 1041–1054, 2019.

[18] J. Mai, X. Xu, G. Xiao, Z. Deng and J. Chen, “PGCA-Net: Progressively aggregating hierarchical features with the pyramid guided channel attention for saliency detection,” *Intelligent Automation & Soft Computing*, vol. 26, no. 4, pp. 847–855, 2020.

[19] K. Dragomiretskiy and D. Zosso, “Variational mode decomposition,” *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] Y. Yu, X. Si, C. Hu and J. Zhang, “A review of recurrent neural networks: LSTM cells and network architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[22] T. He and J. Droppo, “Exploiting LSTM structure in deep neural networks for speech recognition,” in *Proc. ICASSP*, Shanghai, China, pp. 5445–5449, 2016.

[23] D. Zhu, Y. Sun, X. Li and R. Qu, “Massive files prefetching model based on LSTM neural network with cache transaction strategy,” *Computers Materials & Continua*, vol. 63, no. 2, pp. 979–993, 2020.

[24] F. Altche and A. D. L. Fortelle, “An LSTM network for highway trajectory prediction,” in *Proc. ITCS*, Yokohama, Japan, pp. 353–359, 2017.

[25] A. H. Vo, T. Nguyen and T. Le, “Brent oil price prediction using Bi-LSTM network,” *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1307–1317, 2020.

[26] B. Yan, X. Tang, J. Wang, Y. Zhou and G. Zheng, “An improved method for the fitting and prediction of the number of Covid-19 confirmed cases based on LSTM,” *Computers Materials & Continua*, vol. 64, no. 3, pp. 1473–1490, 2020.

[27] G. E. P. Box and D. A. Pierce, “Distribution of residual autocorrelations in autoregressive-integrated moving average time series models,” *Journal of the American Statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.