Engineering Applications of Artificial Intelligence 130 (2024) 107613

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Local and Long-range Convolutional LSTM Network: A novel multi-step wind speed prediction approach for modeling local and long-range spatial correlations based on ConvLSTM

Mei Yu [a,b,c], Boan Tao [a,b,c], Xuewei Li [a,b,c], Zhiqiang Liu [a,b,c], Wei Xiong [d,*]

[a] *College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China*
[b] *Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, 300350, China*
[c] *Tianjin Key Laboratory of Advanced Networking, Tianjin, 300350, China*
[d] *School of Civil Engineering, Tianjin Chengjian University, Tianjin, 300384, China*

ARTICLE INFO

Keywords:
Deep learning
ConvLSTM
Convolution
Attention
Spatiotemporal correlations
Wind speed prediction

ABSTRACT

Accurate wind speed prediction is crucial for enhancing the stability and economic efficiency of power system operation, particularly in wind power grid integration. However, existing methods face challenges as they fail to explicitly model local and long-range spatial correlations simultaneously, thereby limiting the performance of wind speed prediction to a certain extent. To overcome these challenges, this study develops a novel method, namely, LLConvLSTM, from the perspective of modeling local and long-range spatial correlations in wind speed, which leverages Deformable Convolution V2 and Coordinate Attention for multi-step spatiotemporal wind speed prediction. A ConvLSTM encoder–decoder architecture is designed for end-to-end spatiotemporal wind speed prediction. The Residual Deformable Convolution Module (RDCM) increases additional offsets and modulation scales in the spatial sampling locations, enhancing the capability to capture local spatial correlations. Dense Coordinate Attention Module (DCAM) embeds spatial positional information into the channel attention. DCAM improves the representability of long-range spatial correlations. Experimental results based on wind speed data from 253 virtual wind turbines demonstrate that the proposed approach significantly outperforms existing methods throughout the entire year and months. Moreover, the proposed method achieves Mean Squared Error (MSE) of 0.1199, 0.3446 and 0.5798 for multi-step wind speed prediction, representing reductions of 22.47% to 40.91% compared with existing methods. These findings highlight the significance of modeling local and long-range spatial correlations in enhancing the accuracy and stability of wind speed prediction. Future research will design a universal method capable of handling turbine data from any location and emphasize long-term forecasting in wind speed prediction.

1. Introduction

At present, traditional fossil fuels are still important resources that are commonly used (Prema et al., 2021). The excessive use of fossil fuels has led to a sudden increase in carbon emissions, which has exacerbated the greenhouse effect, causing severe pollution to the environment (Sibtain et al., 2022). Unlike fossil fuels, wind energy, as a green, low-carbon, pollution-free renewable energy (Yang et al., 2022), has attracted widespread attention. Wind energy power generation is one of the ways to use wind energy, which can effectively alleviate environmental pollution. However, the wind speed has intense intermittent, volatility and randomness (Ewees et al., 2022), which dramatically affects the stable operation of the power system in the wind power grid. Accurate wind speed prediction helps dispatch the power system reasonably, thereby achieving a stable supply and demand balance, reducing the economic risks of wind power generation, and improving the availability of wind energy (Zhang et al., 2022). Therefore, improving the accuracy of wind speed prediction is of great significance to promoting the large-scale application of wind power generation.

Wind speed data observed at a turbine are not only relevant to its own historical wind speed data (i.e., temporal correlations), but also to wind speed data from other turbines within a specific range (i.e., spatial correlations). By simulating the spatiotemporal variability of wind energy, the model can better characterize the spatiotemporal

* Corresponding author.
E-mail addresses: yumei@tju.edu.cn (M. Yu), taoboan@tju.edu.cn (B. Tao), lixuewei@tju.edu.cn (X. Li), tjubeisong@tju.edu.cn (Z. Liu), xw7745@sohu.com (W. Xiong).

https://doi.org/10.1016/j.engappai.2023.107613
Received 4 March 2023; Received in revised form 24 October 2023; Accepted 23 November 2023
Available online 26 December 2023
0952-1976/

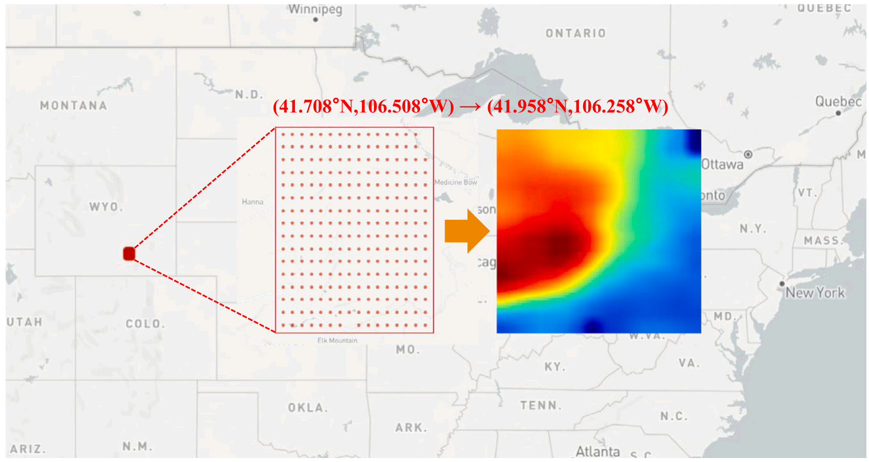

Fig. 1. The wind farm can be designed as a multi-channel gridded image with spatiotemporal features. Among them, each pixel indicates a virtual wind turbine in a space. To facilitate the presentation of feature variations, the multi-channel gridded image is converted to color by applyColorMap in the OpenCV library.

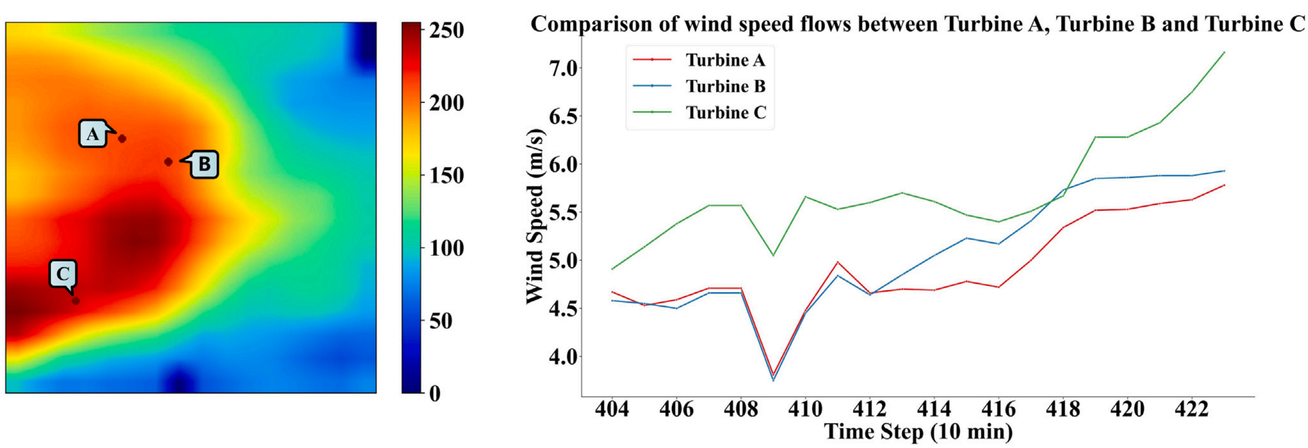

Fig. 2. Wind speed flows of Turbine A and Turbine B show similar wind speed flow dynamics. Likewise, Wind speed flows of Turbine A and Turbine C also show similar wind speed flow dynamics.

correlations of wind speed between different locations and then effectively improve the accuracy of wind speed prediction (Zhu et al., 2018). Fig. 1 shows that the wind farm can be designed as a multi-channel gridded image with spatiotemporal characteristics. Among them, each pixel indicates an virtual wind turbine in a space. Fig. 2 (left) shows a visual image with gridded wind speed data with size 16 × 16. Three turbines are selected and the wind speed variation relationships for these turbines are observed as shown in Fig. 2 (right). Turbine A and Turbine B are outside the normal adjacent locations, but they show similar wind speed flow dynamics, reflecting local spatial correlations. Similarly, the wind speed for Turbine A and Turbine C at distant locations shows similar flow dynamics, which denotes the existence of long-range spatial correlations.

However, convolution with a fixed geometric structure cannot capture the varying relationships between the two turbines, which results in the loss of valid information. Additionally, convolution is challenging to extract long-range spatial location information features. Existing wind speed prediction studies, including physical methods, statistical methods, machine learning methods and deep learning methods, employ various techniques to improve the accuracy of wind speed prediction. However, they have not adequately addressed the modeling of local and long-range spatial correlations in spatiotemporal wind speed prediction.

This study proposes feasible strategies to address these challenges from two perspectives: local spatial correlations and long-range spatial correlations, respectively. To extract local spatial correlations, the Deformable Convolution V2 (DCNv2) (Zhu et al., 2019) augments the 2-D spatial sampling locations with additional offsets to adaptively learn the offsets and modulation amplitudes from the spatial wind speed flow. To construct long-range spatial correlations, the study proposes the utilization of Coordinate Attention (CA) (Hou et al., 2021). Specifically, CA enhances the representation of long-range spatial correlations by embedding the spatial location information of all turbines into channel attention.

Based on the above analysis, the study combines the advantages of Deformable Convolution V2 and Coordinate Attention, proposes a novel ConvLSTM multi-step spatiotemporal wind speed prediction approach, namely, LLConvLSTM, which achieves superior experimental results. Wherein, ConvLSTM encoder–decoder architecture is designed to model the spatiotemporal correlations for achieving end-to-end wind speed prediction. ConvLSTM encoder part extracts the features from the wind speed sequence and encodes them into hidden state and cell state. The prediction sequence is then generated through the decoder part. Residual Deformable Convolution Module (RDCM) incorporates Deformable Convolution V2, which accurately characterizes the local spatial correlations. At the same time, Dense Coordinate Attention Module (DCAM) integrates coordinate attention to enhance the ability to sense long-range spatial correlations. This study fully tests the wind speed data of 253 virtual wind turbines. The experimental results show that LLConvLSTM model is significantly better than existing methods (Potter and Negnevitsky, 2006; Harbola and Coors, 2019; Kisvari et al., 2021; Shen et al., 2022; Liao et al., 2021; Kusiak and Zhang, 2010; Lahouar and Slama, 2017; Velo et al., 2014). The prediction results of LLConvLSTM are closer to the actual value. Ablation experiments further demonstrate that modeling local and long-range spatial correlations could improve performance. The main contributions of this work are as follows:

(1) Based on ConvLSTM encoder–decoder architecture, the study proposes a novel ConvLSTM multi-step spatiotemporal wind speed prediction approach to better realize end-to-end prediction.

(2) Design new RDCM and DCAM can effectively construct local spatial correlations and capture long-range spatial correlations to improve the prediction performance further. This is the first paper to model local and long-range spatial correlations in a deep learning-based spatiotemporal wind speed prediction approach.

(3) Extensive experiments are conducted on wind speed data from 253 virtual wind turbines, and the results show that the proposed approach is superior to state-of-the-art spatiotemporal predictive models. Moreover, ablation experiments further verified the validity of designed components in LLConvLSTM.

2. Literature review

2.1. Methods for wind speed prediction

At present, researchers have developed various wind speed forecasting methods, including physical methods, statistical methods, machine learning methods and deep learning methods.

Physical methods use historical wind speed time series and multiple meteorological factors to achieve wind speed forecasting (Jung and Broadwater, 2014). The most widely used is the numerical weather forecasting (NWP) model (Nor et al., 2014). However, NWP is susceptible to non-stationary and extreme factors and cannot provide reliable wind speed prediction (Zhang et al., 2020) in complex areas. Statistical methods usually refer to time series models, including autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA) (Erdem and Shi, 2011), autoregressive integrated moving average model (ARIMA) (Yunus et al., 2015) and its derivative models. However, various uncertainties in wind speed prediction significantly limit the ability to process nonlinear wind speed data for statistical methods.

Machine learning methods are always more promising than physical and statistical methods because of their potential feature extraction capability (Daut et al., 2017). Hao et al. (2021) used improved grey wolf optimization (IGWO) algorithm to optimize parameters of gradient boosting regression tree (GBRT) to achieve wind speed prediction. Yesilbudak et al. (2013) used k-nearest neighbor (k-NN) classification model with various distance metrics to predict wind speed parameters in an n-tuples input. Yu and Vautard (2022) developed a transfer method for calculating 100 m wind speed using random forests (RF). Liu et al. (2013) developed a hybrid model based on wavelet transform, genetic algorithm (GA), particle swarm optimization (PSO), and multilayer perceptron (MLP) to achieve accurate wind speed prediction. Yeganeh-Bakhtiary et al. (2022) developed a supervised machine learning method, specifically the M5'Decision Tree model, to establish a statistical relationship between predictor and predictand. The capabilities of the M5'Decision Tree model were examined to predict future wind speed and identify spatiotemporal trends in wind characteristics. Peláez-Rodríguez et al. (2022) introduced a novel method for predicting extreme wind speed events using hierarchical classification/regression (HCR) techniques, aiming to enhance the prediction accuracy of extreme wind speed events across various machine learning methods.

However, training these machine learning models requires researchers to extract features manually, relying on priority knowledge in related fields. It is unreliable in practical application scenarios to select suitable features only by hand. Considering the complex and non-stationary characteristics of wind speed, the modeling process of machine learning is complicated, and the model's generalization ability is limited. Therefore, machine learning methods have significant limitations in wind speed prediction.

Deep learning has been widely used in wind speed prediction as a novel artificial intelligence technology. Memarzadeh and Keynia (2020) used the crow search algorithm (CSA) to optimize the structure of long short-term memory (LSTM), thereby improving the accuracy and speed of short-term wind speed prediction. Dolatabadi et al. (2020) developed a hybrid model integrating discrete wavelet packet transform (DWPT) and bidirectional long short-term memory (Bi-LSTM) for learning the internal temporal relationship of wind speed time series. Shen et al. (2022) explored a hybrid model based on CNN-LSTM for multi-step wind speed prediction and proved that the hybrid model is superior to a single model in terms of accuracy and stability. Gan et al. (2021) used an interval prediction model based on the temporal convolutional networks (TCN) to improve the accuracy of wind speed prediction. Wu et al. (2022b) introduced a multidimensional spatiotemporal graph neural network (MST-GNN) that incorporates a Wind-Transformer in the temporal perspective for single-point wind speed prediction. Additionally, they utilize a graph neural network with the Wind-Transformer as a node in the spatial perspective to achieve accurate wind speed prediction at the specific location.

Time-series prediction models are extensively employed in various related domains (Adnan et al., 2021; Ikram et al., 2022b; Adnan et al., 2023; Ikram et al., 2022c,a). Khosravi et al. (2023) used convolutional neural networks, recurrent neural networks and long and short-term memory to accurately predict soil erosion susceptibility in a catchment area. Yuan et al. (2018) proposed a hybrid model, LSTM-ALO, which utilizes the Ant-Lion Optimizer (ALO) to calibrate the parameters of the Long Short-Term Memory network for monthly runoff prediction. Adnan et al. (2022) developed a novel hybrid method, ANFIS-GBO, using the gradient-based optimization (GBO) algorithm to adjust the hyperparameters of the adaptive neuro-fuzzy system (ANFIS) for streamflow prediction in a mountainous river basin.

However, these methods do not consider the spatial correlations inherent in wind speed data. Shi et al. (2015) proposed convolutional LSTM networks (ConvLSTM). Traditional physical methods and statistical methods have significant limitations in handling the spatiotemporal correlations of wind speed. In contrast, modern deep learning methods, particularly ConvLSTM, exhibit remarkable advantages in this regard. By using convolution structures in input-to-state and state-to-state transitions, ConvLSTM can capture underlying spatiotemporal features. ConvLSTM (Xiao et al., 2021; Scheepens et al., 2023) have been successfully used to model the spatiotemporal correlations of wind speed prediction and have obtained better prediction performance. On this basis, the study uses a ConvLSTM encoder–decoder structure to realize end-to-end wind speed prediction.

2.2. Convolutions for local spatial correlations

Because of its excellent performance in fitting nonlinear data, convolution operation has attracted massive researchers' attention in extracting abstract features. However, the data often show irregular change trends in scenes with complex spatial correlations (Zeng et al., 2021). General convolution uses a fixed sampling position to extract the local receptive field, which will lose some critical information. In recent years, researchers have proposed some improvement strategies from the perspective of optimizing spatial sampling locations:

Tang et al. (2020) proposed a multiscale spatial and spectral feature model to capture the discriminative features for hyperspectral image (HSI) classification to obtain spatial information of different scales. Convolution operations of different sizes (for example, 3×3, 5×5, and 7×7) introduce additional parameter computation while adding multiscale features. Hu et al. (2018b) combined a 3-D atrous convolutional neural network with ConvLSTM to capture depth vision features in

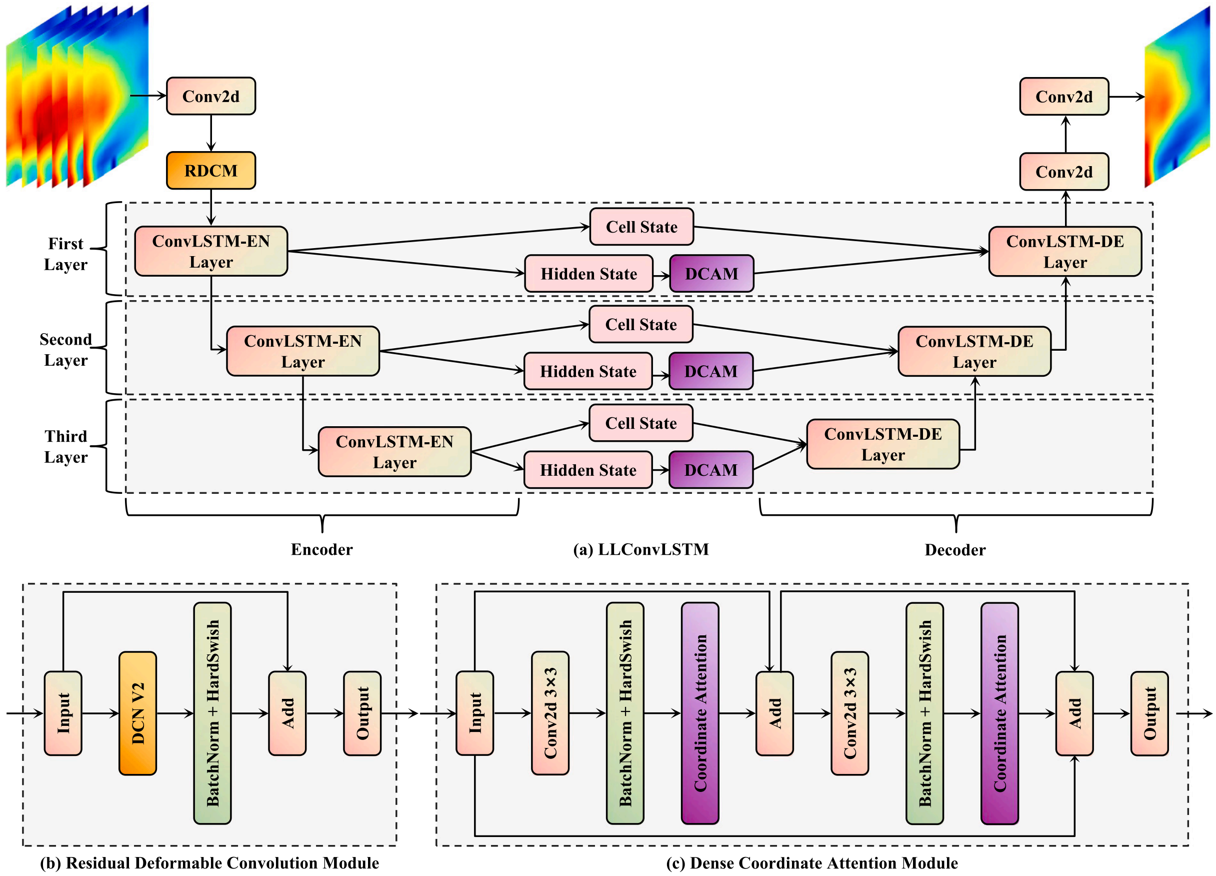

Fig. 3. (a) Illustration of the overall architecture of LLConvLSTM model. (b) Illustration of Residual Deformable Convolution Module. (c) Illustration of Dense Coordinate Attention Module.

video data. However, hollow convolution only samples pixels in strictly symmetric positions. On the contrary, the regions where wind speed changes are usually irregular. In this sense, ensuring that all essential features are collected in symmetric positions is challenging, which may even deteriorate the feature representation of local spatial information.

In order to extract effective features, Dai et al. (2017) proposed Deformable Convolution (DCN). DCN uses additional offsets to augment 2-D spatial sampling locations and adaptively learns offsets through gradient backpropagation. Deformable convolution has achieved significant success in object detection (Wei et al., 2022), image segmentation (Shen et al., 2023), video recovery (Wang et al., 2019) and other domains. However, the sampling locations after deformation may extend well beyond the effective region, affecting the feature by the irrelevant region. Therefore, Zhu et al. (2019) improved the ability to focus on pertinent regions by introducing a modulation mechanism. Unlike standard convolution, which is sensitive to changes in geometric rules (Liu et al., 2020), Deformable Convolution V2 (DCNv2) adaptively learns sampling locations and modulation amplitudes which expand the scope of deformation modeling. By introducing DCNv2, the study designs a residual deformable convolution module (RDCM) better to characterize the local spatial correlations of wind speed data.

2.3. Attention mechanisms for long-range spatial correlations

In recent years, attention mechanisms (Niu et al., 2021) have been used in diverse domains of deep learning. Channel attention refines the feature map by adjusting the relationships between channels. One of the most widely used channel attention is the SE module (Hu et al., 2018a), which adaptively calibrates channel-wise features by explicitly modeling the interdependencies between channels. Nevertheless, the channel attention mechanism ignores the importance of spatial location information. Whereafter, scholars proposed the convolutional block attention module (CBAM) (Woo et al., 2018). CBAM generates attention feature map sequentially along two separate dimensions (channel and space). Zhang and Yang (2021) proposed Shuffle Attention (SA) module. SA utilizes Shuffle Unit to process feature dependencies of multiple sub-features in spatial and channel dimensions parallelly and uses "channel shuffle" to aggregate all sub-features. Although these attention mechanisms consider spatial and channel information, they ignore spatial positional information loss caused by 2-D global pooling. The representation of spatial location information by model structure will directly affect the interpretability of the model's spatial correlations. Therefore, it is difficult for these attention mechanisms to model spatial correlations in wind speed prediction accurately.

For extracting accurate long-range spatial positional information, Hou et al. (2021) proposed a coordinate attention (CA) mechanism. CA embed positional information into channel attention by coordinate information embedding and coordinate attention generation, thereby retaining precise spatial positional information. CA module shows excellent application potential in computer vision, such as ship detection (Wu et al., 2022a) and lung mass segmentation (Chang et al., 2022). Introducing CA mechanism, the study designs a dense coordinate attention module (DCAM) to accurately characterizes the long-range spatial correlations of wind speed data.

3. Method

In this section, the paper introduces the overall architecture of LLConvLSTM model, convolutional long and short-term memory (ConvLSTM), the residual deformable convolution module (RDCM) and dense coordinate attention module (DCAM) in detail.

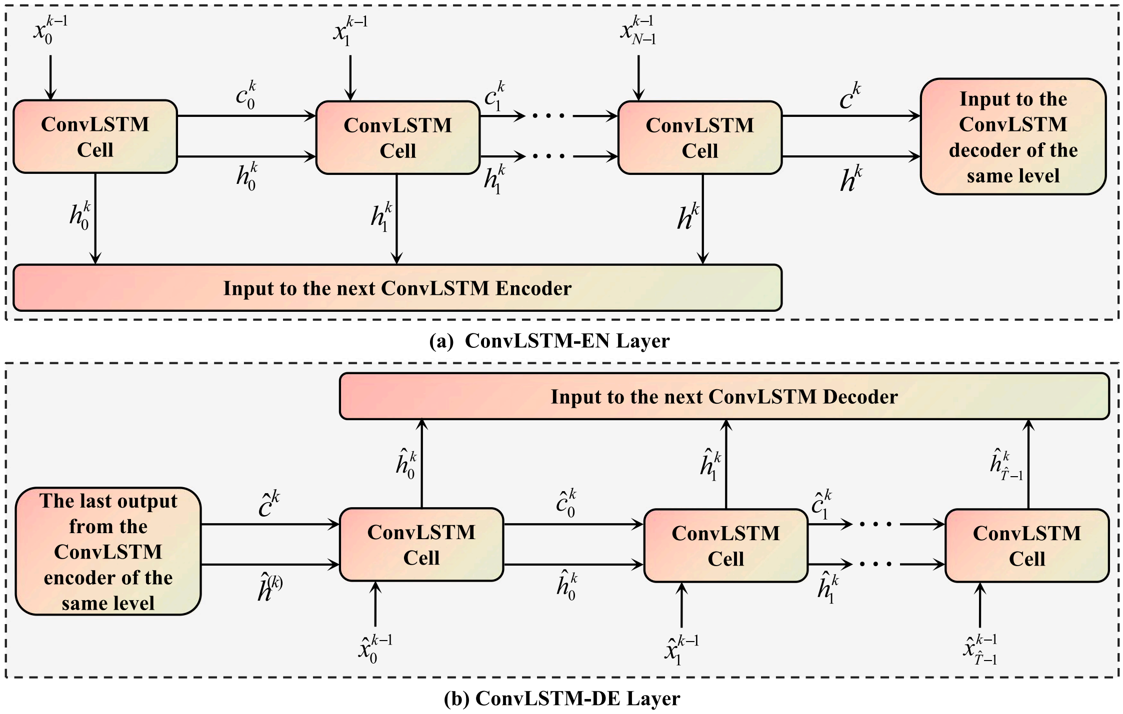

Fig. 4. (a) Illustration of ConvLSTM encoder layer. (b) Illustration of ConvLSTM decoder layer.

3.1. LLConvLSTM Model

The study proposes the LLConvLSTM model to model local and long-range spatial correlations in spatiotemporal wind speed prediction. Fig. 3(a) is the overall architecture of LLConvLSTM model.

(1) Encoder part

Encoder part consists of a 2-D convolution layer, a RDCM, and three ConvLSTM encoder layers. The wind speed feature map is first fed into a 2-D convolution layer to extract features and expand the number of channels. In addition, the encoder part employs RDCM to characterize local spatial correlations by capturing the essential sampling information in the irregular region while maintaining the number of channels in the feature map. With each layer of RDCM inserted, parameters of encoder part increase. Over-adding RDCM does not significantly improve predictive performance. In order to balance computational efficiency and performance, only one layer of RDCM is added after the 2-D convolution layer. Its output is adjusted to a data format acceptable to the ConvLSTM encoder.

The input feature map of ConvLSTM encoder is composed of sequences of given length T. The illustration of ConvLSTM encoder is shown in Fig. 4(a). Through convolution operations, the ConvLSTM cell learns the spatiotemporal relationship of wind speed from input feature map. In the structure of multilayer ConvLSTM encoder layer, the output of $(k-1)$th layer is utilized as the input of kth layer. Meanwhile, to avoid the loss of critical spatial information, the stride is set as 1 in the ConvLSTM cell to maintain the size of feature map. The output of ConvLSTM encoder can be computed as follows:

$$\left(h^k, c^k\right), X^k = \mathcal{F}_{\text{encoder}}\left(X^{k-1}\right) \tag{1}$$

where X^{k-1} and X^k represent the input and output of the kth ConvLSTM encoder layer respectively. X^k consists of the results h_t^k of ConvLSTM, $t \in [0, 1, \dots, T-1]$. For each sample x_t^{k-1} of input data X^{k-1}, $x_t^{k-1} \in \mathbb{R}^{B\times C\times H\times W}$. $\mathcal{F}_{\text{encoder}}(\cdot)$ represents the internal operation of ConvLSTM encoder, $\left(h^k, c^k\right)$ represents the last hidden state and cell state of the kth ConvLSTM encoder layer.

(2) Decoder part

The structure of decoder and encoder parts is approximately similar. The decoder part consists of three DCAM, three ConvLSTM decoders and two 2-D convolution layers, which are used to predict the wind speed sequence of length $\hat{T}$. The illustration of ConvLSTM decoder is shown in Fig. 4(b). The model uses the last hidden state h^k and cell state c^k of ConvLSTM encoder to initialize the hidden state and cell state of ConvLSTM decoder at the same layer, which makes the decoder pay more attention to the details of the input information and reduces the pressure of information carrying. The hidden state $\hat{h}^k$ and cell state $\hat{c}^k$ in the kth ConvLSTM decoder layer can be calculated as follows:

$$\hat{h}^k = \mathcal{F}_{\text{DCAM}}\left(h^k\right) \tag{2}$$

$$\hat{c}^k = c^k \tag{3}$$

where $\mathcal{F}_{\text{DCAM}}(\cdot)$ indicates that the initial hidden state $\hat{h}^k$ of ConvLSTM decoder requires learning the long-range spatial correlations of wind speed through DCAM.

Therefore, the output of the kth ConvLSTM decoder layer can be calculated as follows:

$$\hat{X}^{k-1} = \mathcal{F}_{\text{decoder}}\left(\hat{X}^k, \left(\hat{h}^k, \hat{c}^k\right)\right) \tag{4}$$

where $\hat{X}^k$ and $\hat{X}^{k-1}$ represent the input and output of the kth ConvLSTM decoder layer, respectively. $\mathcal{F}_{\text{decoder}}(\cdot)$ represents the internal operation of ConvLSTM decoder cell. The two 2-D convolution layers receive features from the last ConvLSTM decoder layer to adjust the number of channels and achieve end-to-end prediction.

ConvLSTM encoder–decoder structure utilizes rich information in shallow and deep feature maps so that the model has nonlinear mapping capabilities and can extract the space–time correlations characteristics in historical wind speed data. After adding RDCM and DCAM, LLConvLSTM model integrates the rich local and long-range spatial correlations of the wind speed data. Simultaneously modeling local and long-range spatial correlations highlights the robustness and appropriateness of the novel LLConvLSTM in addressing the challenges of accurate wind speed prediction, providing a promising solution for wind speed forecasting. Additionally, considering that predictions based on artificial intelligence models are heavily influenced by the training data, any variations in the selected training data can introduce significant uncertainties in the model's outputs (Ghiasi et al., 2022). The selection of machine learning techniques can also significantly

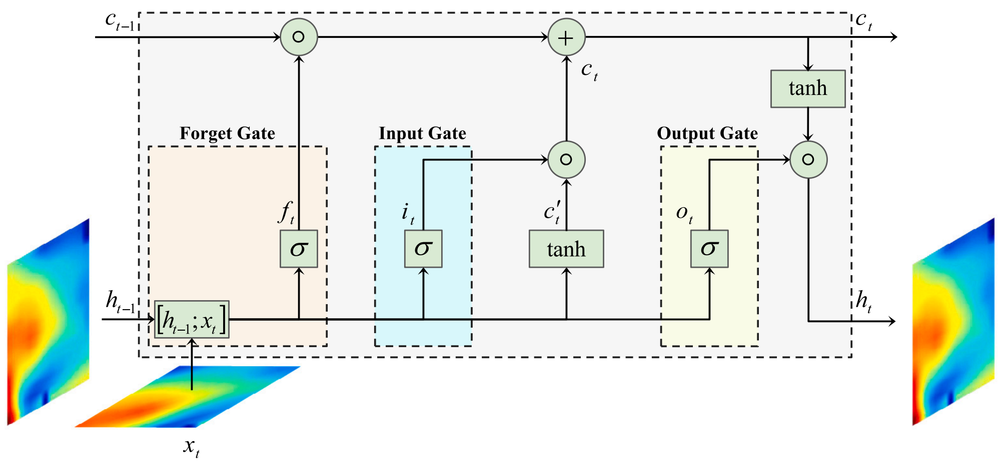

Fig. 5. Illustration of ConvLSTM structure.

impact the quantification capabilities of the model (Donnelly et al., 2022). To make the proposed method applicable to a wider range of data variations, our future work will focus on designing a universal data embedding approach.

3.2. Convolutional long short term memory

Due to the influence of temporal and spatial correlations, wind speed prediction is exceptionally challenging. Therefore, this study introduces ConvLSTM to extract the spatiotemporal characteristics of wind speed data. Researchers have verified the superiority of ConvLSTM to extract spatiotemporal characteristics, including traffic flow prediction (Lin et al., 2020), infectious disease prediction (Paul et al., 2020), and rainfall prediction (Liu et al., 2022). ConvLSTM preserves the merits of LSTM to capture long-term temporal dependencies while incorporating the advantage of convolution operator to capture spatial features. So ConvLSTM could be utilized to predict spatiotemporal wind speed. The structure of ConvLSTM is shown in Fig. 5, and its detailed mathematical formula can be calculated as follows:

$$f_t = \sigma\left(\mathbf{W}_{xf} * x_t + \mathbf{W}_{hf} * h_{t-1} + \mathbf{W}_{cf} \circ c_{t-1} + \mathbf{b}_f\right) \tag{5}$$

$$i_t = \sigma\left(\mathbf{W}_{xi} * x_t + \mathbf{W}_{hi} * h_{t-1} + \mathbf{W}_{ci} \circ c_{t-1} + \mathbf{b}_i\right) \tag{6}$$

$$o_t = \sigma\left(\mathbf{W}_{xo} * x_t + \mathbf{W}_{ho} * h_{t-1} + \mathbf{W}_{co} \circ c_t + \mathbf{b}_o\right) \tag{7}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh\left(\mathbf{W}_{xc} * x_t + \mathbf{W}_{hc} * h_{t-1} + \mathbf{b}_c\right) \tag{8}$$

$$h_t = o_t \circ \tanh\left(c_t\right) \tag{9}$$

where f_t, i_t, and o_t respectively indicate the forgotten gates, input gates, and output gates of ConvLSTM. By computing current input x_t with previous hidden state h_{t-1}, the forgotten gate concludes what information should be deserted from the previous cell state c_{t-1}. The input gate determines what current information should be stored by updating the current cell state c_t. The output gate determines which information should be chosen from c_t to be passed as output to the next ConvLSTM. $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the offset. $*$ and $\circ$ denote convolution operation and Hadamard operation respectively. σ represents the Sigmoid activation function defined in the Eq. (10). tanh represents the hyperbolic tangent activation function defined in the Eq. (11).

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{10}$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{\mathrm{e}^x - \mathrm{e}^{-x}}{\mathrm{e}^x + \mathrm{e}^{-x}} \tag{11}$$

When LSTM processes spatiotemporal data, the input and hidden state are fully connected to each gate-based structure, so capturing the internal spatial correlations is complicated. Compared with LSTM, ConvLSTM introduces convolution operation instead of the usage of full connections where no spatial structural information is encoded. With the convolution operation, ConvLSTM can capture interregional spatial information by integrating historical input and its surrounding neighbors to model the future state of entire region and realize precise wind speed prediction.

3.3. Residual deformable convolution module

In traditional convolution operation, the convolution kernel with a fixed size (e.g., 3×3) is utilized to sample feature maps, restricting the ability to handle complex spatial transformations (Wu et al., 2021). Only some of the information in the receptive field covered by the standard convolution kernel contribute equally to the sampling points for the gridded wind speed feature map. Because different spatial locations may correspond to features with different scale and deformation, the convolution kernel with a fixed size in ConvLSTM cannot effectively learn the changeable relationship in the local area (Li et al., 2021; Zhao et al., 2022), and it is difficult to extend to the complex and non-stationary large wind farm.

In order to solve this problem, the study proposes a residual deformable convolution module (RDCM) based on DCNv2. DCNv2 uses additional offsets learned from the spatial distribution of wind speed to obtain critical sampling locations and modulates the amplitude of input features. The sampling grid can deform freely to simulate local irregular wind speed trends in real scenarios. Residual connection can reduce model complexity to reduce overfitting. Therefore, through RDCM, the receptive field is concentrated on the critical sampling points with similar trends as much as possible, and the model can effectively represent the local spatial correlations of wind speed.

As shown in Fig. 6, the orange points represent the regular sampling grid of standard convolution, and the 2-D offset allows the deformations convolution to select more data containing the required information (dark blue points). Standard convolution first samples using regular grid G over the input feature map $x_{\mathrm{cur\_conv}}$ and then computes the weighted summation of sampled values, where $I = |G|$. For instance, standard convolution defines a 3×3 kernel with dilation 1 :

$$G = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\} \tag{12}$$

The output feature map of standard convolution $y_{\mathrm{std\_conv}}$ at each location p_0 can be calculated as follows:

$$y_{\mathrm{std\_conv}}\left(p_0\right) = \sum_{p_i \in G} \mathbf{w}\left(p_i\right) \cdot x_{\mathrm{cur\_conv}}\left(p_0 + p_i\right) \tag{13}$$

where $p_i \in G$ enumerates the locations in G, $x_{\mathrm{cur\_conv}}\left(p_0 + p_i\right)$ is an arbitrary location in the input feature map, and $\mathbf{w}\left(p_i\right)$ is the weight of standard convolution kernel at p_i.

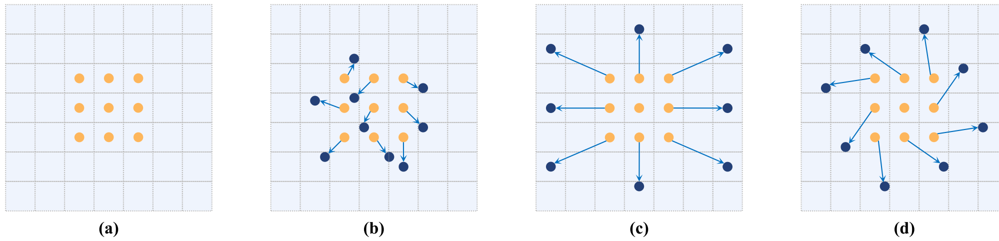

Fig. 6. Illustration of sampling locations in standard convolution and Deformable Convolution V2. (a) regular sampling grid (orange points) of standard convolution. (b) - (d) deformed sampling locations (dark blue points) with augmented offsets (light blue arrows) of Deformable Convolution V2.

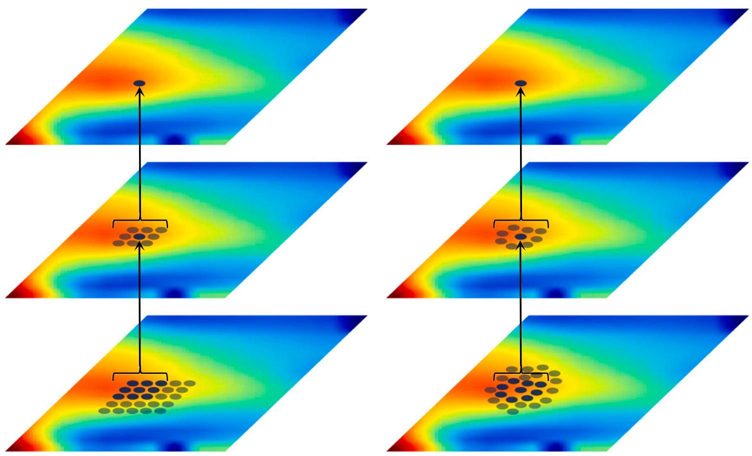

Fig. 7. Illustration of the receptive fields of standard convolution and Deformable Convolution V2. Left: standard convolution. Right: Deformable Convolution V2.

Compared with standard convolution, the output feature map of deformable convolution $y_{\text{def\_conv}}$ at each sampling location p_0 can be calculated as follows:

$$y_{\text{def\_conv}}\left(p_0\right) = \sum_{p_i \in G} \mathbf{w}\left(p_i\right) \cdot x_{\text{cur\_conv}}\left(p_0 + p_i + \Delta p_i\right) \cdot \Delta m_i \tag{14}$$

where Δp_i and Δm_i respectively denote the learnable offset and modulation scalar at p_i. As shown in Fig. 7, Δp_i and Δm_i are obtained through an additional convolution layer Conv applied over the same input feature map. Conv has the same kernel and dilation as the current input feature map $x_{\text{cur\_conv}}$. The output is of $3I$ channels, where the former $2I$ channels correspond to the adaptively learnable offsets $\left\{\Delta p_i\right\}_{i=1}^{I}$. The remaining I channels are fed into via the Sigmoid activation function to obtain the learned modulation scalars $\left\{\Delta m_i\right\}_{i=1}^{I}$. Δp_i is typically fractional and must be integrated via bilinear interpolation.

Fig. 8 shows the difference between the receptive fields of standard convolution and Deformable Convolution V2. The receptive field and sampling locations in standard convolution remain constant across the top feature map. According to the spatial distribution of wind speed, DCNv2 adjusts the receptive field, sampling locations and modulation scalars in order to simulate the local wind speed flow dynamics.

RDCM combines DCNv2, BatchNorm, HardSwish activation function with the residual connection, as shown in Fig. 3(b). The output $y_r \in \mathbb{R}^{C\times H\times W}$ can be calculated as follows:

$$y_r = \mathcal{F}_{\text{BHD}}\left(x_r\right) + x_r \tag{15}$$

where $x_r \in \mathbb{R}^{C\times H\times W}$ denotes the input feature map of RDCM. $\mathcal{F}_{\text{BHD}}(\cdot)$ denotes the integration of BatchNorm, HardSwish and deformable convolution operation with modulation mechanism. HardSwish (Howard et al., 2019) can be implemented as a piece-wise function to reduce the number of memory accesses, thus decreasing the computation cost. Deep learning models typically using HardSwish perform better than ReLU. The formula of HardSwish can be calculated as follows:

$$\text{HardSwish}(x) = \begin{cases} 0, & if\ \ x \le -3 \\ x, & if\ \ x \ge 3 \\ \frac{x\cdot(x+3)}{6}, & otherwise \end{cases} \tag{16}$$

Deformable Convolution V2 is embedded into RDCM, adaptive learning local wind speed flow dynamics. Adding the residual connection realizes the reuse of original input features. Therefore, RDCM can further enhance the capacity to capture local spatial correlations.

3.4. Dense coordinate attention module

Coordinate attention mechanism allows feature maps to embed positional information into channel attention and then capture long-range dependencies along one spatial direction while retaining precise positional information along another spatial direction. Further, this study proposes a dense coordinate attention module (DCAM), which acts on the last hidden state of ConvLSTM encoder at each layer. The structure of coordinate attention is shown in Fig. 9. CA encodes long-range spatial correlations through two specific steps: coordinate information embedding and coordinate attention generation.

Specifically, given an input x with dimension size $C \times H \times W$, two one-dimensional average pooling layers with kernels $(H, 1)$ and $(1, W)$ are utilized for encoding each channel along the horizontal coordinate and the vertical coordinate respectively. For the output of cth channel, the pooling process at height h and width w can be calculated as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0\le i<w} x_c(h, i) \tag{17}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0\le j<h} x_c(j, w) \tag{18}$$

where x_c and z_c respectively denote the input and output of the cth channel in the above formulas. These two transformations separately

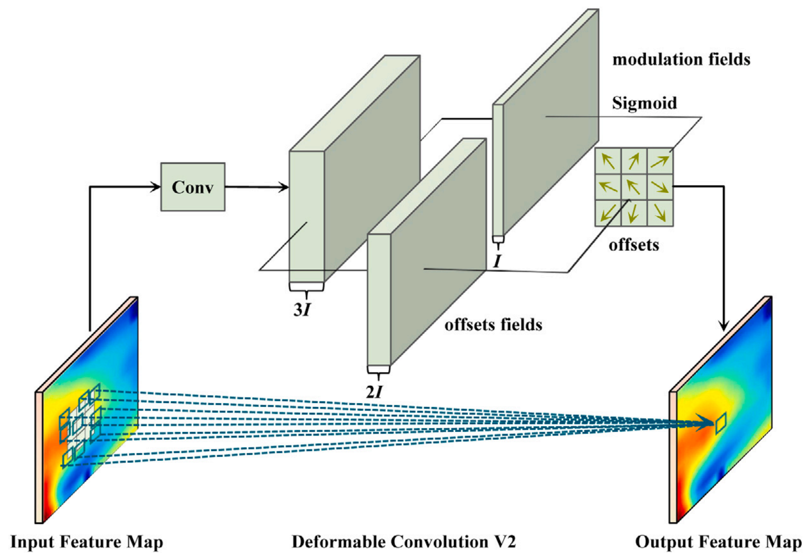

Fig. 8. Illustration of Deformable Convolution V2.

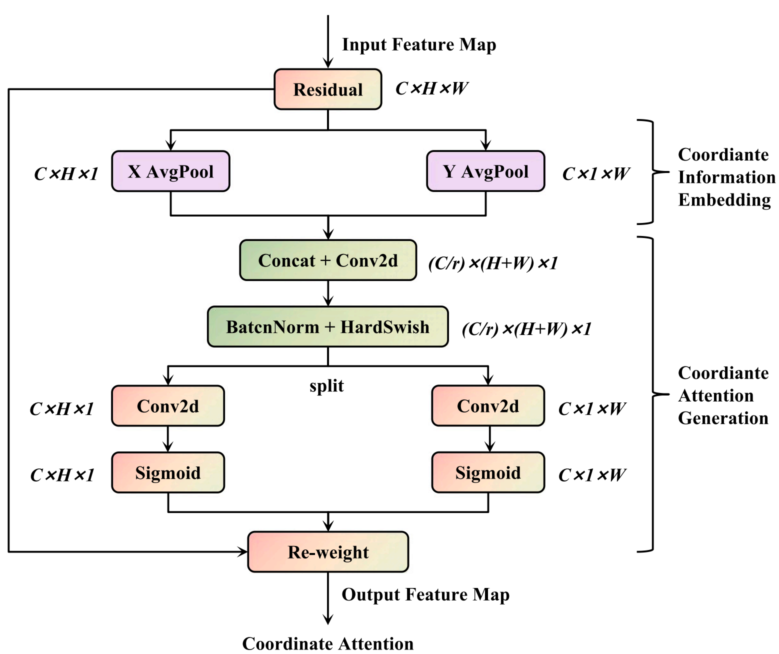

Fig. 9. Illustration of Coordinate Attention.

aggregate features in horizontal and vertical coordinates, generating a pair of directional-aware feature maps, which realize the coordinate information embedding.

In order to make full use of global coordinate information, the two encoding features are concatenated along the horizontal coordinate and sent through a shared convolutional transformation $\mathcal{F}_s$ with kernel 1×1, and it can be calculated as follows:

$$f_s = \text{HardSwish}\left(\mathcal{F}_s\left[z^h; z^w\right]\right) \tag{19}$$

where $[\cdot;\cdot]$ denotes the concatenation operation along the horizontal spatial dimension. $f_s \in \mathbb{R}^{C/r\times(H+W)\times 1}$ denotes the intermediate feature map encoding the locational information in both the horizontal direction and vertical direction. r is the reduction ratio that controls dimension to reduce the model's complexity and is generally set to 32.

Then split f_s along the horizontal spatial dimension into two independent tensors $f^h \in \mathbb{R}^{C/r\times H\times 1}$ and $f^w \in \mathbb{R}^{C/r\times 1\times W}$. Additional two 1×1 convolutional transformations $\mathcal{F}_h$ and $\mathcal{F}_w$ with Sigmoid activation function are utilized to restore the channel number of f^h and f^w to the initial value C, which can be calculated as follows:

$$g^h = \sigma\left(\mathcal{F}_h\left(f^h\right)\right) \tag{20}$$

$$g^w = \sigma\left(\mathcal{F}_w\left(f^w\right)\right) \tag{21}$$

where g^h and g^w denote the attention weights in the horizontal and vertical directions, respectively.

Finally, multiplying the input x with the two attention weights yields the coordinate attention's output y_c, which can be calculated as follows:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{22}$$

Coordinate attention mechanism is embedded in DCAM, as shown in Fig. 3(c). CBH is utilized before the coordinate attention mechanism, and a residual connection is added to get the intermediate result y_m. Assuming the input feature map of DCAM is $x_d \in \mathbb{R}^{C\times H\times W}$, y_m can be calculated as follows:

$$y_m = \mathcal{F}_{\text{CA}}\left(\mathcal{F}_{\text{CBH}}\left(x_d\right)\right) + x_d \tag{23}$$

where $\mathcal{F}_{\text{CA}}(\cdot)$ denotes coordinate attention operation. $\mathcal{F}_{\text{CBH}}(\cdot)$ denotes the integration operation of 2-D convolution, BatchNorm and HardSwish. The intermediate results repeat the above operations and then integrate the initial input and intermediate results to obtain the final output of dense connections. The output $y_d \in \mathbb{R}^{C\times H\times W}$ can be calculated as follows:

$$y_d = \mathcal{F}_{\text{CA}}\left(\mathcal{F}_{\text{CBH}}\left(y_m\right)\right) + y_m + x_d \tag{24}$$

Standard convolution with a fixed kernel can only process position information in adjacent regions. By encoding global information using two complementary 1-D average pooling operations, DCAM can capture spatial correlations in distant regions while avoiding the loss of position information caused by 2-D global pooling used by other channel and spatial attention mechanisms. The following experimental verification shows that introducing DCAM into the proposed approach can improve the accuracy of wind speed prediction.

4. Experiments

4.1. Data preprocessing

Wind energy research requires high-quality wind speed datasets (Draxl et al., 2015). This study uses a grid integration dataset called the Western Wind and Solar Integration Study (WWSIS) dataset (Potter et al., 2008). The WWSIS dataset selects over 30,000 location points for further simulation. To facilitate the experimental evaluation, each location point with ten Vestas V-90 (3MW) turbines can be considered a virtual wind turbine. It produces wind speed data with a spatial resolution of 2 km × 2 km and a temporal resolution of 10 min, namely 144 data points per day.

In the experiments, the dataset from January 2004 to December 2004 is utilized as the training set to train the parameters of the proposed LLConvLSTM model. The remaining dataset from January 2005 to December 2005 is utilized as the verification set to verify the prediction performance. The study selects 253 virtual wind turbines to present a spatiotemporal feature map of size 16*16, which covers the region from 41.708°N to 41.958°N in latitude and 106.508°W to 106.258°W in longitude, as shown in Fig. 1. Blank points are supplemented with zero values. Gridded wind speed data per 10 min is continuous frames and the value in each grid can be considered as the pixel in the feature maps. The study adjusts the shape of the training data and verification data to the required tensor format in the PyTorch framework.

From the common knowledge of wind speed prediction, the longer the past time, the less influence it has on the present. Considering the influence of time step, this study utilizes the previous six consecutive samples to predict the future one-step-ahead, two-step-ahead and three-step-ahead wind speed, namely wind speed in the future 10 min, 20 min and 30 min.

Given the noticeable discrepancy in the wind speed value, the study applies the maximum–minimum normalization to all input sequences before feeding them into the model, which can eliminate the gradient explosion and improve the model convergence speed. The maximum–minimum normalization can be calculated as follows:

$$v' = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{25}$$

where v_{max} and v_{min} denote the maximum and minimum values of the sample data in the training set respectively. In evaluating the prediction performance, the predictive data and the truth must pass through the denormalization operation. The parameters of the denormalization operation are still based on the wind speed range of the original training set.

4.2. Experimental configurations

The study proposes the LLConvLSTM model for wind speed prediction, which is implemented by Pytorch library of Python programming language. To prevent overfitting, the model must randomly shuffle the order of input samples. The model is trained for 100 epochs and the batch size is set to 256. The study trains the model by minimizing the mean square error loss and the Adam (Kingma and Ba, 2014) optimizer with 1×10^{-5} weight decay. Additionally, the gradient is clipped before the optimizer updates the parameters to alleviate gradient explosion or gradient vanishing. The learning rate tuning function is CosineAnnealing, where the initial learning rate is set to 1×10^{-3} and the termination learning rate is set to 1×10^{-5}.

4.3. Evaluation metrics

To verify the validity of the LLConvLSTM model, this study evaluates the model using four standard metrics: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE) and coefficient of determination (R^2). MAE, MSE and RMSE are utilized to measure the deviation of the predicted value from the actual value. The closer MAE, MSE and RMSE are to 0, the better predictive performance the model achieves. The coefficient of determination R^2 is utilized to measure the fitting performance, with a value less than or equal to 1. The closer R^2 is to 1, the better fitting performance the model achieves. The formulas for MAE, MSE, RMSE and R^2 can be calculated as follows:

$$\text{MAE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{26}$$

$$\text{MSE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{27}$$

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{28}$$

$$\text{R}^2(y, \bar{y}, \hat{y}) = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2} \tag{29}$$

where N denotes the total number of samples in the verification set, y_i denotes the actual wind speed value, $\hat{y}_i$ denotes the predicted wind speed value, and $\bar{y}$ denotes the average of all actual wind speed values.

4.4. Comparisons with other methods

4.4.1. Detailed description of the compared methods

In order to verify the superiority of the LLConvLSTM model, the study utilizes a series of prediction methods and compares them with the proposed approach. These methods include Naive Persistence (Potter and Negnevitsky, 2006), CNN (Harbola and Coors, 2019), LSTM (Kisvari et al., 2021), CNN-LSTM (Shen et al., 2022) and four conventional machine learning regression algorithms (Liao et al., 2021; Kusiak and Zhang, 2010; Lahouar and Slama, 2017; Velo et al., 2014).

Naive Persistence method is the baseline for all regression problems, which directly utilizes the wind speed observed values at the previous moment as the future prediction. CNN can handle topological data with a distinct gridded shape and LSTM has been employed successfully for processing sequential data. CNN-LSTM is widely applied as a hybrid model for wind speed prediction. Convolutional operation is responsible for reading and encoding local features in the time series, while the LSTM layer receives the extracted features from the CNN layer as input for multi-step prediction. Furthermore, the experiments use four conventional machine learning regression algorithms to further verify the superiority of the proposed approach. These regression algorithms include gradient boosting regression tree (GBRT), k-nearest neighbor regression (k-NN), random forest (RF) and multilayer perceptron (MLP).

Table 1
Comparison of LLConvLSTM and other methods for one-step-ahead prediction in the 2005 verification set. Best results are shown in bold.

Method	One-step-ahead			
	MAE	MSE	RMSE	R^2
Naive Persistence (Potter and Negnevitsky, 2006)	0.2555	0.2029	0.4504	0.9907
GBRT (Liao et al., 2021)	0.2483	0.1898	0.4357	0.9912
k-NN (Kusiak and Zhang, 2010)	0.2187	0.1819	0.4265	0.9917
RF (Lahouar and Slama, 2017)	0.2149	0.1763	0.4199	0.9919
MLP (Velo et al., 2014)	0.2299	0.1745	0.4177	0.9920
LSTM (Kisvari et al., 2021)	0.2263	0.1751	0.4184	0.9920
CNN (Harbola and Coors, 2019)	0.2292	0.1837	0.4287	0.9916
CNN-LSTM (Shen et al., 2022)	0.2176	0.1702	0.4126	0.9922
LLConvLSTM	**0.1919**	**0.1199**	**0.3463**	**0.9945**

Table 2
Comparison of LLConvLSTM and other methods for two-step-ahead prediction in the 2005 verification set. Best results are shown in bold.

Method	Two-step-ahead			
	MAE	MSE	RMSE	R^2
Naive Persistence (Potter and Negnevitsky, 2006)	0.4478	0.5277	0.7264	0.9759
GBRT (Liao et al., 2021)	0.4368	0.4932	0.7023	0.9768
k-NN (Kusiak and Zhang, 2010)	0.4032	0.4758	0.6898	0.9780
RF (Lahouar and Slama, 2017)	0.3995	0.4664	0.6829	0.9783
MLP (Velo et al., 2014)	0.4104	0.4572	0.6762	0.9782
LSTM (Kisvari et al., 2021)	0.4138	0.4675	0.6837	0.9787
CNN (Harbola and Coors, 2019)	0.4259	0.4997	0.7069	0.9772
CNN-LSTM (Shen et al., 2022)	0.3944	0.4445	0.6667	0.9797
LLConvLSTM	**0.3492**	**0.3446**	**0.5870**	**0.9839**

Table 3
Comparison of LLConvLSTM and other methods for three-step-ahead prediction in the 2005 verification set. Best results are shown in bold.

Method	Three-step-ahead			
	MAE	MSE	RMSE	R^2
Naive Persistence (Potter and Negnevitsky, 2006)	0.6031	0.8720	0.9338	0.9602
GBRT (Liao et al., 2021)	0.5902	0.8117	0.9009	0.9613
k-NN (Kusiak and Zhang, 2010)	0.5578	0.7880	0.8877	0.9632
RF (Lahouar and Slama, 2017)	0.5532	0.7711	0.8781	0.9638
MLP (Velo et al., 2014)	0.5528	0.7529	0.8677	0.9645
LSTM (Kisvari et al., 2021)	0.5594	0.7679	0.8763	0.9650
CNN (Harbola and Coors, 2019)	0.5868	0.8378	0.9153	0.9618
CNN-LSTM (Shen et al., 2022)	0.5474	0.7461	0.8638	0.9660
LLConvLSTM	**0.4752**	**0.5798**	**0.7614**	**0.9729**

In this experiment, Naive Persistence is implemented by Numpy library of Python programming language. CNN, LSTM and CNN-LSTM are implemented by Pytorch library of Python programming language. The four conventional machine learning methods are implemented by Sklearn library of Python programming language (Pedregosa et al., 2011). Meanwhile, the wind speed data are converted into the input formats that these methods can handle.

4.4.2. Comparison of experimental results

Comparison between LLConvLSTM and the other models for one-step-ahead prediction is shown in Table 1. The proposed method achieves MAE, MSE and RMSE of 0.1919, 0.1199 and 0.3463 respectively, generally outperforming other models. The proposed method reduces MAE, MSE and RMSE by 24.89%, 40.91% and 23.11% respectively compared to Naive Persistence, which struggles to model the nonlinear characteristics of wind speed data. Naive Persistence merely employs the observation from the previous time step as the prediction value, leading to a failure in fully capturing the temporal correlations of wind speed and disregarding its spatial correlations. It is worth mentioning that the proposed approach exhibits more promising prediction performance than the deep learning-based method. Compared with CNN, LSTM and CNN-LSTM, the MAE is decreased by 16.27%, 15.20% and 11.81%, and MSE by 34.73%, 31.52%, 29.55% and RMSE by 19.22%, 17.23%, 16.07% respectively. CNN, LSTM, and even CNN-LSTM partially capture the spatiotemporal correlations of wind speed, but they fall far short of LLConvLSTM in this aspect. LLConvLSTM comprehensively perceives spatiotemporal correlations in wind speed prediction and introduces RDCM and DCAM modules to further enhance modeling capabilities for local and long-range spatial correlations. LLConvLSTM also outperformed the other four machine learning regression algorithms in prediction performance. These machine learning methods are computationally expensive, and their performance is sensitive to the choice of hyperparameters. Moreover, they face challenges in interpreting the spatiotemporal correlations in wind speed data. The proposed approach achieves a peak of R^2 of 0.9945, which is closer to 1 compared to other methods, and the fitting is satisfactory. Important factors affecting the integration of wind turbines include the geospatial distribution of turbines and the time-varying wind speed data at each location. With wind energy, turbine power generation is forced by variations in wind speed's spatial and temporal correlations. Accurate wind speed prediction in spatiotemporal correlations is vital for the rational dispatch of power systems. ConvLSTM encoder–decoder architecture maintains and integrates spatiotemporal information from deep and shallow layers. Considering the local and long-range spatial correlations in entire wind farms, the proposed approach further enhances the prediction accuracy by introducing RDCM and DCAM to extract spatial information excellently.

Tables 2–3 show the comparison between LLConvLSTM and other methods for two-step-ahead and three-step-ahead prediction. The proposed approach improves MSE by 34.70% and 33.51% over Naive Persistence in two-step-ahead and three-step-ahead prediction respectively. The experimental results demonstrate that the proposed

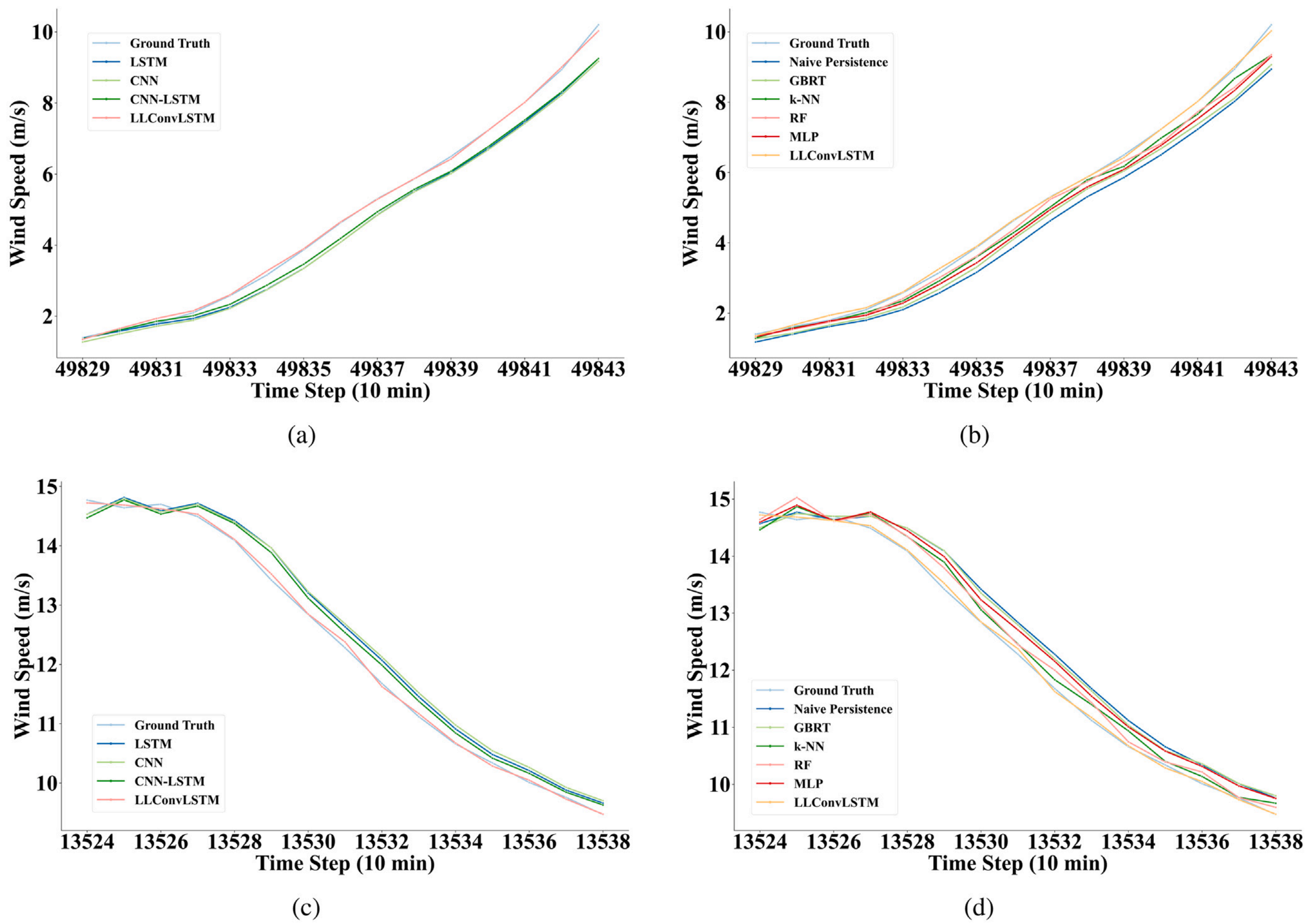

Fig. 10. (a) Comparison of wind speed prediction results between the proposed approach and deep learning methods on Turbine 17039. (b) Comparison of wind speed prediction results between the proposed approach, Naive Persistence and machine learning methods on Turbine 17039. (c) Comparison of wind speed prediction results between the proposed approach and deep learning methods on Turbine 18627. (d) Comparison of wind speed prediction results between the proposed approach, Naive Persistence and machine learning methods on Turbine 18627.

approach still significantly outperforms the other compared methods in multi-step wind speed prediction.

From Fig. 10, the wind speed prediction results are shown for one-step-ahead prediction at Turbine 17039 (106.292°W, 41.725°N) and Turbine 18627 (106.375°W, 41.942°N). It is observed that the prediction results of LLConvLSTM are closer to the actual values compared to other methods. In the two different time frames shown in 10, LLConvLSTM demonstrates a more rapid and stable perception of abrupt wind speed changes compared to other methods, particularly during periods of significant wind speed fluctuations. The proposed method exhibits a significant reduction in time delay effects, whereas these comparative methods only differ in terms of data inputs. This indicates that enhancing the perception of spatial correlations improves the ability to capture wind speed variation trends. This capability enhances the safety of wind power equipment, reducing the risk of equipment damage and ensuring the smooth operation of large-scale wind farms. Sensitive perception of abrupt wind speed variations by LLConvLSTM can provide more accurate information for the daily operation of wind farms, assisting maintenance personnel in their maintenance strategies.

Figs. 11–13 show the visualization of the wind speed prediction results for one-step-ahead at different moments. Compared to other methods, the prediction results of the proposed approach are closest to the actual values in regions with considerable wind speed variation, validating the importance of the proposed approach modeling local and long-range spatial correlations to improve the accuracy of wind speed prediction for wind farms.

During a year, wind speed data varies over 12 months due to atmospheric fluctuations. Therefore, it is necessary to analyze the wind speed prediction performance in 12 months respectively. The validation set contains one-year data, and thus the prediction performance is verified in 12 months respectively. Indexes of one-step-ahead, two-step-ahead and three-step-ahead predictions for each month in 2005 are given in Tables 4–6. From Tables 4–6, the proposed approach achieves promising results in terms of performance indexes for most months, further verifying the superiority and stability of the proposed approach.

4.4.3. Time step analysis

In the previous experiments, based on the common knowledge of wind speed prediction, this study used the previous six consecutive time steps to forecast wind speed. The time step represents a crucial parameter for spatiotemporal wind speed sequence features. Considering the influence of time step size on the performance of spatiotemporal wind speed prediction and its practical significance, this study selected three time steps, time step = 3 (half-hour data), 6 (one-hour data), and 9 (one and a half-hour data), to predict wind speed one-step-ahead and determine the appropriate time step.

According to Table 7, it is evident that the proposed method achieves an MSE of 0.1199 when the time step is 6, which is lower compared to the MSE obtained with time steps of 3 and 9. In terms of RMSE and R^2 evaluation metrics, the model with a time step of 6 also outperforms the models with the other two time steps. When predicting wind speed, selecting an appropriate time dimension is crucial, as too many or too few time steps can impact the model performance. To sum up, this study adopts a time step of 6 for wind speed prediction.

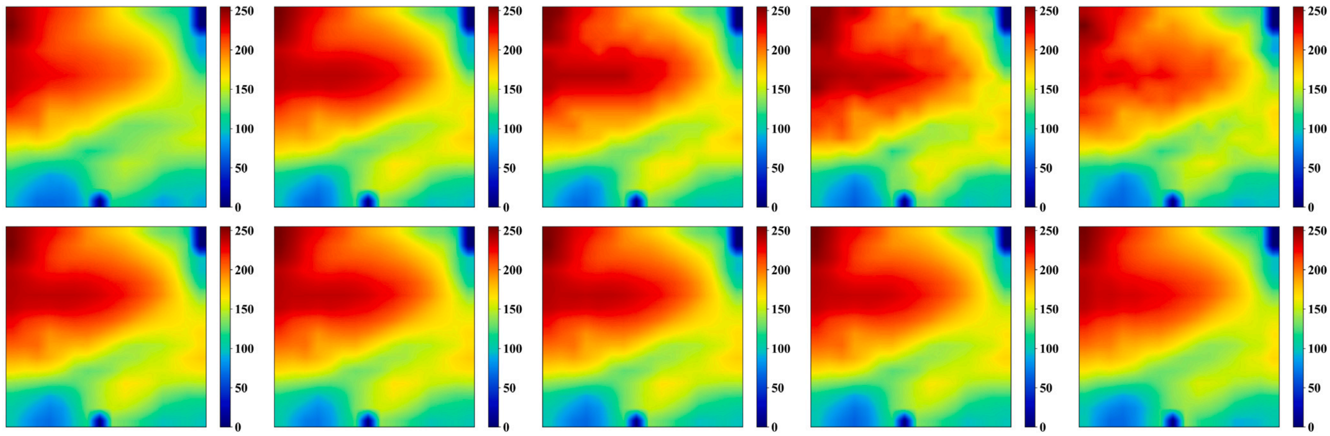

Fig. 11. Visualizations of wind speed prediction results one-step-ahead at timestamp 34657. The first row includes the results of Ground Truth, Naive Persistence, GBRT, *k*-NN and RF respectively. The second row includes the results of MLP, LSTM, CNN, CNN-LSTM and LLConvLSTM respectively.

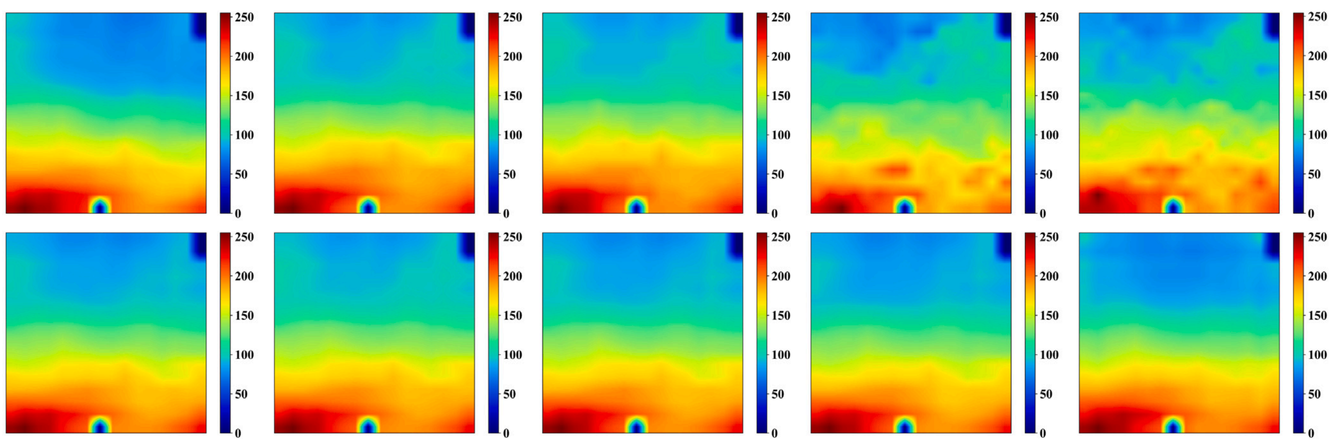

Fig. 12. Visualizations of wind speed prediction results one-step-ahead at timestamp 41019. The first row includes the results of Ground Truth, Naive Persistence, GBRT, *k*-NN and RF respectively. The second row includes the results of MLP, LSTM, CNN, CNN-LSTM and LLConvLSTM respectively.

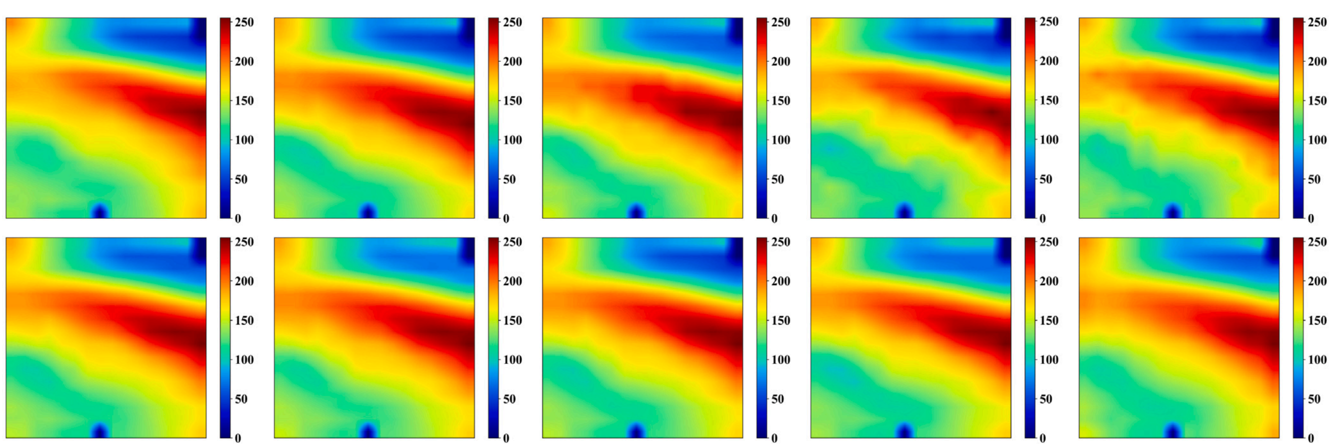

Fig. 13. Visualizations of wind speed prediction results one-step-ahead at timestamp 42535. The first row includes the results of Ground Truth, Naive Persistence, GBRT, *k*-NN and RF respectively. The second row includes the results of MLP, LSTM, CNN, CNN-LSTM and LLConvLSTM respectively.

4.4.4. *Learning rate analysis*

An appropriate learning rate is crucial for achieving optimal model performance. Higher learning rates can lead to oscillations or failure to converge during training, as they might skip the global minimum, resulting in poor model performance. On the other hand, lower learning rates can lead to a slow training process that may not reach optimal results within a reasonable number of training iterations. A learning rate between higher and lower values is a more suitable choice, as it

Table 4
Comparison of LLConvLSTM and other methods for one-step-ahead prediction in the 12 months verification set. Best results are shown in bold.

Metric	Method	One-step-ahead											
		Jan.	Feb.	Mar.	Apr.	May.	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
MAE	Naive Persistence (Potter and Negnevitsky, 2006)	0.1741	0.1687	0.1827	0.2614	0.2966	0.3812	0.3404	0.3117	0.3194	0.2107	0.2239	0.1912
	GBRT (Liao et al., 2021)	0.1761	0.1649	0.1800	0.2489	0.2857	0.3664	0.3278	0.3012	0.3068	0.2017	0.2234	0.1924
	k-NN (Kusiak and Zhang, 2010)	0.1313	0.1235	0.1475	0.2164	0.2699	0.3532	0.3119	0.2856	0.2857	0.1670	0.1784	0.1481
	RF (Lahouar and Slama, 2017)	**0.1293**	**0.1217**	0.1450	0.2128	0.2666	0.3470	0.3069	0.2807	0.2803	0.1647	0.1739	**0.1441**
	MLP (Velo et al., 2014)	0.1515	0.1507	0.1622	0.2315	0.2740	0.3500	0.3155	0.2907	0.2892	0.1876	0.1903	0.1612
	LSTM (Kisvari et al., 2021)	0.1486	0.1446	0.1565	0.2262	0.2711	0.3482	0.3118	0.2880	0.2870	0.1818	0.1870	0.1606
	CNN (Harbola and Coors, 2019)	0.1509	0.1446	0.1593	0.2297	0.2722	0.3529	0.3141	0.2878	0.2896	0.1835	0.1948	0.1659
	CNN-LSTM (Shen et al., 2022)	0.1415	0.1298	0.1490	0.2136	0.2624	0.3400	0.3029	0.2787	0.2777	0.1684	0.1861	0.1563
	LLConvLSTM	0.1356	0.1262	**0.1354**	**0.1990**	**0.2221**	**0.2842**	**0.2486**	**0.2302**	**0.2349**	**0.1588**	**0.1724**	0.1518
MSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.0673	0.0690	0.0771	0.1991	0.2429	0.4385	0.3290	0.2804	0.3219	0.1189	0.1499	0.1360
	GBRT (Liao et al., 2021)	0.0660	0.0640	0.0742	0.1800	0.2267	0.4094	0.3095	0.2607	0.2992	0.1071	0.1445	0.1325
	k-NN (Kusiak and Zhang, 2010)	0.0462	0.0476	0.0619	0.1682	0.2286	0.4141	0.3095	0.2583	0.3047	0.0921	0.1258	0.1206
	RF (Lahouar and Slama, 2017)	0.0453	0.0466	0.0605	0.1617	0.2239	0.4009	0.3001	0.2506	0.2941	0.0908	0.1200	0.1163
	MLP (Velo et al., 2014)	0.0505	0.0533	0.0620	0.1627	0.2179	0.3896	0.2978	0.2508	0.2820	0.0945	0.1193	0.1091
	LSTM (Kisvari et al., 2021)	0.0501	0.0522	0.0606	0.1638	0.2169	0.3914	0.2982	0.2526	0.2862	0.0937	0.1176	0.1130
	CNN (Harbola and Coors, 2019)	0.0508	0.0529	0.0612	0.1717	0.2303	0.4197	0.3074	0.2634	0.2950	0.1011	0.1290	0.1176
	CNN-LSTM (Shen et al., 2022)	0.0479	0.0472	0.0584	0.1557	0.2133	0.3815	0.2898	0.2456	0.2769	0.0883	0.1197	0.1136
	LLConvLSTM	**0.0433**	**0.0435**	**0.0467**	**0.1196**	**0.1486**	**0.2481**	**0.1901**	**0.1580**	**0.1827**	**0.0722**	**0.0939**	**0.0897**
RMSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.2595	0.2627	0.2777	0.4462	0.4929	0.6622	0.5736	0.5295	0.5674	0.3448	0.3872	0.3688
	GBRT (Liao et al., 2021)	0.2568	0.2529	0.2723	0.4243	0.4762	0.6398	0.5563	0.5106	0.5470	0.3272	0.3801	0.3640
	k-NN (Kusiak and Zhang, 2010)	0.2149	0.2181	0.2488	0.4101	0.4781	0.6435	0.5563	0.5083	0.5520	0.3034	0.3547	0.3473
	RF (Lahouar and Slama, 2017)	0.2128	0.2159	0.2460	0.4021	0.4732	0.6332	0.5478	0.5006	0.5423	0.3014	0.3464	0.3410
	MLP (Velo et al., 2014)	0.2248	0.2308	0.2489	0.4033	0.4668	0.6242	0.5457	0.5008	0.5310	0.3074	0.3453	0.3302
	LSTM (Kisvari et al., 2021)	0.2238	0.2284	0.2461	0.4047	0.4658	0.6256	0.5461	0.5026	0.5350	0.3060	0.3429	0.3362
	CNN (Harbola and Coors, 2019)	0.2253	0.2299	0.2474	0.4144	0.4799	0.6479	0.5544	0.5132	0.5432	0.3179	0.3592	0.3429
	CNN-LSTM (Shen et al., 2022)	0.2189	0.2173	0.2416	0.3946	0.4618	0.6177	0.5383	0.4956	0.5262	0.2972	0.3460	0.3371
	LLConvLSTM	**0.2081**	**0.2085**	**0.2161**	**0.3458**	**0.3855**	**0.4981**	**0.4361**	**0.3975**	**0.4274**	**0.2687**	**0.3064**	**0.2994**
R^2	Naive Persistence (Potter and Negnevitsky, 2006)	0.9980	0.9971	0.9965	0.9850	0.9808	0.9620	0.9639	0.9740	0.9712	0.9907	0.9901	0.9944
	GBRT (Liao et al., 2021)	0.9980	0.9973	0.9967	0.9864	0.9821	0.9645	0.9660	0.9758	0.9732	0.9916	0.9904	0.9945
	k-NN (Kusiak and Zhang, 2010)	0.9986	0.9980	0.9972	0.9873	0.9819	0.9641	0.9660	0.9761	0.9727	0.9928	0.9917	0.9950
	RF (Lahouar and Slama, 2017)	0.9986	0.9980	0.9973	0.9878	0.9823	0.9653	0.9671	0.9768	0.9737	0.9929	0.9920	0.9952
	MLP (Velo et al., 2014)	0.9985	0.9977	0.9972	0.9877	0.9828	0.9663	0.9673	0.9768	0.9748	0.9926	0.9921	0.9955
	LSTM (Kisvari et al., 2021)	0.9985	0.9978	0.9973	0.9876	0.9828	0.9661	0.9673	0.9766	0.9744	0.9927	0.9922	0.9953
	CNN (Harbola and Coors, 2019)	0.9985	0.9978	0.9972	0.9870	0.9818	0.9636	0.9663	0.9756	0.9736	0.9921	0.9914	0.9952
	CNN-LSTM (Shen et al., 2022)	0.9985	0.9980	0.9974	0.9882	0.9831	0.9670	0.9682	0.9773	0.9752	0.9931	0.9921	0.9953
	LLConvLSTM	**0.9987**	**0.9982**	**0.9979**	**0.9910**	**0.9882**	**0.9785**	**0.9791**	**0.9854**	**0.9837**	**0.9943**	**0.9938**	**0.9963**

Table 5
Comparison of LLConvLSTM and other methods for two-step-ahead prediction in the 12 months verification set. Best results are shown in bold.

Metric	Method	Two-step-ahead											
		Jan.	Feb.	Mar.	Apr.	May.	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
MAE	Naive Persistence (Potter and Negnevitsky, 2006)	0.3197	0.3114	0.3298	0.4668	0.5041	0.6433	0.5756	0.5279	0.5522	0.3806	0.4060	0.3500
	GBRT (Liao et al., 2021)	0.3207	0.3065	0.3252	0.4489	0.4893	0.6220	0.5584	0.5147	0.5343	0.3678	0.4008	0.3475
	k-NN (Kusiak and Zhang, 2010)	0.2605	0.2488	0.2872	0.4092	0.4810	0.6165	0.5492	0.5095	0.5149	0.3247	0.3400	0.2880
	RF (Lahouar and Slama, 2017)	0.2595	0.2474	0.2848	0.4066	0.4768	0.6103	0.5437	0.5042	0.5101	0.3218	0.3356	0.2850
	MLP (Velo et al., 2014)	0.2915	0.2803	0.2981	0.4173	0.4722	0.5997	0.5417	0.5044	0.5067	0.3406	0.3561	0.3100
	LSTM (Kisvari et al., 2021)	0.2944	0.2827	0.2986	0.4198	0.4725	0.6050	0.5415	0.5044	0.5112	0.3420	0.3679	0.3191
	CNN (Harbola and Coors, 2019)	0.3026	0.2906	0.3104	0.4374	0.4841	0.6205	0.5546	0.5110	0.5266	0.3553	0.3797	0.3311
	CNN-LSTM (Shen et al., 2022)	0.2656	0.2532	0.2803	0.4006	0.4618	0.5915	0.5290	0.4925	0.4989	0.3205	0.3417	0.2897
	LLConvLSTM	**0.2528**	**0.2380**	**0.2543**	**0.3630**	**0.3970**	**0.5061**	**0.4403**	**0.4099**	**0.4315**	**0.2944**	**0.3161**	**0.2826**
MSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.2193	0.2198	0.2354	0.5501	0.5897	1.0507	0.8158	0.6894	0.8174	0.3368	0.4207	0.3783
	GBRT (Liao et al., 2021)	0.2145	0.2073	0.2275	0.4970	0.5522	0.9727	0.7656	0.6448	0.7536	0.3087	0.4032	0.3628
	k-NN (Kusiak and Zhang, 2010)	0.1661	0.1643	0.2040	0.4637	0.5701	0.9858	0.7731	0.6612	0.7613	0.2764	0.3483	0.3240
	RF (Lahouar and Slama, 2017)	0.1651	0.1621	0.2005	0.4553	0.5568	0.9649	0.7521	0.6456	0.7522	0.2703	0.3400	0.3204
	MLP (Velo et al., 2014)	0.1823	0.1763	0.1984	0.4535	0.5290	0.9246	0.7366	0.6255	0.7089	0.2706	0.3513	0.3192
	LSTM (Kisvari et al., 2021)	0.1851	0.1797	0.1994	0.4640	0.5364	0.9510	0.7489	0.6388	0.7279	0.2772	0.3606	0.3313
	CNN (Harbola and Coors, 2019)	0.1921	0.1911	0.2094	0.5067	0.5732	1.0290	0.7884	0.6706	0.7854	0.3041	0.3822	0.3544
	CNN-LSTM (Shen et al., 2022)	0.1641	0.1614	0.1881	0.4407	0.5211	0.9067	0.7206	0.6198	0.7055	0.2606	0.3320	0.3034
	LLConvLSTM	**0.1406**	**0.1381**	**0.1509**	**0.3539**	**0.3995**	**0.6792**	**0.5171**	**0.4347**	**0.5425**	**0.2136**	**0.2700**	**0.2891**
RMSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.4683	0.4688	0.4852	0.7417	0.7679	1.0250	0.9032	0.8303	0.9041	0.5803	0.6486	0.6151
	GBRT (Liao et al., 2021)	0.4631	0.4553	0.4769	0.7050	0.7431	0.9862	0.8750	0.8030	0.8681	0.5556	0.6350	0.6023
	k-NN (Kusiak and Zhang, 2010)	0.4076	0.4053	0.4517	0.6809	0.7550	0.9929	0.8793	0.8131	0.8725	0.5257	0.5901	0.5692
	RF (Lahouar and Slama, 2017)	0.4063	0.4026	0.4478	0.6747	0.7462	0.9823	0.8672	0.8035	0.8673	0.5199	0.5831	0.5660
	MLP (Velo et al., 2014)	0.4270	0.4199	0.4454	0.6734	0.7273	0.9616	0.8582	0.7909	0.8420	0.5202	0.5927	0.5649
	LSTM (Kisvari et al., 2021)	0.4302	0.4239	0.4465	0.6812	0.7324	0.9752	0.8654	0.7992	0.8532	0.5265	0.6005	0.5756
	CNN (Harbola and Coors, 2019)	0.4383	0.4371	0.4576	0.7118	0.7571	1.0144	0.8879	0.8189	0.8862	0.5514	0.6182	0.5953
	CNN-LSTM (Shen et al., 2022)	0.4051	0.4017	0.4337	0.6639	0.7219	0.9522	0.8489	0.7873	0.8399	0.5105	0.5762	0.5509
	LLConvLSTM	**0.3750**	**0.3716**	**0.3885**	**0.5949**	**0.6320**	**0.8241**	**0.7191**	**0.6593**	**0.7366**	**0.4622**	**0.5196**	**0.5377**
R^2	Naive Persistence (Potter and Negnevitsky, 2006)	0.9934	0.9907	0.9894	0.9585	0.9533	0.9090	0.9105	0.9361	0.9269	0.9736	0.9721	0.9844
	GBRT (Liao et al., 2021)	0.9935	0.9912	0.9898	0.9625	0.9563	0.9157	0.9160	0.9403	0.9326	0.9758	0.9733	0.9851
	k-NN (Kusiak and Zhang, 2010)	0.9950	0.9931	0.9908	0.9650	0.9549	0.9146	0.9152	0.9388	0.9319	0.9784	0.9769	0.9867
	RF (Lahouar and Slama, 2017)	0.9950	0.9931	0.9910	0.9656	0.9559	0.9164	0.9175	0.9402	0.9327	0.9788	0.9775	0.9868
	MLP (Velo et al., 2014)	0.9945	0.9925	0.9911	0.9658	0.9581	0.9199	0.9192	0.9421	0.9366	0.9788	0.9767	0.9869
	LSTM (Kisvari et al., 2021)	0.9944	0.9924	0.9910	0.9650	0.9575	0.9176	0.9178	0.9408	0.9349	0.9783	0.9761	0.9864
	CNN (Harbola and Coors, 2019)	0.9942	0.9919	0.9906	0.9618	0.9546	0.9109	0.9135	0.9379	0.9298	0.9762	0.9747	0.9854
	CNN-LSTM (Shen et al., 2022)	0.9950	0.9932	0.9915	0.9667	0.9588	0.9215	0.9209	0.9426	0.9369	0.9796	0.9780	0.9875
	LLConvLSTM	**0.9957**	**0.9942**	**0.9932**	**0.9733**	**0.9684**	**0.9412**	**0.9433**	**0.9597**	**0.9515**	**0.9833**	**0.9821**	**0.9881**

allows the model to converge to the global minimum within appropriate epoches. The learning rate ranging from 0.01 to 0.0007 resulted in stable training and validation losses for the model. We conducted experiments by adjusting the learning rate and the results are presented in Table 8. From Table 8, it is evident that evaluation metrics, such as MAE, MSE, RMSE and R^2, are affected by the learning rate. To sum up, we selected the optimal learning rate of 0.001 for the proposed method.

4.5. Ablation experiments

The ablation experiment verifies the validity of RDCM and DCAM in Table 9. For the benchmark model without RDCM and DCAM, its future three-step predictions reached 0.1491, 0.3868, and 0.6550, respectively. For the model with the introduction of RDCM, the MSE is decreased by 12.81%, 6.95% and 6.53% respectively. Meanwhile,

Table 6
Comparison of LLConvLSTM and other methods for three-step-ahead prediction in the 12 months verification set. Best results are shown in bold.

Metric	Method	Three-step-ahead											
		Jan.	Feb.	Mar.	Apr.	May.	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
MAE	Naive Persistence (Potter and Negnevitsky, 2006)	0.4476	0.4379	0.4546	0.6378	0.6647	0.8382	0.7539	0.6894	0.7318	0.5252	0.5635	0.4859
	GBRT (Liao et al., 2021)	0.4509	0.4351	0.4504	0.6166	0.6467	0.8107	0.7322	0.6741	0.7082	0.5118	0.5566	0.4824
	k-NN (Kusiak and Zhang, 2010)	0.3841	0.3700	0.4138	0.5762	0.6450	0.8137	0.7306	0.6788	0.6949	0.4676	0.4909	0.4184
	RF (Lahouar and Slama, 2017)	0.3834	0.3690	0.4105	0.5734	0.6396	0.8054	0.7233	0.6725	0.6879	0.4635	0.4860	0.4149
	MLP (Velo et al., 2014)	0.3961	0.3877	0.4077	0.5781	0.6261	0.7902	0.7138	0.6609	0.6805	0.4720	0.4936	0.4199
	LSTM (Kisvari et al., 2021)	0.4106	0.3967	0.4139	0.5775	0.6274	0.7933	0.7130	0.6623	0.6834	0.4759	0.5101	0.4415
	CNN (Harbola and Coors, 2019)	0.4389	0.4254	0.4412	0.6146	0.6488	0.8195	0.7370	0.6776	0.7098	0.5063	0.5423	0.4739
	CNN-LSTM (Shen et al., 2022)	0.3925	0.3812	0.4040	0.5663	0.6204	0.7835	0.7051	0.6562	0.6748	0.4655	0.4920	0.4198
	LLConvLSTM	**0.3473**	**0.3308**	**0.3527**	**0.4939**	**0.5366**	**0.6776**	**0.5909**	**0.5521**	**0.5882**	**0.4092**	**0.4344**	**0.3826**
MSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.4190	0.4171	0.4303	0.9395	0.9359	1.6358	1.2940	1.0830	1.2990	0.6001	0.7545	0.6447
	GBRT (Liao et al., 2021)	0.4121	0.3977	0.4171	0.8489	0.8752	1.4929	1.2040	1.0102	1.1806	0.5565	0.7226	0.6110
	k-NN (Kusiak and Zhang, 2010)	0.3383	0.3295	0.3888	0.8003	0.9111	1.5141	1.2230	1.0489	1.1946	0.5117	0.6323	0.5464
	RF (Lahouar and Slama, 2017)	0.3362	0.3254	0.3814	0.7879	0.8855	1.4775	1.1887	1.0230	1.1757	0.5000	0.6193	0.5367
	MLP (Velo et al., 2014)	0.3421	0.3397	0.3651	0.7885	0.8399	1.4355	1.1671	0.9905	1.1267	0.4951	0.6131	0.5174
	LSTM (Kisvari et al., 2021)	0.3532	0.3426	0.3673	0.7965	0.8477	1.4644	1.1776	0.9997	1.1552	0.5015	0.6405	0.5553
	CNN (Harbola and Coors, 2019)	0.3902	0.3844	0.4007	0.8864	0.9100	1.6065	1.2583	1.0585	1.2583	0.5600	0.7067	0.6218
	CNN-LSTM (Shen et al., 2022)	0.3357	0.3284	0.3593	0.7720	0.8342	1.4219	1.1529	0.9812	1.1266	0.4874	0.6173	0.5226
	LLConvLSTM	**0.2594**	**0.2511**	**0.2793**	**0.5964**	**0.6558**	**1.0985**	**0.8462**	**0.7320**	**0.8983**	**0.3862**	**0.4772**	**0.4671**
RMSE	Naive Persistence (Potter and Negnevitsky, 2006)	0.6473	0.6458	0.6560	0.9693	0.9674	1.2790	1.1375	1.0407	1.1398	0.7747	0.8686	0.8029
	GBRT (Liao et al., 2021)	0.6419	0.6306	0.6458	0.9214	0.9355	1.2219	1.0973	1.0051	1.0866	0.7460	0.8500	0.7816
	k-NN (Kusiak and Zhang, 2010)	0.5816	0.5740	0.6235	0.8946	0.9545	1.2305	1.1059	1.0242	1.0930	0.7154	0.7951	0.7392
	RF (Lahouar and Slama, 2017)	0.5798	0.5704	0.6175	0.8876	0.9410	1.2155	1.0903	1.0114	1.0843	0.7071	0.7870	0.7326
	MLP (Velo et al., 2014)	0.5849	0.5828	0.6042	0.8880	0.9165	1.1981	1.0803	0.9952	1.0615	0.7037	0.7830	0.7193
	LSTM (Kisvari et al., 2021)	0.5943	0.5853	0.6061	0.8924	0.9207	1.2101	1.0852	0.9999	1.0748	0.7082	0.8003	0.7452
	CNN (Harbola and Coors, 2019)	0.6247	0.6200	0.6330	0.9415	0.9539	1.2675	1.1217	1.0288	1.1218	0.7483	0.8406	0.7885
	CNN-LSTM (Shen et al., 2022)	0.5794	0.5731	0.5994	0.8786	0.9133	1.1924	1.0737	0.9906	1.0614	0.6982	0.7857	0.7229
	LLConvLSTM	**0.5094**	**0.5011**	**0.5285**	**0.7723**	**0.8098**	**1.0481**	**0.9199**	**0.8556**	**0.9478**	**0.6215**	**0.6908**	**0.6835**
R^2	Naive Persistence (Potter and Negnevitsky, 2006)	0.9873	0.9824	0.9806	0.9291	0.9259	0.8583	0.8580	0.8997	0.8838	0.9528	0.9500	0.9734
	GBRT (Liao et al., 2021)	0.9875	0.9832	0.9812	0.9359	0.9307	0.8707	0.8679	0.9064	0.8944	0.9564	0.9521	0.9748
	k-NN (Kusiak and Zhang, 2010)	0.9898	0.9861	0.9825	0.9396	0.9279	0.8689	0.8658	0.9028	0.8932	0.9599	0.9581	0.9775
	RF (Lahouar and Slama, 2017)	0.9898	0.9862	0.9828	0.9405	0.9299	0.8720	0.8696	0.9052	0.8949	0.9609	0.9589	0.9779
	MLP (Velo et al., 2014)	0.9896	0.9856	0.9836	0.9405	0.9335	0.8757	0.8719	0.9083	0.8992	0.9612	0.9593	0.9787
	LSTM (Kisvari et al., 2021)	0.9893	0.9855	0.9835	0.9399	0.9329	0.8732	0.8708	0.9074	0.8967	0.9607	0.9575	0.9771
	CNN (Harbola and Coors, 2019)	0.9882	0.9837	0.9820	0.9331	0.9280	0.8608	0.8619	0.9020	0.8875	0.9562	0.9531	0.9744
	CNN-LSTM (Shen et al., 2022)	0.9898	0.9861	0.9838	0.9417	0.9340	0.8768	0.8735	0.9091	0.8992	0.9618	0.9591	0.9785
	LLConvLSTM	**0.9921**	**0.9894**	**0.9874**	**0.9550**	**0.9481**	**0.9049**	**0.9071**	**0.9322**	**0.9197**	**0.9698**	**0.9684**	**0.9808**

Table 7
The experimental results of our proposed method in different time steps. Best results are shown in bold.

Time step	One-step-ahead			
	MAE	MSE	RMSE	R^2
3	**0.1915**	0.1201	0.3466	0.9945
6	0.1919	**0.1199**	**0.3463**	**0.9948**
9	0.1925	0.1201	0.3466	0.9945

Table 8
The experimental results of our proposed method in different learning rates. Best results are shown in bold.

Learning rate	One-step-ahead			
	MAE	MSE	RMSE	R^2
0.01	0.2056	0.1403	0.3746	0.9939
0.005	0.2048	0.1330	0.3647	0.9942
0.0025	0.1960	0.1247	0.3531	0.9946
0.002	0.1932	0.1214	0.3484	**0.9947**
0.001	**0.1919**	**0.1199**	**0.3463**	0.9945
0.0009	0.1988	0.1241	0.3523	0.9946
0.0008	0.2049	0.1270	0.3564	0.9941
0.0007	0.2109	0.1355	0.3681	0.9941

Table 9
Comparison of predictive values of different modules. Best results are shown in bold.

Method	One-step-ahead				Two-step-ahead				Three-step-ahead			
	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE	R^2
ConvLSTM	0.2126	0.1491	0.3862	0.9932	0.3675	0.3868	0.6220	0.9820	0.5068	0.6550	0.8093	0.9691
ConvLSTM-RDCM	0.2009	0.1300	0.3606	0.9940	0.3551	0.3599	0.5999	0.9833	0.4881	0.6122	0.7824	0.9714
ConvLSTM-DCAM	0.1947	0.1257	0.3546	0.9942	0.3524	0.3556	0.5963	0.9836	0.4829	0.5982	0.7734	0.9719
LLConvLSTM	**0.1919**	**0.1199**	**0.3463**	**0.9945**	**0.3492**	**0.3446**	**0.5870**	**0.9839**	**0.4752**	**0.5798**	**0.7614**	**0.9729**

the model with the introduction of DCAM is decreased by MSE by 15.69%, 8.07% and 8.67% respectively. Introducing either module also reduces the MAE and RMSE. It is demonstrated that RDCM and DCAM outperform the benchmark model from the perspective of local and long-range spatial correlations focusing on wind speed information, respectively. With the simultaneous introduction of RDCM and DCAM, the MSE of the proposed approach are 0.1199, 0.3446 and 0.5798 respectively. Compared with the benchmark model, the MSE are reduced by 19.58%, 10.91% and 11.48% respectively. Similarly, the proposed approach outperforms the benchmark model in other indexes.

The results of ablation experiments indicate that models with the simultaneous introduction of RDCM and DCAM can significantly improve the accuracy of wind speed prediction. It is mainly because the local and long-range spatial features extracted from the original spatial correlations via the ConvLSTM encoder–decoder structure are not obvious. The importance of the LLConvLSTM model for extracting local and long-range spatial correlations to improve wind speed prediction is further demonstrated.

5. Conclusions

With the ever-increasing popularity of wind power generation, improving the accuracy of wind speed prediction is extremely critical for the operation and maintenance of wind power grid-connected systems. In this study, a novel ConvLSTM multi-step spatiotemporal wind speed prediction approach, namely, LLConvLSTM, is proposed to predict wind speed and its prediction performance is verified. The model adequately incorporates spatiotemporal information to improve the accuracy of wind speed forecasting. In particular, ConvLSTM encoder–decoder architecture is designed to extract spatiotemporal wind speed information for achieving end-to-end prediction. RDCM is applied to accurately characterize the local spatial correlations in the wind speed flows. DCAM enhances the capability to sense long-range spatial correlations in the wind speed flows. Extensive experimental validations are carried out on wind speed data from 253 virtual wind turbines. LLConvLSTM outperforms other existing models in all four evaluation metrics (MAE, MSE, RMSE, and R^2) in one-step-ahead, two-step-ahead, and three-step-ahead prediction over the whole year and each month. MSE of the proposed method is reduced by 40.91%, 34.70% and 33.51% respectively compared with Naive Persistence. Moreover, ablation experiments further verified the validity of designed components in the proposed method. Hyperparameters such as the time step and learning rate are also investigated.

This study cleverly utilized information from wind turbines located at regular positions to construct feature maps, enabling modeling the spatialtemporal correlations of wind speed flows. However, the existing method have limitations in dealing with turbines distributed in extremely irregular and scattered locations. Moreover, long-term wind speed prediction has not been considered. As one of future endeavor, a pervasive method will be designed to construct feature maps that can handle turbine data at arbitrary locations. This method aims to capture more comprehensive turbine information, thereby further improving the accuracy of spatialtemporal wind speed prediction. Secondly, in the current research field of wind speed prediction, models based on long-term forecasting (Malhan and Mittal, 2022; Duan et al., 2022) have shown the capability to accurately achieve multi-step wind speed prediction. Due to the influence of climate and seasonal variations, accurate long-term wind speed prediction (Cai et al., 2023; Ran et al., 2023) becomes a challenging task. Precise long-term forecasting is equally crucial for the planning and operation of the wind energy industry, facilitating the advancement and application of sustainable energy, particularly in the context of large-scale grid-connected wind power implementation. This study specifically examines wind speed prediction for one-step-ahead, two-step-ahead, and three-step-ahead. Based on the proposed method, another future work will develop modules dedicated to long-term forecasting. These modules aim to enhance the accuracy of wind speed predictions over longer time scales.

CRediT authorship contribution statement

Mei Yu: Conceptualization, Validation, Project administration, Funding acquisition. **Boan Tao:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Xuewei Li:** Validation, Resources, Writing – review & editing. **Zhiqiang Liu:** Conceptualization, Validation, Writing – review & editing. **Wei Xiong:** Conceptualization, Validation, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61976155).

References

Adnan, R.M., Mostafa, R.R., Dai, H.L., Heddam, S., Kuriqi, A., Kisi, O., 2023. Pan evaporation estimation by relevance vector machine tuned with new metaheuristic algorithms using limited climatic data. Eng. Appl. Comput. Fluid Mech. 17 (1), 2192258.

Adnan, R.M., Mostafa, R.R., Elbeltagi, A., Yaseen, Z.M., Shahid, S., Kisi, O., 2022. Development of new machine learning model for streamflow prediction: Case studies in Pakistan. Stoch. Environ. Res. Risk Assess. 1–35.

Adnan, R.M., Mostafa, R.R., Kisi, O., Yaseen, Z.M., Shahid, S., Zounemat-Kermani, M., 2021. Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization. Knowl.-Based Syst. 230, 107379.

Cai, Z., Dai, S., Ding, Q., Zhang, J., Xu, D., Li, Y., 2023. Gray wolf optimization-based wind power load mid-long term forecasting algorithm. Comput. Electr. Eng. 109, 108769.

Chang, C.Y., Lin, T.K., Lin, C.W., Cheng, H.T., 2022. Application of TransUNet for segmenting lung mass from chest X-ray image. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, pp. 175–176.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 764–773.

Daut, M.A.M., Hassan, M.Y., Abdullah, H., Rahman, H.A., Abdullah, M.P., Hussin, F., 2017. Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review. Renew. Sustain. Energy Rev. 70, 1108–1118.

Dolatabadi, A., Abdeltawab, H., Mohamed, Y.A.R.I., 2020. Hybrid deep learning-based model for wind speed forecasting based on DWPT and bidirectional LSTM network. IEEE Access 8, 229219–229232.

Donnelly, J., Abolfathi, S., Pearson, J., Chatrabgoun, O., Daneshkhah, A., 2022. Gaussian process emulation of spatio-temporal outputs of a 2D inland flood model. Water Res. 225, 119100.

Draxl, C., Clifton, A., Hodge, B.M., McCaa, J., 2015. The wind integration national dataset (wind) toolkit. Appl. Energy 151, 355–366.

Duan, Z., Liu, H., Li, Y., Nikitas, N., 2022. Time-variant post-processing method for long-term numerical wind speed forecasts based on multi-region recurrent graph network. Energy 259, 125021.

Erdem, E., Shi, J., 2011. ARMA based approaches for forecasting the tuple of wind speed and direction. Appl. Energy 88 (4), 1405–1414.

Ewees, A.A., Al-qaness, M.A., Abualigah, L., Abd Elaziz, M., 2022. HBO-LSTM: optimized long short term memory with heap-based optimizer for wind power forecasting. Energy Convers. Manage. 268, 116022.

Gan, Z., Li, C., Zhou, J., Tang, G., 2021. Temporal convolutional networks interval prediction model for wind speed forecasting. Electr. Power Syst. Res. 191, 106865.

Ghiasi, B., Noori, R., Sheikhian, H., Zeynolabedin, A., Sun, Y., Jun, C., Hamouda, M., Bateni, S.M., Abolfathi, S., 2022. Uncertainty quantification of granular computing-neural network model for prediction of pollutant longitudinal dispersion coefficient in aquatic streams. Sci. Rep. 12 (1), 4610.

Hao, J., Zhu, C., Guo, X., 2021. Short-term wind speed forecasting based on weighted spatial correlation and improved GWO-GBRT algorithm. In: 2021 IEEE 5th Conference on Energy Internet and Energy System Integration. EI2, IEEE, pp. 2685–2690.

Harbola, S., Coors, V., 2019. One dimensional convolutional neural network architectures for wind prediction. Energy Convers. Manage. 195, 70–75.

Hou, Q., Zhou, D., Feng, J., 2021. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13713–13722.

Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Hu, J., Shen, L., Sun, G., 2018a. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141.

Hu, Z., Turki, T., Phan, N., Wang, J.T., 2018b. A 3D atrous convolutional long short-term memory network for background subtraction. IEEE Access 6, 43450–43459.
Ikram, R.M.A., Dai, H.L., Al-Bahrani, M., Mamlooki, M., et al., 2022a. Prediction of the FRP reinforced concrete beam shear capacity by using ELM-CRFOA. Measurement 205, 112230.
Ikram, R.M.A., Dai, H.L., Ewees, A.A., Shiri, J., Kisi, O., Zounemat-Kermani, M., 2022b. Application of improved version of multi verse optimizer algorithm for modeling solar radiation. Energy Rep. 8, 12063–12080.
Ikram, R.M.A., Ewees, A.A., Parmar, K.S., Yaseen, Z.M., Shahid, S., Kisi, O., 2022c. The viability of extended marine predators algorithm-based artificial neural networks for streamflow prediction. Appl. Soft Comput. 131, 109739.
Jung, J., Broadwater, R.P., 2014. Current status and future advances for wind speed and power forecasting. Renew. Sustain. Energy Rev. 31, 762–777.
Khosravi, K., Rezaie, F., Cooper, J.R., Kalantari, Z., Abolfathi, S., Hatamiafkoueieh, J., 2023. Soil water erosion susceptibility assessment using deep learning algorithms. J. Hydrol. 618, 129229.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Kisvari, A., Lin, Z., Liu, X., 2021. Wind power forecasting–A data-driven method along with gated recurrent neural network. Renew. Energy 163, 1895–1909.
Kusiak, A., Zhang, Z., 2010. Short-horizon prediction of wind power: A data-driven approach. IEEE Trans. Energy Convers. 25 (4), 1112–1122.
Lahouar, A., Slama, J., 2017. Hour-ahead wind power forecast based on random forests. Renew. Energy 109, 529–541.
Li, J., Huang, L., Wei, Z., Zhang, W., Qin, Q., 2021. Multi-task learning with deformable convolution. J. Vis. Commun. Image Represent. 77, 103109.
Liao, H., Mei, H., Hu, G., Wu, B., Wang, Q., 2021. Machine learning strategy for predicting flutter performance of streamlined box girders. J. Wind Eng. Ind. Aerodyn. 209, 104493.
Lin, Z., Li, M., Zheng, Z., Cheng, Y., Yuan, C., 2020. Self-attention convlstm for spatiotemporal prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 07. pp. 11531–11538.
Liu, H., Tian, H.q., Chen, C., Li, Y.f., 2013. An experimental investigation of two Wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization. Int. J. Electr. Power Energy Syst. 52, 161–173.
Liu, W., Wang, Y., Zhong, D., Xie, S., Xu, J., 2022. ConvLSTM network-based rainfall nowcasting method with combined reflectance and radar-retrieved wind field as inputs. Atmosphere 13 (3), 411.
Liu, Z., Yang, B., Duan, G., Tan, J., 2020. Visual defect inspection of metal part surface via deformable convolution and concatenate feature pyramid neural networks. IEEE Trans. Instrum. Meas. 69 (12), 9681–9694.
Malhan, P., Mittal, M., 2022. A novel ensemble model for long-term forecasting of wind and hydro power generation. Energy Convers. Manage. 251, 114983.
Memarzadeh, G., Keynia, F., 2020. A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets. Energy Convers. Manage. 213, 112824.
Niu, Z., Zhong, G., Yu, H., 2021. A review on the attention mechanism of deep learning. Neurocomputing 452, 48–62.
Nor, K.M., Shaaban, M., Rahman, H.A., 2014. Feasibility assessment of wind energy resources in Malaysia based on NWP models. Renew. Energy 62, 147–154.
Paul, S.K., Jana, S., Bhaumik, P., 2020. A multivariate spatiotemporal spread model of COVID-19 using ensemble of ConvLSTM networks. MedRxiv 2020–04.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.
Peláez-Rodríguez, C., Pérez-Aracil, J., Fister, D., Prieto-Godino, L., Deo, R., Salcedo-Sanz, S., 2022. A hierarchical classification/regression algorithm for improving extreme wind speed events prediction. Renew. Energy 201, 157–178.
Potter, C.W., Lew, D., McCaa, J., Cheng, S., Eichelberger, S., Grimit, E., 2008. Creating the dataset for the western wind and solar integration study (USA). Wind Eng. 32 (4), 325–338.
Potter, C.W., Negnevitsky, M., 2006. Very short-term wind forecasting for tasmanian power generation. IEEE Trans. Power Syst. 21 (2), 965–972.
Prema, V., Bhaskar, M.S., Almakhles, D., Gowtham, N., Rao, K.U., 2021. Critical review of data, models and performance metrics for wind and solar power forecast. IEEE Access.
Ran, M., Huang, J., Qian, W., Zou, T., Ji, C., 2023. EMD-based gray combined forecasting model-application to long-term forecasting of wind power generation. Heliyon.
Scheepens, D.R., Schicker, I., Hlaváčková-Schindler, K., Plant, C., 2023. Adapting a deep convolutional RNN model with imbalanced regression loss for improved spatio-temporal forecasting of extreme wind speed events in the short to medium range. Geosci. Model Dev. 16 (1), 251–270.
Shen, Z., Fan, X., Zhang, L., Yu, H., 2022. Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network. Ocean Eng. 254, 111352.
Shen, N., Wang, Z., Li, J., Gao, H., Lu, W., Hu, P., Feng, L., 2023. Multi-organ segmentation network for abdominal CT images based on spatial attention and deformable convolution. Expert Syst. Appl. 211, 118625.
Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Adv. Neural Inf. Process. Syst. 28.
Sibtain, M., Bashir, H., Nawaz, M., Hameed, S., Azam, M.I., Li, X., Abbas, T., Saleem, S., 2022. A multivariate ultra-short-term wind speed forecasting model by employing multistage signal decomposition approaches and a deep learning network. Energy Convers. Manage. 263, 115703.
Tang, X., Meng, F., Ma, J., Zhang, X., Liu, F., Peng, Q., Jiao, L., 2020. Hyperspectral image classification based on multiscale spatial and spectral feature network. In: IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 838–841.
Velo, R., López, P., Maseda, F., 2014. Wind speed estimation using multilayer perceptron. Energy Convers. Manage. 81, 1–9.
Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C., 2019. Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
Wei, X., Wei, Y., Lu, X., 2022. RMDC: Rotation-mask deformable convolution for object detection in top-view fisheye cameras. Neurocomputing 504, 99–108.
Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. CBAM: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.
Wu, Q., Feng, D., Cao, C., Zeng, X., Feng, Z., Wu, J., Huang, Z., 2021. Improved mask R-CNN for aircraft detection in remote sensing images. Sensors 21 (8), 2618.
Wu, P., Huang, H., Qian, H., Su, S., Sun, B., Zuo, Z., 2022a. SRCANet: Stacked residual coordinate attention network for infrared ship detection. IEEE Trans. Geosci. Remote Sens. 60, 1–14.
Wu, Q., Zheng, H., Guo, X., Liu, G., 2022b. Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks. Renew. Energy 199, 977–992.
Xiao, Y., Yin, H., Zhang, Y., Qi, H., Zhang, Y., Liu, Z., 2021. A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction. Int. J. Intell. Syst. 36 (5), 2036–2057.
Yang, W., Tian, Z., Hao, Y., 2022. A novel ensemble model based on artificial intelligence and mixed-frequency techniques for wind speed forecasting. Energy Convers. Manage. 252, 115086.
Yeganeh-Bakhtiary, A., EyvazOghli, H., Shabakhty, N., Kamranzad, B., Abolfathi, S., 2022. Machine learning as a downscaling approach for prediction of wind characteristics under future climate change scenarios. Complexity 2022.
Yesilbudak, M., Sagiroglu, S., Colak, I., 2013. A new approach to very short term wind speed prediction using k-nearest neighbor classification. Energy Convers. Manage. 69, 77–86.
Yu, S., Vautard, R., 2022. A transfer method to estimate hub-height wind speed from 10 meters wind speed based on machine learning. Renew. Sustain. Energy Rev. 169, 112897.
Yuan, X., Chen, C., Lei, X., Yuan, Y., Muhammad Adnan, R., 2018. Monthly runoff forecasting based on LSTM–ALO model. Stoch. Environ. Res. Risk Assess. 32, 2199–2212.
Yunus, K., Thiringer, T., Chen, P., 2015. ARIMA-based frequency-decomposed modeling of wind speed time series. IEEE Trans. Power Syst. 31 (4), 2546–2556.
Zeng, W., Lin, C., Liu, K., Lin, J., Tung, A.K., 2021. Modeling spatial nonstationarity via deformable convolutions for deep traffic flow prediction. IEEE Trans. Knowl. Data Eng..
Zhang, Y., Li, Y., Zhang, G., 2020. Short-term wind power forecasting approach based on Seq2Seq model using NWP data. Energy 213, 118371.
Zhang, Q.L., Yang, Y.B., 2021. Sa-net: Shuffle attention for deep convolutional neural networks. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 2235–2239.
Zhang, Y., Zhao, Y., Shen, X., Zhang, J., 2022. A comprehensive wind speed prediction system based on Monte Carlo and artificial intelligence algorithms. Appl. Energy 305, 117815.
Zhao, S., Zhang, S., Lu, J., Wang, H., Feng, Y., Shi, C., Li, D., Zhao, R., 2022. A lightweight dead fish detection method based on deformable convolution and YOLOV4. Comput. Electron. Agric. 198, 107098.
Zhu, Q., Chen, J., Zhu, L., Duan, X., Liu, Y., 2018. Wind speed prediction with spatio–temporal correlation: A deep learning approach. Energies 11 (4), 705.
Zhu, X., Hu, H., Lin, S., Dai, J., 2019. Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9308–9316.