

Learning Temporal and Spatial Correlations Jointly: A Unified Framework for Wind Speed Prediction

Qiaomu Zhu, *Student Member, IEEE*, Jinfu Chen, Dongyuan Shi, *Member, IEEE*, Lin Zhu, *Member, IEEE*, Xiang Bai, *Senior Member, IEEE*, Xianzhong Duan, *Member, IEEE*, Yilu Liu, *Fellow, IEEE*

Abstract—Leveraging both temporal and spatial correlations to predict wind speed remains one of the most challenging and less studied areas of wind speed prediction. In this paper, the problem of predicting wind speeds for multiple sites is investigated by using the spatio-temporal correlation. We proposed a deep architecture termed predictive spatio-temporal network (PSTN), which is a unified framework integrating a convolutional neural network (CNN) and a long short-term memory (LSTM). Initially, the spatial features are extracted from the spatial wind speed matrices (SWSMs) by the CNN at the bottom of the model. Then, the LSTM captures the temporal dependencies among the spatial features extracted from contiguous time points. Finally, the predicted wind speeds are given by the last state of the top layer of the LSTM, which are generated by using the spatial features and temporal dependencies. Though composed of two kinds of architectures, PSTN is trained with one loss function in an end-to-end (E2E) manner, which can learn temporal and spatial correlations jointly. Experiments for short-term predictions are conducted on real-world data, whose results demonstrate that PSTN outperforms prior methods.

Index Terms-- Convolutional neural networks, deep learning, spatial and temporal correlations, wind speed prediction

I. INTRODUCTION

WHILE providing clean energy to humankind, wind generation also poses significant challenges to the security and economy of power systems due to its intermittency, uncertainty, and stochasticity [1]. Accurate wind speed prediction has long been a pressing issue for large-scale wind power integration [2].

The community has made tremendous efforts in the area of wind speed prediction and scored remarkable achievements over several decades. However, the majority of previous work focuses merely on prediction for one single site by only using the temporal correlation. As another nature of wind speed, the spatial correlation has not drawn as much attention as temporal correlation. Though recognized as one of the most promising approaches, wind prediction using spatio-temporal correlation has still not been effectively implemented. Recently, stimulated by the enrichment of spatio-temporal data of wind farms, wind speed prediction with spatio-temporal correlation has been becoming a hot research topic in the electrical and meteorological communities.

Wind speed prediction with spatio-temporal correlation can be applied not only to wind turbines within a wind farm but also to wind farms scattered over a relatively extensive geographic terrain [3]. In general, the time scale [4], [5] and space scale (i.e., geographical scale) are two critical

parameters, which should be coordinated with the research objects. Specifically, when conducting wind speed prediction for multiple wind turbines within a small geographical scale (e.g., a wind farm), the time scale is often set to be small. These tasks are often short-term (30 mins to 6 hrs) and very short-term (a few seconds to 30 mins) predictions. The wind speed prediction for wind farms or wind farm groups scattered over a broad geographic scale often has relatively longer prediction horizons, such as middle term (6-24 hrs) and long term (1-7 days or more). Existing methods on wind speed prediction with spatio-temporal correlation can be mainly divided into four categories:

1) Physical methods [6]–[9] formulate the problem of wind speed prediction by modeling complete hydrodynamic and thermodynamic equation sets using environmental and geographic parameters, including temperature, humidity, surface roughness, etc., which often have good performance over longer horizons (from several hours to dozens of hours) [9]. Although physical methods can describe the nature of atmospheric motion well, their applications are still hampered by such factors as huge computational burdens, high dependency on parameters, limited temporal and spatial resolution, the impossibility of accounting for local topography, etc. [5].

2) Statistical methods use available historical observations to establish the stochastic approximation between the predictions and wind speed measurements [10], which can avoid the difficulties of understanding the physical mechanism. These methods include time-series analysis [11], [12], the Kriging interpolation method [13], [14], Von Mises distribution [15], etc., which are much simpler in implementation when compared to physical methods.

3) Probabilistic methods, based on the probabilistic analysis of historical measurements, use probability density estimation to represent wind resource characteristics. For example, Karaki et al. [16], [17] conducted prediction by using the probability distribution table which is established based on basic probability theory. TASTU et al. [18] introduced and evaluated a methodology allowing the issuing of probabilistic wind prediction optimally accounting for geographically dispersed information. Zhang et al. [19] discussed private information sharing for spatio-temporal forecasting. Moreover, adaptive resampling [20], time-adaptive quantile regression [21], time-adaptive quantile-copula [22] also brought insightful ideas into this field.

4) Artificial intelligence (AI) methods address the problem of wind speed from the machine learning perspective, namely learning the intricate mapping between the inputs and outputs

from massive historical data. AI methods are capable of capturing the complex nonlinearity and uncertainty of the wind speed time series. Various AI methods have emerged, including the fuzzy expert system [23], multi-layer perceptron (MLP) [24], adaptive general regression neural network (AGRNN) [25], fast training neural network [26], recurrent neural network (RNN) [27], etc.

However, most AI methods reported previously possess three main failings: 1) Most are shallow in architecture, which would be not enough in representing the complicated mapping between the inputs and outputs [28]. Moreover, the generalization ability of shallow architectures may be constrained when solving classification and regression problems [29]. 2) Many AI methods lack the targeted mechanism for spatial correlation. For instance, the observations from the neighboring sites are fed into the model indiscriminately. Although spatio-temporal information is introduced in this way, the effectiveness of spatial correlation is undoubtedly weakened. 3) Most only predict the wind speed at each site separately, neglecting to share prediction knowledge among related sites. It is better to conduct predictions for multiple sites simultaneously since there is a lot of shared information among the spatially correlated sites.

Recently, the community has made a good start in applying deep learning for wind speed prediction. For example, Zhang et al. [30] and Wang et al. [31] constructed models for wind speed prediction based on the deep belief network (DBN). Khodayar et al. [32] proposed wind speed models based on the stacked autoencoder (SAE) and incorporated with the rough set theory. Hu et al. [33] proposed a transfer learning framework for wind speed prediction based on deep neural networks. Though they achieved promising performances, these methods do not account for the spatial correlation. In this paper, we attempt to construct a deep architecture for wind speed prediction that is capable of leveraging both temporal and spatial correlations. The proposed prediction model, termed the predictive spatio-temporal network (PSTN), combines a convolutional neural network (CNN) for spatial features extraction and a long short-term memory (LSTM) for capturing temporal dependencies. Trained in an end-to-end (E2E) manner [34], PSTN is capable of learning temporal and spatial correlations jointly and automatically. Meanwhile, our model incorporates multi-task learning (MTL) [28]. Namely, wind speed prediction is conducted for multiple sites simultaneously rather than a single site. Consequently, PSTN can be seen as a pattern learning model that has the ability to predict a spatio-temporal sequence. As a case study, we test PSTN with real-world data for short-term wind speed predictions. The experimental results demonstrate that the proposed PSTN outperforms the baselines.

The main contributions of this paper can be summarized as follows: 1) Following the idea of MTL, a new problem of predicting wind speeds for multiple sites simultaneously is proposed. Here, this problem is formulated as a spatio-temporal sequence prediction task and is solved by leveraging both temporal and spatial correlations. 2) PSTN, a novel deep architecture for wind speed prediction, is presented, which integrates a CNN and an LSTM into a unified framework. This model achieves spatial feature extraction and captures

temporal dependency automatically and intrinsically. To the best of our knowledge, we are the first to employ deep learning methods for wind speed prediction by leveraging both spatial and temporal correlations. 3) Numerical experiments are conducted on real-world wind-speed data, and the results show that the proposed model outperforms prior methods in short-term predictions.

The rest of this paper is organized as follows. Section II analyses the temporal and spatial correlations of wind speed briefly. Section III formulates the wind speed prediction for multiple sites as a spatio-temporal sequence prediction task. Section IV provides background theories on CNN and LSTM. Section V presents the unified framework for wind speed prediction which is capable of learning spatial and temporal correlations jointly. The experimental results are discussed in Section VI. The conclusion is made in Section VII.

II. TEMPORAL AND SPATIAL CORRELATIONS

In the wind velocity vector field, the spatio-temporal function describing the wind speed at any site is continuous in most cases [35]. On the one hand, the wind speed at a certain site (e.g., a wind farm, a wind turbine, etc.) is related to its historical values, which is referred to as the temporal correlation. On the other hand, the wind speeds at multiple sites within a certain geographic scale are not statistically independent, instead, they have spatial correlations with others. Therefore, the knowledge for wind speed prediction is contained in both temporal and spatial correlations which can both be utilized to enhance prediction accuracy.

To illustrate the temporal and spatial correlations intuitively, a 10×10 array (i.e., consisting of 10 rows and 10 columns) of wind turbines is used as an example. Each site (i.e., turbine) in the array is indexed by a two-dimensional rectangular coordinate (i, j) where i, j denotes the row index and column index respectively. Meanwhile, mutual information (MI), a fundamental statistical approach to determine both linear and non-linear correlations [32], is applied to measure the temporal and spatial correlations for the wind speed time series. The MI between two variables measures how much information is known about one variable via the observation of the other variable [36]. Suppose there is a pair of wind speed time series (X, Y) , and the MI between X and Y , denoted as $I(X, Y)$, is given by (1). The larger $I(X, Y)$ is, the stronger the correlation between X and Y is.

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \quad (1)$$

where

$$\begin{cases} H(X) = - \sum_{x \in X} P_X(x) \log(P_X(x)) \\ H(X, Y) = - \sum_{x \in X, y \in Y} P_{XY}(x, y) \log(P_{XY}(x, y)) \end{cases} \quad (2)$$

$H(X)$ represents the entropy of X , $H(X, Y)$ represents the joint entropy of X and Y , and $P(\cdot)$ denotes the probability density function.

Suppose the site (5,6) is the reference, calculate the MI between the wind speed time series at (5,6) and each site in the array. The MI indices are displayed by a gray-scale image (of the size of 10×10) where each block corresponds to a certain site, as shown in Fig. 1(a). The spatial correlation is apparent

and revealed by the figure, which mainly lies in two aspects: 1) the wind speed at (5,6) is related to those at all other sites within the wind turbine array. 2) the wind speed at (5,6) has strong correlations with those at the sites geographically close to (5,6) while it has relatively weakly correlations with those at sites far away from (5,6). Therefore, within a certain geographical range, the spatial correlation of wind speed is absolute while its degree is relative.

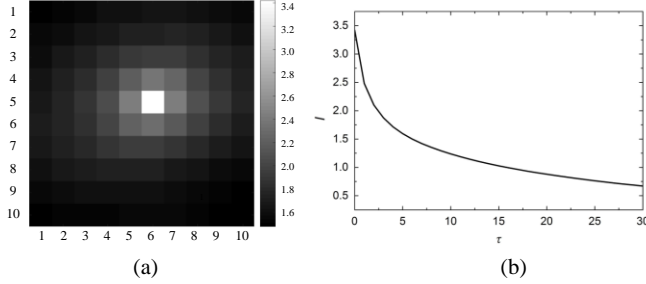


Fig. 1. Spatial and temporal correlations. (a) MI indices between wind speed time series at (5,6) and other sites. (b) MI indices between wind speed time series at (5,6) and its time-lag time series.

It is worth mentioning that the spatial correlations among the sites are also related to wind direction. The sites in the prevailing wind direction usually have stronger correlations than those not in the prevailing wind direction [37]. Moreover, some factors, such as the terrain, surface roughness, etc., also have an impact on the spatial correlations. Therefore, spatial correlations are quite complicated and are expected to be recognized in a targeted manner.

Likewise, calculating the MI between the wind speed time series at one certain site and its time-lag series [32] illustrates the temporal correlation. Fig. 1(b) shows the MI indices at (5,6) with time-lag τ from 0 to 30. The figure indicates that the wind speed time series has autocorrelations, namely, the value of wind speed is related to the historical values, especially the relatively recent ones.

To summarize, the temporal and spatial correlations are two critical aspects of wind speed, which could not be separated. Taking both temporal and spatial correlations into account is the key to enhancing the wind speed prediction accuracy.

III. PROBLEM FORMULATION

As analyzed above, the wind speeds at the sites within a certain geographic terrain may be highly correlated. There is a lot of sharing of information among the predictions for each site. Therefore, it is promising to integrate multiple single-site predictions in the manner of multi-task learning (MTL). MTL learns several tasks at the same time with the aim of mutual benefit, which is a paradigm in machine learning [38]. In this paper, we are concerned with the problem of predicting wind speeds for multiple sites simultaneously, rather than the prediction for a single site.

In essence, predicting wind speed at multiple sites is a spatio-temporal sequence prediction problem, which will be discussed in two-dimensional space. Generally, suppose our research object is an array, uniformly spaced or non-uniformly spaced, consisting of M rows and N columns over a spatial region which could be represented by an $M \times N$ grid, as shown in Fig. 2(a). Each site is indexed by a two-dimensional

rectangular coordinate (i,j) ($1 \leq i \leq M$, $1 \leq j \leq N$) where i and j denote the row index and column index respectively. For each site in the array, the wind speed is a typical one-dimension (1D) time series, as depicted in Fig. 2(b). Therefore, the wind speed time series for the whole array is composed of $M \times N$ time series, which is a spatio-temporal sequence.



Fig. 2. (a) The research object over a spatial region is represented by an $M \times N$ grid. (b) Wind speed time series at the site (1, N).

At time t , the wind speed at (i,j) is denoted as $x(i,j)_t$. A spatial wind speed matrix (SWSM) $\mathbf{x}_t \in \mathbf{R}^{M \times N}$ is defined to represent the wind speeds at all sites in the array at t , i.e.,

$$\mathbf{x}_t = \begin{bmatrix} x(1,1)_t & x(1,2)_t & \cdots & x(1,N)_t \\ x(2,1)_t & x(2,2)_t & \cdots & x(2,N)_t \\ \vdots & \vdots & & \vdots \\ x(M,1)_t & x(M,2)_t & \cdots & x(M,N)_t \end{bmatrix} \quad (3)$$

The SWSM contains rich spatial information. On the one hand, the SWSM is made up of the observations from all sites. On the other hand, the values of wind speed are in the same order as their corresponding sites in the array. Therefore, given an SWSM, at least two clues can be inferred directly: 1) the adjacent relationships between any two sites. 2) the approximate relative distance between any two sites. It is worth noting that a single SWSM doesn't refer to any temporal information because all of its elements are observed at the same time. Therefore, a spatio-temporal sequence describing the wind speed of the array is constructed by organizing the SWSMs in chronological order. That is, $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\}$, where $L \geq 1$ denotes the length of the sequence, as shown in Fig.3. An analogy with multimedia might prove helpful in understanding the spatio-temporal sequence, that is, the spatio-temporal sequence can be regarded as a video, while any SWSM is a frame of the sequence which can be regarded as a frame in the video, i.e., an image.

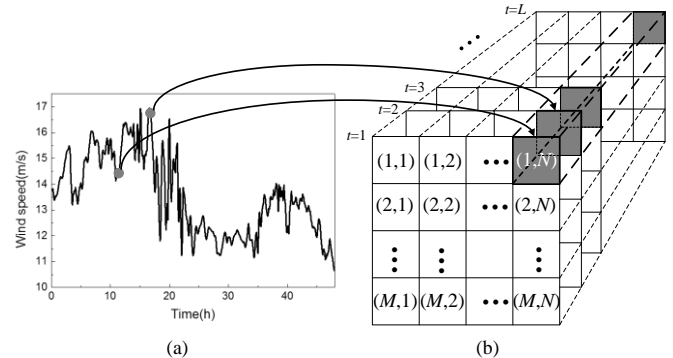


Fig. 3. (a) The wind speed time series at the site (1, N). (b) The spatio-temporal sequence describing the wind speed of the array.

The task of wind speed prediction for an array is to predict the SWSM instead of a single value of wind speed, which

naturally becomes a spatio-temporal sequence prediction problem. Specifically, at time t , an SWSM $\hat{\mathbf{x}}_{t+\tau}$ can be predicted based on the k previous SWSMs ($k \leq t$), i.e.,

$$\hat{\mathbf{x}}_{t+\tau} = f(\mathbf{x}_{t-k+1}, \mathbf{x}_{t-k+2}, \dots, \mathbf{x}_t | \boldsymbol{\theta}) \quad (4)$$

where τ denotes the prediction horizon, f represents the explicit or implicit function of the prediction model, and $\boldsymbol{\theta}$ represents the parameter vector of the prediction model. Note that, when $M=N=1$, this problem will be cast as a conventional wind speed prediction, i.e., predicting wind speed for a single site. Therefore, compared to the wind speed prediction for a single site, the prediction for multiple sites is a more general case.

IV. BACKGROUND THEORIES

A. Convolutional Neural networks

CNNs, one of the typical deep learning models, have trainable convolutional kernels and local neighborhood pooling operations, which are alternately applied on the raw inputs, extracting hierarchy spatial features [39]. Due to the superior performance on spatial information handling, CNNs have been successfully applied in various fields, including video analysis [40], image recognition [41], language processing [42], etc. Fig.4 depicts the typical structure of CNN. There are three types of components constructing the CNN, including the convolution layer, pooling layer, and fully-connected layer.

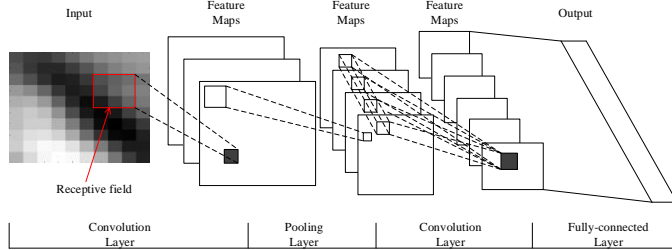


Fig. 4. The typical structure of CNN.

At the convolution layer, the input feature maps are convolved with learnable kernels (also termed filters) firstly and then transformed through a non-linear mapping operated by the activation function. The rectangular region each convolutional kernel covers is termed a receptive field [43], shown as Fig.4, referring to the term from biological vision. Generally, the convolution layer is computed by

$$\mathbf{x}_q^l = g\left(\sum_{p \in \mathbf{M}_q} \mathbf{x}_p^{l-1} * \mathbf{k}_{pq}^l + \mathbf{b}_q^l\right) \quad (5)$$

where \mathbf{x}_q^l denotes the q th feature map of the l th layer, \mathbf{k}_{pq}^l denotes the trainable kernel of the l th layer, \mathbf{b}_q^l denotes the bias vector, \mathbf{M}_q denotes a selection of inputs, $*$ denotes the convolution operation, and g denotes the activation function. Through the receptive field strategy, the stronger dependencies among topological nearby values can be exploited effectively.

The pooling layer aims to reduce the dimension of the input feature maps, which is achieved by a pooling operation, given as

$$\mathbf{x}_q^l = \text{down}(\mathbf{x}_q^{l-1}) \quad (6)$$

where $\text{down}(\cdot)$ represents the sub-sampling function, such as max-pooling and average-pooling.

The fully-connected layer flattens the two-dimensional (2-D) feature maps and put through the activation function to form the 1-D output, i.e.,

$$\mathbf{x}^l = g(\mathbf{w}^l \mathbf{x}^{l-1} + \mathbf{b}^l) \quad (7)$$

where \mathbf{w}^l and \mathbf{b}^l represents the weight matrix and bias vector respectively.

With the three special designs, including receptive fields, local connectivity, and shared weights, CNNs are capable of extracting spatial features effectively.

B. Long Short-term Memory

LSTM [44], a variant of the recurrent neural network (RNN), has special designs for overcoming the gradient vanishing problem that troubles conventional RNNs. LSTMs have shown their strength in handling sequential data, and have been applied successfully in various tasks, such as image captioning [45], language modeling [46], video analysis [47], etc.

An LSTM [48] with one layer, illustrated in Fig. 5, consists of an internal memory cell \mathbf{c}_t and three multiplicative gates, including the forget gate \mathbf{f}_t , input gate \mathbf{i}_t , and output gate \mathbf{o}_t . Each time, after receiving the input \mathbf{x}_t , LSTM updates its internal state \mathbf{c}_t by using the current input \mathbf{x}_t and previous internal state \mathbf{c}_{t-1} , namely, $\mathbf{c}_t = g(\mathbf{x}_t, \mathbf{c}_{t-1})$. The final state \mathbf{h}_t will be determined by \mathbf{c}_t and \mathbf{o}_t . Controlled by the input and output gates, the memory cell is able to store the previous information for a long period of time. Meanwhile, the states stored in the memory cell can be cleared by the forget gate. The formulations of the components in the LSTM are given by (8) to (10). In this way, the LSTM is capable of capturing the temporal dependencies of the sequence.

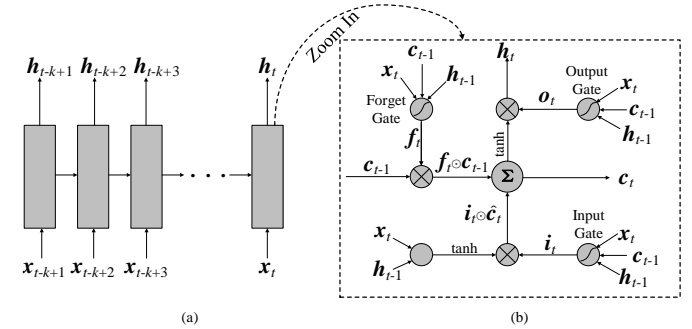


Fig. 5. (a) The working scheme of a single-layer LSTM. (b) The structure of a basic LSTM unit.

$$\begin{bmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \end{bmatrix} = \sigma \left(\begin{bmatrix} \mathbf{W}_{ix} & \mathbf{W}_{ih} & \mathbf{W}_{ic} \\ \mathbf{W}_{fx} & \mathbf{W}_{fh} & \mathbf{W}_{fc} \\ \mathbf{W}_{ox} & \mathbf{W}_{oh} & \mathbf{W}_{oc} \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \\ \mathbf{c}_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{b}_i \\ \mathbf{b}_f \\ \mathbf{b}_o \end{bmatrix} \right) \quad (8)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot (\mathbf{W}_{cx} \mathbf{x}_t + \mathbf{W}_{ch} \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (9)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (10)$$

where \mathbf{W}_{ix} , \mathbf{W}_{ih} , \mathbf{W}_{ic} , \mathbf{W}_{fx} , \mathbf{W}_{fh} , \mathbf{W}_{fc} , \mathbf{W}_{ox} , \mathbf{W}_{oh} , \mathbf{W}_{oc} , \mathbf{W}_{cx} , \mathbf{W}_{ch} are weight matrices for the corresponding inputs, \mathbf{b}_i , \mathbf{b}_f , \mathbf{b}_o , \mathbf{b}_c are the bias vectors, \odot denotes the Hadamard product, and σ represents the activation function.

A deep LSTM can be constructed by stacking multiple LSTM layers, which often works better than a single layer [49] and has been applied to solve many real-world sequence modeling problems [50], [51]. The back-propagation through time (BPTT) algorithm is widely applied to training LSTMs,

which is effective for minimizing the objective functions for the RNNs and LSTMs.

V. METHODOLOGY

In this section, we first propose the basic idea for spatio-temporal sequence modeling. Then, a deep architecture learning temporal and spatial information jointly is presented.

A. The strategy for spatio-temporal sequence modeling

Taking into account both temporal and spatial correlations is the key to achieving the spatio-temporal sequence prediction. Here, we propose a two-stage strategy for modeling the spatio-temporal sequence, that is, extracting the spatial features first, followed by capturing the temporal dependencies among the extracted spatial features. In this way, the problem can be further formulated as

$$\hat{\mathbf{x}}_{t+\tau} = f(\mathbf{x}_{t-k+1}, \mathbf{x}_{t-k+2}, \dots, \mathbf{x}_t) \approx f_{tem}(s_{t-k+1}, s_{t-k+2}, \dots, s_t) \quad (11)$$

$$= f_{tem}(f_{spa}(\mathbf{x}_{t-k+1}, \mathbf{x}_{t-k+2}, \dots, \mathbf{x}_t))$$

where $s_{t-k+1}, s_{t-k+2}, \dots, s_t$ denote the spatial features extracted from $\mathbf{x}_{t-k+1}, \mathbf{x}_{t-k+2}, \dots, \mathbf{x}_t$ respectively, f_{tem} and f_{spa} represent the implicit function of temporal dependency capturing and spatial feature extraction respectively. Following this strategy, the spatial feature extraction aims to learn the spatial correlations of different sites at the same frame (i.e., the SWSM). The task of temporal dependency capturing is to model the temporal dependencies among the spatial features extracted from different frames.

B. The proposed network architecture

Following the aforementioned two-stage strategy, the wind prediction model for multiple sites can be made up of two functional modules, namely, one (called spatial model) for extracting spatial feature, and the other (called temporal model) for capturing temporal dependency.

In this paper, CNN is employed as the spatial model, which provides two main benefits: 1) Owing to the topology-preserving property of kernels [52], CNNs can receive SWSMs (which are 2D tensors) directly and without element rearrangement, which enables CNNs to exploit the spatial property on SWSMs thoroughly. 2) With deep architecture, CNNs extract the spatial features automatically and hierarchically. That is, feature learning can incorporate progressively larger spatial regions [53].

Due to the sequential nature of the chronologically extracted spatial features, a deep LSTM is utilized as the temporal model in this paper, whose advantages are three-fold: 1) LSTM has a robust capability of capturing both long-term and short-term temporal dependencies within a sequence, which can satisfy the needs for different prediction horizons. 2) LSTM can back-propagate error differentials to its input, which allow us to integrate the spatial model and temporal model into a unified framework. 3) LSTM can share parameters through time, which enables PSTN to maintain a constant number of parameters regardless of the length of the input sequence. Therefore, PSTN can control the number of the parameters in a reasonable range, which significantly benefits training the model.

The network architecture of the proposed model is shown

in Fig.6(b), where the SWSMs are displayed by gray-scale images, as illustrated in Fig.6(a). This architecture is a unified framework integrating a CNN and a deep LSTM. At the bottom of the model, the CNN automatically extracts the spatial features which have embedded the information revealing the spatial correlations among the different sites. On top of the CNN, an LSTM is built to capture the temporal dependency among the spatial features, chronologically extracted by the CNN. Finally, the predicted SWSM is given by the last state of the top layer of the LSTM, which is generated by using the spatial features and temporal dependencies. It is worth mentioning that SWSMs cannot be output by the LSTM directly due to the fact that the outputs of LSTM are 1D vectors. Thus, a one-to-one correspondence between the elements in the SWSM and those in the 1D output vectors needs building. In this way, the SWSM can be constructed by the outputs of the model according to the location of the elements.

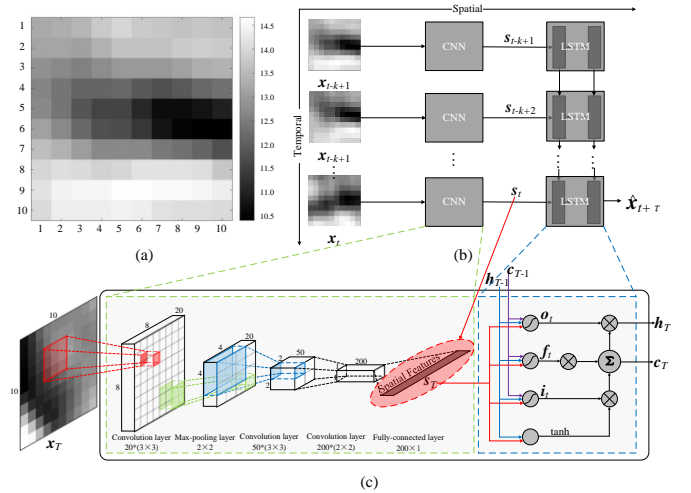


Fig. 6. (a) A gray-scale image with the size of 10×10 , which is generated by an SWSM. Each block in the gray-scale image corresponds to an element in the SWSM, and the blocks are in the same order to their corresponding elements in the SWSM. (b) The network architecture. In spatial dimension, the spatial features are extracted by the CNN. In temporal dimension, the LSTM receives the spatial features in chronological order. (c) The detailed schematic diagram showing the architecture and connections amongst PSTN at a single time step T . Meanwhile, the diagram presents a succinct information flow within as well as between time steps. In the example, CNN adopts a 5-layer structure and LSTM is equipped with one layer.

To clearly illustrate the work process of PSTN, a detailed schematic diagram is shown as Fig.6(c). Meanwhile, the prediction flow chart is represented in Fig.7. PSTN predicts the target SWSM based on k previous SWSMs. Concretely, at every time step $T = t-k+1, t-k+2, \dots, t$:

- The CNN at the bottom of PSTN receives an SWSM and extracts the spatial features (i.e., s_T) through its hierarchical architecture.
- Then, the LSTM is fed the spatial features extracted by the CNN at present (i.e., s_T) and the outputs of LSTM at previous time step (i.e., h_{T-1} and c_{T-1}). After calculating, the LSTM updates and outputs the final state and inner state at present time step, i.e. h_T and c_T .

At the final time step $T = t$, the predicted SWSM is given by the final state of the top layer of LSTM, i.e., $\hat{\mathbf{x}}_{t+\tau} = h_t$.

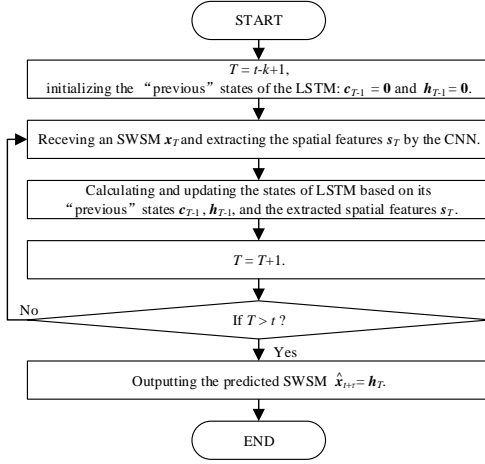


Fig. 7. The flow chart of wind prediction by PSTN.

C. Spatial correlation domain and receptive field

To clearly illustrate the process of spatial feature extraction in PSTN, we will discuss two concepts here, i.e., spatial correlation domain and receptive field. The connection between these two concepts is also the key to understanding the wind speed prediction with spatio-temporal correlation achieved by PSTN.

To predict the wind speed at one certain site, we are expected to introduce the wind speed data from a limited and specific region, defined as spatial correlation domain in this paper, instead of the whole array. The wind speeds at the sites within the same spatial correlation domain are strongly correlated, while the wind speeds at the sites from different spatial correlation domains have a relatively weaker correlation. Suppose the spatial correlation domain of the site (i, j) is a square region covering (i, j) , denoted as \mathbf{G} as shown in Fig. 8(a). Due to the spatial correlation of the wind speed, the basic idea of spatial features extraction is to extract features from each local region and finally to form complete spatial features.

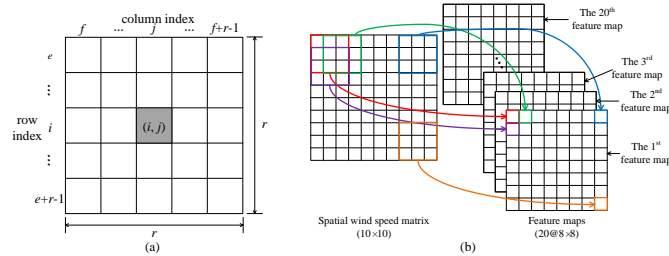


Fig. 8. (a) A spatial correlation domain \mathbf{G} with the size of $r \times r$. e, f is the start row index and start column index respectively, and r denotes the spatial size of the spatial correlation domain. The larger r is, the larger the spatial correlation domain is. Generally, i and j are expected to satisfy the following two inequalities, $e \leq i \leq e+r-1$, and $f \leq j \leq f+r-1$. (b) The input process. Suppose the size of the input, i.e. the SWSM, is 10×10 . There are 20 filters with a receptive field with the size of 3×3 in the first layer of the CNN. Thus, 20 feature maps with size 8×8 ($8=10-3+1$), marked as $20@8 \times 8$, are generated. The elements are in the same order as their corresponding spatial correlation domains on the SWSM.

In biological and computer vision, the image features, e.g., orientation, motion, disparity, etc., are initially extracted over small local regions of visual space, i.e., the receptive fields, by the front end of a vision system. The pixels within a receptive field are considered to have a strong spatial correlation with each other, while they are weakly related to those out of the

receptive field. Logically, the receptive field is consistent with the spatial correlation domain. The processing of spatial information from the SWSMs starts in the bottom layer of the CNN. The receptive field can be regarded as a sliding window, which will dock every spatial correlation domain in turn. In this way, the spatial information, such as distance, prevailing wind direction, etc., is captured by the convolutions and capsuled in the parameters of the CNN. Intuitively, the input process can be illustrated as Fig.8(b). Likewise, the subsequent convolution layer extracts the spatial features from the feature maps generated by its lower layer. In this way, spatial features of larger regions are hierarchically learned by CNN based on spatial features of smaller regions. That is, when we have multiple convolutional layers, the initial layer extract more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more complex and more suited to the problem of wind speed prediction. In this way, spatial features of larger regions are hierarchically learned by CNN based on spatial features of smaller regions [54].

D. Training algorithm

Though the model is composed of two different kinds of network architectures, i.e., the CNN and LSTM, it can be jointly trained with one loss function. Note that, the loss function in this task is constructed by the differences between the predicted SWSMs and their corresponding ground-truth values, instead of those between the single values in the wind speed prediction for a single site. Thus, Frobenius-norm is employed to measure the difference between two matrices in this paper. Given the historical data and the ground truth of prediction task, the loss function J is defined as

$$J = \frac{1}{M \times N} \frac{1}{|\mathbf{P}|} \sum_{t \in \mathbf{P}} \|\mathbf{x}_{t+\tau} - \hat{\mathbf{x}}_{t+\tau}\|_F^2 \quad (12)$$

where \mathbf{P} denotes the collection of time points when the predictions are conducted of training samples, $|\mathbf{P}|$ denotes the number of training samples, $\|\cdot\|_F$ represents the Frobenius-norm, $\mathbf{x}_{t+\tau}$ is the ground truth while $\hat{\mathbf{x}}_{t+\tau}$ is the predicted value. f can be found by adjusting the parameters θ during training. The aim of training is to find the optimal parameters $\hat{\theta}$ which minimize the loss function.

In the LSTM, error differentials are propagated in the opposite directions of the arrows shown in Fig.7(a), i.e., back-propagation through time (BPTT). Then, the error differentials are fed back to the bottom layer of the LSTM, then propagated to the top layer of the CNN, and further propagated to the other layers of CNN. In this way, every layer in the model can adjust the parameters by the guiding of error differentials. It is worth mentioning that, for the LSTM (the temporal model), the process of the parameter optimization is involved in spatial information which is encapsulated in the errors between the predicted SWSMs and their ground truth. Likewise, for the CNN (the spatial model), the error differential guiding the parameter adjusting also contains the temporal information since it propagated from the LSTM. In this way, the processes of spatial and temporal correlation learning are coupled, which enables the whole framework to learn the spatial and temporal correlation jointly and intrinsically. There are several

commonly used BPTT algorithms which can be employed, e.g., stochastic gradient descent (SGD), Adadelata, RMSprop, Adam, etc [55]. After training, the prediction model is capable of predicting the spatio-temporal sequence.

VI. CASE STUDY

To evaluate the effectiveness of PSTN, the experiments of short-term prediction are conducted on real-world data. Moreover, the performance comparison between PSTN and NWP is provided.

A. Data set

The Wind Integration National Dataset (WIND), provided by the National Renewable Energy Laboratory (NREL), contains wind speed data with the time-resolution of 5-min for more than 126,000 sites in the continental United States for the year 2007-2013. Based on WIND, a 10×10 wind turbine array within a wind farm in the state of Wyoming is selected for short-term prediction, as shown in Fig.9. The annual dataset of 2012 is selected, and the time-resolution of the dataset is transferred to 10 min. Thus, there are 52,704 frames in the dataset. The maximum wind speed is 38.55m/s, and the minimum is 0.02m/s.

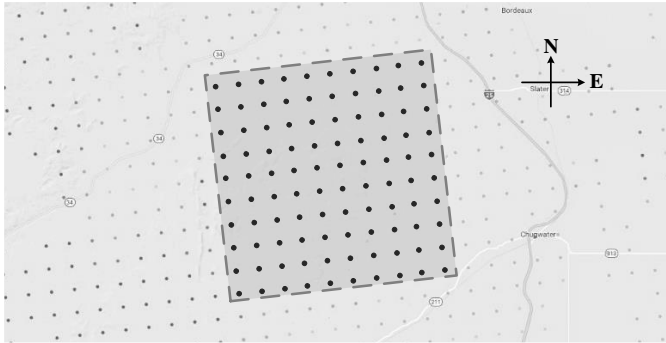


Fig. 9. A 10×10 wind turbine array in the state of Wyoming.

B. Evaluation indices

The root mean squared error (RMSE) and mean absolute percentage error (MAPE) are widely employed as the evaluation indices in the task of wind speed prediction for a single site. Specifically, for the site (i,j) , the RMSE (denoted as $\varepsilon_r(i,j)$) and MAPE (denoted as $\varepsilon_m(i,j)$) are given by

$$\varepsilon_r(i,j) = \sqrt{\frac{1}{|\mathcal{Q}|} \sum_{t \in \mathcal{Q}} (x(i,j)_{t+\tau} - \hat{x}(i,j)_{t+\tau})^2} \quad (13)$$

$$\varepsilon_m(i,j) = \frac{1}{|\mathcal{Q}|} \sum_{t \in \mathcal{Q}} \frac{|x(i,j)_{t+\tau} - \hat{x}(i,j)_{t+\tau}|}{|x(i,j)_{t+\tau}|} \times 100\% \quad (14)$$

where $\hat{x}(i,j)_{t+\tau}$ and $x(i,j)_{t+\tau}$ represents the predicted and ground-truth wind speed at the site (i,j) respectively, \mathcal{Q} denotes the collection of time points when the predictions are conducted of testing or validation samples.

In this paper, the prediction results are matrices rather than single values. Naturally, two novel evaluation indices RMSE for array (RMSE-A), denoted as ε_R , and MAPE for array (MAPE-A), denoted as ε_M , are defined, i.e.,

$$\varepsilon_R = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \varepsilon_r(i,j)^2} \quad (15)$$

$$\varepsilon_M = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \varepsilon_m(i,j) \quad (16)$$

The Pearson correlation, denoted as COR, is also employed as an evaluation index, which is defined as

$$COR = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N r(i,j) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{\text{cov}(x_{t+\tau}(i,j), \hat{x}_{t+\tau}(i,j))}{\sigma_{x_{t+\tau}(i,j)} \sigma_{\hat{x}_{t+\tau}(i,j)}} \quad (17)$$

where $r(i,j)$ represents the Pearson correlation for the site (i,j) , $\text{cov}(X,Y)$ represents the covariance between X and Y , and σ_X represents the standard deviation of X . The bigger the COR is, the better the performance the model has.

Moreover, the maximum instantaneous error (MIE) is defined to measure the individual error control ability of the models, expressed as

$$\varepsilon_{MIE} = \max_{t \in \mathcal{Q}}(\varepsilon(t)) = \max_{t \in \mathcal{Q}} \{\|\mathbf{x}_{t+\tau} - \hat{\mathbf{x}}_{t+\tau}\|_F\} \quad (18)$$

where $\varepsilon(t) = \|\mathbf{x}_{t+\tau} - \hat{\mathbf{x}}_{t+\tau}\|_F$ represents the instantaneous absolute error.

To summarize, RMSE-A, MAPE-A, and COR are employed to evaluate the overall error control ability of models, while MIE is employed to evaluate the individual error control ability.

C. Baseline algorithms

To verify the superiority of the proposed PSTN, two statistical models – the persistence method (PR) and vector autoregression (VAR) – and four AI models – MLP, RNN, LSTM, and predictive deep CNN (PDCNN) – are employed as baselines.

PR, a simple and standard benchmark algorithm for short-term prediction, uses an assumption that the predicted data is the same as the data at the present time. PR is not employed in the long-term prediction tasks due to its poor performance with long prediction horizons.

VAR, the natural extension of AR model, is one of the most successful, flexible, and easy to use models for the analysis of multivariate time series [56]. The SWSMs are flattened to 1D vectors to fed to the VAR. The model selection of VAR is according to the Akaike information criteria [56].

Without the structure adaptive to time-varying, MLP needs to flatten all the inputs at different time points into a unified vector, that is, modeling the time sequence from the spatial perspective. MLP is composed of three layers, i.e., the input layer, hidden layer, and output layer. The input layer and hidden layer adopt ReLU [57] as the activation function, while the output employs the sigmoid activation function to protect it from sparsity. The dimensions of the input and output layers equal the dimensions of input and output vectors respectively. The dimension of the hidden layer is a hyper-parameter to be selected from $\{1000, 1100, \dots, 2000\}$ through model selection. Moreover, MLP is trained by Adam algorithm.

Differently from MLP, RNN and LSTM are able to capture temporal dependencies in a temporally dynamic manner [58], which have a strong capability of capturing temporal correlation within the time series. Like the MLP, RNN and LSTM cannot be fed with the 2D matrices (i.e., the SWSM)

directly, which receives 1D vectors flattened by the SWSMs. The RNN and LSTM both adopt tanh as their activation functions. Except for the input dimension, other hyper-parameters of the baseline RNN and LSTM are the same as those of the LSTM in PSTN.

PDCNN [59] is an assemble architecture for wind speed prediction, which use CNNs and an MLP to model spatial and temporal correlations respectively. In PDCNN, the CNNs adopt the same structure as that of PSTN. The dimension of the output layer is 100, and the dimension of the secondary top layer is a hyper-parameter which is selected from {100, 200, ..., 500}. Moreover, all baselines are implemented in Python.

D. Model selection and implementation details

It is still an open issue how to determine the optimal hyper-parameters of different deep-learning models for different problems. In this paper, the hyper-parameters are determined by the combination of the artificial experience and machine searching.

Specifically, in each prediction task, the dataset is divided into three subsets, including the training set, validation set, and testing set, which contains the first 60%, the following 10%, and the last 30% frames respectively. The training set and testing set serve for model training and testing respectively, while the validation set is used for model selection and over-fitting prevention. The candidate models that have a different hyper-parameter combination are trained with the training set first, and then evaluated with the validation set. The model producing the best performance in terms of the two indices (i.e., ε_R and ε_M) is regarded as being governed by the optimal hyper-parameters, which will be selected to compare with baseline models. Of course, the hyper-parameters of the baselines are determined in this way as well.

PSTN is trained by RMSprop with the epoch for training set to 100. Also, early-stopping is employed to enhance the model's generalization and prevent the model from over-fitting. We implement the network within the Keras framework with TensorFlow backend [60]. Experiments are carried out on a 64-bit PC with Intel core i7-7820 CPU/32.00 GB RAM.

E. Short-term prediction

There are 31,622, 5,270, and 15,812 in the training set, validation set, and testing set respectively. The configuration of the model for short-term prediction is summarized in Table I. The hidden dimension of the 6th layer (i.e., the first LSTM layer) is a hyper-parameter to be selected from {100, 200, 300, 400, 500}. Specially, no activation functions are applied to the 5th layer that just functions as a flattening layer, namely, to transform the 2D feature maps to 1D vectors fed to the LSTM.

TABLE I

THE CONFIGURATION OF THE MODEL FOR SHORT-TERM PREDICTION

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

-	Input	SWSMs of size 10×10
---	-------	---------------------

To evaluate the short-term prediction performance of the models, experiments with the prediction horizon τ ranging from 10 min to 3 h are conducted. The look-back period is 3 hours for all models, that is, all models use the previous 18 SWSMs (i.e., $x_{t-17}, x_{t-16}, \dots, x_t$) to predict one SWSM in the future. The prediction performance of models with optimal hyper-parameters is evaluated on the testing set by the four indices, the results are shown in the Table II to V.

TABLE II

RMSE-A OF PREDICTION MODELS FOR SHORT-TERM PREDICTION(M/S)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	0.564	0.992	1.331	2.078	3.301	3.721
VAR	0.542	0.926	1.163	1.720	2.649	3.294
MLP	0.553	0.964	1.197	1.684	2.461	2.782
RNN	0.547	0.936	1.038	1.473	2.106	2.311
LSTM	0.543	0.919	0.995	1.236	1.892	2.108
PDCNN	0.537	0.883	0.961	1.194	1.804	2.062
PSTN	0.531	0.864	0.927	1.139	1.643	1.924

TABLE III

MAPE-A OF PREDICTION MODELS FOR SHORT-TERM PREDICTION (%)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	4.405	8.038	11.021	17.850	27.814	36.139
VAR	4.117	7.742	10.435	15.064	24.995	33.047
MLP	4.121	7.694	10.082	15.345	24.903	32.385
RNN	4.005	7.453	9.248	13.186	21.495	28.741
LSTM	3.948	7.074	9.045	12.038	20.274	25.379
PDCNN	3.902	6.927	8.003	11.856	19.954	23.058
PSTN	3.674	6.496	7.532	11.203	18.240	22.273

TABLE IV

COR OF PREDICTION MODELS FOR SHORT-TERM PREDICTION

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	0.965	0.920	0.886	0.811	0.708	0.624
VAR	0.967	0.921	0.890	0.824	0.722	0.656
MLP	0.970	0.923	0.900	0.826	0.743	0.663
RNN	0.972	0.927	0.901	0.829	0.751	0.668
LSTM	0.973	0.929	0.907	0.835	0.758	0.689
PDCNN	0.973	0.936	0.918	0.845	0.765	0.706
PSTN	0.980	0.947	0.920	0.859	0.782	0.729

TABLE V

MIE OF PREDICTION MODELS FOR SHORT-TERM PREDICTION(M/S)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	20.197	27.230	34.354	48.942	58.577	64.084
VAR	20.003	25.262	30.065	40.653	46.870	48.074
MLP	20.190	26.194	30.720	38.282	46.421	49.692
RNN	19.910	26.246	30.576	37.725	43.717	48.250
LSTM	18.604	25.575	29.544	36.879	42.336	46.100
PDCNN	18.241	25.493	28.466	36.458	41.381	43.636
PSTN	18.035	25.323	27.887	35.692	40.030	41.745

The results demonstrate that the proposed PSTN holds the dominant position over other models regarding these three error indices, while PR produces the worst prediction results. PR performs comparably to others when with ultra-short prediction horizon (e.g. 10-min), while it yields fairly poor results with relatively longer prediction horizons. For example, in the 10-min ahead prediction task, the ε_R of PR are 2.0% lower than MLP, while this value increases to 34.8% when performing 3-hour ahead prediction. VAR outperforms PR but is still inferior to those AI models in terms of the error indices. MLP shows inferior performance to RNN and LSTM, which indicates that it is more efficient to capture the temporal dependencies in a temporally dynamic way rather than a static

spatial fashion [61]. Compared to RNN, LSTM shows more competitive performance in terms of all error indices. This mainly benefits from the unique working mechanism of LSTM, because using the gates and the memory cell to update information will prevent the model from gradient vanishing. The two deep learning models, PDCNN and PSTN, considerably outperform other models in terms of these error indices. Specifically, PDCNN improves the average ε_R , ε_M and ε_{MIE} by 3.2%, 5.2% and 2.7% respectively compared to LSTM, 11.5%, 12.4% and 6.2% respectively compared to RNN, and 22.8%, 22.0% and 8.4% respectively compared to MLP. PSTN improves the average ε_R , ε_M and ε_{MIE} by 8.7%, 10.7% and 5.2% respectively compared to LSTM. Such improvements reach 16.5%, 17.5% and 8.6% compared to RNN, and 27.1%, 26.6% and 10.8% compared to MLP. In terms of COR, PSTN also shows the best performance over all prediction horizons. It implies that the deep architecture integrating temporal and spatial models is useful in learning the patterns of the spatio-temporal sequence. Moreover, with the extension of the prediction horizon, the advantage that PSTN has becomes more significant. Specifically, compared to LSTM, the improvements PSTN shows in ε_R , ε_M and ε_{MIE} are 2.2%, 7.0% and 3.1% respectively when predicting with the 10-min horizon, those reach to 8.8%, 12.2% and 9.4% respectively in the 3-hour ahead prediction task. The reasons for this may lie in two aspects: 1) The temporal dependencies of wind speed sequence become obviously lower when the prediction horizon is relatively larger since the wind speed is highly variable. Therefore, the spatial correlation is going to play a more critical role when the prediction horizon increases. 2) The baselines do not handle the spatial correlation in a targeted way, which weakens the spatial information embedded in the SWSMs to some extent. Moreover, PDCNN shows inferior performance to PSTN, which indicates that the LSTM can capture the temporal dependencies among the extracted spatial features more effectively than MLP. Therefore, the PSTN has better capability of utilizing the temporal and spatial correlations simultaneously, which can predict the wind speed based on richer knowledge.

To intuitively demonstrate the performance of PSTN, the instantaneous absolute error (refers to Eqn. (17)) of the seven models over 50 contiguous samples in a 60-min ahead prediction task is displayed in Fig10. The curve corresponding to PSTN is almost lower than other curves over these samples, which means the SWSMs predicted by PSTN are close to the ground truth. Based on these results, it is safe to say that the performance of the proposed model is much better than others.

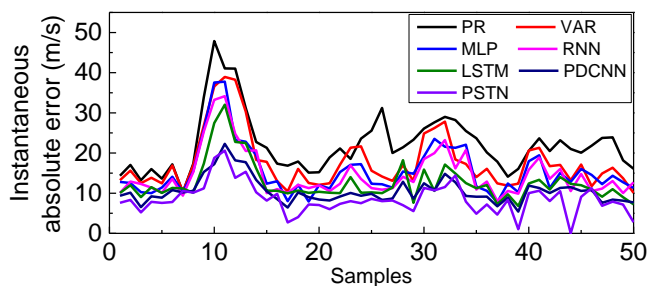


Fig.10. Instantaneous absolute error in 60-min ahead predictions.

To further inspect the learning process of the models, the

loss function J (refers to Eqn. (12)) is recorded. Fig. 11 depicts the values of J for MLP, RNN, LSTM, PDCNN, and PSTN during the learning process for the 60-min ahead prediction. The figure demonstrates that PSTN converges much faster than others and can finally converge to a lower level. Therefore, it is suitable to conclude that learning temporal and spatial correlations jointly benefits the model's training, and PSTN is the best model to predict highly varying wind speed in both prediction accuracy and generalization property.

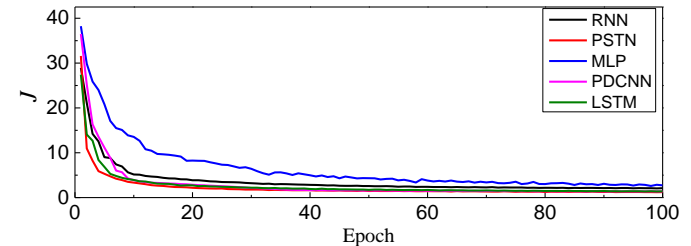


Fig. 11. The monitoring curve of J during the learning process for the 60-min ahead prediction.

F. Comparison with NWP

To further investigate the effectiveness of PSTN, the numerical weather prediction (NWP) is introduced as a baseline. The wind forecast dataset in WIND was derived from the Weather Research and Forecasting (WRF) numerical prediction model version 3.4.1 [62] and provided high-resolution NWP output with the horizons of 1h, 4h, 6h, and 24h. More details of the WRF setup and output were provided in [63]. Here, the performance of PSTN in the prediction tasks with horizons of 1h, 4h, 6h, and 24h are compared with that of NWP given by WIND. The experiments results are shown in Table VI. Moreover, National Oceanic and Atmospheric Administration (NOAA) and European Centre for Medium-Range Weather Forecasts (ECMWF) can also provide NWP data free of charge.

TABLE VI
PERFORMANCE OF PSTN AND NWP

Indices	Models	Prediction Horizon (h)			
		1	4	6	24
RMSE-A (m/s)	PSTN	1.139	2.311	2.909	4.932
	NWP	1.242	2.347	2.754	4.618
MAPE-A (%)	PSTN	11.203	26.406	33.021	41.422
	NWP	12.038	27.143	32.271	38.705
COR	PSTN	0.859	0.783	0.701	0.402
	NWP	0.844	0.760	0.724	0.461
MIE (m/s)	PSTN	35.692	47.253	53.250	72.003
	NWP	37.274	48.293	51.252	67.304

As shown in the table, when the prediction horizon is no more than 6 hours, PSTN shows competitive performance to NWP. Concretely, in the tasks of 1-h and 4-h ahead prediction, PSTN slightly outperforms NWP in terms of all the four indices. There are two main reasons: 1) The wind is strongly affected by local topography and roughness. NWP generally does not model the local topography and roughness very meticulously in short-term predictions due to the heavy computational burden, which limits its prediction accuracy. However, for PSTN, the local effect is taken into account by using measurement data at the sites. That is, the influences of the physical phenomenon, including wake effect, geostrophic wind, and ground roughness, etc., are learned from the massive historical data, which helps PSTN take various

influence factors into consideration comprehensively. 2) The spatial resolution of NWP is often too coarse. Hence, the downscaling techniques, such as the computational fluid dynamics (CFD) based microscale model, are required to downscale the prediction data. In this case, the systematic error and calculation error are likely introduced into the results. Therefore, in the predictions with horizons less than 6 hours, NWPs are less employed than the AI approaches [10].

NWP overtakes PSTN when the prediction horizon is 6 hours, however, the performance gap is fairly small. When the prediction horizon reaches 24 hours, NWP takes a more significant advantage over PSTN in terms of all those indices. The reason mainly lies in three aspects: 1) The temporal dependency among the wind speed time series is reducing as the prediction horizon expands. PSTN make predictions only based on the historical data of wind speed, hence, has less robustness than NWP, especially when the wind speed fluctuates dramatically [32]. 2) NWP employs prior information about the Earth-system and more data from several sources, while PSTN conducts predictions only based on the observations of wind speed. Other variables involved in NWP, such as temperature, pressure, etc., would provide valuable knowledge for prediction, especially in the predictions with long prediction horizons. 3) Compared to PSTN, NWP employs data from a larger geographic range. Specifically, in the case study, the data training PSTN are all from the 100 sites shown in Fig.9, while NWP model is constructed based on the data from a region covering but not limited to the 100 sites. For instance, the parameters describing the boundary conditions in the physical model are from the surrounding areas. Therefore, the advantage of the NWP can be enhanced by the data from a larger geographical range more and more significantly as the prediction horizon increase. Therefore, NWP beats PSTN in the prediction with the horizon over 6 hours.

To summarize, PSTN shows competitive performance to NWP with a horizon no more than 6h. When the prediction horizon reaches 24 hours, NWP holds a dominant position comparing to PSTN. PSTN still has bright application prospects since it is much simpler in implementation than NWP and its performance can be boosted by utilizing more data resources.

G. Time complexity

In terms of time complexity of the learning algorithms, we discuss two main aspects, i.e., training time complexity and testing time complexity. The average time for training and testing the five AI models (i.e., MLP, RNN, LSTM, PDCNN, and PSTN) are shown in Table VII.

TABLE VII

TIME COMPLEXITY OF PREDICTION MODELS (s)					
Models	MLP	RNN	LSTM	PDCNN	PSTN
Training	1018.724	1723.889	2173.804	1544.791	2322.684
Testing	1.578	1.732	3.834	2.758	3.191

In the prediction tasks, PSTN requires more computational time in the training process than others. This is mainly because there is an LSTM at the top of PSTN, which is usually computationally expensive in training due to the multiplicative gates and the recurrent connections [64]. For this reason, the baseline LSTM spends 2,173.804s for training, second only to

PSTN. Compared to LSTM, RNN needs less training time since there is no multiplicative gate. Without the recurrent connections, MLP and PDCNN require less training time than the others. Though requiring more time for training, PSTN still satisfies the practical applications of wind speed prediction since its training time is in an acceptable range.

In terms of testing, MLP is the fastest model due to the feed-forward algorithm for producing the output which has negligible computation burden [32]. Compared to MLP, it takes more time for PDCNN to output the results because of the convolutional calculations. The average testing time of PSTN in the predictions is 3.191s, which means it takes only 0.204ms for each predicted sample. Compared to PSTN, the baseline LSTM needs more time to produce the predicted results. This is because the main computational overhead of PSTN is from the calculation of the LSTM instead of the CNN. In PSTN, the inputs of the LSTM are the spatial features extracted by CNN, whose dimensions are much smaller than that of the input of the baseline LSTM. Therefore, the CNN reduces the data dimensionality, which enhances the computational efficiency of PSTN.

H. Sensitive analysis of the capacities of the datasets

The capacities of the training, validation, and testing sets have an impact on the performance of deep learning models. As illustrated in Fig.12, suppose there are S frames in the total dataset, l and r are two parameters related to the dataset split, where $0 < l < 1$ and $0 < r < 1$. Concretely, the first $l \times S$ frames are used for training, the following $r(1-l) \times S$ frames are used for validation, and the last $(1-r)(1-l) \times S$ frames are used for testing. Naturally, a larger l results in more training frames, and with a certain l , a smaller r results in more testing frames.

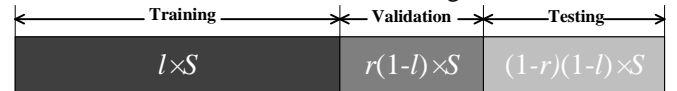


Fig.12 The capacities of the training, validation, and testing sets.

To check how the proposed PSTN performs under different capacities of the datasets, the experiments with different l and r are conducted. Specifically, l is selected from {40%, 50%, 60%, 70%, 80%}, and r is selected from {15%, 20%, 25%, 30%, 35%}. The average performance, in terms of average RMSE-A, over 10-min to 3-h prediction horizons are shown in Fig. 13.

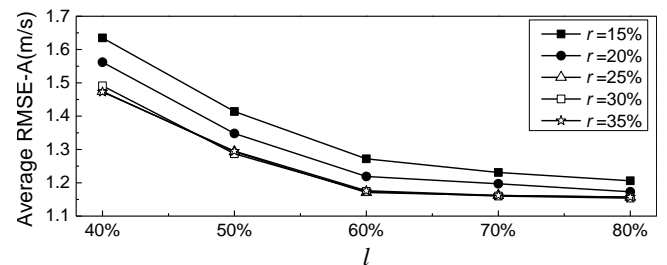


Fig. 13. The average performance of PSTN.

The results demonstrate that l and r both influence the performance of PSTN. Overall, with a certain r , the average RMSE-A becomes smaller and smaller with the increase of l . This mainly because with more training data, the parameter estimates of the model have smaller variance. With a certain l , the average RMSE-A are roughly decreased as r increases.

This mainly because, with more validation samples, the model selection can be more precise, which lays a foundation for a better performance. However, when $r = 30\%$ and $r = 35\%$, the average RMSE-A is fairly close to that with $r = 25\%$. Moreover, when $l \geq 60\%$, the performance is not improved slightly with the increase of r . The reason is that the performance is approaching the best and has less potential for improvement. The analysis of different evaluation indices also come to similar conclusions. It is worth pointing out that, the time complexity of PSTN has an approximately linear correlation with the capacity of the dataset. That is, more training samples results in a longer training time, and more testing samples leads to a longer testing time. According to the results, $l = 60\%$ and $r = 25\%$ is a reasonable choice in the case study.

I. Discussions

Though PSTN has produced promising prediction performance, there are still two main open problems that should be considered in order to apply it for wind speed prediction: 1) model selection is a common dilemma in deep learning, whose purpose is to determine the optimal hyper-parameters. It is always impractical to explore the hyper-parameters through exhaustive grid search because of the significant computation burden and considerable time consuming. In this paper, involving human experience in the model selection can significantly reduce the computational resource and time. However, the model's performance may be constrained since the hyper-parameters determined in this way are likely not optimal. It is necessary to develop more practical methods for model selection that can make a trade-off between the computational cost and a model's performance. 2) Governed by many more parameters, the deep architectures often need more time for training. Giving the larger model scale and more training samples, the training time may be much more considerable. Therefore, there is a pressing need to develop new techniques to reduce the computation time, such as GPU computation, multi-core systems, parallel and distributed systems, etc.

VII. CONCLUSION AND FUTURE WORK

In this paper, a deep architecture PSTN is presented for wind speed prediction, which integrates spatial features extraction and temporal dependencies capture into a unified framework. At the bottom of the model, the CNN automatically extracts the spatial features from the SWSMs. On the top of the model, an LSTM is employed for capturing the temporal dependencies among the feature frame outputted by the CNN. The model is trained with a loss function in an end-to-end manner. Consequently, the model is able to achieve wind speed prediction by leveraging both temporal and spatial correlations. The superiority of the proposed model is demonstrated by the short-term predictions on real-world data.

It is worth pointing out that the proposed PSTN is a general deep architecture suitable for spatio-temporal data, which can also be applied in other fields, e.g., photovoltaic power prediction, wave power generation prediction, etc. For future work, we will investigate how to apply PSTN and its modified

versions to solving other prediction tasks. For example, the deformed CNN will be adopted to enhance the flexibility for dealing with variate topologies. Moreover, there appears to be a widespread belief that using combinations of input variables can result in more suitable representations for learning algorithms and improve performance in knowledge discovery and data mining applications. We also plan to investigate how to account more available meteorological variables in the task of wind speed prediction, such as temperature, air pressure, humidity, etc. Based on the preliminary research, we believe that it is a feasible way to address this problem by using multi-modal learning, e.g., employing the multi-channel CNN to leverage multiple variables and then utilizing the recurrent neural network to capturing the temporal dependencies. It will be challenging to achieve the input variable selection and train the more complex deep model effectively.

VIII. APPENDIX

An additional study case for short-term (10-min to 3-h ahead) predictions is provided in the followings. The settings and the model selection rules are the same as those in Section VI.

A 10×10 wind turbine array in California is selected, as shown in Fig. 14. The wind speed data was collected in 2012. There are 31,622, 5,270, and 15,812 in the training set, validation set, and testing set respectively. The maximum and minimum wind speed is 33.646m/s and 0.015m/s respectively. The experiment results are shown in Table VIII, to XI.

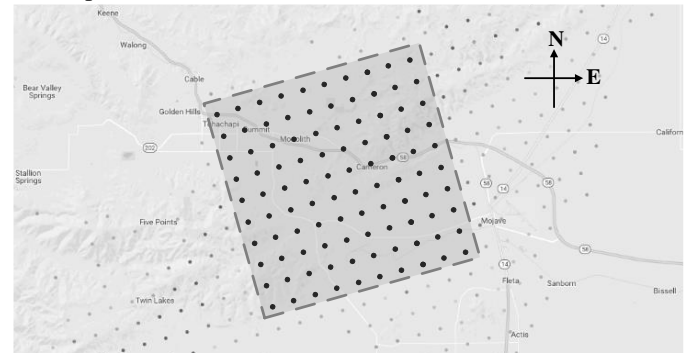


Fig. 14. A 10×10 wind turbine array in the state of California.

TABLE VIII
RMSE-A OF PREDICTION MODELS FOR SHORT-TERM PREDICTION(M/S)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	0.692	0.986	1.247	2.461	3.651	3.943
VAR	0.677	0.961	1.101	1.648	2.715	3.314
MLP	0.605	0.892	0.986	1.417	2.510	2.813
RNN	0.582	0.863	0.947	1.362	2.203	2.549
LSTM	0.576	0.851	0.928	1.336	2.021	2.297
PDCNN	0.550	0.836	0.915	1.120	1.932	2.113
PSTN	0.547	0.802	0.895	1.114	1.833	1.961

TABLE IX
MAPE-A OF PREDICTION MODELS FOR SHORT-TERM PREDICTION (%)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	8.013	10.252	11.974	18.250	30.114	37.832
VAR	7.612	9.575	10.304	15.756	25.003	32.475
MLP	6.844	8.085	9.549	14.125	24.058	29.760
RNN	6.033	7.461	9.035	13.037	21.603	24.251
LSTM	5.611	7.231	8.845	12.542	20.846	23.779
PDCNN	5.049	7.114	8.307	12.153	19.304	23.362
PSTN	4.385	6.023	7.510	11.127	18.038	22.735

TABLE X
COR OF PREDICTION MODELS FOR SHORT-TERM PREDICTION

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	0.943	0.901	0.870	0.770	0.675	0.583
VAR	0.944	0.906	0.873	0.781	0.688	0.610
MLP	0.948	0.913	0.881	0.793	0.694	0.629
RNN	0.950	0.921	0.894	0.809	0.714	0.644
LSTM	0.956	0.927	0.901	0.816	0.734	0.657
PDCNN	0.965	0.933	0.913	0.829	0.750	0.672
PSTN	0.969	0.938	0.919	0.840	0.767	0.690

TABLE XI
MIE OF PREDICTION MODELS FOR SHORT-TERM PREDICTION(M/S)

Model	Prediction Horizon (min)					
	10	20	30	60	120	180
PR	32.113	37.364	41.495	49.019	59.520	67.385
VAR	23.394	26.731	30.253	39.042	47.428	49.283
MLP	21.631	25.304	28.473	38.930	47.115	50.362
RNN	20.052	24.117	26.835	35.073	44.374	48.204
LSTM	19.324	24.038	25.104	34.250	42.645	47.375
PDCNN	18.392	23.184	24.329	32.304	41.650	44.955
PSTN	17.846	23.046	24.107	30.246	40.195	42.374

IX. REFERENCES

[1] A. Kusiak, H. Zheng and Z. Song, "Short-term prediction of wind farm power: a data mining approach," *IEEE Trans. Energy Convers.*, vol. 24, no. 1, pp. 125-136, Jan. 2009.

[2] Y. Ren, P. N. Suganthan and N. Srikanth, "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods," *IEEE Trans. Sustain. Energy*, vol. 6, no. 1, pp. 236-244, Jan. 2015.

[3] M. Khalid and A. V. Savkin, "A method for short-term wind power prediction with multiple observation points," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 579-586, May 2012.

[4] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renew. Sustain. Energy Rev.*, vol. 31, no. 2, pp. 762-777, Mar. 2014.

[5] S. S. Soman, H. Zareipour, O. Malik and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *Proc. North American Power Symposium*, Arlington, TX, USA, Sept. 2010, pp. 1-8.

[6] L. Landberg, "Short-term prediction of the power production from wind farms," *J Wind Eng. Ind. Aerod.*, vol. 80, no. 2, pp. 207-220, Mar. 1999.

[7] M. Lange and F. Ulrich, "Physical approach to short-term wind power prediction," *Springer Sci. Bus. Media*, 2006.

[8] L. Landberg, "Short-term prediction of local wind conditions," *J Wind Eng. Ind. Aerod.*, vol. 89, no. 4, pp. 235-245, Mar. 2001.

[9] N. Chen, Z. Qian, I. T. Nabney and X. Meng, "Wind power forecasts using gaussian processes and numerical weather prediction," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 656-665, March 2014.

[10] K. Bhaskar and S. N. Singh, "AWNN-assisted wind power forecasting using feed-forward neural network," *IEEE Trans. Sustain. Energy*, vol. 3, no. 2, pp. 306-315, Apr. 2012.

[11] Y. Wu and F. Porté-Agel, "Simulation of turbulent flow inside and above wind farms: Model validation and layout effects," *Boundary-Layer Meteorology*, vol. 146, no. 2, pp. 181-205, Apr. 2013.

[12] J. Tastu, P. Pinson, E. Kotwa, H. Madsen and H. A. Nielsen, "Spatio-temporal analysis and modeling of short-term wind power forecast errors," *Wind Energy*, vol. 14, no.1, pp. 43-60, Jan. 2011.

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

[14] J. Hur and R. Baldick, "Spatial prediction of wind farm outputs using the Augmented Kriging-based Model," in *Proc. IEEE Power Energy Society Gener. Meet.*, San Diego, CA, USA, Jul. 2012, pp. 1-7.

[15] J. A. Carta, C. Bueno and P. Ram íez, "Statistical modelling of directional wind speeds using mixtures of von Mises distributions: case study," *Energy convers. manage.*, vol. 49, no. 5, pp. 897-907, May 2008.

[16] S. H. Karaki, R. B. Chedid, and R. Ramadan. "Probabilistic performance assessment of wind energy conversion systems," *IEEE Trans. Energy Convers.*, vol. 14, no. 2, pp. 217-224., Jun. 1999.

[17] S. H. Karaki, R. B. Chedid and R. Ramadan. "Probabilistic performance assessment of autonomous solar-wind energy conversion systems," *IEEE Trans. Energy Convers.* vol. 14, no. 3, pp. 766-772, Jun. 1999.

[18] J. Tastu, P. Pinson, P. J. Trombe and H. Madsen, "Probabilistic forecasts of wind power generation accounting for geographically dispersed information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 480-489, Jan. 2014.

[19] Y. Zhang and J. Wang, "A distributed approach for wind power probabilistic forecasting considering spatio-temporal correlation without direct access to off-site information," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5714-5726, Sept. 2018.

[20] P. Pinson and G. Kariniotakis, "Conditional Prediction Intervals of Wind Power Generation," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1845-1856, Nov. 2010.

[21] J. K. Møller, H. A. Nielsen and H. Madsen. "Time-adaptive quantile regression." *Comput. Stat. Data Analysis*, vol. 52, no. 3, pp. 1292-1303, Jan. 2008.

[22] R. J. Bessa, V. Miranda, A. Botterud, Z. Zhou and J. Wang, "Time-adaptive quantile-copula for wind power probabilistic forecasting," *Renew. Energy*, vol. 40, no. 1, pp. 29-39, Apr. 2012.

[23] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis and P. S. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation." *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 352-361, Jun. 2004.

[24] D. A. Bechrakis and P. D. Sparis, "Correlation of wind speed between neighboring measuring stations," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 400-406, Jun. 2004.

[25] S. Robert, L. Foresti and M. Kanevski, "Spatial prediction of monthly wind speeds in complex terrain with adaptive general regression neural networks," *Intl. J. of Climatology*, vol. 33, no. 7, pp. 1793-1804, Jul. 2013.

[26] B. Saavedra-Moreno, S. Salcedo-Sanz, L. Carro-Calco, J. Gascon-Moreno, S. Jimenez-Fernandez and L. Prieto. "Very fast training neural-computation techniques for real measure-correlate-predict wind operations in wind farms." *J Wind Eng. Ind. Aerod.*, vol. 116, no. 1, pp. 49-60, May 2013.

[27] T. G. Barbounis and J. B. Theocharis, "Locally recurrent neural networks optimal filtering algorithms: application to wind speed prediction using spatial correlation," in *Proc. IEEE Intl. Joint Conf. Neural Netw.*, Montreal, Que., Canada, Jul. 2005, pp. 2711-2716.

[28] W. Huang, G. Song, H. Hong and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191-2201, Oct. 2014.

[29] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[30] C. Y. Zhang, C. L. P. Chen, M. Gan and L. Chen, "Predictive deep Boltzmann machine for multiperiod wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416-1425, Oct. 2015.

[31] H. Z. Wang, G. B. Wang, G. Q. Li, J. C. Peng and Y. T. Liu, "Deep belief network based deterministic and probabilistic wind speed forecasting approach," *Appl. Energy*, vol. 182, no. 8, pp. 80-93, Nov. 2016.

[32] M. Khodayar, O. Kaynak and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770-2779, Dec. 2017.

[33] Q. Hu, R. Zhang and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renew. Energy*, vol. 85, pp. 83-95, Jan. 2016.

[34] Z. Wang, C. Liu, D. Cheng, L. Wang, X. Yang and K. T. Cheng, "Automated detection of clinically significant prostate cancer in mp-MRI images based on an end-to-end deep neural network," *IEEE Trans. Med. Imag.*, vol. 37, no. 5, pp. 1127-1139, May 2018.

[35] Y. Xue, N. Chen, S. Wang, F. Wen, Z. Lin and Z. Wang, "Review on wind speed prediction based on spatial correlation," *Auto. Electrical Power Syst.*, vol. 41, no. 10, pp. 161-169, May., 2017. (in Chinese)

[36] T. Trappenberg, J. Ouyang and A. Back, "Input variable selection: mutual information and linear mixing measures," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 37-46, Jan. 2006.

[37] S. M. Chan, D. C. Powell, M. Yoshimura and D. H. Curtice, "Operations requirements of utilities with wind power generation," *IEEE Trans. Power Apparatus Syst.*, vol. PAS-102, no. 9, pp. 2850-2860, Sept. 1983.

[38] R. Caruana, *Multitask Learning*, New York, NY, USA: Springer-Verlag, 1998.

[39] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221-231, Jan. 2013.

[40] W. H. Yun, D. Lee, C. Park, J. Kim and J. Kim, "Automatic recognition of children engagement from facial video using convolutional neural networks," *IEEE Trans. Affect. Comput.* early access, 2018.

[41] J. Zuo, G. Xu, K. Fu, X. Sun and H. Sun, "Aircraft type recognition based on segmentation with deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 282-286, Feb. 2018.

[42] W. Xiong, et al., "Toward human parity in conversational speech recognition," *IEEE/ACM Trans. Audio Speech Language Process.*, vol. 25, no. 12, pp. 2410-2423, Dec. 2017.

[43] Z. Miao, K. Fu, H. Sun, X. Sun and M. Yan, "Automatic water-body segmentation from high-resolution satellite images via deep networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 4, pp. 602-606, Apr. 2018.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[45] Q. Wu, C. Shen, P. Wang, A. Dick and A. V. D. Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1367-1381, Jun. 1 2018.

[46] M. Sundermeyer, H. Ney and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio Speech Language Process.*, vol. 23, no. 3, pp. 517-529, Mar. 2015.

[47] W. Du, Y. Wang and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Trans. Imag. Process.*, vol. 27, no. 3, pp. 1347-1360, Mar. 2018.

[48] X. Shi, Z. Chen, H. Wang and D. Y. Yeung, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Informat. Process. Syst.*, Istanbul, Turkey, Nov. 2015, pp. 802-810.

[49] G. Hinton, L. Deng, D. Yu, et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82-97, Nov. 2012.

[50] I. Sutskever, O. Vinyals, Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Informat. Process. Syst.*, Kuching, Malaysia, Nov. 2014, pp. 3104-3112.

[51] K. Xu, J. Ba, R. Kiros, et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proc. Intl. Conf. Mach. Learning*, Lille, France, Jul. 2015, pp. 2048-2057.

[52] R. Li, S. Wang, F. Zhu and J. Huang, "Adaptive graph convolutional neural networks," *arXiv, available:1801.03226*, 2018.

[53] H. C. Shin, M. R. Orton, D. J. Collins, S. J. Doran and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930-1943, Aug. 2013.

[54] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, et al., "Deep hierarchies in the primate visual cortex: what can we learn for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1847-1871, Aug. 2013.

[55] Y. Dauphin, D. V. Harm, and B. Yoshua. "Equilibrated adaptive learning rates for non-convex optimization." in *Proc. Adv. Neural Informat. Process. Syst.*, Istanbul, Turkey, Nov. 2015, pp. 1504-1512.

[56] E. Zivot and Jiahui Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-PLUS®*, Springer, New York, NY USA, pp. 385-429, 2006.

[57] K. L. Du and M. N. Swamy. *Neural networks and statistical learning*. Springer Science & Business Media, 2013.

[58] A. Graves, A. R. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Intl. Conf. Acoustics Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645-6649.

[59] Q. Zhu, J. Chen, L. Zhu, X. Duan and Y. Liu, "Wind speed prediction with spatio-temporal correlation: a deep learning approach," *Energies*, vol. 11, no. 4, pp. 1-18, Apr. 2018.

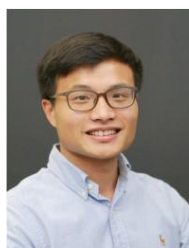
[60] M. Abadi, P. Barham, J. Chen, et al., "TensorFlow: a system for large-scale machine learning," [online], available: ariXiv:1605.08695v2, 2016.

[61] H. Strobel, S. Gehrmann, H. Pfister and A. M. Rush, "LSTMVis: a tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 667-676, Jan. 2018.

[62] W. C. Skamarock and J. B. Klemp, "A time-split nonhydrostatic atmospheric model for weather research and forecasting applications," *J. Comput. Phys.* vol. 227, no. 7, pp. 3465-3485, Mar. 2008.

[63] C. Draxl, B. M. Hodge, A. Clifton and J. McCaa., "Overview and meteorological validation of the wind integration national dataset toolkit," Technical Report, NREL/TP-5000-61740, Golden, CO: National Renewable Energy Laboratory, 2015.

[64] H. Sak, A. Senior and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," in *Proc. An. Conf. Intl. Speech Commun. Assoc.*, Singapore, Sept. 2014, pp. 338-342.



Qiaomu Zhu (S'2016) received the B.S. degree in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2014. He is currently pursuing Ph.D. degree at HUST. He is also a visiting student researcher at University of Tennessee (UTK), Knoxville, TN, USA.

His research interests include deep learning, data analytics in smart grid and transient stability assessment.



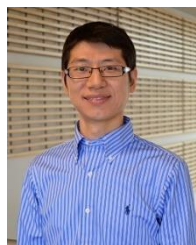
Jinfu Chen received both a B.S. and Ph.D. in Electrical Engineering in Huazhong University of Science and Technology (HUST), Wuhan, China in 1996 and 2002 respectively. Currently, he is an Associate Professor in the School of Electrical and Electronic Engineering at HUST.

His research interests include electric power system analysis and operation, microgrid, and distributed generation.



Dongyuan Shi (M'06) received both a B.S. and Ph.D. in Electrical Engineering in Huazhong University of Science and Technology (HUST), Wuhan, China in 1996 and 2002 respectively. Currently, he is a Professor with the School of Electrical and Electronic Engineering at HUST.

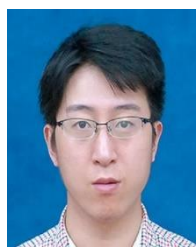
His research interests include electric power system analysis and operation, integrated energy system, and cyber security.



Lin Zhu (M'11) received the B.S. and Ph.D. degrees in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2005 and 2011 respectively. He is currently a Research Assistant Professor with the Department of Electrical Engineering and Computer Science, University of Tennessee (UTK), Knoxville, TN, USA.

His current research interests include protective relaying, substation automation, and power system

dynamics and control.



Xiang Bai (SM'16) received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications and the Vice Director of the National Center of Anti-Counterfeiting Technology with HUST.

His current research interests include computer vision, deep learning, object recognition, shape analysis, and scene text recognition.



Xianzhong Duan (M'03) received the B.S. and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1987 and 1992, respectively, all in electrical engineering. He is currently a Professor with the School of Electrical and Electronic Engineering, HUST.

His current research interests include power system analysis and operation, voltage stability, FACTS, and power system automation.



Yilu Liu (S'88–M'89–SM'99–F'04) received her M.S. and Ph.D. degrees from the Ohio State University, Columbus, in 1986 and 1989. She received the B.S. degree from Xian Jiaotong University, China. Dr. Liu is currently the Governor's Chair at the University of Tennessee, Knoxville and Oak Ridge National Laboratory (ORNL). Dr. Liu is elected as the member of National Academy of Engineering in 2016. She is also the Deputy Director of the DOE/NSFcofunded engineering research center CURENT. Prior to joining UTK/ORNL, she was a Professor at Virginia Tech. She led the effort to create the North American power grid Frequency Monitoring Network (FNET) at Virginia Tech, which is now operated at UTK and ORNL as GridEye.

Her current research interests include power system wide-area monitoring and control, large interconnection-level dynamic simulations, electromagnetic transient analysis, and power transformer modeling and diagnosis.