Renewable Energy 198 (2022) 267–282

Contents lists available at ScienceDirect

Renewable Energy

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

Wind speed forecasting with correlation network pruning and augmentation: A two-phase deep learning method

Yang Yang [a], Jin Lang [b,*], Jian Wu [c], Yanyan Zhang [d], Lijie Su [b], Xiangman Song [b]

[a] National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang, 110819, China
[b] Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education, 110819, China
[c] Liaoning Key Laboratory of Manufacturing System and Logistics Optimization, Northeastern University, Shenyang, 110819, China
[d] Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Northeastern University, Shenyang, 110819, China

ARTICLE INFO

Keywords:
Two-phase deep learning method
Wind speed prediction
Cross-correlation function and correlation network
Network pruning and network augmentation
Fractional quadratic optimization

ABSTRACT

To ensure the operational reliability of power systems, it is important for wind speed signal forecasting systems of wind turbines to be efficient, accurate and stable. This paper proposes a two-phase deep learning structure with network augmentation and pruning. By introducing the cross-correlation and quasi-convex optimization, a fractional quadratic programming problem and related convex optimization models are constructed to generate the augmented data for this proposed internal network; by pruning weakly correlated convolution channels, the redundant features of its external network are reduced. Furthermore, the closed-form solution of the convex optimization model is derived, which reduces the computational complexity considerably from $O(n^*\log(2N))$ to $O(n)$. The proposed approach has been extensively validated using the real data of the wind farm in China. The results of the numerical experiments demonstrate that the proposed method achieves the superior performance in the training flexibility, model accuracy, stability, and interpretability.

1. Introduction

Energy system decarburization has been a critical measure for combating global climate change and the fossil energy crisis [1]. According to International Renewable Energy Agency data published in 2021, the global installed wind-generation capacity reached 733 GW at the end of 2020, representing an increase of 18% compared to that in 2019 [2]. Apparently, wind power is being presented as a solution to the demands for carbon neutral energy development and is highly anticipated and persistently supported by governments worldwide [3,4]. Wind farms build wind turbines (WTs) to convert wind energy to electrical energy. To effectively utilize wind energy, their control systems need to be fast, robust and reliable [5]. Various control methods have been proposed to optimize control systems based on wind turbine signal prediction (WTSP) corresponding to the resulting energy [6,7]. However, the nonstationarity and uncertainty of wind changes in nature pose a challenge to the control of wind farms. Specifically, the control system of a wind farm requires not only accurate and reliable forecasting capabilities to consider short-term random fluctuations and long-term uncertain periodicity of wind energy but also high efficiency and stable operation with limited computing power. These capabilities can promote the carbon-neutral energy transition, increase the utilization of renewable clean energy sources such as wind and solar energy, and enhance the safety of the grid.

In recent years, many methods have been proposed to study the periodicity and variability of wind based on temporal and spatial correlations. Nevertheless, leveraging both temporal correlations and spatial correlations is still one of the most challenging areas in wind speed prediction [8,9]. Data augmentation can reduce overfitting to improve the accuracy and stability of wind speed prediction [10–13]. A sparse neural network [14] and pruning neural network [15] can improve the reliability of the model and reduce redundant parameters while still maintaining good performance in large-scale, wind speed spatiotemporal prediction. When the abovementioned methods are integrated into a framework, the intractable problem of balancing training time, training data scale, model complexity, model accuracy and

Abbreviations: CNN, Convolutional neural network; LSTM, Long short-term memory; SVM, Support vector machine; RNN, Recurrent neural network; WT, Wind turbine; NWP, Numerical weather prediction; WTSF, Wind turbine signal forecasting; WSSF, Wind speed signal forecasting; WPSF, Wind power signal forecasting.

* Corresponding author.
E-mail addresses: yang_cmu@icloud.com (Y. Yang), langjin@ise.neu.edu.cn (J. Lang), wujian@mail.neu.edu.cn (J. Wu), zhangyanyan@ise.neu.edu.cn (Y. Zhang), sulijie@ise.neu.edu.cn (L. Su), songxiangman@ise.neu.edu.cn (X. Song).

https://doi.org/10.1016/j.renene.2022.07.125
Received 9 June 2021; Received in revised form 17 April 2022; Accepted 25 July 2022
Available online 10 August 2022

stability for short-term wind speed prediction exists.

Physical modelling, statistical modelling and deep learning methods are current controversial topics. Physical models, including numerical weather prediction (NWP), use hydro and thermodynamic models to incorporate weather data into the prediction of wind speed with certain initial values and boundary conditions [16–18]. NWP has been extensively introduced as augmented information for data correction in wind forecasting systems [19–21].

Statistical methods generally concern the correlation between wind speed sequences and explanatory variables [22,23]. Correlated data and correlation networks based on correlation coefficients, such as Pearson correlation [24] and Kendall correlation [25], are widely utilized to improve forecasting performance. In addition, when exploring the ambiguous data association, the data-driven fuzzy models [23,26,27] and statistical probabilistic models [28–30] are the effective choices to improve modelling robustness by constructing augmented information.

Deep neural networks stand out in terms of extracting the nonlinear correlated characteristics of wind speed [31,32]. Long short-term memory (LSTM)-based and convolutional neural network (CNN)-based models are two kinds of classical deep neural networks. CNN-based models show good generalization capacity for spatial features of short-term wind speed prediction [33,34], while LSTM-based methods and their variants stand out for solving for the long-term dependence [35,36]. Their results for determining temporal and spatial wind speed correlations revealed that deep neural networks have a better learning ability than shallow models [37]. The data representation of neural networks can not only inherit the relationship from the physical and statistical methods but also absorb more potentially correlated data into the model [38–40]. Especially for missing signal processing, the correlation of sequences in the data representation supports the optimization of deep neural networks [41]. Therefore, neural networks can be pruned increasingly quickly by removing connections or nodes to reduce the redundancy of model parameters [42].

Although deep learning methods have been progressively applied to wind speed signal forecasting (WSSF) systems, the model's interpretability and training flexibility still fall short regarding practical challenges, such as intermittent winds and hardware limitations. The two-phase deep learning method based on a hybrid network has achieved high performance in actual wind power operation [43]. Hybrid networks combine the advantages of LSTM-based networks and CNN-based networks on spatiotemporal series problems [44,45]. Sequence-to-sequence (Seq2Seq) is a hybrid network based on encoder-decoder machine translation processing that maps an input of a sequence to an output of a sequence with a tag and attention value [46]. The Seq2Seq model is suitable for extracting complex features from high-dimensional data and effectively capturing the future trend of sequences. For wind speed sequential prediction, the Seq2Seq model effectively avoids the random, short-term interference and long-range dependence problems caused by periodicity [47,48].

On account of the abovementioned strengths, a complex, augmented and pruned deep neural network based on an encoder-decoder structure is proposed for short-term WSSF with the following perks:

1. The parameter optimization of the deep learning model becomes more stable and efficient after inputting the correlated spatiotemporal data into the encoder network.
2. The network augmentation and network pruning improve the adaptability of the deep learning framework to various hardware conditions.
3. In view of the ubiquitous bad data phenomenon in power systems, the bidirectional encoder-decoder structure shows better visualization and interpretability, which improves the security and reliability of power systems.

The remainder of this paper is organized as follows: Section 2 provides the preliminaries of the cross-correlation function and pruning of CNNs. Sections 3 and 4 are theory sections. In Section 3, an optimized correlation model and the constrained CNNs are established for data augmentation and network pruning, respectively, of WTSP. In Section 4, the two-phase encoder-decoder structure is proposed to integrate the augmented and pruned networks with a Seq2Seq network. In Section 5, two numerical experiments are designed to show the training performance and forecasting performance of the proposed method. The statistical results are discussed in Section 6, and Section 7 concludes the paper.

2. Preliminaries

2.1. Cross-correlation function

Cross-correlation [49] is a measure of the similarity between the two series shown in Formula (1-2), where N is the number of data points in each data series, x_i is the ith data point of the first data series, y_i is the ith data point of the second data series, and $r_{XY}(\ell)$ is a correlation with the signal lag ℓ. $\rho_{XY}(\ell)$ is the most commonly employed version of the cross-correlation. As a function of the displacement of one relative to the other, cross-correlation has a vital role in research on the transformation of wind energy and the dispatching of electricity [50,51]. By considering different degrees of wind speed correlation, researchers have explored their impacts on the reliability of incorporating wind energy conversion systems [52,53]. For time series analysis, it is common practice to normalize the cross-correlation function to obtain a time-dependent, Pearson product-moment correlation coefficient to enhance forecasting performance in accuracy and stability [54].

$$r_{XY}(\ell) = \sum_{i=1}^{N} x_i y_{i-\ell} \tag{1}$$

$$\rho_{XY}(\ell) = \left|\frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}\right| = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_{i-\ell}-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_{i-\ell}-\bar{y})^2}} \tag{2}$$

where cov represents the covariance of the two samples, and σ represents the variance of the sample.

2.2. Pruning CNNs

Among the many parameters in neural networks, certain parameters are redundant and do not contribute greatly to the output [55]. CNNs have demonstrated extraordinarily good performance for large-scale, spatiotemporal data streams. However, with an increase in the size of the networks, CNNs cannot be widely deployed to devices with limited computing power. The emerging topic of CNN pruning not only strives to address this problem but also enhances the interpretability and visibility of neural networks [56].

Considering the fully connected CNNs for the feature extraction of a time series, their datasets are stored on multiple channels in a decentralized manner. Assume that each CNN consists of N channels and that the dataset of the kth channel is given by $D_k = \{(X_k, Y_k), k = 1, \ldots, N\}$, where $X_k = [x_1, x_2, \ldots, x_{T_k}]^T \in R^{T_k \times 1}$, and $Y_k \in R^{T_k \times m}$. For the kth channel, the CNN model is transformed into matrix form

$$f_k = H_k \beta_k \tag{3}$$

where the hidden matrix $H_k = [h_1, h_2, \ldots, h_L] \in R^{T_k \times L}$ can be described by

$$H_k = \sigma(X_k W + b) \tag{4}$$

where σ is a type of activation function, and $W = [w_1, w_2, \ldots, w_L]$ and $b = [b_1, b_2, \ldots, b_L]$ are the input weights and bias matrix, respectively.

To prune the input channels from N to the required N_0 ($0 \le N_0 \le N$)

while minimizing reconstruction error [57], output weights $\beta_k = [\beta_1, \beta_2, \ldots, \beta_L]^T \in R^{L\times m}$ are evaluated by the standard, regularized, least squares problem with the $L0$-norm penalty constraint.

$$\widehat{\beta}_k = \underset{\beta_k}{\operatorname{argmin}} \frac{1}{2}\|Y_k - H_k\beta_k\|_2^2 \tag{5}$$

$$s.t. \quad \|\beta_k\|_0 \le N_0 \tag{6}$$

where $\|u\|_p = (\sum_{i=1}^{N}|u_i|)^{1/p}$ is the standard L_p-norm.

3. Augmented and pruned networks for wind speed signal forecasting

With the aim of augmented data representation and the pruned CNN for the wind speed signal based on the correlation analysis, an optimized correlation model for corpus augmentation and a CNN pruning model for spatiotemporal feature extraction are established. A fast, two-phase encoder-decoder structure is proposed for integrating an augmented, pruned CNN and LSTM-based network, as shown in Fig. 1. The spatiotemporal features of the historical signal data of WTs are extracted by CNNs, and then the networks are pruned and sorted based on the correlation between two sequences. The trained CNNs are then stored in a sparse data structure. When the WSSF model is trained online, the training data and associated pretrained convolution layers are concatenated into the Seq2Seq structure to form the two-phase encoder-decoder network. Fig. 2 presents a schematic of the proposed two-phase encoder-decoder networks for the sequential point forecaster with a data generator of the augmented samples.

3.1. Optimized correlation model for data augmentation

Recent studies noticed that the cross-correlation function contributes to the stability and precision of deep neural networks due to its equivalence between cross-correlation and convolution in mathematics [37, 58,59]. The augmented corpus and networks are helpful to restrict the training scope of neural networks, reduce the vanishing gradients of deep learning models, and enhance learning stability with more spatiotemporal periodic data [52–60]. Fig. 3 illustrates the wind speed variation and correlation of the WTs over two days; (b) shows the imaged data of (a), and (c) shows the correlation among wind speed signals. It is efficient to identify less important samples for convolutional sampling based on the weak correlation among the training samples (light coloured areas in Fig. 3(c)).

Let $X = [x_1, \ldots, x_N]^T \in \mathbb{R}^{N\times 1}$ and $Y = [y_{1-\ell}, \ldots, y_{N-\ell}]^T \in \mathbb{R}^{N\times 1}$ be two time series. X is the recent short-term wind speed series of the target wind turbine, and Y is the correlated historical data, which records the continuous sampling values of wind speed from the same or another wind turbine. Suppose the sample values at some point in sequence X are missing or need to be predicted. Let $X^* = \{x_k\} \subseteq X$ be a collection of the missing data, where k is the serial number of the unknown points. Assume that the time series X and Y are linearly correlated. To pursue the maximum linearity, an optimized correlation model (OCM) with linear constraints is defined as follows:

$$\begin{aligned} \max_{X^*} \quad & f(X^*) = \left|\frac{\operatorname{cov}(X,Y)}{\sigma_X\sigma_Y}\right| \\ s.t. \quad & AX^* \le 0 \end{aligned} \tag{7}$$

where $f(X^*)$ is an optimization function to maximize the linearity between two time series. $\left|\frac{\operatorname{cov}(X,Y)}{\sigma_X\sigma_Y}\right|$ represents the absolute value of Pearson correlation. In terms of the WSSF, the objective function in Eq. (7) represents the correlation between the current wind speed series and the historical series, which is usually derived from the short-term temporal and spatial relationship and the long-term periodic relationship between two time series. The constraints on the variables are derived from existing knowledge of the wind speed and WTs, including the physical law of nature, operating rules of the power grid, and performance design of the WTs.

By introducing the spatial correlation of multilocation time series and natural constraints, the Pearson product moment correlation coefficient is firstly transformed into a fractional quadratic programming model. Furthermore, the novel fractional quadratic programming is adapted to generate the knowledge-based corpus to guide the deep neural networks for short-term WSSF. However, the solution of the fractional quadratic programming is usually complicated. Therefore, the purpose of this paper is to construct an equivalent convex optimization model for this mathematical programming model, which can obtain efficient and stable numerical solutions.

It is challenging to present efficient solutions to the abovementioned nonlinear fractional programming problems [61]. Based on the research of Zhang et al. [62], the OCM can be equivalently written as a quadratic fractional programming problem (QFP):

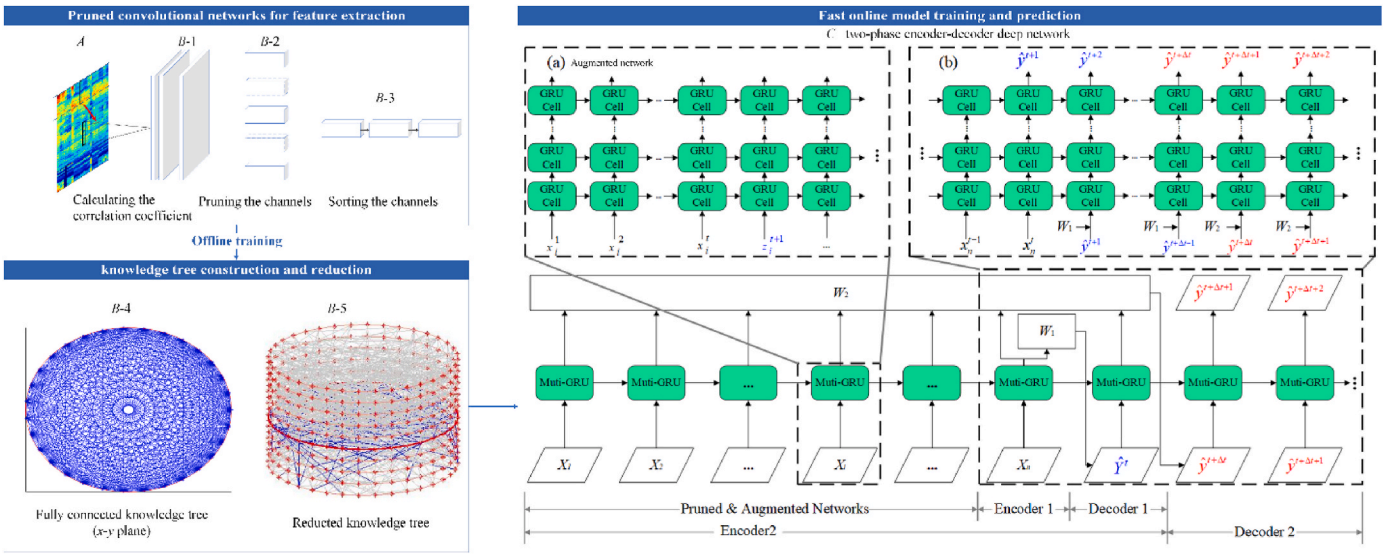

Fig. 1. Workflow of fast, two-phase encoder-decoder networks for integrating pruned CNN-based and LSTM-based networks.

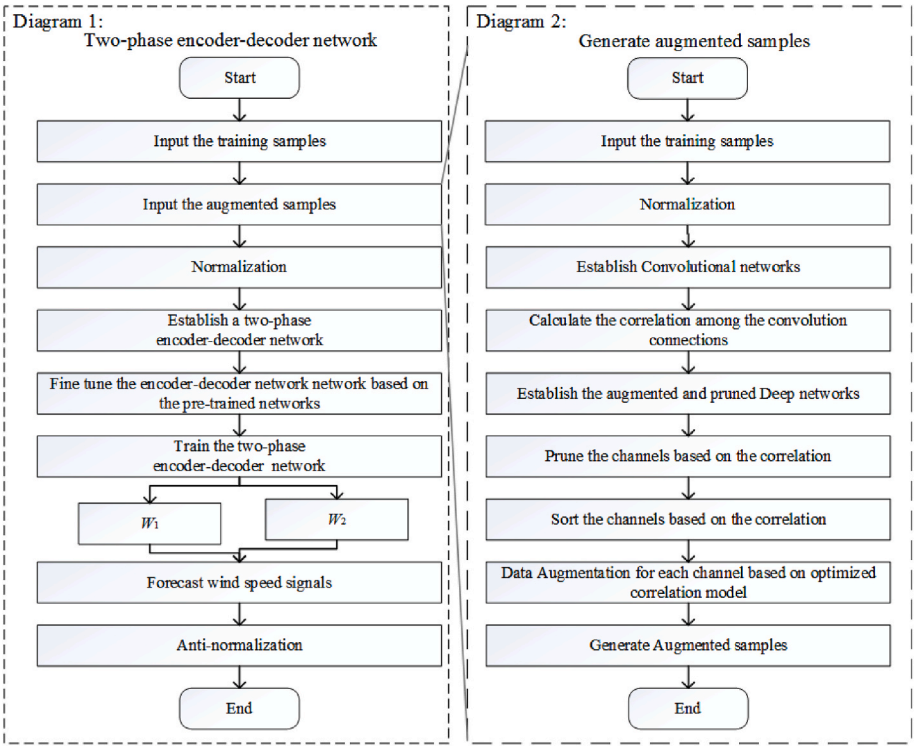

Fig. 2. Schematic of the proposed sequential point forecaster and generation process of the augmented samples for WSSF.

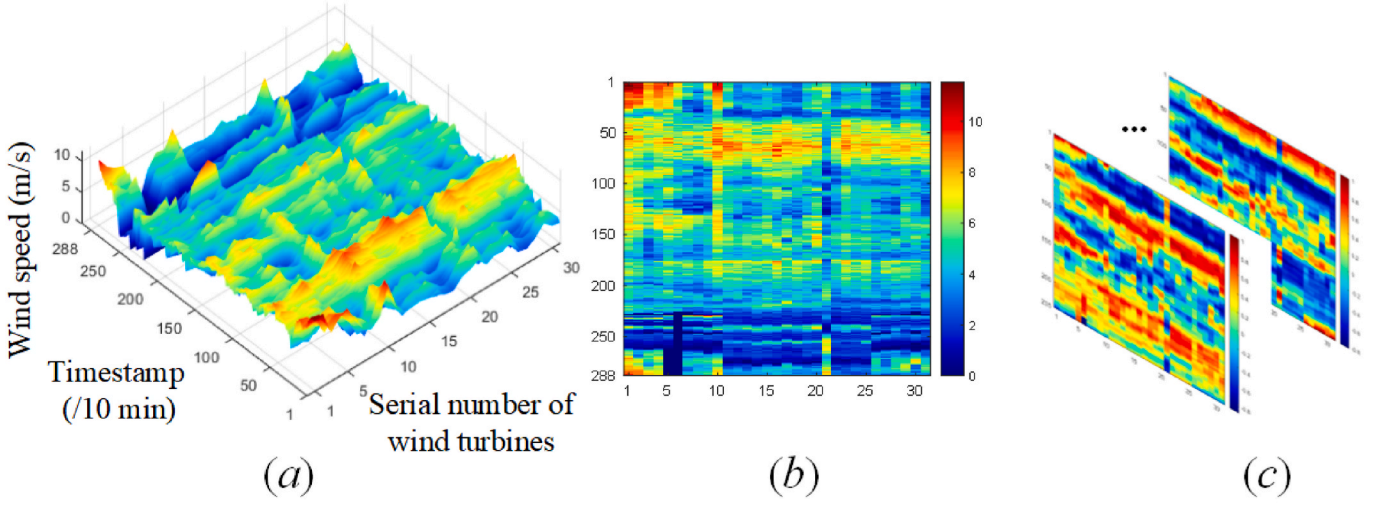

Fig. 3. Two-dimensional features of training data: (a) space-time wind speed signals, (b) imaged data of (a), and (c) correlation among wind speed signals with a one-dimensional convolutional kernel.

$$\max_{X^*} \; f_0(X^*) = \frac{1}{n^2\sigma_Y^2} \frac{\left(\sum_{i=1}^{n}\sum_{j>i}^{n}(x_i - x_j)(y_i - y_j)\right)^2}{\sum_{i=1}^{n}\sum_{j>i}^{n}(x_i - x_j)(x_i - x_j)} \tag{8}$$

$$s.t. \qquad AX^* \le 0$$

where $f_0(X^*)$ is an optimization function equivalent to $f(X^*)$, namely the squared function $f(X^*)$.

A quasi-convex optimization technique is employed to solve the QFP problem as follows: Furthermore, the proposed QFP problem is transformed into a family of semidefinite programming (SDP) problems [63] with the bisection method to solve the convex subproblem, as shown in Formulas (9-11). Furthermore, it is proved that the global optimal solution exhibits a closed-form solution for the QFP model and the corresponding convex model with a complexity of O(1) in each dimension, which accelerates the generation of the augmented data from traditional O(n*log(2N)) to O(n). The expression is shown in Formulas (12-22), and the relevant proof is shown in Appendix A.

$$f_0(x) = \frac{p(x)}{q(x)} \tag{9}$$

$$\varphi_t(x) = p(x) - tq(x) \tag{10}$$

$$\begin{aligned} \max_{X^*} \quad & g(X^*) = X^{*T}AX^* + BX^* + C \\ s.t. \quad & DX \le 0 \end{aligned} \tag{11}$$

$$\widehat{x}_i = -\frac{b_\rho}{2a_\rho}, \tag{12}$$

$$a_\rho = m^4 c_4 c_6 \rho^2 - c_3^2, \tag{13}$$

$$b_\rho = 2c_1 c_3 - m^4 c_4 c_7 \rho^2, \tag{14}$$

$$\rho^2 = \frac{4c_1 c_3 c_7 - 4c_1^2 c_6 - 4c_3^2 c_5}{m^4 c_4 c_7^2 - 4m^4 c_4 c_5 c_6}, \tag{15}$$

$$c_1 = (m-1)^2 c_2 + \sum_{i=1}^{m-1}[(x_i - x_m)y_i], \tag{16}$$

$$c_2 = \mathrm{cov}(X', Y'), \tag{17}$$

$$c_3 = \sum_{i=1}^{m-1}(x_i - x_m), \tag{18}$$

$$c_4 = S(X)^2 \tag{19}$$

$$c_5 = S(Y')^2, \tag{20}$$

$$c_6 = \frac{m-1}{m^2}, \tag{21}$$

$$c_7 = \frac{2}{m^2 \sum_{i=1}^{m-1} Y_i}, \tag{22}$$

Here, m denotes the lengths of the linearly correlated variables X and Y. Let $X' = [x_1, \ldots, x_{m-1}]^T$ $Y' = [y_1, \ldots, y_{m-1}]^T$, and x_m be unknown data of variable X; here, S represents the standard deviation.

An illustrative example is shown in Fig. 4 to explain the data complement and augmentation capabilities of the OCM. Assume that (X, Y) is a pair of correlated time series with the unknown points x_i^* and y_i^* shown in Fig. 4. Combining the autocorrelation and cross-correlation of sequences (X, Y) with the constraints on variables derived from the existing knowledge, the unknown variables in sequences (X, Y) can be completed by the proposed QFP problem based on the OCM. For WTSP problems, such as WSSF and wind power signal forecasting (WPSF), the wind speed signals of WTs at adjacent regions are correlated in a physical mechanism. Due to the long-term and short-term periodicity, changes in the physical properties of the wind produce changes in the forecasting model over time. Therefore, the wind speed changes randomly and variously.

3.2. Pruning CNNs based on data correlation

Considering the high autocorrelation and cross-correlation among the wind speed signals of the WTs, weakly correlated channels in fully connected CNNs are pruned to reduce the redundant parameters. These channels are sorted by correlation and are reconstructed into a pruned network as the input features, as shown in Fig. 5.

The fully connected CNNs and the connections between correlated sequences are calculated offline using the historical data. The historical dataset is referred to as the support set. From stage A to stage B, the convolutional layers are trained with a one-dimensional convolutional kernel between historical WT signals and recent WT signals. The weight vector of the convolutional kernel is the same as that of the OCM. In addition, the correlation coefficient between two sequences is calculated. From stage B to stage C, the channels in the convolutional layers are pruned based on the correlation threshold. The remaining channels are then sorted according to the value of the correlation coefficient. In addition, due to the autocorrelation of wind speed series, using the fine-tuning method for the proposed networks can effectively introduce the pruning CNN from the previous training moment into the current online training process.

The convolutional connections of the training samples are then restricted with a threshold of the correlation coefficient,

$$\beta_i \in \{\beta_i = 0 | \rho(X_i, X_{t+\Delta t}) \le \theta_0 \quad \}, \tag{23}$$

where the training sample $X_{t+\Delta t}$ represents the predicted data.

After the process of pruning the weak channels of the CNNs with the lower bound of the correlation coefficient, the pruned networks are taken as the input features for the encoder-decoder structure in Section 4.

4. Two-phase encoder-decoder networks based on augmented and pruned correlation networks

A novel, two-phase encoder-decoder structure is proposed to hybridize the spatial networks and temporal networks. The internal networks adopt the Seq2Seq model for the classical sequential forecasting tasks, and the external networks collect the correlated sequences from the CNNs to provide auxiliary input for the internal networks. The Seq2Seq model [64] is a widely utilized encoder-decoder structure for sequential forecasting tasks. Fig. 6 shows a classical structure of the Seq2Seq models based on LSTM units.

The Seq2Seq model is a conditional language model evolved from recurrent neural network (RNN)-based encoder-decoder frameworks [45]. The encoder process generally utilizes a stack of several recurrent units that receive a single element of the input sequence, which collects information for that element and propagates it forward. Vector W is then output as a semantic representation vector of the input sequence. The decoder is responsible for generating the specified sequence according to the semantic vector. The output of the previous moment will be

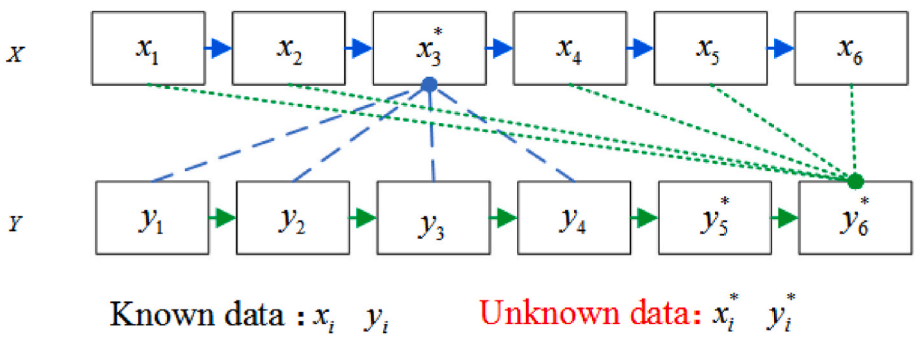

Fig. 4. Data augmentation based on the cross-correlation.

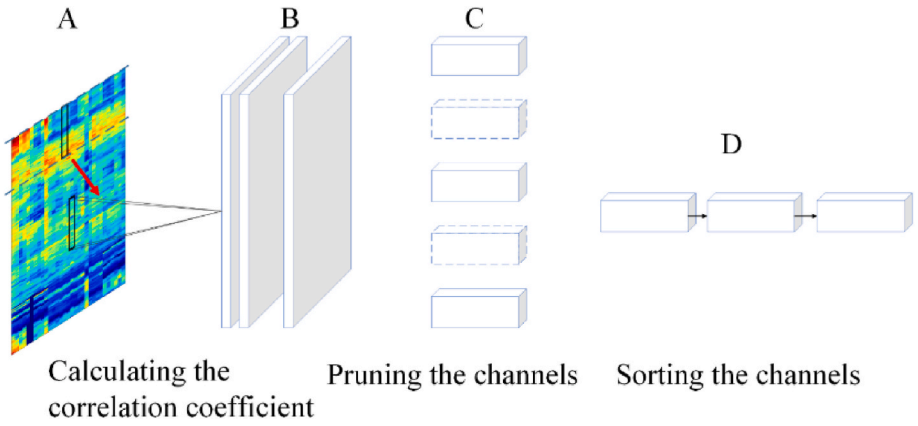

Fig. 5. Pruning and transforming the fully connected CNNs into the input networks of the Seq2Seq networks.

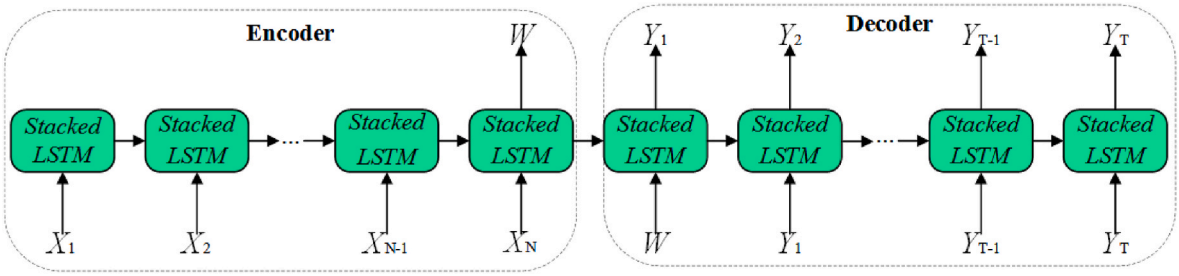

Fig. 6. Classical Seq2Seq structure.

utilized as the input of the current moment; the semantic vector W is only involved in the operation as the initial state; and subsequent operations are unrelated to the semantic vector W.

For short-term wind speed prediction, a pruned and augmented CNN-LSTM-LSTM (PACNN)-LSTM-LSTM) model is proposed as follows: The model consists of two-phase encoder-decoder models, as shown in Fig. 7. The inner structure (encoder 1 and decoder 1) is the standard encoder-decoder framework. Encoder 1 is a time series model that extracts short-term features based on the autocorrelation of recent wind speed data, while encoder 2 is a spatial model that can extract multimodal historical signals correlated to recent data. Decoder 2 can give the predicted sequence $\widehat{Y}_{t+\Delta t+n} = (\hat{y}^{t+\Delta t+1}, \ldots, \hat{y}^{t+\Delta t+n})$ beyond the prediction time horizon at time t.

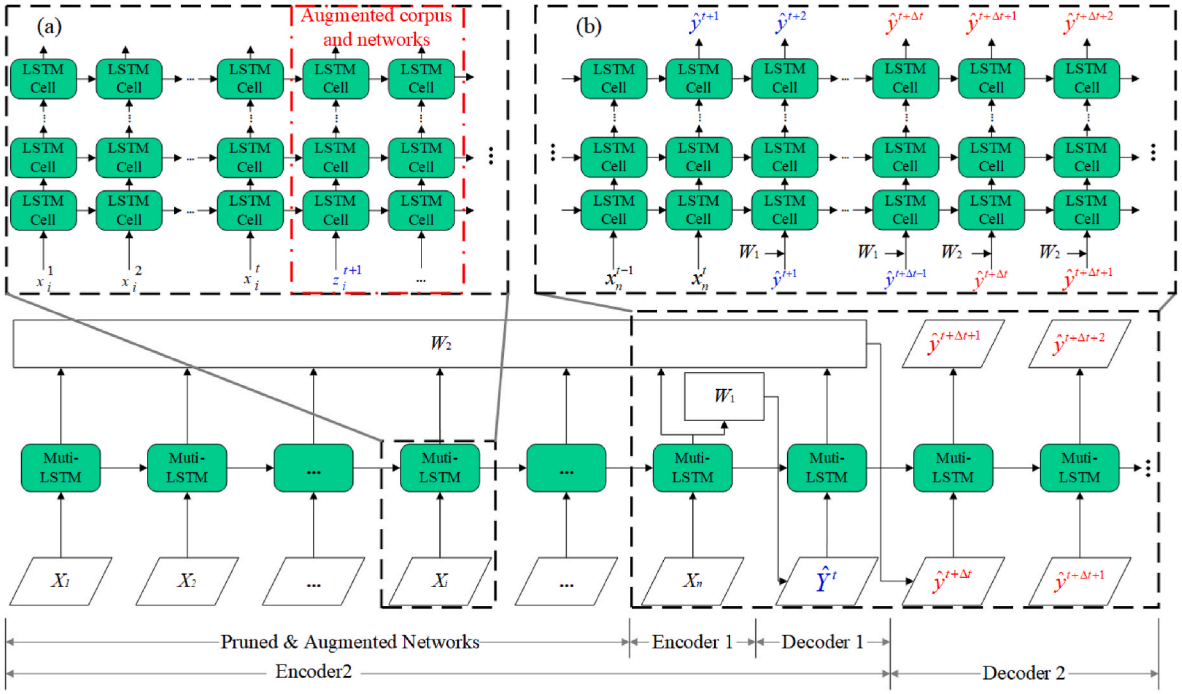

Fig. 7. Structure of the two-phase encoder-decoder model: (a) augmented network for a data sample in phase A and (b) online forecasting process for phase B.

The proposed encoder 1 has three functions in WSSF applications. Generally, the output $\widehat{Y}_{t+\Delta t} = (\widehat{y}^{t+1}, ..., \widehat{y}^{t+\Delta t-1})$ of decoder 1 is the set of sequential predictions within the prediction time horizon Δt. The data correction problems [14,20] of the WTSP can be generalized to $\widehat{Y}_{t+\Delta t} = (..., \widehat{y}^{t}, \widehat{y}^{t+1}, ..., \widehat{y}^{t+\Delta t-1}, \widehat{y}^{t+\Delta t}, ...)$, which provides the data correction before and after the predicted period. Therefore, the internal encoder-decoder network is established to be a bidirectional data correction network.

Encoder 2 consists of a pruned network and augmented network, encoder 1 and decoder 1, respectively. The pretrained CNN layers that correlate with the recent sequence are selected, and each channel of the layers is augmented based on the OCM, which forms the pruned and augmented networks in Fig. 7. The input vector X_1, …, $X_{n\text{-}1}$ represents the sparse correlated sequences of sequence X_n arranged from the least relevant to the most relevant. Each input vector includes a historical wind speed sequence and its augmented sequence $Z = \{z_i^{t+1}, z_i^{t+2}, ...\}$ calculated by the OCM.

The augmented data increase the dimension of the input matrix X and length of the LSTM cell, thereby adding more correlated parameters to limit the expression of the output matrix Y, shown as follows:

$$\text{LSTM:}\quad Y = \varphi\left(\omega^T X + b\right), \tag{24}$$

$$\text{Augmented LSTM:}\quad Y = \varphi\left(\omega^T \begin{bmatrix} X \\ Z \end{bmatrix} + b\right) \tag{25}$$

By pruning weakly correlated connections, the stability of the network is improved. The PA-CNN-LSTM model is described as follows:

$$\operatorname{argmin} \frac{1}{2N} \left\| Y - \sum_{i=1}^{N} \beta_i \widetilde{X}_i W_i^T \right\|_F^2, \tag{26}$$

$$\text{subject to}\quad \|\beta\|_0 \le N_0, \tag{27}$$

$$\beta_i \in \left\{ \beta_i = 0 | \rho(X_i, X_0) \le \theta_0 \right\}, \tag{28}$$

$$\varphi(V_i) = 0, \rho(V_i, V_0) \le \theta, \tag{29}$$

$$\widetilde{X}_k = [X_k, Z_k]^T \in R^{T_k + T_a \times d}, \tag{30}$$

$$Z_k = \left[z_1, z_2, \ldots, z_{\Delta T_k}\right]^T \in R^{\Delta T_k \times d} \tag{31}$$

where $\|M\|_F$ represents the Frobenius norm of matrix M. In terms of model pruning, it can be seen that the PA-CNN-LSTM model deletes the weak correlation tensor, reduces the dimension of X in the original problem, and reduces the solving space, thus improving the convergence efficiency. In the aspect of data augmentation, the augmented information Z_k provides the initial trajectory of the predicted sequence. Furthermore, the initial trajectory based on mathematical programming provides the deep learning model with more features from statistics and indirect constraints on WSSF and WPSF.

The Seq2Seq structure is suitable for integrating relevant information, statistical graphical models, and deep learning models for sequential tasks [45]. Based on the pretrained CNNs, the Seq2Seq model enhances model interpretability through the visual correlation between the recent data and the reference data. Due to the symmetry of the statistical correlation coefficient, the proposed two-phase Seq2Seq framework is suitable for optimization by bidirectional networks [65]. Other popular choices for optimizing Seq2Seq models include attention mechanisms [66] and shortcut connections or residual architectures [67].

In addition, several variations of the proposed model exist. Without the augmented corpus z_{it}, the proposed model is an augmented CNN-LSTM-LSTM (A-CNN-LSTM-LSTM) model. If Encoder 2 is hidden, the remainder of the structure is a classical LSTM-based structure.

5. Case study

5.1. Numerical experiment settings

The proposed model is evaluated by using the actual data from a wind farm in Northeast China. Sixty-six correlated wind turbines (WTs) are investigated, and the wind speed data are collected with a sampling interval of 10 min. The WT signal data are divided into a training dataset, test dataset and support dataset. The training dataset covers the days from the 1st to the 25th of each month, and the remaining days comprise the testing dataset. The support dataset is the dataset 20 days prior to each sample, which is the input for the pruned and augmented networks.

For the augmented networks, the length of the augmented sequence is set to 6, which means that every 10 min of the WT signals in the next 1 h are completed by the OCM to generate the augmented input. The output sequence of decoder 1 is set to $\widehat{Y}_{t+\Delta t} = (\widehat{y}^{t}, \widehat{y}^{t+1}, ..., \widehat{y}^{t+\Delta t-1}, \widehat{y}^{t+\Delta t})$, which involves the one-step data correction of the original signals.

The time series cross-validation method [68] is employed to train the parameters of all the models. The relationship between the length of the support dataset and the convergence of the PACNN-LSTM-LSTM model was investigated in numerical experiment I.

The mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), R squared (R^2) values and training time are recorded as the evaluation metrics. The calculations for the MAPE, RMSE and R^2 evaluation scores are shown in Formulas (32)-(35).

$$MAE = \frac{1}{T} \sum_{i=1}^{T} |\widehat{y}_i - y_i|, \tag{32}$$

$$MAPE = \left| \frac{\widehat{y}_i - y_i}{y_i} \right| \times 100\%, \tag{33}$$

$$R^2 = \left(1 - \frac{\sum_{i=1}^{T} (\widehat{y}_i - y_i)^2}{\sum_{i=1}^{T} (\bar{y} - y_i)^2} \right), \tag{34}$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^{T} (\widehat{y}_i - y_i)^2}, \tag{35}$$

where T is the total length of the time series, y_i is the actual wind speed in the test set, $\widehat{y}_i$ is the prediction and $\bar{y}$ is the average of the real values. The mean RMSE is applied to compare the results of multistep prediction.

$$MRMSE = \frac{1}{N} RMSE, \tag{36}$$

where N is the number of test samples.

The RMSE function with L2-norm serves as the loss function formulated in Formula (28). For the two-phase Seq2Seq structure, $RMSE(\omega; X, Y_{t+\Delta t})$ ensures the temporal relationship within the prediction time horizon for decoder 1, and $RMSE(\omega; X, Y_{t+\Delta t+n})$ improves the accuracy of end point prediction for decoder 2. In this experiment, the parameters of the loss functions are set to $\alpha_1 = \alpha_2 = 0.5$ and $\lambda = 0.01$.

$$L(\omega; X, Y_{t+\Delta t}, Y_{t+\Delta t+n}) = \alpha_1 * RMSE(\omega; X, Y_{t+\Delta t+n}) + \alpha_2 * RMSE(\omega; X, Y_{t+\Delta t}) + \lambda \|\omega\|_2^2 \tag{37}$$

In numerical experiment II, the effect of corpus augmentation, pruned convolutional layer and spatiotemporal series are validated. First, the effect of augmented data on the prediction is illustrated by comparing the PACNN-LSTM-LSTM model and A-CNN-LSTM-LSTM

model. A ConvLSTM-LSTM model [69] based on the encoder-decoder structure is employed to compare the effects of the unpruned convolutional layer and pruned convolutional layer. As one of the classical spatiotemporal series methods, the support vector machine (SVM) [70] is selected as the contrast model for comparing the forecasting performance.

The prediction algorithms are conducted on a personal computer with an Intel(R) Core (TM) i7-9750H 2.6-GHz CPU, a 16.00 GB 2667 MHz RAM and an 8.00 GB GeForce RTX 2070 GPU. Considering the optimal computing resources for different scales of the networks, the pruned network is carried on a central processing unit (CPU), while the unpruned network is run on a graphics processing unit (GPU).

5.2. Numerical experiment I: convergence verification

The convergence and error evaluation scores of the proposed method are evaluated by introducing three optimization algorithms, namely, Adam [71], Root Mean Squared Propagation (RMSprop) [72], and gradient descent (GD) [73]. To verify the influence of the proposed augmented method on the convergence performance, the support sets are set to three lengths: (a) 2 days, (b) 10 days, and (c) 20 days. Fig. 8 shows the convergence effect of the three optimizers and the influence of the augmented data on the convergence process of the optimizers.

In terms of model convergence, the loss functions based on the three optimizers can converge quickly. Although the GD algorithm is set with a high learning rate, its convergence process is the slowest. RMSProp starts with the slowest convergence rate, but after 150 iterations, the optimization of the loss function improves significantly. Compared with the RMSprop and GD algorithms, the initial gradient descent rate of the Adam algorithm is the fastest, and with an increase in the number of iterations, the global stability of the convergence process is optimal. Therefore, the Adam algorithm is taken as the optimization method of the proposed model for precision comparison experiments. Based on the Adam optimization algorithm, Fig. 9 shows the training MAPE, R^2 and RMSE curves based on four datasets.

5.3. Numerical experiment II: comparative experiment with the benchmark results

To verify the effectiveness, stability and accuracy of the proposed methods, the datasets were divided by the time series cross-validation method [68]. Table 1 lists the dominant statistical characteristics of the four datasets. It is common that the wind speed signals of the four seasons exhibit great fluctuations and variability. Compared with the data of four seasons, the data fluctuation of Season 2 and Season 3 increases significantly, which is recognized as an imbalanced data problem [74]. Table 2 shows the point prediction metrics of the four models. Fig. 10 compares the multistep predictions of ConvLSTM-LSTM and the proposed method.

The PACNN-LSTM-LSTM model performs the best in terms of the RMSE, MAPE and R^2 of the four datasets. Without the augmented networks based on the OCM, PCNN-LSTM-LSTM exhibited the best MAE in the Season 1 and Season 4 datasets.

In terms of MAPE, the PACNN-LSTM-LSTM model shows a more stable forecast effect when the signals are near 0. Considering the sparsity of the model, the sparse model based on network pruning is more suitable for addressing data fluctuations. In addition, the SVM model performs overfitting on the Season 2 dataset; overfitting is commonly caused by the random fluctuation of the nonstationary time series and the difference between the training set and the test set.

Considering the RMSE and R^2 scores of the training and test datasets, the augmented data and network do have an important role in the stability and accuracy of the model for the multivariate multistep prediction problem.

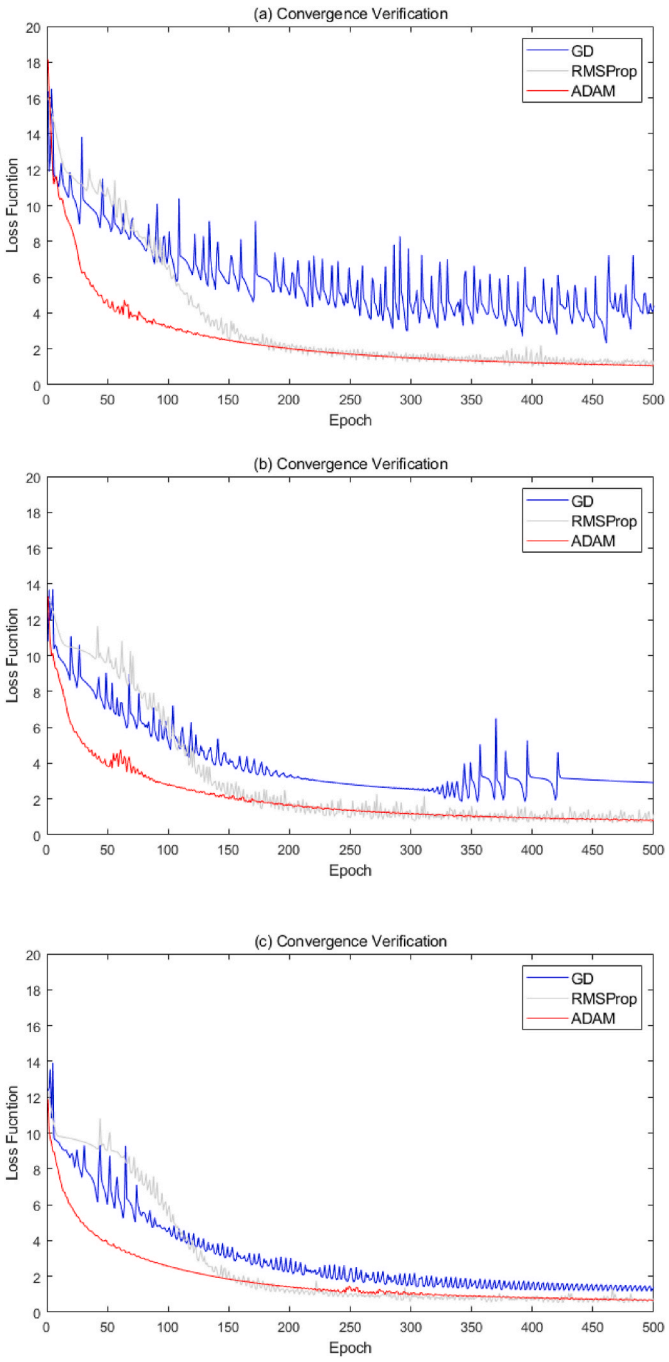

Fig. 8. Convergence curve of three optimizers.

6. Discussion

6.1. Convergence and stability

In terms of convergence, the results of numerical experiment I demonstrate the convergence of the proposed pruned and augmented network based on three optimization methods. Among them, the Adam optimizer exhibits the best convergence properties. Numerical experiment II shows that the correlated temporal and spatial data are able to improve the accuracy and stability of the LSTM-based encoder-decoder network for the multistep WSSF problem. Compared with that of the ConvLSTM-LSTM model, the RMSE of the multistep prediction of the proposed PACNN-LSTM-LSTM is more convergent, as shown in Fig. 10.

It is found that the augmented data also affect the convergence and

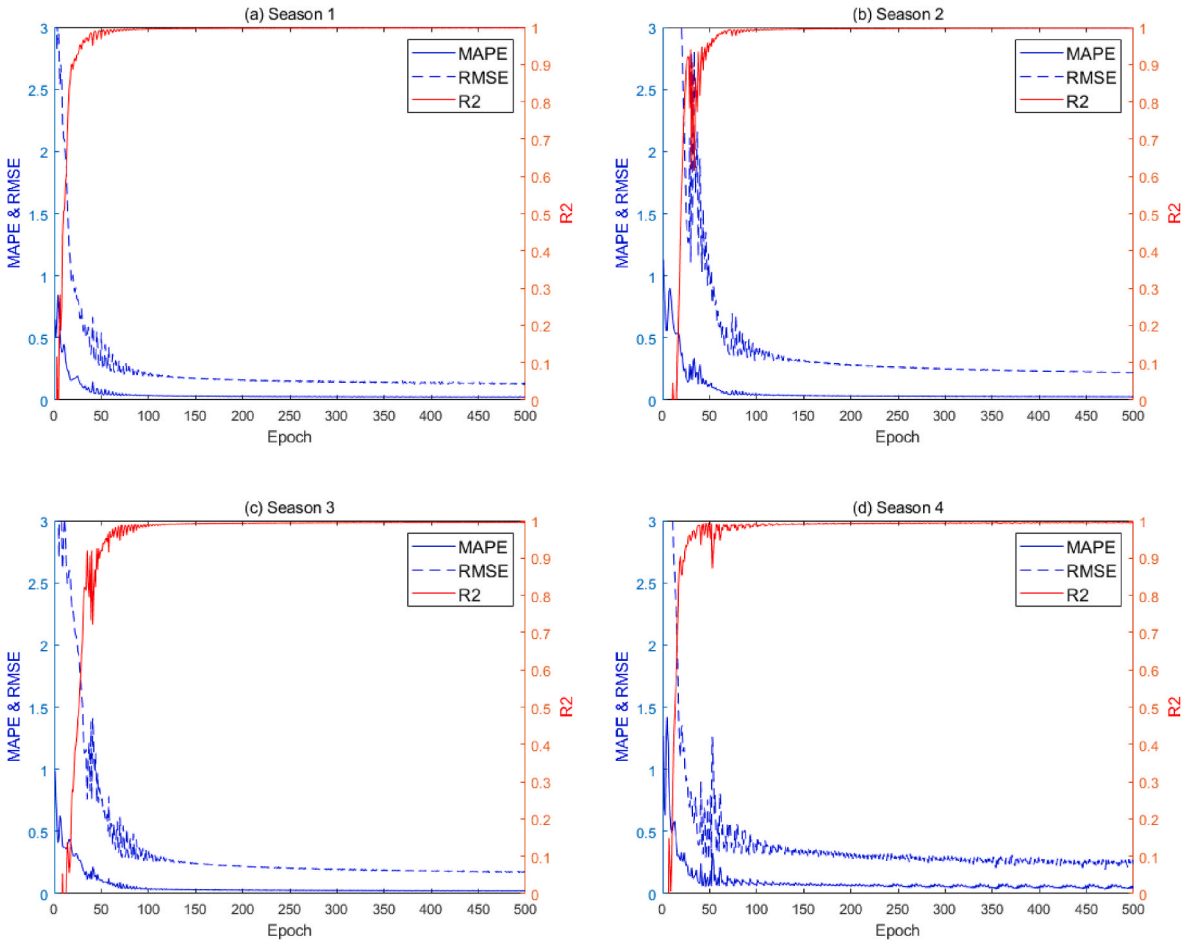

Fig. 9. Training performance curves of the four seasons.

Table 1
Characteristics of the datasets.

Period	Mean (m/s)	Std.	Skewness	Kurtosis	Max. (m/s)	Min. (m/s)	Q-25 (m/s)	Q-50 (m/s)	Q-75 (m/s)
Season 1	6.4789	3.93253	0.871	0.903	27.4	0.1	3.29	5.95	9.13
Season 2	8.857	4.7567	0.33	−0.663	24.66	0.24	4.99	8.35	12.43
Season 3	8.2071	3.93161	0.497	−0.076	34.84	0.13	5.28	7.92	10.7
Season 4	6.9262	3.81219	0.812	1.5	29.86	0.19	4.26	6.75	9.08

Table 2
Error evaluation scores of 1-h ahead forecasting results.

Dataset	Method	MAE	MAPE	RMSE	R^2
Season 1	PACNN-LSTM-LSTM	0.1231	3.61%	0.1350	0.9985
	PCNN-LSTM-LSTM	0.2805	7.12%	0.3027	0.9925
	ConvLSTM-LSTM	0.3485	10.88%	0.4689	0.9545
	SVM	0.7115	18.53%	0.9402	0.8713
Season 2	PACNN-LSTM-LSTM	0.3208	12.89%	0.3615	0.9672
	PCNN-LSTM-LSTM	0.2819	14.25%	0.3753	0.9647
	ConvLSTM-LSTM	1.1084	47.31%	1.3698	0.2776
	SVM	2.6423	67.17%	1.7320	−1.2972
Season 3	PACNN-LSTM-LSTM	0.4483	5.29%	0.7544	0.9700
	PCNN-LSTM-LSTM	0.4785	5.48%	0.8065	0.9657
	ConvLSTM-LSTM	1.3490	15.77%	1.3363	0.7471
	SVM	1.7309	28.01%	1.8062	0.6198
Season 4	PACNN-LSTM-LSTM	0.1986	8.19%	0.2259	0.9849
	PCNN-LSTM-LSTM	0.1968	8.38%	0.2299	0.9844
	ConvLSTM-LSTM	0.4234	24.77%	0.5952	0.7637
	SVM	0.7837	42.56%	1.1122	0.3976

stability of the network [75]. With an increase in the support set capacity, the correlation degree of the augmented corpus increases, and the stability of the GD algorithm is notably improved from (a)–(c) in Fig. 8. In addition, the augmented networks restrict the training scope of neural networks by using the OCM to introduce more factors from the correlation between the historical data and recent data. Therefore, the augmented corpus has a stabilizing role in the process of parameter optimization.

In addition, due to sparsity and data correction, the proposed network is suitable for solving imbalanced WSSF data. As shown in Fig. 9, the RMSE and MAPE of the endpoint prediction values drop rapidly and almost reach the bottom within 100 epochs, which indicates that the correlated data contribute to parameter optimization.

6.2. *Interpretability*

The input sequence and output sequence of the cross-correlation function and encoder-decoder structure are both bidirectional. The bidirectional structure provides a visual data representation of a correlation network referred to as the knowledge tree, as shown in Fig. 11. The knowledge tree is a kind of sparse data structure that reduces the

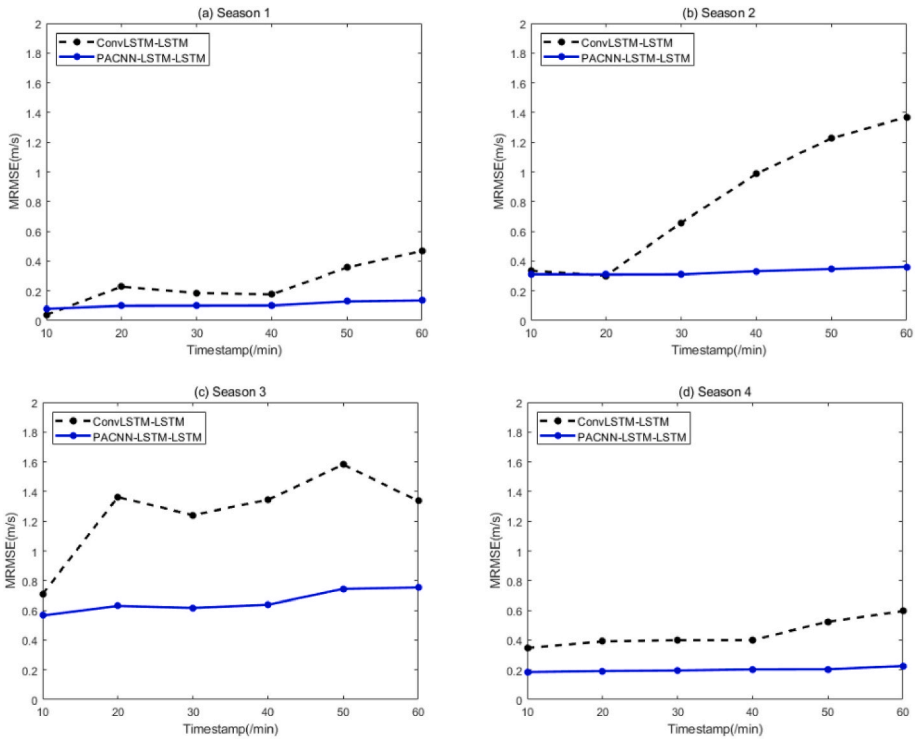

Fig. 10. Comparison of MRMSEs for multistep prediction of the two methods (ConvLSTM-LSTM and PACNN-LSTM-LSTM).

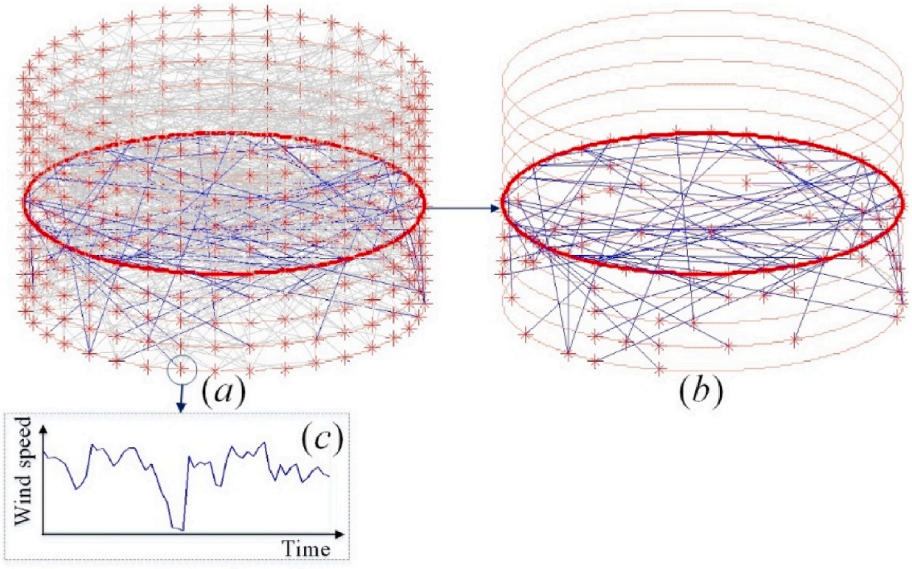

Fig. 11. Knowledge tree: (a) a tree structure containing wind speed sequence and correlation information, (b) a layer extracting data flow for one time point, and (c) a node including the relevant spatiotemporal data.

interference of noise sequences and contradictory data to machine learning models. Each forecasting output can be traced to its corresponding input samples, including correlations and other meteorological information. If necessary, the output can be modified, deleted, or added manually. Researchers can observe the influence of the relevant spatiotemporal sequences on the forecasting model.

6.3. *Training flexibility and hardware adaptability*

The training processes of convolutional network pruning and the two-phase encoder-decoder are separated. The complex convolution process has been split into time layers in daily offline training, thus greatly improving the time efficiency. Compared with the ConvLSTM-LSTM trained on GPUs, the PACNN-LSTM-LSTM trained on a CPU has more advantages in training efficiency, and its training time is shortened by 46%, as shown in Fig. 12.

The proposed multiblock deep networks can adapt to a variety of hardware conditions. Considering the time demand and hardware requirements, a typical encoder-decoder structure is extended to the two-phase encoder-decoder network. This network separates spatiotemporal feature extraction from model training and prediction. Through the offline trained knowledge tree based on CNNs and fine-tuned, online

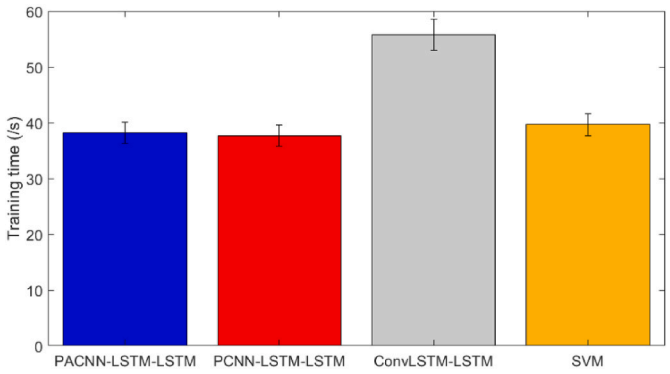

Fig. 12. Training times of four models.

sequence training and prediction, deep networks can be applied to a wide range of hardware conditions and meet the daily work requirements of wind farms. A novel idea is to match each network block to the optimal GPU or CPU computing units to address the problems of computing limitation or computing waste.

6.4. Convex optimization of the cross-correlation model

The computational complexity of the proposed convex optimization technique for the OCM model is O(n). Compared with the computational complexity O(n*log(2N)) of the bisection method used to solve the SDP model, the proposed convex optimization method is more efficient for the fast data augmentation and data correction.

7. Conclusions

The network pruning and data augmentation based on cross-correlation can effectively improve the traditional encoder-decoder structure that is selected for wind speed signal forecasting. In this paper, a convex optimized correlation model and a pruning CNN are established to construct a sparse encoder network for the proposed two-phase encoder-decoder deep learning structure. The proposed structure inherits the classical encoder-decoder structure and extends spatial-temporal feature extraction based on the augmented data and pruning CNN. The results of the numerical experiment demonstrate that the proposed two-phase deep learning network improves the stability, accuracy, and training efficiency for the signal prediction of the spatial-temporal wind turbine by hybridizing the pruned network and augmented network.

Funding

This work was supported by the Major Program of National Natural Science Foundation of China (71790614), the National Natural Science Foundation of China (72072029) and the 111 Project (B16009).

CRediT authorship contribution statement

Yang Yang: Methodology, Software, Data curation, Writing – original draft. **Jin Lang:** Project administration, Data curation, Validation, Supervision, Writing – review & editing, Funding acquisition. **Jian Wu:** Methodology, Data curation, Validation, Writing – review & editing. **Yanyan Zhang:** Investigation, Writing – review & editing, Funding acquisition. **Lijie Su:** Methodology, Writing – review & editing. **Xiangman Song:** Data curation, Writing – review & editing.

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.

Acknowledgements

The authors gratefully acknowledge the financial support for this work from the Major Program of National Natural Science Foundation of China (71790614), the National Natural Science Foundation of China (72072029) and the 111 Project (B16009). The authors also sincerely appreciate Professor Lixin Tang at Northeastern University, Professor Nikolaos Sahinidis at Georgia Institute of Technology, Professor Defeng Sun at Hong Kong Polytechnic University and Professor Toh Kim-Chuan at National University of Singapore for providing valuable suggestions and comments on this work.

Appendix A

The appendix is divided into two sections. In the first section, a nonlinear fractional programming problem (A-1) and its general solution are introduced. The second section first describes the proposed quadratic fraction programming problem. On this basis, an efficient numerical method based on closed form solution is proposed, and the corresponding derivation is presented. Then, the proposed problem is generalized to the multiple blocks of variables.

A1 Nonlinear fractional programming and nonlinear parameter programming problem

A nonlinear fractional programming problem and a nonlinear parametric programming problem are introduced for the first time in the paper [61] (Refer to Dinkelbach [61], p. 493). Let E^n be the Euclidean space of dimension n,and let S be a compact and convex subset of E^n. Let $p(x) : \mathbb{C} \rightarrow \mathbb{R}$ and $q(x) : \mathbb{C} \rightarrow \mathbb{R}$ be continuous functions of $x \in S$, where $\mathbb{C}$ represents the convex set. Assume that $q(x) > 0$ for all. $x \in S$.

A concave-fractional programming problem (A-1) and its corresponding parametric programming problem (A-2) are taken into account:

$$\max\{p(x) / q(x) | x \in S\} \tag{A-1}$$

$$\max\{p(x) - tq(x) | x \in S\} \text{ for } t \in E^1. \tag{A-2}$$

Paper [61] proved that problems (A-1) and (A-2) have solutions. Since $p(x)$ and $q(x)$ are continuous, S is compact, and the singular points defined by $q(x) = 0$ are excluded.

Lemma 1. (*Refer to Dinkelbach* [57], *p.* 494. *theorem*): $t_0 = p(x_0)/q(x_0) = \max\{p(x)/q(x) | x \in S\}$ *if, and only if,* $f_0(t_0) = f_0(t_0, x_0) = \max\{p(x) - t_0 q(x) | x \in S\} = 0$.

Furthermore, the theorem is still valid if "max" is replaced with "min".

In this section, $p(x)$ *is assumed to be concave, and* $q(x)$ *is assumed to be convex for* $x \in S$.

In terms of optimization, Agrawal and Boyd classified this concave-fractional programming problem (A-1) as the following disciplined quasiconvex programming (Refer to Agrawal [76], p. 6.).

$$\begin{aligned} &\min\ f_0(x)\\ &s.t.\quad f_i(x)\le 0, i=1,\ldots,m\\ &\quad Ax=b. \end{aligned} \tag{A-3}$$

According to Lemma 1, a property of disciplined quasiconvex programming is obtained, as shown in Proposition 1. According to Proposition 1, its minimized form $\min_{x\in S} f_0(x)=-\frac{p(x)}{q(x)}$ *can be represented via a family of convex functions* $\varphi_t(x) = -(p(x) - tq(x))$.

Proposition 1. (*refer to Agrawal [76], p. 3.*). ***Quasiconvex representation via a family of convex functions***: *The sublevel sets of a quasiconvex function can be represented as inequalities of convex functions. In this sense, every quasiconvex function can be represented by a family of convex functions. If a function* $f:\mathbb{C}\to\mathbb{R}$ *is quasiconvex, then there exists a family of convex functions* $\varphi_t:\mathbb{C}\to\mathbb{R}$, *indexed by* $t\in\boldsymbol{R}$, *such that*

$$f(x)\le t \Leftrightarrow \varphi_t(x)\le 0$$

The indicator functions for the sublevel sets of f,

$$\varphi_t(x)=\begin{cases}0 & f(x)\le t\\ \infty & otherwise,\end{cases}$$

generate one such family.

A convex feasible problem (A-4) is then proposed to obtain the optimal value of the quasiconvex problem (A-3). Let p^* *denote the optimal value of the quasiconvex optimization problem (A-3). If the feasibility problem*

$$\begin{aligned} &\text{find}\ \ x\\ &s.t.\qquad \varphi_t(x)\le 0\\ &\quad f_i(x)\le 0, i=1,\ldots,m\\ &\quad Ax=b, \end{aligned} \tag{A-4}$$

is feasible, then $p^*\le t$. *Conversely, if problem (A-4) is infeasible, then we can conclude that* $p^*\le t$. *Problem (A-4) is a convex feasibility problem, since the inequality constraint functions are convex and the equality constraints are linear. Thus, we can check whether the optimal value* p^* *of a quasiconvex optimization problem is less than or more than a given value t by solving the convex feasibility problem (A-4). If the convex feasibility problem is feasible, then we have* $p^*\le t$, *and any feasible point x is feasible for the quasiconvex problem and satisfies* $f_0(x)\le t$. *If the convex feasibility problem is infeasible,* $p^*\ge t$.

In terms of the solution of the quasiconvex problem, Algorithm 1 based on the bisection method is proposed as follows: The setting of the initial interval for Algorithm 1 and the detailed proof of convergence are presented in the paper (See Dinkelbach [61], p. 495 or Boyd [63], p. 145-146).

Algorithm 1. *Bisection method for quasiconvex optimization.*

Algorithm 1 Bisection method for quasiconvex optimization.

given $l\le p^*$, $u\ge p^*$, tolerance $\xi>0$.

repeat

1. $t:=(l+u)/2$.
2. Solve the convex feasibility problem (A-4).
3. **if** (A-4) is feasible, $u:=t$; **else** $l:=t$.

until $u-l\le\xi$.

A2 Quadratic Fractional Programming

a) *Quadratic fractional programming problem based on the OCM model*

The proposed OCM model is a quadratic fractional programming problem with the boundary constraints. Let the time series $X^{'}=[x_1,\ldots,x_N]^T\in\mathbb{R}^{N\times 1}$ *be a historical wind speed sequence of a wind turbine and let* $X^*=[x_{N+1},\ldots,x_{N+\Delta T}]^T\in\mathbb{R}^{\Delta T\times 1}$ *be a time series that represents the augmented data of* $X^{'}$. $X=[X^{'},X^*]^T\in\mathbb{R}^{(N+\Delta T)\times 1}$ *represents the integrated sequence. Let* $\rho^2(X^*)\in\{f_0(x)|f_0(x):\mathbb{C}\to[0,1]\}$ *be the quasi-convex optimized function of the Pearson cross-correlation. The proposed fractional quadratic programming problem is formulated as*:

$$\begin{aligned} &\max_{X^*}\quad f_0(X^*;X^{'},Y)=\frac{M(X^*)}{D(X^*)}=\frac{\mathrm{cov}(X,Y)\cdot\mathrm{cov}(X,Y)}{\mathrm{cov}(X,X)\cdot\mathrm{cov}(Y,Y)}\\ &s.t.\ f_i(X^*)\le 0,\quad i=1,\ldots,m, \end{aligned} \tag{A-5}$$

where the boundary constraints $f_i(X^*) \in \{f(x)|f(x): \mathbb{C} \rightarrow \mathbb{R}\}$ *include the natural speed bounds of wind and the feasible upper and lower bounds of wind speed according to the design of wind turbines.*

By introducing a fixed constant t in each iteration, the quadratic fractional optimization problem (A-5) *can also be transformed into a quadratic convex problem* (A-6) *based on* Algorithm 1. *The value of the constant t is then gradually adjusted after each iteration until it approaches the optimal solution to the original problem. The interval of t is known, that is,* $0 \leq t \leq 1$, *as the physical meaning of the constant t is the square of the correlation coefficient* (ρ^2).

Based on the OCM model (A-5), *the quadratic convex problem is shown as follows.*

$$\begin{aligned} &\min_{X^* \in S} \ \varphi_t(X^*) = p(X^*) - tq(X^*) \\ &s.t. \ f_i(X^*) \leq 0, \qquad i = 1, \ldots, m, \end{aligned} \tag{A-6}$$

where $\varphi_t : \mathbb{C} \rightarrow \mathbb{R}$ *is a convex representation of sublevel sets of* f_0 *for all* $t \geq 0$. $p(X^*) = X^* M_1 X^* + M_2 X^* + M_3$ *and* $q(X^*) = X^* D_1 X^* + D_2 X^* + D_3$ *are the quadratic functions of the augmented variables. The convex feasibility problem* (A-6) *can then be solved by semidefinite programming methods* [77] *based on the following theorem.*

Lemma 2. (*refer to Lin* [78], *p.* 1008-1009, *theorem no.* 3.1): *Let* $f(X) = X^T AX + BX + C = 0$ *be an n-dimensional quadratic equation, where A is an n-order, nonzero, real-valued symmetric matrix with* $rank(A) = r$, $X = \begin{pmatrix} x_1 \\ \cdots \\ x_n \end{pmatrix} \in \mathbb{R}^{n \times 1}$, $B = \begin{pmatrix} b1 \\ \cdots \\ b_n \end{pmatrix}^T \in \mathbb{R}^{1 \times n}$, $C \in \mathbb{R}$. *a) If A is positive semidefinite,* $f(X)$ *is a convex function for all* $X \in S$; *b) If A is negative semidefinite,* $f(X)$ *is a concave function for all* $X \in S$.

Furthermore, for the unconstrained OCM model and OCM model with simple boundary constraints, there is an analytical expression, and it is proven that the analytical expression of the one-dimensional case is also the optimal solution of the high-dimensional case. The related proposition and proof are given as follows:

Theorem 1. *According to the* Eqs. 12–22 *in* Section 3.1, *the expression of the analytical solution in each dimension is equivalent to the analytical solution obtained by separately solving the corresponding one-dimensional problem.*

Let X and Y be two time series that have a significant linear correlation, where the sample size of X and Y is n. All samples in the sequence Y are known, while the sequence X consists of a known sequence $X^{'}$ *and an unknown augmented sequence* X^*. *Let* $n^{'}$ *and* n^* *be the sample sizes of* $X^{'}$ *and* X^*, *respectively. Thus, the sample size of X is also equal to* $n = n^{'} + n^*$.

Lemma 3. *If Theorem* 1 *is true, when* $n^* = k$ ($k \geq 1, k \in N^+$), *then it is also true when* $n^* = k + 1$.

Proof. *If Lemma* 3 *is not true, then there exist two different optimal solutions for* x_h *and* $h = 1, \ldots, k$ *from quadratic convex problems* (A-6) *when the simple size of* X^* *is k and k*+1, *shown as follows.*

$$\begin{aligned} &{x_h}^{(k+1)} = \mathrm{argmin}\varphi_t\left({X^*}^{(k+1)}\right), \\ &{x_h}^{(k)} = \mathrm{argmin}\varphi_t\left({X^*}^{(k)}\right), \\ &{x_h}^{(k)} \neq {x_h}^{(k+1)}, \end{aligned} \tag{A-7}$$

where the top corner $\cdot^{(k)}$ *represents the sample size of the augmented sequence* X^*. *Therefore, the inequality* ${x_h}^{(k)} \neq {x_h}^{(k+1)}$ *in* (A-7) *is true if and only if Lemma* 3 *is not true.*

Let E^n *be an n-dimensional Euclidean space, where X and Y are two hyperplanes belonging to* E^n, *namely,* $X, Y \in E^n$. *According to Zhang's work on the correlation coefficient* (*refer to Zhang et al.* [58], *p.* 988), *the function of the transformed correlation coefficient* $\rho(X, Y) : E^n \rightarrow R$ *is a distance metric between the hyperplane X and hyperplane Y, as shown in Eq.* (A-8).

$$\rho_{XY}^2 = \frac{\mathrm{cov}(X,Y) \cdot \mathrm{cov}(X,Y)}{\mathrm{cov}(X,X) \cdot \mathrm{cov}(Y,Y)} \\ \frac{\left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j>i}^{n}(x_i - x_j)(y_i - y_j)\right)^2}{\left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j>i}^{n}(x_i - x_j)(x_i - x_j)\right)\left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j>i}^{n}(x_i - x_j)(x_i - x_j)\right)} \tag{A-8}$$

When the sample size $n \geq 3$ *and the distance metric* ρ *are determined, the distance relationship between all the sample points* x_i *and* y_i *is uniquely determined in Euclidean space* E^n. *Therefore, combined with Lemma* 1, *the inequality* ${x_h}^{(k)} \neq {x_h}^{(k+1)}$ *in* (A-7) *is not true. Consequently, Lemma* 3 *is proved.*

Lemma 4. *When* $n = 2$, *the solutions from the corresponding one-dimensional quadratic optimization problem are solutions to the two-dimensional quadratic optimization problem.*

Let two sequences X and Y, $X = [p_1, p_2, \ldots, x_1, x_2]$ *and* $Y = [q_1, q_2, \ldots, q_{n-1}, q_n]$, *be highly linearly dependent. The length of X and Y is n, where the last* 2 *dimensions of X,* x_1 *and* x_2 *are the unknowns written as follows: In a two-dimensional OCM optimization problem, the values of* x_1 *and* x_2 *can be inferred according to the maximization of the Pearson correlation coefficient.*

According to the formula for variance and covariance, $\mathrm{cov}(X, Y)$, $\mathrm{cov}(Y, Y)$ *and* $\mathrm{cov}(X, X)$ *are shown as:*

$$\mathrm{cov}(X,Y) = c_8 + c_9 x_1 + c_{10} x_2,$$

$$\mathrm{cov}(Y,Y) = c_{11} + c_{12} x_1^2 + c_{13} x_1 + c_{14} x_2^2 + c_{15} x_2 + c_{16} x_1 x_2,$$

$$\mathrm{cov}(X,X) = c_{17},$$

where $c_8 \sim c_{17}$ *are constants. Based on Lemma* 2, *the quadratic convex problem* (A-6) *is equal to the parametric programming problem as:*

$$X^T AX + BX + C = 0, \tag{A-9}$$

where A is a second–order, nonzero, real-valued symmetric matrix;

$$A = \begin{bmatrix} -c_9^2 + \rho^2 c_{12} c_{17} & -c_9 c_{10} + \frac{1}{2}\rho^2 c_{16} c_{17} \\ -c_9 c_{10} + \frac{1}{2}\rho^2 c_{16} c_{17} & -c_{10}^2 + \rho^2 c_{14} c_{17} \end{bmatrix},$$

$$B = \begin{bmatrix} -2c_8 c_9 + \rho^2 c_{14} c_{17} \\ -2c_8 c_{10} + \rho^2 c_{15} c_{17} \end{bmatrix}^T,$$

$$C = -c_8^2 + \rho^2 c_{11} c_{17}.$$

It is found that the Eq. (A-9) holds when the one-dimensional solutions Eqs. 12–22 *in* Section 3.1 *for* x_1 *and* x_2 *are introduced. Therefore, Lemma* 4 *is proved.*

Finally, based on Lemma 3 *and Lemma* 4, *Theorem* 1 *is proven by mathematical induction.*

b) *A more general convex quadratic programming problem and its solver*

In WSSF and WSPF problems, wind speed and wind power are often affected by the scheduling rules of the power grid and the control rules of WTs. These rules tend to affect a certain number of power units on a subregional basis. The mathematical programs based on data analytics and optimization in smart industry [79] *are defined as convex quadratic fractional programming with block constraints. For solving the SDP problem in Eq. (A-9) or more general problems with N blocks of variables, a MATLAB software package SDPNAL+* [80] *is developed based on a Newton-conjugate gradient (CG), augmented Lagrangian method* [81] *and a majorized, semismooth, Newton-CG, augmented Lagrangian Method* [82].

The SDP problem with N blocks of variables is defined as:

$$\begin{aligned} \min \quad & \sum_{j=1}^{N} \langle C^{(j)}, X^{(j)} \rangle \\ s.t. \quad & \sum_{j=1}^{N} A^{(j)}\left(X^{(j)}\right) = b, \qquad l \le \sum_{j=1}^{N} B^{(j)}\left(X^{(j)}\right) \le u, \\ & X^{(j)} \in K^{(j)}, \quad X^{(j)} \in P^{(j)}, \quad j = 1, \ldots, N, \end{aligned} \tag{A-10}$$

where $A^{(j)}: X^{(j)} \to \mathbb{R}^m$, *and* $B^{(j)}: X^{(j)} \to \mathbb{R}^p$ *are given linear maps and* $P^{(j)} := \{X^{(j)} \in X^{(j)} \,|\, L^{(j)} \le X^{(j)} \le U^{(j)}\}$ *and* $L^{(j)}$, $U^{(j)} \in X^{(j)}$ *are given symmetric matrices where the elements are allowed to take the values* $-\infty$ *and* ∞, *respectively. Here,* $X^{(j)} = S^{n_j}(R^{n_j})$, *and* $K^{(j)} = X^{(j)}$ *or* $K^{(j)} = S_+^{n_j}(R_+^{n_j})$. *For subsequent expositions, note that when* $X^{(j)} = S^{n_j}$, *the linear map* $A^{(j)}: S_+^{n_j} \to R^m$ *can be expressed in the form of*

$$A^{(j)}\left(X^{(j)}\right) = \left[\langle A_1^{(j)}\left(X^{(j)}\right)\rangle, \ldots, \langle A_m^{(j)}\left(X^{(j)}\right)\rangle\right]^T,$$

where $A_1^{(j)}, \ldots, A_m^{(j)} \in S^{n_j}$ *are given constraint matrices. The corresponding adjoint* $(A^{(j)})^*: R^m \to S^{n_j}$ *is then given by*

$$\left(A^{(j)}\right)^* y = \sum_{k=1}^{m} y_k A_k^{(j)}$$

References

[1] C. Magazzino, M. Mele, N. Schneider, A machine learning approach on the relationship among solar and wind energy production, coal consumption, GDP, and CO2 emissions, Renew. Energy 167 (2021) 99–115, https://doi.org/10.1016/j.renene.2020.11.050.

[2] International Renewable Energy Agency (IRENA), U.A.E. Abu Dhabi, Renewable Capacity Statistics, 2021. https://irena.org/publications/2021/March/Renewable-Capacity-Statistics-2021. (Accessed 12 December 2021), 2021.

[3] United Nations, Framework Convention on climate change (UNFCCC), Paris, France, adoption of the Paris agreement. https://undocs.org/FCCC/CP/2015/L.9/Rev.1, 2015. (Accessed 1 June 2021).

[4] World Bank, The Development Research Center of the State Council, China 2030: Building a Modern, Harmonious, and Creative Society, World Bank, Washington, DC, 2013.

[5] J. Chatterjee, N. Dethlefs, Scientometric review of artificial intelligence for operations & maintenance of wind turbines: the past, present and future, Renew. Sustain. Energy Rev. 144 (2021), 111051, https://doi.org/10.1016/j.rser.2021.111051.

[6] R. Kumar, M. Ismail, W. Zhao, M. Noori, A.R. Yadav, S. Chen, V. Singh, W. A. Altabey, A.I.H. Silik, G. Kumar, J. Kumar, A. Balodi, Damage detection of wind turbine system based on signal processing approach: a critical review, Clean Technol. Environ. Policy 23 (2021) 561–580, https://doi.org/10.1007/s10098-020-02003-w.

[7] Y. Liu, R. Ferrari, P. Wu, X. Jiang, S. Li, J.W.V. Wingerden, Fault diagnosis of the 10MW floating offshore wind turbine benchmark: a mixed model and signal-based approach, Renew. Energy 164 (2021) 391–406, https://doi.org/10.1016/j.renene.2020.06.130.

[8] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai, X. Duan, Y. Liu, Learning temporal and spatial correlations jointly: a unified framework for wind speed prediction, IEEE Trans. Sustain. Energy 11 (2020) 509–523, https://doi.org/10.1109/tste.2019.2897136.

[9] Y. Xu, G. Yang, J. Luo, J. He, H. Sun, A multi-location short-term wind speed prediction model based on spatiotemporal joint learning, Renew. Energy 183 (2021) 148–159, https://doi.org/10.1016/j.renene.2021.10.075.

[10] F. Vahedifard, A. Ermagun, K. Mortezaei, A. AghaKouchak, Integrated data could augment resilience, Science 363 (2019) 134, https://doi.org/10.1126/science.aaw2236.

[11] Y. Hou, Y. Liu, W. Che, T. Liu, Sequence-to-sequence Data Augmentation for Dialogue Language Understanding, 2018, 01554 arXiv:1807.

[12] T. Liu, H. Wei, K. Zhang, Wind power prediction with missing data using Gaussian process regression and multiple imputation, Appl. Soft Comput. 71 (2018) 905–916, https://doi.org/10.1016/j.asoc.2018.07.027.

[13] L. Liu, J. Wang, Super multi-step wind speed forecasting system with training set extension and horizontal–vertical integration neural network, Appl. Energy 292 (2021), 116908, https://doi.org/10.1016/j.apenergy.2021.116908.
[14] H. Liu, C. Chen, Multi-objective data-ensemble wind speed forecasting model with stacked sparse autoencoder and adaptive decomposition-based error correction, Appl. Energy 254 (2019), 113686, https://doi.org/10.1016/j.apenergy.2019.113686.
[15] S.K. Yeom, P. Seegerer, S. Lapuschkin, A. Binder, S. Wiedemann, K.R. Müller, W. Samek, Pruning by explaining: a novel criterion for deep neural network pruning, Pattern Recogn. 115 (2021), 107899, https://doi.org/10.1016/j.patcog.2021.107899.
[16] M. Khalid, A.V. Savkin, A method for short-term wind power prediction with multiple observation points, IEEE Trans. Power Syst. 27 (2012) 579–586, https://doi.org/10.1109/TPWRS.2011.2160295.
[17] S. Al-Yahyai, Y. Charabi, A. Gastli, Review of the use of numerical weather prediction (NWP) models for wind energy assessment, Renew. Sustain. Energy Rev. 14 (2010) 3192–3198, https://doi.org/10.1016/j.rser.2010.07.001.
[18] N. Chen, Z. Qian, I.T. Nabney, X. Meng, Wind power forecasts using Gaussian processes and numerical weather prediction, IEEE Trans. Power Syst. 29 (2014) 656–665, https://doi.org/10.1109/TPWRS.2013.2282366.
[19] S. Hu, Y. Xiang, H. Zhang, S. Xie, J. Li, C. Gu, W. Sun, J. Liu, Hybrid forecasting method for wind power integrating spatial correlation and corrected numerical weather prediction, Appl. Energy 293 (2021), https://doi.org/10.1016/j.apenergy.2021.116951.
[20] H. Wang, S. Han, Y. Liu, J. Yan, L. Li, Sequence transfer correction algorithm for numerical weather prediction wind speed and its application in a wind power forecasting system, Appl. Energy 237 (2019) 1–10, https://doi.org/10.1016/j.apenergy.2018.12.076.
[21] Z. Di, J. Ao, Q. Duan, J. Wang, W. Gong, C. Shen, Y. Gan, Z. Liu, Improving WRF model turbine-height wind-speed forecasting using a surrogate- based automatic optimization method, Atmos. Res. 226 (2019) 1–16, https://doi.org/10.1016/j.atmosres.2019.04.011.
[22] A. Kusiak, W. Li, Estimation of wind speed: a data-driven approach, J. Wind Eng. Ind. Aerod. 98 (2010) 559–567, https://doi.org/10.1016/j.jweia.2010.04.010.
[23] I.G. Damousis, M.C. Alexiadis, J.B. Theocharis, P.S. Dokopoulos, A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation, IEEE Trans. Energy Convers. 19 (2) (2004) 352–361, https://doi.org/10.1109/TEC.2003.821865.
[24] I. Jebli, F.Z. Belouadha, M.I. Kabbaj, A. Tilioua, Prediction of solar energy guided by pearson correlation using machine learning, Energy 224 (2021), 120109, https://doi.org/10.1016/j.energy.2021.120109.
[25] S. Rehman, Long-term wind speed analysis and detection of its trends using Mann–Kendall test and linear regression method, Arabian J. Sci. Eng. 38 (2013) 421–437, https://doi.org/10.1007/s13369-012-0445-5.
[26] F. Liu, R. Li, A. Dreglea, Wind speed and power ultra short-term robust forecasting based on Takagi–Sugeno fuzzy model, Energies 12 (18) (2019) 3551, https://doi.org/10.3390/en12183551.
[27] F. Liu, R. Li, Y. Li, Y. Cao, D. Panasetsky, D. Sidorov, Short-term wind power forecasting based on T-S fuzzy model, in: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), IEEE, Xi'an (China), 2016, pp. 414–418, https://doi.org/10.1109/APPEEC.2016.7779537.
[28] J. Tastu, P. Pinson, P.J. Trombe, H. Madsen, Probabilistic forecasts of wind power generation accounting for geographically dispersed information, IEEE Trans. Smart Grid 5 (2014) 480–489, https://doi.org/10.1109/tsg.2013.2277585.
[29] M. He, L. Yang, J. Zhang, V. Vittal, A spatio-temporal analysis approach for short-term forecast of wind farm generation, IEEE Trans. Power Syst. 29 (2014) 1611–1622, https://doi.org/10.1109/tpwrs.2014.2299767.
[30] M. Wytock, Z. Kolter, Sparse Gaussian conditional random fields: algorithms, theory, and application to energy forecasting, in: Proceedings of the 30th International Conference on Machine Learning, JMLR:W&CP, Atlanta, Georgia, 2013, pp. 1265–1273.
[31] Y. Wang, Y. Yu, S. Cao, X. Zhang, S. Gao, A review of applications of artificial intelligent algorithms in wind farms, Artif. Intell. Rev. 53 (2020) 3447–3500, https://doi.org/10.1007/s10462-019-09768-7.
[32] F. Sun, T. Jin, A hybrid approach to multi-step, short-term wind speed forecasting using correlated features, Renew. Energy 186 (2022) 742–754, https://doi.org/10.1016/j.renene.2022.01.041.
[33] Y. Liu, H. Qin, Z. Zhang, S. Pei, Z. Jiang, Z. Feng, J. Zhou, Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model, Appl. Energy 260 (2020), 114259, https://doi.org/10.1016/j.apenergy.2019.114259.
[34] Y.Y. Hong, C.L.P.P. Rioflorido, A hybrid deep learning-based neural network for 24-h ahead wind power forecasting, Appl. Energy 250 (2019) 530–539, https://doi.org/10.1016/j.apenergy.2019.05.044.
[35] B. Gu, T. Zhang, H. Meng, J. Zhang, Short-term forecasting and uncertainty analysis of wind power based on long short-term memory, cloud model and non-parametric kernel density estimation, Renew. Energy 164 (2021) 687–708, https://doi.org/10.1016/j.renene.2020.09.087.
[36] Z. Zhang, L. Ye, H. Qin, Y. Liu, C. Wang, X. Yu, X. Yin, J. Li, Wind speed prediction method using shared weight long short-term memory network and Gaussian process regression, Appl. Energy 247 (2019) 270–284, https://doi.org/10.1016/j.apenergy.2019.04.047.
[37] Y. Chen, S. Zhang, W. Zhang, J. Peng, Y. Cai, Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting, Energy Convers. Manag. 185 (2019) 783–799, https://doi.org/10.1016/j.enconman.2019.02.018.
[38] X.J. Chen, J. Zhao, X.Z. Jia, Z.L. Li, Multi-step wind speed forecast based on sample clustering and an optimized hybrid system, Renew. Energy 165 (2021) 595–611, https://doi.org/10.1016/j.renene.2020.11.038.
[39] Q. Hu, R. Zhang, Y. Zhou, Transfer learning for short-term wind speed prediction with deep neural networks, Renew. Energy 85 (2016) 83–95, https://doi.org/10.1016/j.renene.2015.06.034.
[40] T. Peng, C. Zhang, J. Zhou, M.S. Nazir, Negative correlation learning-based RELM ensemble model integrated with OVMD for multi-step ahead wind speed forecasting, Renew. Energy 156 (2020) 804–819, https://doi.org/10.1016/j.renene.2020.03.168.
[41] F. Qu, J. Liu, Y. Ma, D. Zang, M. Fu, A novel wind turbine data imputation method with multiple optimizations based on GANs, Mech. Syst. Signal Process. 139 (2020), 106610, https://doi.org/10.1016/j.ymssp.2019.106610.
[42] N. Azzaya, K. Sanggil, Pruning method using correlation of weight changes and weight magnitudes in CNN, Int. J. Fuzzy Log. Intell. Syst. 18 (2018) 333–338, https://doi.org/10.5391/IJFIS.2018.18.4.333.
[43] Z. Tang, G. Zhao, T. Ouyang, Two-phase deep learning model for short-term wind direction forecasting, Renew. Energy 173 (2021) 1005–1016, https://doi.org/10.1016/j.renene.2021.04.041.
[44] D. Wei, J. Wang, X. Niu, Z. Li, Wind speed forecasting system based on gated recurrent units and convolutional spiking neural networks, Appl. Energy 292 (2021), 116842, https://doi.org/10.1016/j.apenergy.2021.116842.
[45] H. Liu, X. Mi, Y. Li, Z. Duan, Y. Xu, Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional gated recurrent unit network and support vector regression, Renew. Energy 143 (2019) 842–854, https://doi.org/10.1016/j.renene.2019.05.039.
[46] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: I. Sutskever, O. Vinyals, Q.V. Le (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 3104–3112.
[47] Y. Zhang, Y. Li, G. Zhang, Short-term wind power forecasting approach based on Seq2Seq model using NWP data, Energy 213 (2020), 118371, https://doi.org/10.1016/j.energy.2020.118371.
[48] S. Du, T. Li, Y. Yang, S.J. Horng, Multivariate time series forecasting via attention-based encoder–decoder framework, Neurocomputing 388 (2020) 269–279, https://doi.org/10.1016/j.neucom.2019.12.118.
[49] T.R. Derrick, J.M. Thomas, Time series analysis: the cross-correlation function, in: N. Stergiou (Ed.), Innovative Analyses of Human Movement, Human Kinetics Publishers, Champaign, IL, 2004, pp. 189–205.
[50] H.H. Goh, S.W. Lee, Q.S. Chua, K.C. Goh, K.T.K. Teo, Wind energy assessment considering wind speed correlation in Malaysia, Renew. Sustain. Energy Rev. 54 (2016) 1389–1400, https://doi.org/10.1016/j.rser.2015.10.076.
[51] D.A. Bechrakis, P.D. Sparis, Correlation of wind speed between neighboring measuring stations, IEEE Trans. Energy Convers. 19 (2004) 400–406, https://doi.org/10.1109/TEC.2004.827040.
[52] W. Wangdee, R. Billinton, Considering load-carrying capability and wind speed correlation of WECS in generation adequacy assessment, IEEE Trans. Energy Convers. 21 (2006) 734–741, https://doi.org/10.1109/TEC.2006.875475.
[53] J. Liu, X. Wang, Y. Lu, A novel hybrid methodology for short-term wind power forecasting based on adaptive neuro-fuzzy inference system, Renew. Energy 103 (2017) 620–629, https://doi.org/10.1016/j.renene.2016.10.074.
[54] Y. Jiang, G. Huang, X. Peng, Y. Li, Q. Yang, A novel wind speed prediction method: hybrid of correlation-aided DWT, LSSVM and GARCH, J. Wind Eng. Ind. Aerod. 174 (2018) 28–38, https://doi.org/10.1016/j.jweia.2017.12.019.
[55] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning Convolutional Neural Networks for Resource Efficient Inference, 2016, 06440 arXiv:1611.
[56] G. Li, J. Wang, H.W. Shen, K. Chen, G. Shan, Z. Lu, CNNPruner: pruning convolutional neural networks with visual analytics, IEEE Trans. Visual. Comput. Graph. 27 (2021) 1364–1373, https://doi.org/10.1109/tvcg.2020.3030461.
[57] C. Liu, H. Wu, Channel pruning based on mean gradient for accelerating Convolutional Neural Networks, Signal Process. 156 (2019) 84–91, https://doi.org/10.1016/j.sigpro.2018.10.019.
[58] Y.Y. Hong, T.R.A. Satriani, Day-ahead spatiotemporal wind speed forecasting using robust design-based deep learning neural network, Energy 209 (2020), 118441, https://doi.org/10.1016/j.energy.2020.118441.
[59] Z. Lin, X. Liu, Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network, Energy 201 (2020), 117693, https://doi.org/10.1016/j.energy.2020.117693.
[60] F. Marchetti, F. Becattini, L. Seidenari, A.D. Bimbo, Multiple trajectory prediction of moving agents with memory augmented networks, IEEE Trans. Pattern Anal. Mach. Intell. 99 (2020) 1, https://doi.org/10.1109/TPAMI.2020.3008558.
[61] W. Dinkelbach, On nonlinear fractional programming, Manag. Sci. 13 (1967) 492–498, https://doi.org/10.1287/mnsc.13.7.492.
[62] Y. Zhang, H. Wu, L. Cheng, Some new deformation formulas about variance and covariance, in: 2012 Proceedings of International Conference on Modelling, Identification and Control, IEEE, Wuhan, Hubei, 2012, pp. 987–992.
[63] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, MA, 2004.
[64] K.C.B.V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.
[65] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2018, pp. 4171–4186.
[66] S. Zhang, Y. Chen, J. Xiao, W. Zhang, R. Feng, Hybrid wind speed forecasting model based on multivariate data secondary decomposition approach and deep learning algorithm with attention mechanism, Renew. Energy 174 (2021) 688–704, https://doi.org/10.1016/j.renene.2021.04.091.
[67] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, 2016, pp. 770–778.
[68] C. Bergmeir, J.M. Benítez, On the use of cross-validation for time series predictor evaluation, Inf. Sci. 191 (2012) 192–213, https://doi.org/10.1016/j.ins.2011.12.028.
[69] X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, MIT Press, Montreal, Canada, 2015, pp. 802–810.
[70] M. Liu, Z. Cao, J. Zhang, L. Wang, C. Huang, X. Luo, Short-term wind speed forecasting based on the Jaya-SVM model, Int. J. Electr. Power Energy Syst. 121 (2020), 106056, https://doi.org/10.1016/j.ijepes.2020.106056.
[71] D. Kingma, J. Ba, Adam: a Method for Stochastic Optimization, 2014 arXiv: 1412.6980.
[72] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA Neural netw, Mach. Learn. 4 (2012) 26–31.
[73] S. Hochreiter, A.S. Younger, P.R. Conwell, Learning to learn using gradient descent, in: G. Dorffner, H. Bischof, K. Hornik (Eds.), Artificial Neural Networks — ICANN 2001, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 87–94.
[74] L. Chen, G. Xu, Q. Zhang, X. Zhang, Learning deep representation of imbalanced SCADA data for fault detection of wind turbines, Measurement 139 (2019) 370–379, https://doi.org/10.1016/j.measurement.2019.03.029.
[75] N.M. Hewahi, Neural network pruning based on input importance, J. Intell. Fuzzy Syst. 37 (2019) 2243–2252, https://doi.org/10.3233/JIFS-182544.
[76] A. Agrawal, S. Boyd, Disciplined quasiconvex programming, Opt Lett. 14 (2020) 1643–1657, https://doi.org/10.1007/s11590-020-01561-8.
[77] R.H. Tütüncü, K.C. Toh, M.J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Math. Program. 95 (2003) 189–217, https://doi.org/10.1007/s10107-002-0347-5.
[78] L.G. Lin, Y.W. Liang, W.Y. Hsieh, Convex quadratic equation, J. Optim. Theor. Appl. 186 (2020) 1006–1028, https://doi.org/10.1007/s10957-020-01727-5.
[79] L.X. Tang, Y. Meng, Data analytics and optimization for smart industry, Front. Eng. Manag. 8 (2021) 157–171, https://doi.org/10.1007/s42524-020-0126-0.
[80] D. Sun, K.C. Toh, Y. Yuan, X.Y. Zhao, SDPNAL+: a Matlab software for semidefinite programming with bound constraints (version 1.0), Optim. Methods Software 35 (2020) 87–115, https://doi.org/10.1080/10556788.2019.1576176.
[81] X.Y. Zhao, D. Sun, K.C. Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, SIAM J. Optim. 20 (2010) 1737–1765, https://doi.org/10.1137/080718206.
[82] L.Q. Yang, D.F. Sun, K.C. Toh, SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints, Mathemtical Program. Comput. 7 (2015) 331–366, https://doi.org/10.1016/j.measurement.2019.03.029.